



# Object Recognition using Fractal Geometry and Fuzzy Logic

A thesis presented by

**Dmitry A. Dubovitskiy**

Faculty of Computer Science and Engineering

In partial fulfillment of the requirements
for the degree of

**Doctor of Philosophy**

De Montfort University
Leicester England

October 2004

# Abstract

This thesis describes a novel approach to the object recognition problem for incoherent images using fractal geometry and fuzzy systems. Although the applications of this approach are general, in this work the method is applied to the evaluation of cytological states associated with cervix uteri diseases, skin cancer and a surface inspection system for quality control in the steel industry. These applications are the basis for industrial work undertaken during the development of this thesis. In each of these applications, the object recognition problem and its evaluation in terms of conventional human based assessment is particularly difficult in which conventional image processing and pattern recognition methods fail. The reason for this failure is due to the textural characteristics of the images under inspection. In this thesis, we consider the application of fractal geometry to provide a description of the textures associated with the nuclei of a cell for example and using fuzzy logic to categorise the patterns based on a set of fractal parameters. This requires segmentation of the nuclei from unwanted features such as the cytoplasm, white blood cells and bacteria or foreign inclusions on metal surface etc. which is undertaken using novel methods based on conventional image segmentation techniques. For fractal parameters calculation with the better accuracy is offered new approach. Eventually the number of new compact algorithms for two dimensional objects processing on an image is developed.

This thesis addresses the issue of a potential future pattern recognition technology, namely the possibility of utilizing the textural properties of a pattern to make decisions about the condition and characteristics of a sample.

# Acknowledgements

I would like to acknowledge my supervisors Professor J M Blackledge, Loughborough University, Professor V Deviatkov and Professor A Chernikov, (Bauman) Moscow State Technical University, Professor Brian Foxon, Dr B John and Dr X Li for their help throughout the process of undertaking this research project. I would also express my gratitude to De Montfort University and Microsharp Corporation Limited for their financial support throughout the duration of my PhD studies and to Professor Jonathan Brostoff (Kings College, London University), Dr Alastair Deery (Royal Free Hospital, London), Professor Irina Shabalova (Russian Medical Academy of Postgraduate Education, Moscow) and Dr Peter Richards (University College Hospital, London) for their help with regard to data provision and analysis. Finally, some of this work (surface inspection system) was supported by Novolipetsk Iron and Steel Corporation, Russia.

Finally, I would like to acknowledge my gratitude to all of my working colleagues who provided a helpful and friendly working environment during my time in England and Russia and to my family for their support throughout.

# Contents

# List of Figures

## Notation

| | |
|---|---|
| $(x, y)$ | Spatial coordinates (continuous) |
| $f(x, y), s(x, y)$ | Function, image |
| $n(x, y)$ | Additive noise, in image |
| $(k_x, k_y)$ | Spatial frequency of image |
| $F(k_x, k_y)$, $S(k_x, k_y)$, $N(k_x, k_y)$ | Image of function in frequency domain |
| $(m, n)$ | Spatial coordinates (digital number) |
| $\{f_{m,n}\}$, $\{s_{m,n}\}$, $\{n_{m,n}\}$ | Digital image |
| $(u, v)$ | Spatial frequency of digital image |
| $\{F_{u,v}\}$, $\{S_{u,v}\}$, $\{N_{u,v}\}$ | Image of digital image |
| $\hat{L}[f(x, y)]$ | Linear operator |
| $\delta(x)$ | Delta function $\delta(x) = \begin{cases} 1, & x = 0, \\ 0, & x \neq 0 \end{cases}$ |
| $\mathrm{Re}(z)$ | True part of a complex number $z$ |
| $\mathrm{Im}(z)$ | Imaginary part of a complex number $z$ |
| $z^*$ | Conjugate complex number $z^* = \mathrm{Re}(z) - i\,\mathrm{Im}(z)$ |
| $\mathrm{sinc}(x)$ | Function $\sin(x)/x$ |
| $\mathrm{comb}(x)$ | Function $\sum_{m=-\infty}^{\infty} \delta(x - n\Delta x)$, where $n$ — integer |
| $DFT$ | Discrete Fourier transform |
| $IDFT$ | Inverse discrete Fourier transform |
| $\otimes$, $\otimes\otimes$ | One-dimensional, two-dimensional convolution |
| $\odot$, $\odot\odot$ | One-dimensional, two-dimensional correlation |
| $\nabla f(x, y)$ | Gradient $\nabla f(x, y) \equiv \left[\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\right]$ |
| $\nabla^2 f(x, y)$ | Laplacian |

$$\Delta f(x, y) \equiv \nabla\left[\nabla f(x, y)\right] = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}.$$

# List of Tables

# Chapter 1

# Introduction

## 1.1 The Role of Machine Vision

Modern society has brought about a number of changes in human activity. In recent years, there has been a rapid advance in the science and technology of information processing and analysis. With developing technology and computational capacity, information processing, communication and decision making has become significant. Increasingly large streams of information have to be processed, compactly represented, estimated and saved to a data base.

Over the last forty years there was a significant development in object recognition and machine vision technologies [44]. It is now difficult to imagine an assembly line without a computer-assisted quality control and robots with image recognition hardware and software for example. The efficiency of automation defines the manufacturing competitiveness in the global economy. The military sector relies on high-technology intelligence gathering techniques and self-guided missiles such as the cruise missile which uses automatic image recognition of the ground surface to guide its self to a target. Space research programs require heavy duty image processing in order to search for new planets beyond our solar systems for example [48]. Automated systems are widely used in the health care industry for diagnostics in cytology, pathology, mammography and computed tomography. Geologists are equipped with computer systems processing satellite images of the earths surface for the identification of new mineral deposits.

Humans are very advanced in interpreting visual data from the natural world

and the recognition of faces and animals for example and other sophisticated objects can not be handled by computers to anyway near the accuracy of a human. Thus, humans can perform complex pattern recognition tasks very well; however, the technological drive is to build machines which can perform such tasks more accurately (where possible), faster and more cheaply than humans as well as to release humans from the drudgery of routine pattern recognition processes [65]. There are also issues concerned with purely technological tasks such as reading 2D bar codes at which humans are poor. Object recognition is the discipline of building such machines.

One characteristic of human object recognition is that it is mainly learnt. We cannot describe the rules we use to recognise a particular face, and will probably be unable to describe it well enough for anyone else to use the description for recognition. On the other hand, botanists can give the rules they use to identify flowering plants. There is a subject known as machine learning which has emerged from the artificial intelligence and computer science communities. It is concerned with distilling structures from labeled examples, although the labels are usually 'true' and 'false'. Machine Learning is generally taken to encompass automatic learning procedures based on logical or binary operations, that learn a task from a series of examples [76, 2]. It aims to generate classifying expression simple enough to be understood easily by humans. They must mimic human reasoning sufficiently well to provide insight into the decision process. Like statistical approaches, background knowledge may be exploited in development, but operation is assumed without human intervention.

The information revolution can be compared in its impact to that of the industrial revolution of the nineteenth century. The value of information has changed in that both the constructive and destructive powers of information flow are very different compared with the situation less than just hundred years ago. For example, in the industrial sector, information or technical data provides a primary competitive advantage and contributes considerably to company market value. Clearly, critical information is vital for national security and financial organizations. Accurate knowledge of public opinion allows political leaders to react rapidly in their policies or programs. Thus, political power is now critically dependent on the flow of information.

Organizations in both the public and private sectors have become increasingly

dependent on electronic data processing. Large amounts of digital data are now gathered and stored in computer data bases and transmitted between computers and terminal devices linked together in complex communications networks [30, 75].

Another feature of the 'information society' is its prioritization with respect to time. The speed of information flow has developed considerably over the past ten years. As with high speed vehicles, skilled control becomes increasingly important as information flow increases in the rapidity of its exchange; the world wide web being an environment in which information flow is now very difficult to control effectively. An important development has been the representation of data in a convenient form for automatic inspection. This can be explained in terms of an artificial agent informed by an appropriate knowledge data base. This kind of support is very useful to the interpretation of picture data bases: medical applications, landscape scanning and different kinds of surface inspection systems. The example of knowledge in a specific field of philosophy can be compared with the mathematical goal of first and second derivation. The first represents speed (information) and second, knowledge of information. Control of the latter also generates new information which can be helpful for researchers, engineering systems and improvement in the quality of life.

In this work, a new type of pattern recognition system is considered which focuses on embedding a new way to solve problems in the interpretation of complex digital images. Regardless of the embedding method, there are several requirements that the embedding technique must satisfy. First, the accuracy of recognition should retain as much as possible the quality of an original human decision. This means that it should not be distinguishable between the result obtained from a group of experts and that obtained artificially; thus, satisfying the test for artificial intelligence [20, 50]. Second, once the most appropriate image processing has been established, implementation should be computational efficient and provide an optimum output time. Also, it is highly desirable that the recognition process can be made simple but accurate. In addition, if the boundary between two classes is blurred, then the recognition process should to have a natural intuitive engine. Finally, the recognition should be robust to various types of distortion which may occur during the capture and/or transmission of the image.

Let us now consider an introduction to the theoretical background of the subject, its mathematical and computational basis and applications. In the following sections, each part is designed to provide an overview of a particular area making reference to the notes provided.

## 1.2 Object Recognition

For a typical object recognition system, the determination of the class is only one of the aspects of the overall task. In general, pattern recognition systems receive data in the form of 'raw' measurements which collectively form a stimuli vector [25, 80]. Uncovering relevant attributes in the features present within the stimuli vector is typically an essential part of such systems. An ordered collection of such relevant attributes which more faithfully or more clearly represent the underlying features of the object is assembled into a feature vector.

Class is only one of the attributes that may or may not have to be determined depending on the nature of the problem. The attributes may be discrete values, Boolean entities, syntactic labels or analog values. Learning in this context amounts to the determination of rules of associations between the features and attributes of an object.

Practical image recognition systems generally contain several stages in addition to the recognition engine itself.The recognition represents information processing that is realized by some converter of the information (by an intellectual information channel), having an input and output. On input, such a system establishes information on the properties of an objects. On output, the information shows which class or feature of an object is assigned.

### 1.2.1 Methods of pattern recognition and their characteristic

The tasks of construction and application of formal operations for numerical or character representation of objects of a real or ideal world is the basis of pattern recognition. The equivalence relations express a fit of evaluated objects to any class considered to be independent semantic units. The recognition classes of equivalence can be set by the user in the construction of an algorithm, which uses

own pithy representations or uses external padding information on a likeness and difference of objects in the context of a solved task; the basis for phrase 'recognition with the teacher'. When a computerized system decides on the task of classification without engaging external learning information, it is called as automatic classification or 'recognition without the teacher'. The majority of algorithms for pattern recognition require the engagement of a number of considerable computational capabilities, which can be provided only with high-performance computer equipment [2].

Distinguished parametric, nonparametric methods and others evolve groups of methods, outgoing from the historically folded schools and directions in the given area. A typical review of pattern recognition techniques involves the following typology of methods as used for pattern recognition:

(i) Methods based on a principle of separation [94] which shows good results in modeled conditions but does not work in real life where there is indistinct borders between classes.

(ii) Statistical methods [67] which work very well but require a significant amount of memory for storage of the original data and it is difficult to implement on current elements.

(iii) Methods constructed on the basis of 'potential functions' [44] which work under ideal conditions but in real life are difficult to control using natural objects.

(iv) Methods by calculation the estimations (voting) [26];

(v) Methods based on alphabet of propositional calculus, in particular on the machinery of logic algebra. [51, 26]

The combination of the last two methods provide relatively good results as researched in this thesis which introduces concepts from Fractal Geometry and Fuzzy Logic.

The representation of knowledge on the data domain with the use of any of formal algorithms for pattern recognition can be divided into two main areas:

(i) Intensional - the scheme of connections between attributes.

(ii) Extensional, with the help of particular facts (objects, examples).

These fundamental ways of knowledge representation allow for the following classification of methods of pattern recognition:

(i) Intensional methods based on operations with features.

(ii) Extensional methods based on operations with objects.

**Intensional methods**

The distinctive feature of intensional methods is the element of operations at construction and the application of algorithms for pattern recognition using a number characteristics of features and their communications. Such elements can be separate values or intervals of values of a property, mean values and dispersions, matrix of property connections and so on, upon which, the operations expressed in the analytical or design stage are made. Thus, the objects in such methods are not considered as integrated information units and act in the role of indicators for an estimation of the interaction and behaviour of the attributes.

The group of intensional methods of pattern recognition is extensive, and its division into subclasses has a particular measure conditional character:

(i) Methods based on estimations of densities function of values of features.

(ii) Methods based on the suppositions about the class of decision functions.

(iii) Logical methods.

(iv) Linguistic (structural) methods.

**Extensional methods**

For the methods of the given group, as against an intensional direction, the independent diagnostic value is attached to each studied object in a greater or smaller measure. In essence, these methods are close to the clinical approach, which considers individuals, not as a ranged space on this or that index but as a chain of objects and as integrated systems, each of which is unique with a singular diagnostic value. The basic operations in the recognition (identification) of abnormal situations are those operations that are allied to the definition of a likeness and distinction of an object. The cells of such a group of methods

play a role in diagnostic precedents. Thus, depending on the conditions of a particular task, the role of separate precedents can vary in the broadest limits: from principal determinism to rather oblique participation during recognition. In turn, the conditions of a task may require, for a successful solution, participation of different quantities of diagnostic precedents; from one in each recognized class to a full sample size, and also different ways of calculation of measures of a likeness and distinction of an object. These requirements dictate a further divisional extension based on the following subclasses:

(i) Method of matching with the prototype

(ii) Method of the k-proximate neighbors

(iii) Algorithms of calculation of estimations 'voting'

(iv) Collectives of the inference engines

### 1.2.2 A comparative analysis of the methods of pattern recognition

Let us consider via a comparative approach the methods described above with regard to their adequacy to the requirements of identification and classification of complex field patterns such as those associated with cytology for example. For the solution of substantial tasks from groups of methods of an intensional direction, the practical value is represented by parametric methods and methods based on the type of decision functions. Parametric methods provide the fundamentals for a conventional methodology based on the design of an index [15]. The application of these methods for real tasks is connected to the superposition of strong limitations on data structure, resulting in linear diagnostic models with very approximate estimations of their parameters. With usage of methods based on suppositions about a category of decision functions, the developer is also compelled to revert to linear models. These are governed by the high dimensional space of features that are a characteristic for substantial tasks, which increase of the degree of a polynominal decision function, leading to a large propagation in the number of terms and a problem concerning improvements in the quality of recognition. Thus, planning for the area of potential application for intensional

methods of recognition is of major importance and we need to consider a well defined approach and 'road map' when using a conventional methodology of linear diagnostic models [40].

The properties of linear diagnostic models, in which the diagnostic index is characterized by the fluidization of the total number of starting features has been investigated. The results of these models (with appropriate normalization) are interpreted in terms of the 'distance' from the researched objects to some hyper plane on the parameter space. This is equivalent to taking the projections of an object along a set of straight line in given space. Thus, linear models are adequate only for simple geometrical configurations on areas of a space of properties in which the objects of different diagnostic classes are mapped. For more composite distributions, these models, in essence, can not show many structural features of the experimental data although such features are capable of conveying valuable diagnostic information.

In addition to the above, it is further necessary to estimate appearance in any substantial task concerned with simple multivariate structures (in particular, multivariate normal distributions). Frequent diagnostic classes are based on constructs of external criteria that automatically entails a geometrical non-uniformity of the data of classes in a space of features. It especially concerns 'vital' criteria being met in practice. In such conditions, the application of linear models alerts only the most 'rough' of legitimate features obtained from experimental results.

The application of extensional methods requires suppositions about the structure of the experimental data in addition to those internal classes that should exist for one or several groups by characterized by similar objects. However, the objects of different classes should differ from each other to a limited extent. Any final dimension of 'learning' should be designed to support the fact that there are random differences between objects. As measures of likeness, different measures of proximity (distance) in an object space of features need to be applied. Therefore, the effect of using extensional methods of pattern recognition depends on: the success of how the measures of proximity are determined; how the objects of a learning sampling (the objects with a known classification) carry out the role of diagnostic precedents. The successful solution of task data must provid results approaching theoretically accessible limits of efficiency for the recogni-

tion process. The high technical complexity and its practical implementation is offset by the advantages of extensional methods of pattern recognition. For a multi-dimension parameter space, the simple task of finding a pair of proximity points yields a problem. Also, there remains the problem of excessive storage with regard to the recognition classes.

The theoretical problems associated with the application of extensional methods of recognition are connected to problems related to the search for informative groups of features. Finding the optimal metrics for measurement of a likeness that are distinct enough for such objects and analysis of their structure is vital. At the same time, the successful solution must allow not only for an effective recognition algorithm, but also provide a method that transforms extensional knowledge on empirical facts to intensional knowledge on legitimate features of their structure [71].

The transfer from extensional to intensional knowledge occurs at an early stage, when the formal recognition algorithm(s)is already constructed along with its efficiency criteria. The learning mechanisms are then considered subject to efficiency gains wherever possible. These mechanisms are connected with the analysis of geometrical structures on the data yielding results that are appropriate for the transformation of objects characterizing one or another diagnostic class into one typical (prototype) representative. As discussed before, this is equivalent to an assignment using a linear diagnostic scale. Also, it is possible that each diagnostic class needs to be exchanged by several objects interpreted in terms of typical representatives of some subclasses. Typical representatives are equivalent to the construction of a list of linear scales.

It is necessary to emphasize the existence of two, and only two, groups of methods of recognition: operating with features; operating with objects; the latter case being the more natural of the two. From this point of view, neither of these methods (taken in isolation) allow for the generation of a completely adequate representation in the data domain. In between these methods, there are a number of 'intercoupling' techniques and perspective systems of recognition should provide for the implementation of both methods.

In the given work it is considered three different area of application for image recognition system. All area rather identical from the point of view on the structural characteristic and decision-making of objects, difference consists only

in objects segmentation algorithms. In the modern literature there is no description of universal images segmentation mechanisms. The segmentation part of image processing can be separated in the section for further development.

## 1.3 Fractal Geometry

Fractal Geometry is considered in detail in Chapter 3.3. Here a brief overview is provided to help put the read in context.

If we are to consider the totality of real objects in our world in a form that is ambient to us, we need to come to terms with the fact that the majority of them consist of self-similar forms [8]. Self-similarity is not a simple whim of the nature, but a form that provides the most stable state of objects subject to external actions. For example, the human lung is arranged in such a (self-similar) manner that damage to any part of the lung allows for the prolonged functionality in a way that is only possible because of the self-similarity of its design. This feature of our world is described mathematically through the theory of self-similar and self-affine objects, i.e. Fractal Geometry [41, 8].

Fractal geometry is a consequence of the computing revolution, the rapid advances in digital processing and in particular, computer graphics [62]. Algorithms used to generate and analyse fractals and chaotic fields depend on:

- understanding a physical field and typically a non-linear dynamical system [70];
- a clear and concise (mathematical) definitions of the field properties [87].

The applications of fractal geometry is increasingly useful in many different area of science and engineering including [39, 65, 69]:

- Signal Processing - Time Series Analysis, Speech Recognition;
- Image Processing - Fractal Compression, Fractal Dimension Segmentation;
- Simulation - Terrain Modelling, Image Synthesis, Music, Stochastic Fields
- Financial - Fractal and Multi-Fractal Market Analysis, Futures Markets

- Medicine - Histology, Cytology, Monitoring, Epidemiology
- Military - Visual Camouflage, Covert Digital Communications

The research of different dynamic systems is connected with stage prediction of a system behavior and for this purpose of the Fractal and Chaos theory very well describes the laws of a natural world. The different fields of study of Fractals/Chaos are represented in Figure 1.1.

The origins of fractals and chaos go back many years and it is informative to provide a brief history of the subject which is given below.

- 1800-1897: K W T Weierstrass Nowhere Differentiable Functions
- 1854-1912: J H Poincare Non-Deterministic (Chaotic) Dynamics
- 1900s: Julia, Koch & others Julia sets (simple fractals) - arise in connection with the iteration of a function of a complex variable
- 1886-1971: P Levy Fractal Random Walks (Random Fractals)
- 1960s: E Lorenz Nonlinear Systems Dynamics and Chaos
- 1970s: B Mandelbrot Mandelbrot sets and the development of a general theory on the 'Fractal Geometry of Nature' (1977 [7], 1983[8])
- 1980 - date: Exponential growth in researchers, books, publications, patents and new businesses etc. Study of Deterministic and Random Fractals with a wide range of applications in science, engineering and IT (e.g. image compression - Mike Barnsley, Iterated Systems Inc.)

Euclidean Geometry is based on the theorems and results associated with simple objects - triangles, squares, circles, lines etc. Euclid was the first Greek mathematician to compound the mathematical ideas of the ancient world in his book 'The Elements' Figure 1.2. In Euclidean geometry, a number of abstract concepts are introduced, e.g. two parallel lines being defined as lines that meet at infinity. The underlying philosophy of Euclidean geometry is to combine primitive objects in order to construct complex ones. This is the basis of most man-made objects, computational geometry, pattern recognition systems etc [68]. It is also one of the underlying principle of science in which an understanding of a complex

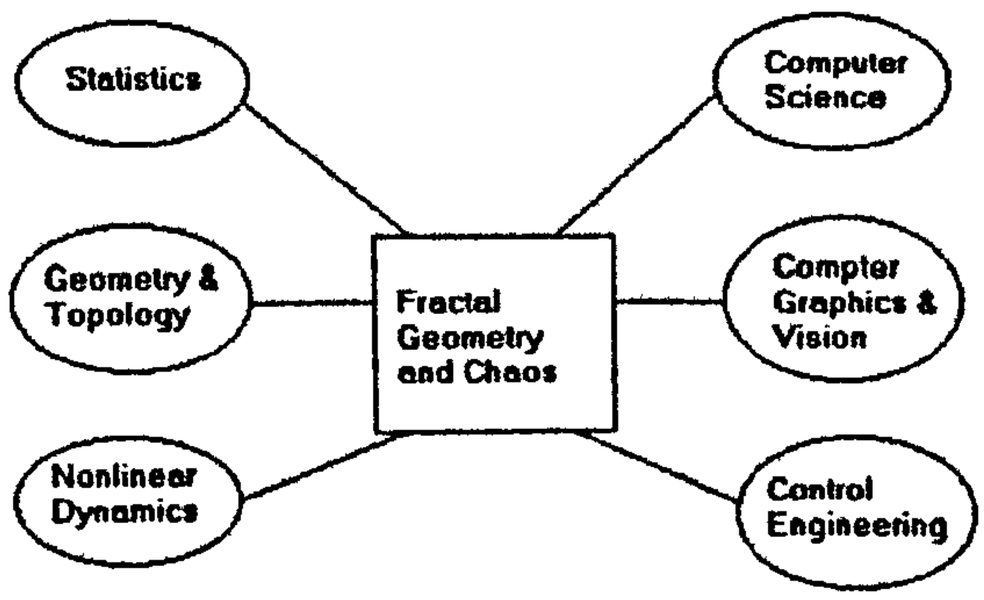



Figure 1.1: Different fields of study associated with fractals and chaos

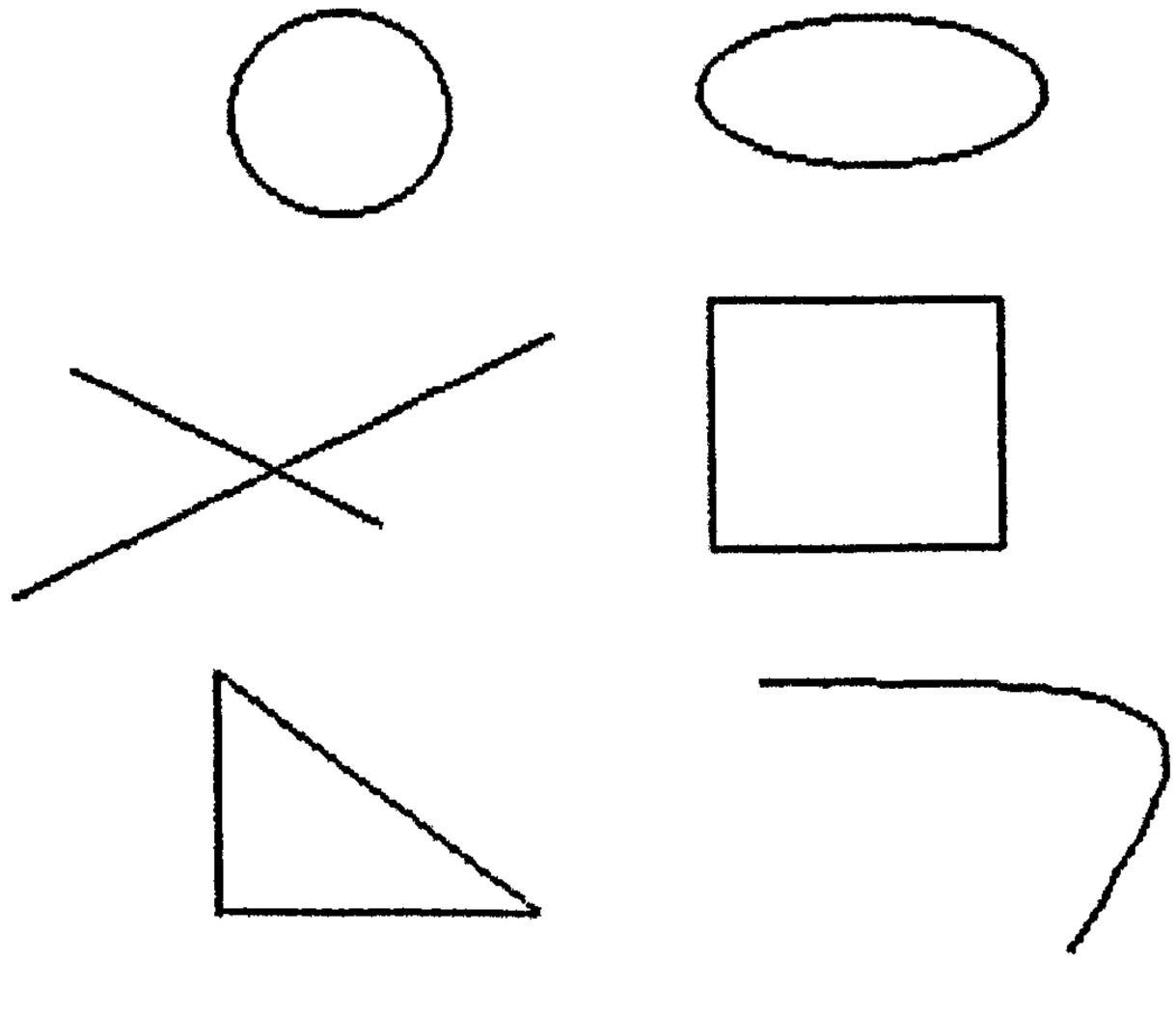


Figure 1.2: Examples of Euclidean Objects

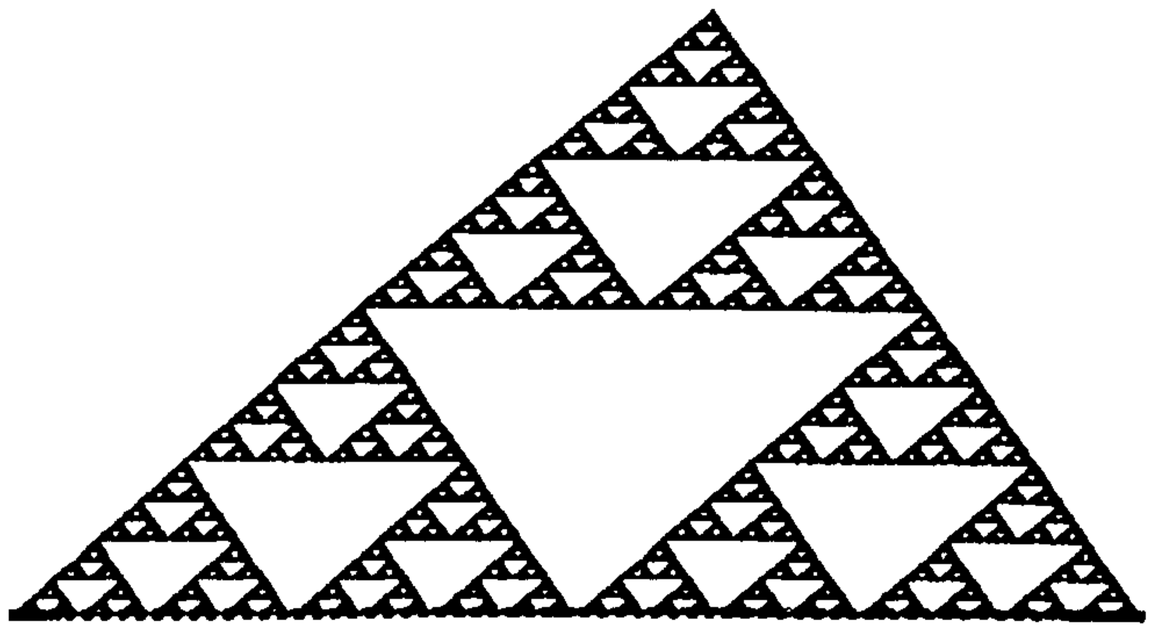


Figure 1.3: Fractal Object

phenomenon is addressed by first attempting to interpret just a small element of the phenomenon.

Geometry, with its roots in ancient Egypt and Greece, first dealt with the mathematically simplistic forms of spheres, cones, cubes etc. These exact forms, however, rarely occur naturally. A geometry suitable for describing natural objects - Fractal Geometry - was constructed this century and has only relatively recently (over the past twenty years) been researched properly [77]. This revolutionary field deals with shapes of infinite detail, such as coastlines, the branching of a river delta or nebulous forms of clouds for example and allows us to define and measure the properties of such objects. This measure is compounded in a metric called the Fractal Dimension [53].

Fractals arise in many diverse areas, from the complexity of natural phenomenon to the dynamic behaviour of nonlinear systems. Their striking wealth of detail has given them an immediate presence in our collective consciousness. Fractals are the subject of research by artists and scientists alike, making their study one of the most important activities of the late 20th century [49, 52].

A typical example of a fractal object is the the Sierpinski Triangle given in Figure 1.3.

Fractal geometry is based on the theorems and results associated with complex objects with repeating patterns - objects that are scale invariant[8, 42]. Some abstract concepts need to be introduced, e.g. repeating patterns that continue to infinity. However, unlike Euclidean geometry, the underlying philosophy is to construct an object by finding a simple underlying structure and then repeat this structure again and again. This is the basis of natural objects and systems.

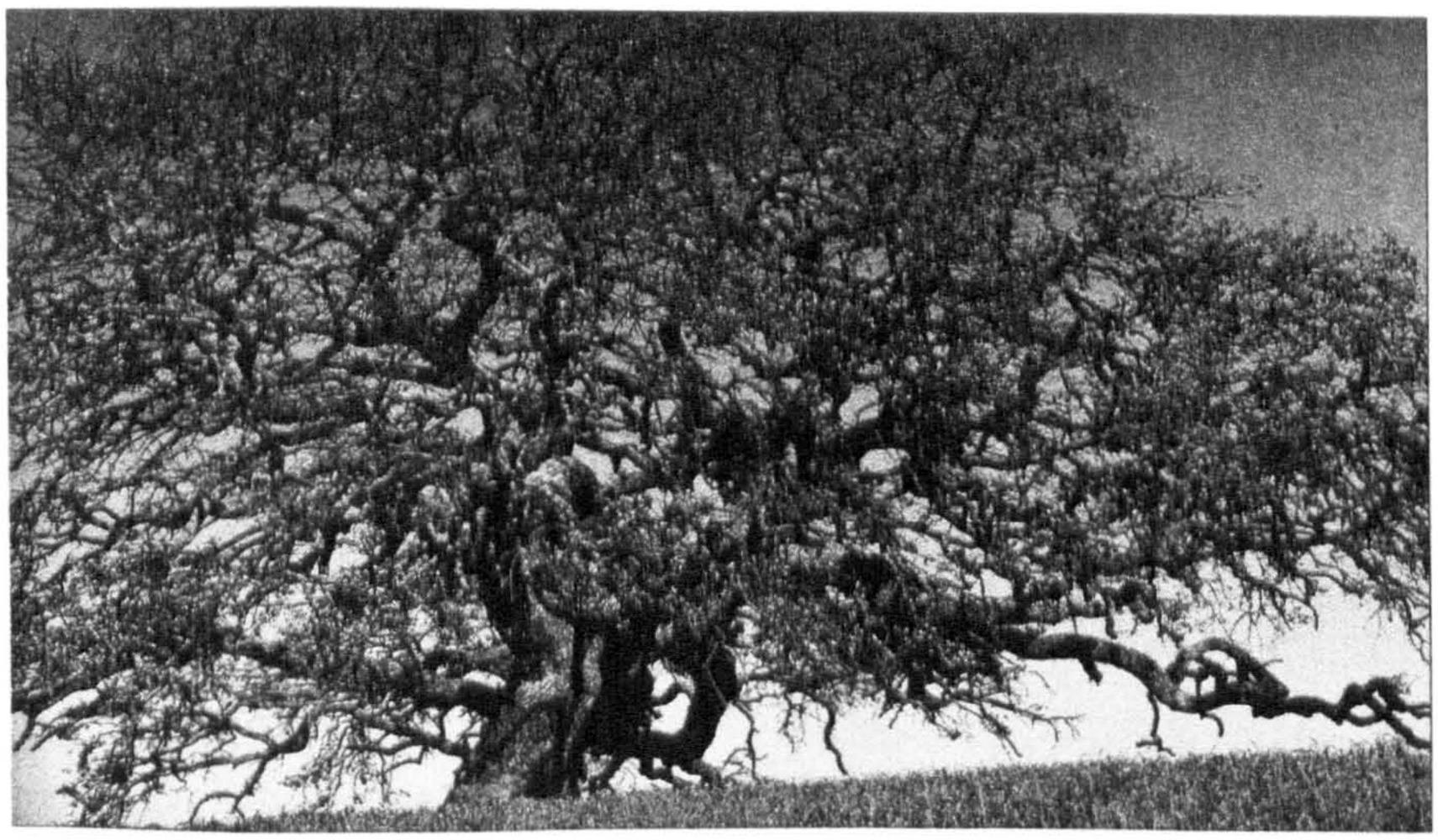


Figure 1.4: A Tree

It is also the basis for describing texture. Indeed, Mandelbrot , arguably the 'father' of fractal geoemtry, made a point that *Texture is an elusive notion which mathematicians and scientists find hard to grasp* and *Much of Fractal Geometry can be considered to be an intrinsic study of texture* The Fractal Geometry of Nature can be observed in many natural objects, some examples of which are provided below. The underlying philosophy of Fractal Geometry is concerned with the principle of self-similarity an example being given in Figure 1.7. Self-similarity over limited ranges of scale is very common in nature.

Texture has been understood and used by artists and musicians for many years. Impressionist paintings are studies of texture(Figure 1.8 and 1.9). The music of Debussy, Ravel, Scriabin and others is the study of musical texture. The painting and music of many cultures exhibits properties that are fractal. There are many examples of self-similarity in Chinese Philosophy; for example, the Chinese Proverb *In every way one can see the shape of the sea*, i.e. although very complex, the sea surface has dynamic structures which are similar at different scales and behave in a similar way with time. Further, 'pictures' of the same idea at different scales occur in terms of images of different physical systems. For example, Copernicus was the first to considered the idea that the planets orbit the sun. Kepler considered the orbit of moons about the planets. Neils Bohr was the first to consider a picture in which electrons orbit the nucleus (except for Quantum Mechanicists who know better).

The word Fractal was introduced by B Mandelbrot in the 1970s [7]. "The term

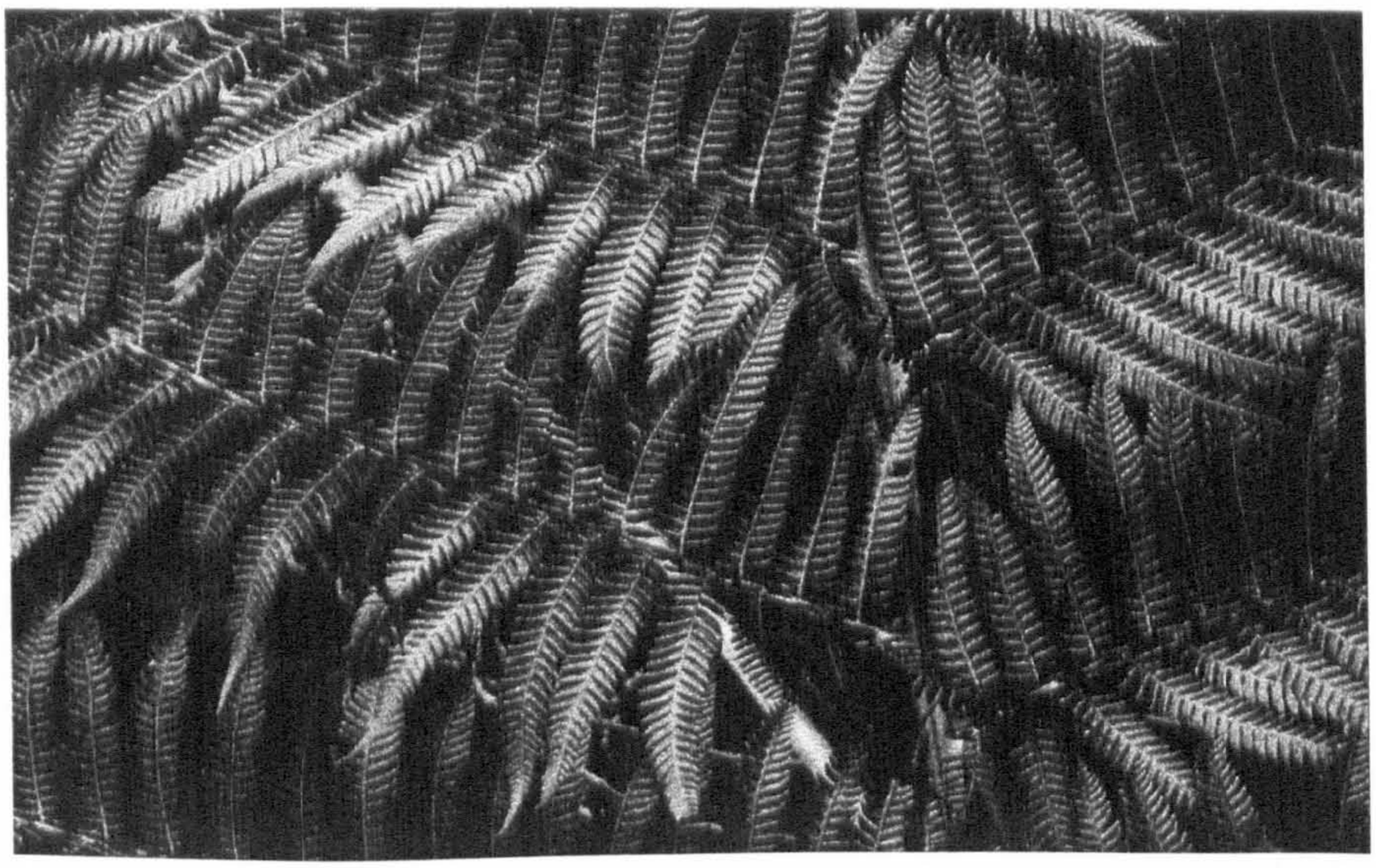


Figure 1.5: A Fern

Figure 1.6: Forrest

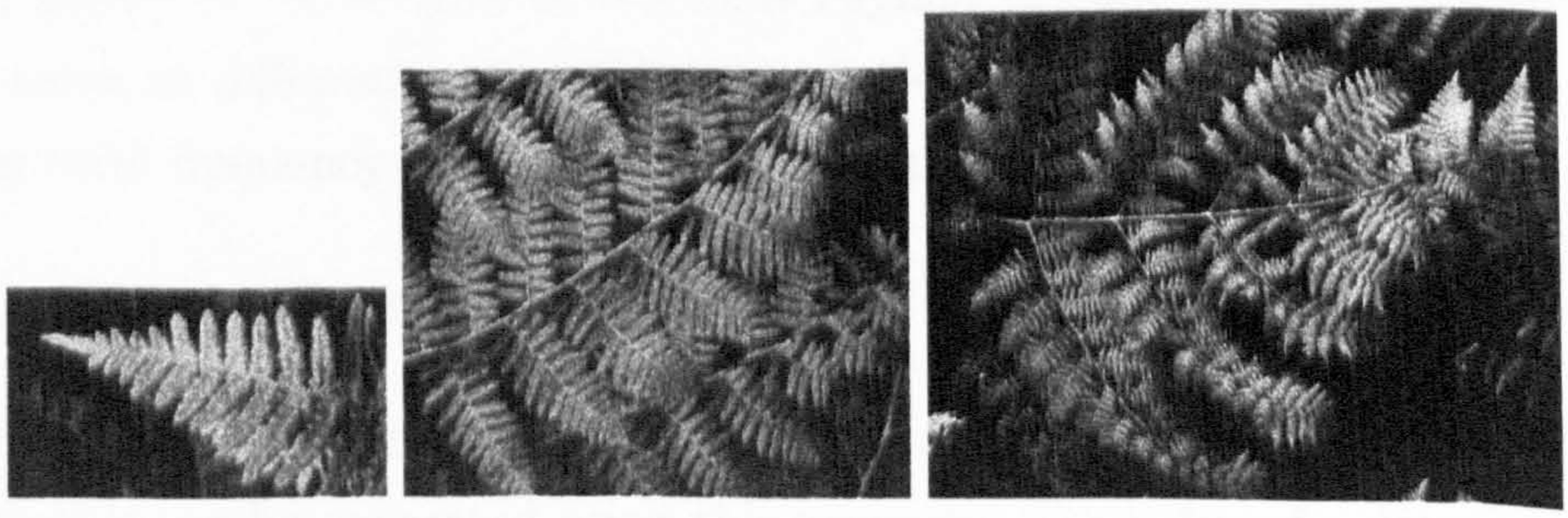


Figure 1.7: Self - Similarity

# DIAGRAM ON THIS PAGE EXCLUDED UNDER INSTRUCTION FROM THE UNIVERSITY

Figure 1.8: Islamic Art: Stylised Versions of Self-Repeating Patterns

*fractal* is derived from the Latin adjective *fractus*. The corresponding Latin verb *frangere* means 'to break', to create irregular fragments. In addition to 'fragmaneted' *fractus* should also mean 'irregular', both meanings being preseved in *fragment*". The critical states of physical systems are closely connected to the theory of fractals. Critical states occur during phase transitions. In a critical state, it is the characteristics of the interconnectivity between elements that is important and not the elements themselves. The interconnectivity is often characterised by self-similar structures. In this sense, fractal geometry is the geometry of physical systems undergoing phase transitions. Critical states are governed by the universal power law:

$$\text{System(size)} = \text{constant} \times (\text{size})^{-q} \tag{1.1}$$

where $q$ is a non-integer value. Fractals geometry is also connected to genetics. For example, the distribution of base pairs in DNA is statistically self-affine, i.e. the frequency of occurrence of Adenine-Thymin (0) and Cytosine-Guanine (1) is the same at different scales. DNA is a self-affine bit stream. A fractal has a fundamental frequency distribution or spectrum (size of detail):

$$\text{PowerSpectrum} = \text{constant} \times (\text{frequency})^{-q} \tag{1.2}$$

Any data field that has this characteristic can be analysed in terms of a fractal and fractals can be generated using this property. The value of $q$ determines the roughness of the fractal (texture) and is a non-integer.

# DIAGRAM ON THIS PAGE EXCLUDED UNDER INSTRUCTION FROM THE UNIVERSITY

Figure 1.9: Self-Similarity by M C Escher

# DIAGRAM ON THIS PAGE EXCLUDED UNDER INSTRUCTION FROM THE UNIVERSITY

Figure 1.10: Texture by Claude Monet

Figure 1.11: Texture by Jackson Pollock

Figure 1.12: Texture and Medicine

# DIAGRAM ON THIS PAGE EXCLUDED UNDER INSTRUCTION FROM THE UNIVERSITY

Figure 1.13: Self-Similarity and J S Bach

### 1.3.1 Fractals and Chaos

Fractal Geometry and Chaos are related through a variety of features:

- Chaotic systems exhibit self-similar structures when analysed or imaged in an appropriate way, i.e. an appropriate phase (e.g. a Feigenbaum map) [79, 54].
- Chaotic systems are typically the product of developing a non-linear model to describe a physical process and discretizing the result, e.g. the Verhulst process for (discrete) population ($x_i$) growth is $x_{i+1} = x_i + Rate\,(x_i - x_i^2)$ whose behaviour is quantified by the Feigenbaum diagram given in Figure 1.15.

In a sense, chaotic systems are in unstable equilibrium; even the slightest change to the initial conditions of the system at time $t$ leads the system to a very different outcome at some arbitrary later time. Such systems are said to have a sensitive dependence on initial conditions.

Some system models such as those for the motion of planets within our solar system contain many variables, and yet are still relatively accurate. With chaotic systems, however, even when there are hundreds of thousands of variables involved, no accurate prediction of their behaviour can be made. For example, the weather is known to be a chaotic system. Despite the best efforts of beleaguered meteorologists to forecast the weather, they very frequently fail, especially at lo-

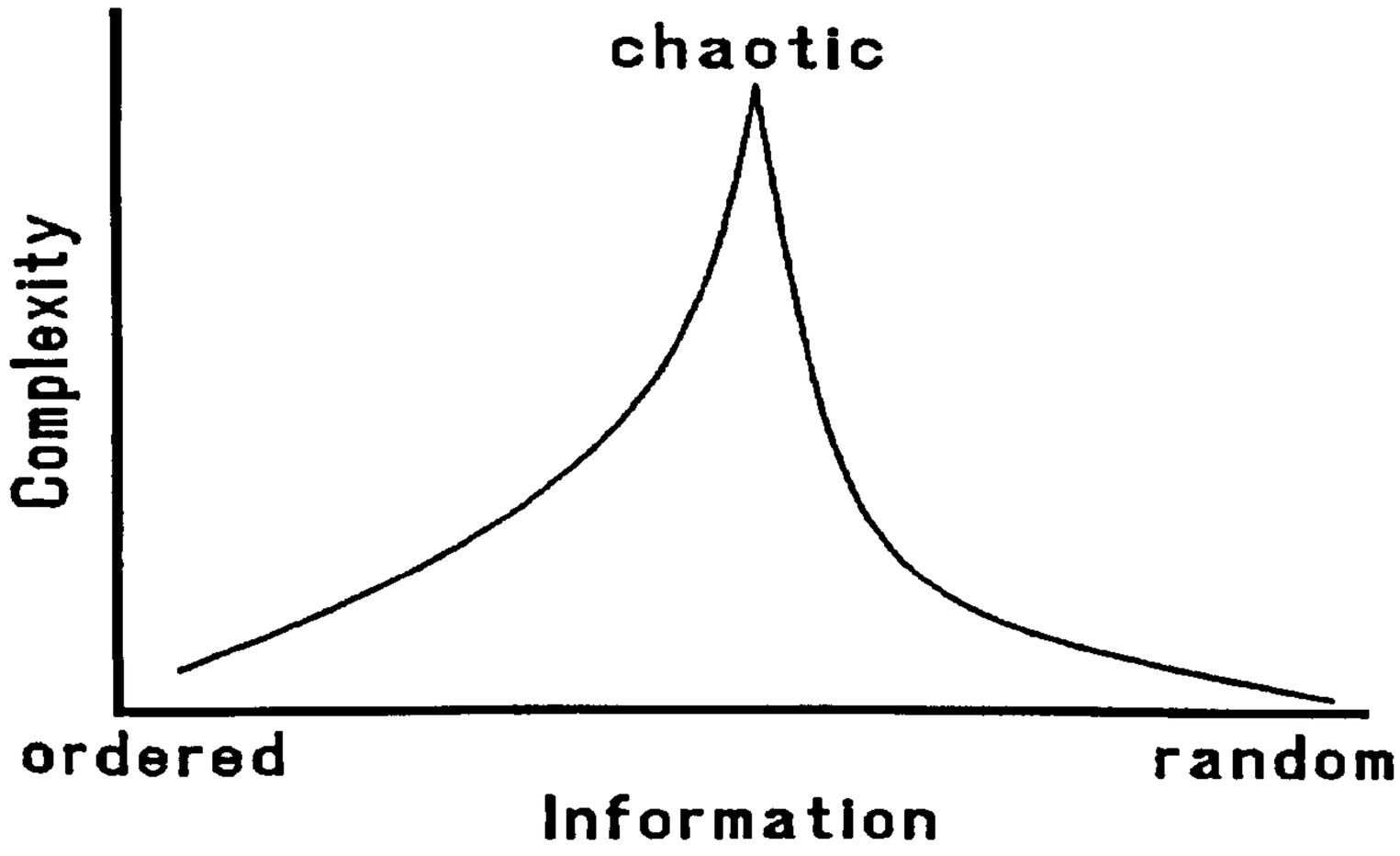



Figure 1.14: Fractals and Chaos: Complexity and Information

cal levels. There is a famous anecdote about the movement of a butterfly's wings in Tokyo affecting the weather in New York. This is typical of a chaotic system and illustrates its sensitive dependence on initial conditions [86].

Chaotic systems appear in virtually every aspect of life [79]. Traffic patterns tend to be chaotic, the erratic behaviour of even one car can create an accident or traffic jam that can affect thousands of others. The stock market is a chaotic system because the behaviour of one investor, depending on the political situation or corporation, can alter prices and supply. Politics, particularly the politics of non-democratic societies, is also chaotic in the sense that a slight change in the behaviour of a dominant individual can effect the behaviour of millions. In this sense, democracy can be defined as 'chaos limiting'. In general, chaos is the study of situations in which the slightest actions can have far-reaching repercussions.

### 1.3.2 Example of a Feigenbaum Diagram or 'Map'

By way of a short introduction to chaotic systems, let consider the properties of the Feigenbaum diagram which has become an important icon of chaos theory [97]. The diagram is a computer generated image and is necessarily so. That is to say that the details of it can not be obtained without the aid of a computer. Consequently, the mathematical properties associated with its structure would have remained elusive without the computer. This applies to the investigation of most chaotic systems whose properties are determined as much by numerical

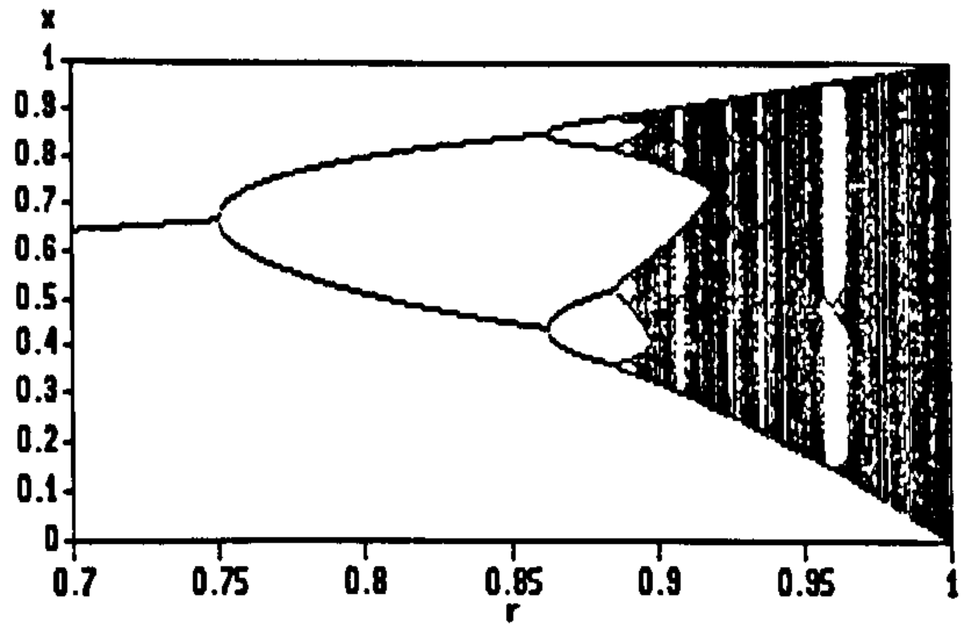



Figure 1.15: The Feigenbaum diagram illustrating routes to chaos for the Verhulst process

experimentation as they are through the rigors of functional and stability analysis.

One essential structure seen in the Feigenbaum diagram (an example of which is given in Figure 1.15) is the branching which, in this example, portrays the dynamical behaviour of the iterator $x \rightarrow ax(1-x)$. Out of the major stem, we see two branches **bifurcation**, and out of these branches we see two more and so on. This is the period-doubling regime of the iterator.

For $a = 4$, we have chaos and the points of the final state densely fill the complete interval, i.e. at $a = 4$, chaos governs the whole interval from 0 to 1 (of the dependent axis). This image is called the Feigenbaum diagram because it is intimately connected with the ground breaking work of the physicist Mitchell Feigenbaum at the Jet Propulsion Laboratory in the USA where he still works. Another point to note, is that the chaotic region for $0.9 < r < 1$ bifurcating structures are found at smaller scales (not visible in Figure 1.15) which resemble the structures shown for $0.3 < r < 0.9$. In other words, the Feigenbaum diagram (like many other 'phase space' diagrams) exhibits self-similar characteristics. This diagram is therefore an example of a fractal. In general, chaotic systems, if analysed in the appropriate phase space, are characterised by self-similar structures. Chaotic systems therefore produce fractal objects and can be analysed in terms of the fractal geometry that characterizes them.

### 1.3.3 Example of a Chaos Generator: The Verhulst Process

This model describes the development of some population, influenced by some external environment. It assumes that the population growth rate depends on the current size of the population. We first normalise the population count by introducing $x = P/N$ where $P$ denotes the current population count and $N$ is the maximum population count in a given environment. The range of $x$ is then from 0 to 1. Let us index $x$ by $n$, i.e. write $x_n$ to refer to the size of the population at time steps $n = 0, 1, 2, ...$ The growth rate is then measured by $x_{n+1}/x_n$. Verhulst postulated that the growth rate at time $n$ should be proportional to $1 - x_n$ (the fraction of the environment that is not yet used up by the population at time $n$). Thus, we can consider a population growth model based on

$$\frac{x_{n+1}}{x_n} \propto 1 - x_n$$

or after introducing a constant $a$ and rearranging the result,

$$x_{n+1} = a x_n (1 - x_n)$$

which yields the so called logistic model. Note, this is the model used to generate the Feigenbaum diagram discussed in Section 1.3.2, i.e. the iterator $x \rightarrow ax(1-x)$. Clearly, this process depends on two parameters: $x_0$ which defines the initial population size (seed value) and $a$ which is a parameter of the process. One can expect that this process (as with any conventional process that can be described by a set of algebraical or differential equations), is of three kinds:

(i) It can converge to some value x.

(ii) It can be periodic.

(iii) It can diverge and tend to infinity.

However, this is not the case. The Verhulst generator, for certain parameter values, is completely chaotic, i.e. it continues to be indefinitely irregular, irrespective of the initial condition used. This behaviour is compounded in the Feigenbaum diagram (Figure 1.15) and is due to the nonlinearity of the iterator. In general, we can define four classes of behaviour depending on value of parameter $r$.

(i) $0 < r < R_1$: the process converges to some value $p$.

(ii) $R_1 < r < R_2$: the process is period.

(iii) $R_2 < r < R_3$: the process is chaotic.

(iv) $R_3 < r < 0$: the process tends to infinity.

The specific values of $R_1, R_2$ and $R_3$ depends on the seed value, but the general pattern remains the same. The region $R_2 < r < R_3$ can be used for pseudo chaotic number generation.

Another feature of this process is its sensitivity to the initial conditions. This effect is one of the central ingredients of what is called deterministic chaos. The main idea here is that any (however small) change in the initial conditions leads, after many iterations, to a completely different resulting processes. In this sense, we cannot predict the development of this process at all due the impossibility of infinitely exact computations. However, we need to strictly determine the rounding rules which are used in generating a chaotic sequence in order to receive the same results on different systems.

Many other chaos generators exist. In most cases they are compounded by iterative processes which are inherently nonlinear. This is not to say that all nonlinear processes produce chaos, but that chaotic processes are usually a result of nonlinear systems.

### 1.3.4 Definition of a Fractal

Unfortunately, a good and general definition of a fractal is elusive. Any particular definition either excludes sets that are thought of as fractals or includes sets that are not thought of as fractals. The definition of a 'fractal' should be regarded in the same way as the biologist regards the definition of 'life'. There is no hard and fast definition, but just a list of properties and characteristics of a living thing [29, 41, 53, 59]. In the same way, it seems best to regard a fractal as a set that has properties such as those listed below, rather than to look for a precise definition which will almost certainly exclude some interesting cases.

If we consider a set $F$ to be a fractal, then it should possess at least some of the following properties:

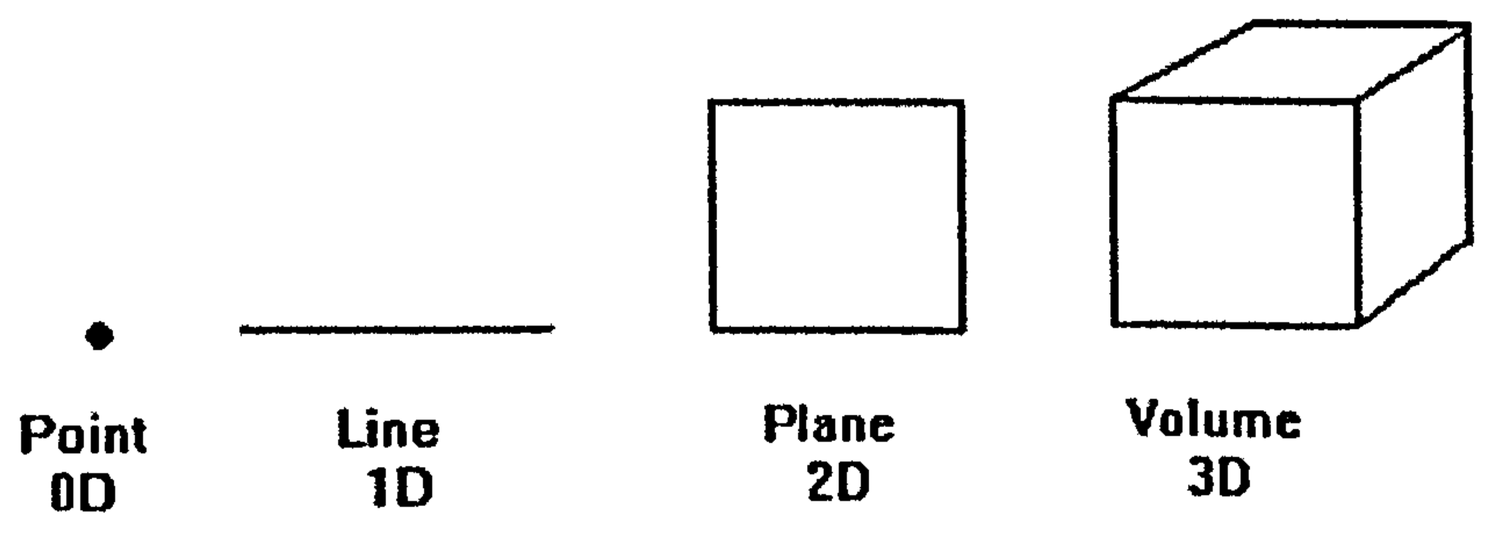



Figure 1.16: Points, Lines, Planes, Volumes and Common (Integer) Dimensions

(i) $F$ has detail at every scale.

(ii) $F$ is (exactly, approximately, or statistically) self-similar.

(iii) The 'Fractal Dimension' of $F$ is greater than its topological dimension.

### 1.3.5 The Similarity or Fractal Dimension

Central to fractal geometry is the concept of self-similarity, which means that some types of mainly naturally occurring objects look similar at different scales [41, 29, 8]. We are all used to the concept of dimensions 1,2 and 3. The 4th dimension or time is also now commonly accepted thanks to the work of Albert Einstein in the early twentieth century [3]. Higher dimensions, i.e. 5,6,7,8, are abstractions but nevertheless of fundamental significance in modern theoretical physics, e.g. string theory. Just as we cannot imagine or visualise higher integer dimensional spaces does not mean that these spaces do not exist. Humans are necessarily limited in there ability of imagine higher dimensional spaces because of the limitations of their natural senses - in particular, vision. Coming to terms with 3 and 4 dimensions has taken its time. Further, the limited way in which limited senses inform the human mind, necessarily places a limit on the way in which the same mind can interpret the world (including the work of other); the philosophical basis of the work of Emmanual Kant in the 17th century. Representing dimension in art has been the focus of the develop of painting over many years. This development is illustrated in Figures 1.17, 1.18, 1.19 and 1.20.

# DIAGRAM ON THIS PAGE EXCLUDED UNDER INSTRUCTION FROM THE UNIVERSITY

Figure 1.17: Medievil pre-renaissance art: 2D - flat paintings with distortions in natural perspective

Figure 1.18: Renaissance art: 3D - coming to terms with perspective in paintings and taking on three dimensional form (i.e. re-birth of Greek/Roman concepts and philosophy)

# DIAGRAM ON THIS PAGE EXCLUDED UNDER INSTRUCTION FROM THE UNIVERSITY

Figure 1.19: Cubist art: trying to express 4D in paintings

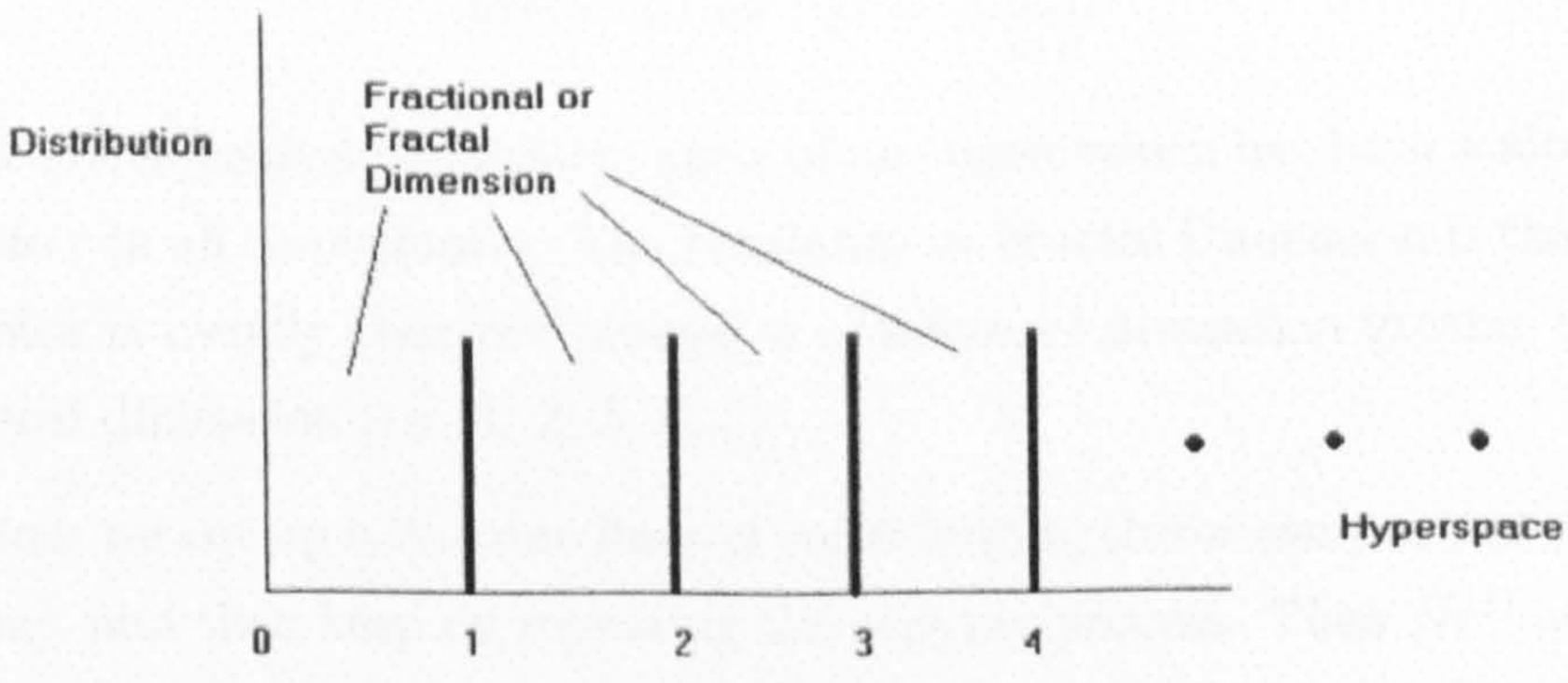



Figure 1.20: Computer graphics: attempts being made to represent hyper-space

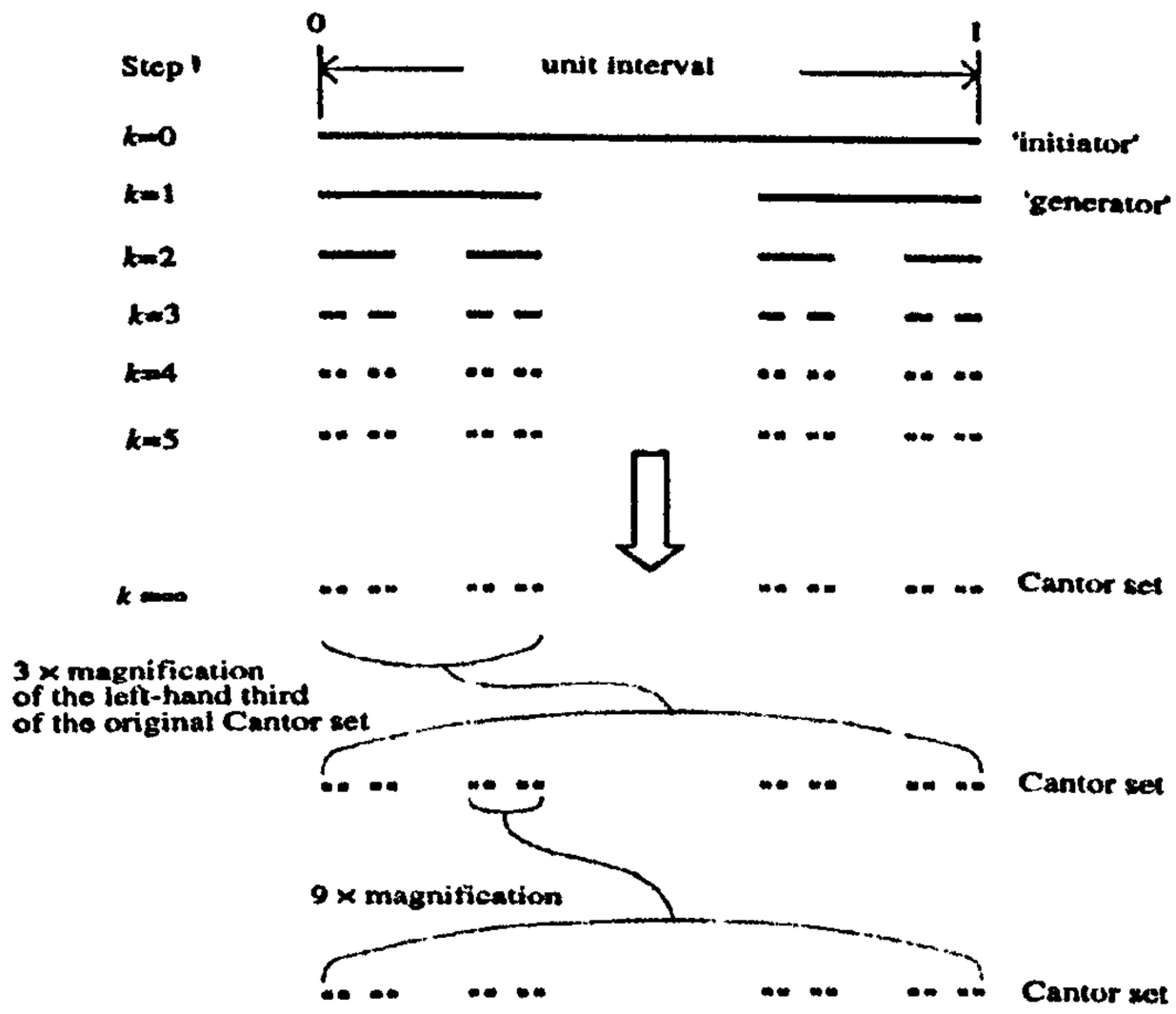



Figure 1.21: A Cantor Set

Self-similar objects are compounded by a parameter called the 'Similarity Dimension' or the 'Fractal Dimension', $D$ [8]. Suppose we cut up some simple Euclidean objects, make exact copies of them and then keep on repeating the copying process. Let N be the number of copies we make at each stage and let $r$ be the length of each of the copies - the scaling ratio. Then $Nr^D = 1$ where D=1, 2, 3. Thus, we arrive at the definition

$$Nr^D = 1 \quad \text{or} \quad D = -\frac{\ln N}{\ln r} \tag{1.3}$$

where $N$ is the number of distinct copies of an object which has been scaled down by a ratio $r$ in all co-ordinates. The Similarity or Fractal Dimension is that value of D which is usually (but not always) a non-integer dimension greater than its topological dimension (i.e. 1, 2, 3, 4,...).

Suppose we cut up a line into lines of equal length, throw away one of the lines (deletion), and then keep on repeating the copying process. Then $Nr^D = 1$ with $0 < D < 1$. This process produces, for example, a Cantor set or Fractal Dust (see Figure 1.21)[39, 69, 65, 70]. $D$ is a fractional dimension - the Fractal or Similarity Dimension.

The number of copies N made each time the copying process is repeated is 2. Scaling ratio r is 1/3. To compute $D$, noting that $Nr^D = 1$, we take logarithms

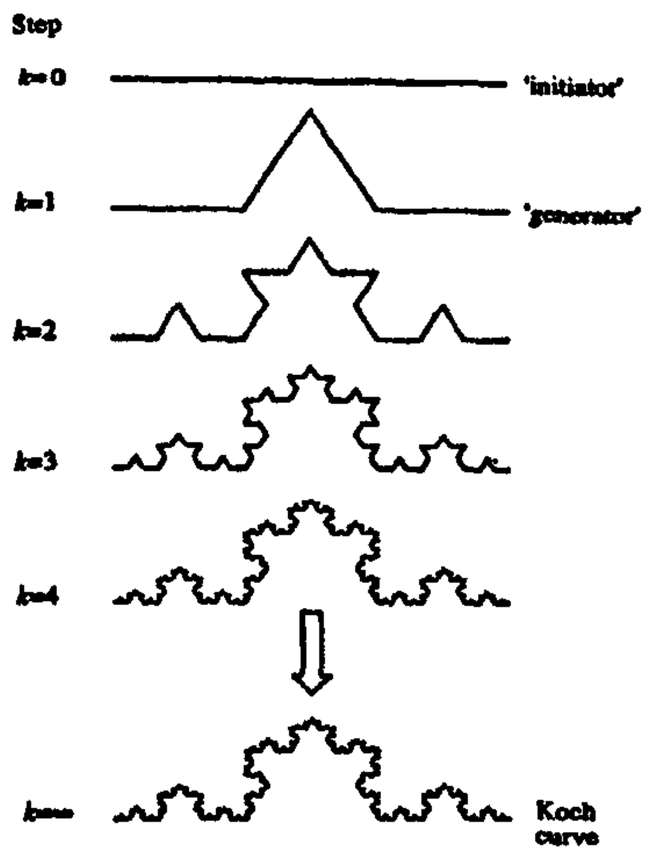



Figure 1.22: The Triadic von Koch Curve

giving

$$D = -\frac{\log(N)}{\log(r)} \tag{1.4}$$

For this Fractal Dust:

$$D = -\log(2)/\log(1/3) = \log(2)/\log(3) = 0.6309297536\ldots$$

The lower the value of $D$, the less complex the fractal dust is until for $D = 0$ we just get a single point.

Suppose we now cut up a line into lines of equal length, add one (or a number) of the lines (addition), and then keep on repeating the copying process. Then $Nr^D = 1$ with $1 < D < 2$. This process produces fractal curves of which there are many types, e.g. the von Koch curve given in Figure 1.22. Here, $N = 4$ and $r = 1/3$ at each step and $D = \log(4)/\log(3) = 1.261859507....$

Consider another, where $N = 8$ and $r = 1/4$ at each step. Then $D = \log(8)/\log(4) = 1.5$. Observe that the complexity or texture of this quadratic Koch curve (Figure 1.23) is greater than that of the triadic - a result that is compounded in the increased value of $D$.

The Sierpinski Triangle (Figure 1.24) has $N = 3$, $r = 1/2$ at each step and $D = \log(3)/\log(2) = 1.584962501....$

The Sierpinski Carpet (Figure 1.25) has $N = 8$, $r = 1/3$ at each step and $D = \log(8)/\log(3) = 1.892789261....$

Now suppose we cut up a cube into cubes of equal shape, delete one (or a number) of the shapes (deletion), and then keep on repeating the copying process.

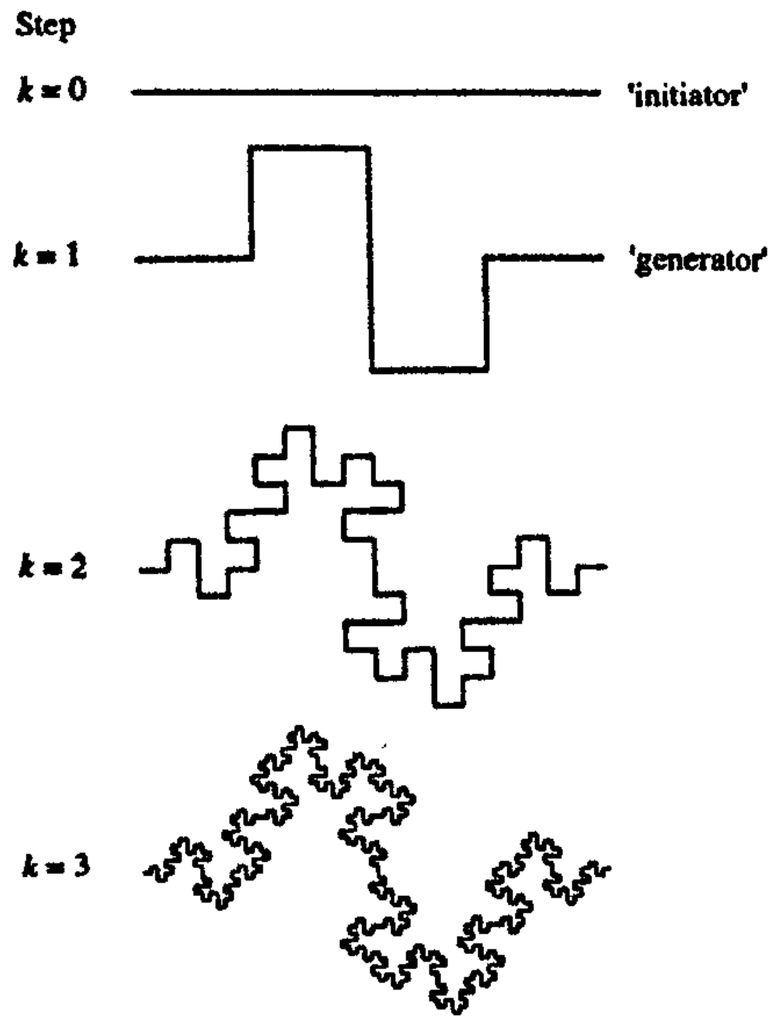
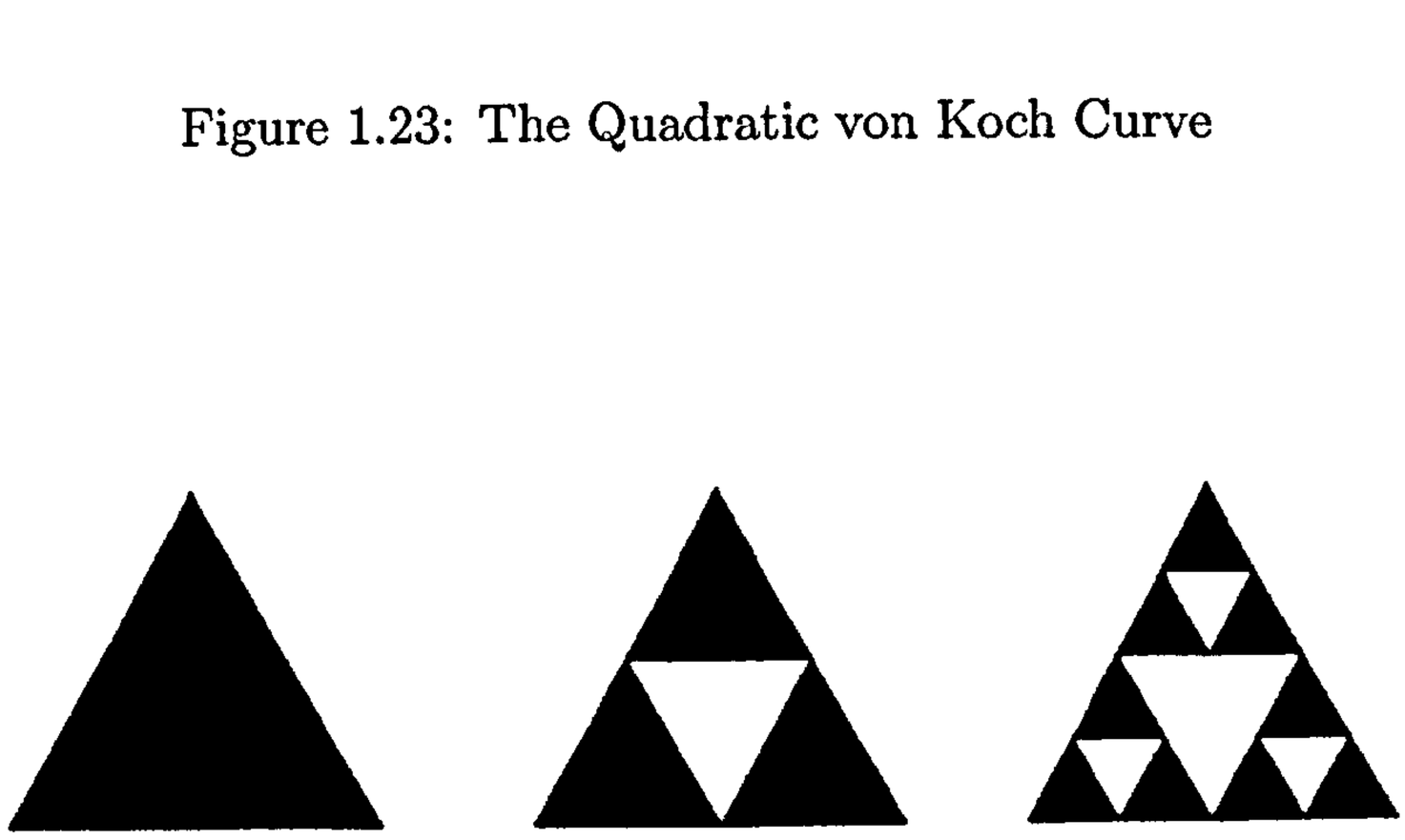



Figure 1.23: The Quadratic von Koch Curve



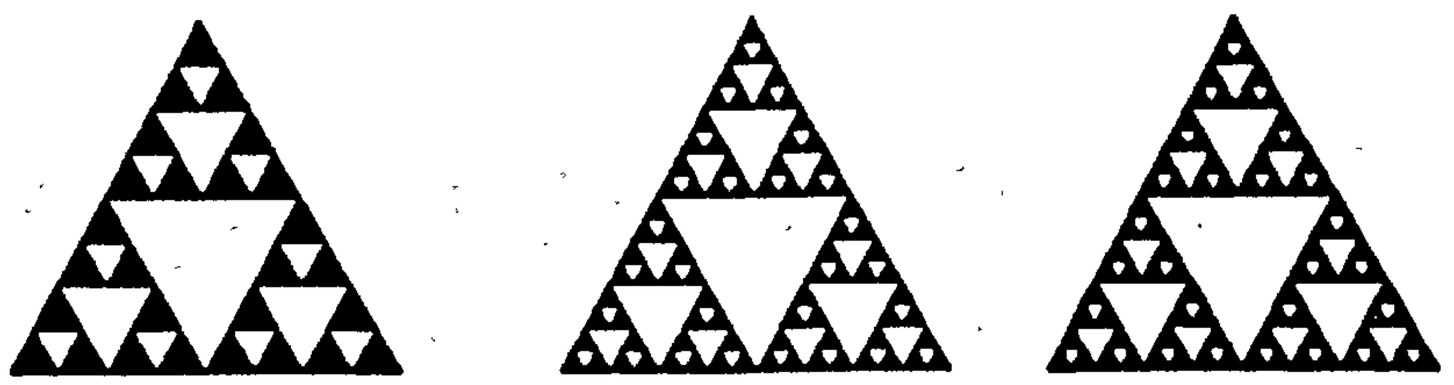




Figure 1.24: The Sierpinski Triangle

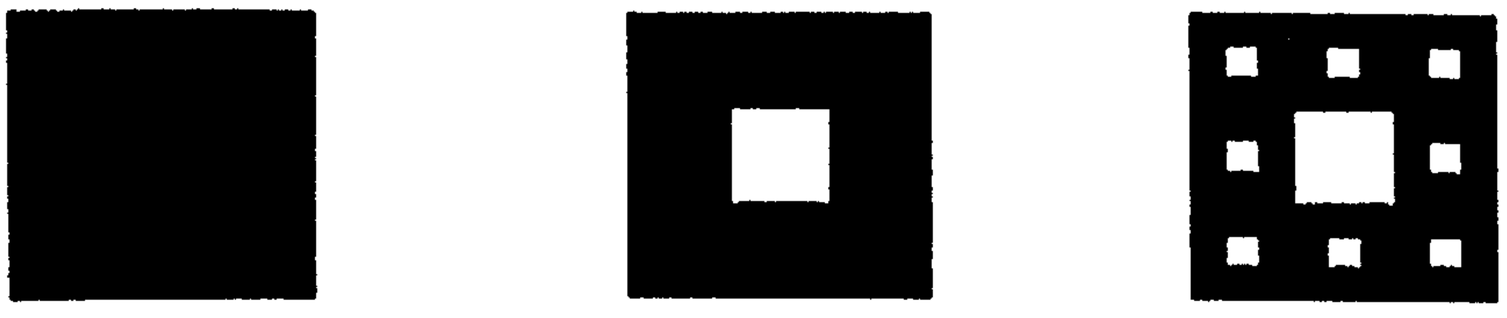



Figure 1.25: The Sierpinski Carpet

Then $Nr^D = 1$ with $2 < D < 3$. This process produces fractal surfaces (see Figure 1.26) such as the Menger sponge - an object with infinite surface area and zero volume where $N = 20$, $r = 1/3$ and $D = \log(20)/\log(3) = 2.726833028...$ These ideas can be repeated with an infinite variety of 'variations on a theme' and lead to fractal types which are compounded in the following table.

The basic equations describing Self-Similarity and Self-Affinity of Fractal Functions $f(x)$ are as follows:

Self-Similarity: Change the scale of a function $f(x)$ by a factor $\lambda$ say and the

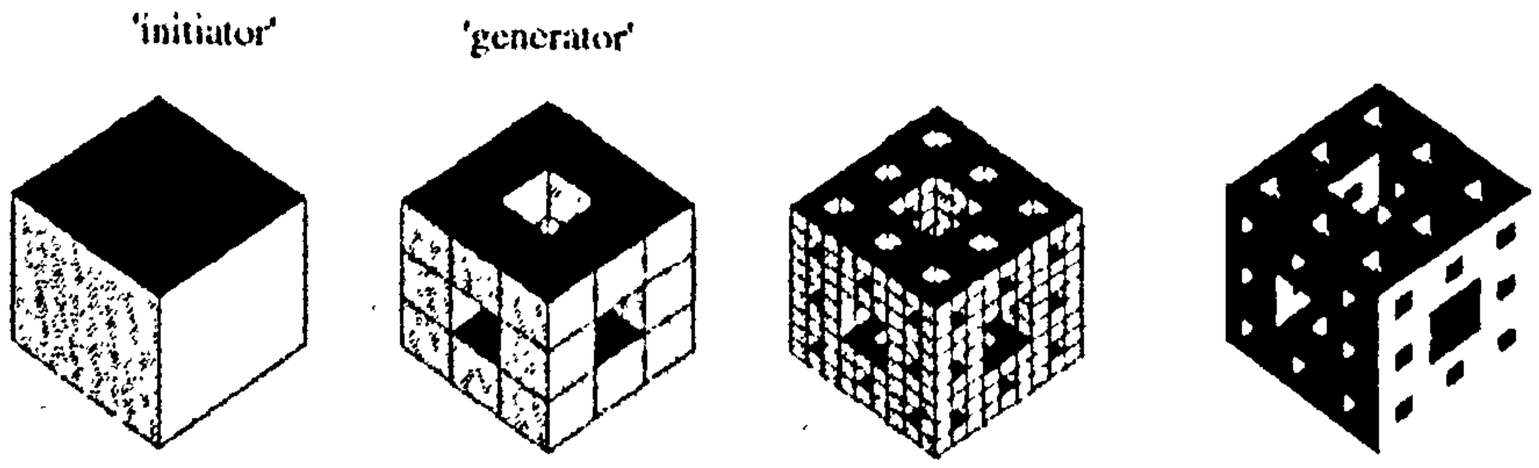



Figure 1.26: The Menger Sponge

Table 1.1: Types of Fractals and Their Fractal Dimension

| Fractal Type | Fractal Dimension |
|---|---|
| **Fractal Dust** | $0 < D < 1$ |
| **Fractal Curves** | $1 < D < 2$ |
| **Fractal Surfaces** | $2 < D < 3$ |
| **Fractal Volumes** | $3 < D < 4$ |
| **Fractal Time** | $4 < D < 5$ |

result is a smaller version, reduced in size by $\lambda$

$$f(\lambda x) = \lambda f(x)$$

Self-Affinity: Change the scale of a function $f(x)$ by a factor $\lambda$ say and the result is a smaller version, reduced in size by $\lambda^q$

$$f(\lambda x) = \lambda^q f(x)$$

where $q > 0$ is an arbitrary constant.

There is a close relationship between fractal objects and fractal functions or signals. Many regular fractal objects or 'curves' can be generated with a specified fractal dimension. These regular fractal curves are not functions of the type $f(x)$. The associated fractal function $f(x)$ can be generated through parametrisation, e.g. chord length parametrisation. This is illustrated in the figures that follow

The frequency spectrum $F$ of a fractal signal is proportional to $(Frequency)^{-q}$ or

$$F = \frac{c}{k^q}$$

where $c$ is a constant and $k$ denotes the (spatial) frequency. Taking logs of both side, we get

$$log(F) = C - qlog(k)$$

where $C = log(c)$ This equation describes a straight line with a negative gradient determined by the value of $q$ (Figure 1.29).

The parameter $q$ is known as the Fourier Dimension. The relation between $D$ and $q$ for fractal functions is given by $q = \frac{(5-2D)}{2}$ and range $1 < D < 2$. The Fourier Dimension determines the roughness, texture or frequency characteristics or spectrum of the fractal. There are two distinct types of fractals which exhibit

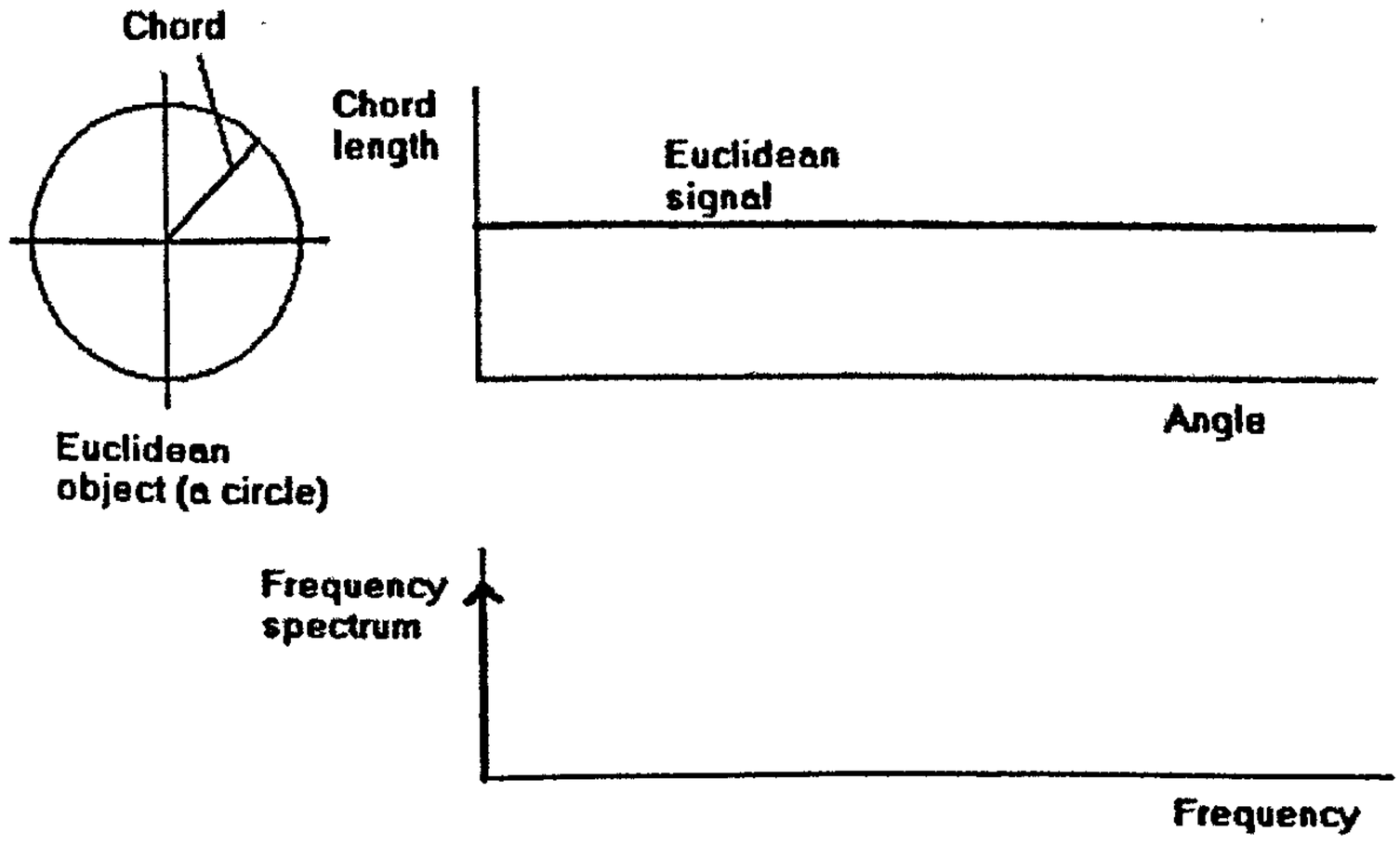



Figure 1.27: Euclidean Objects and Euclidean Signals After Parametrization

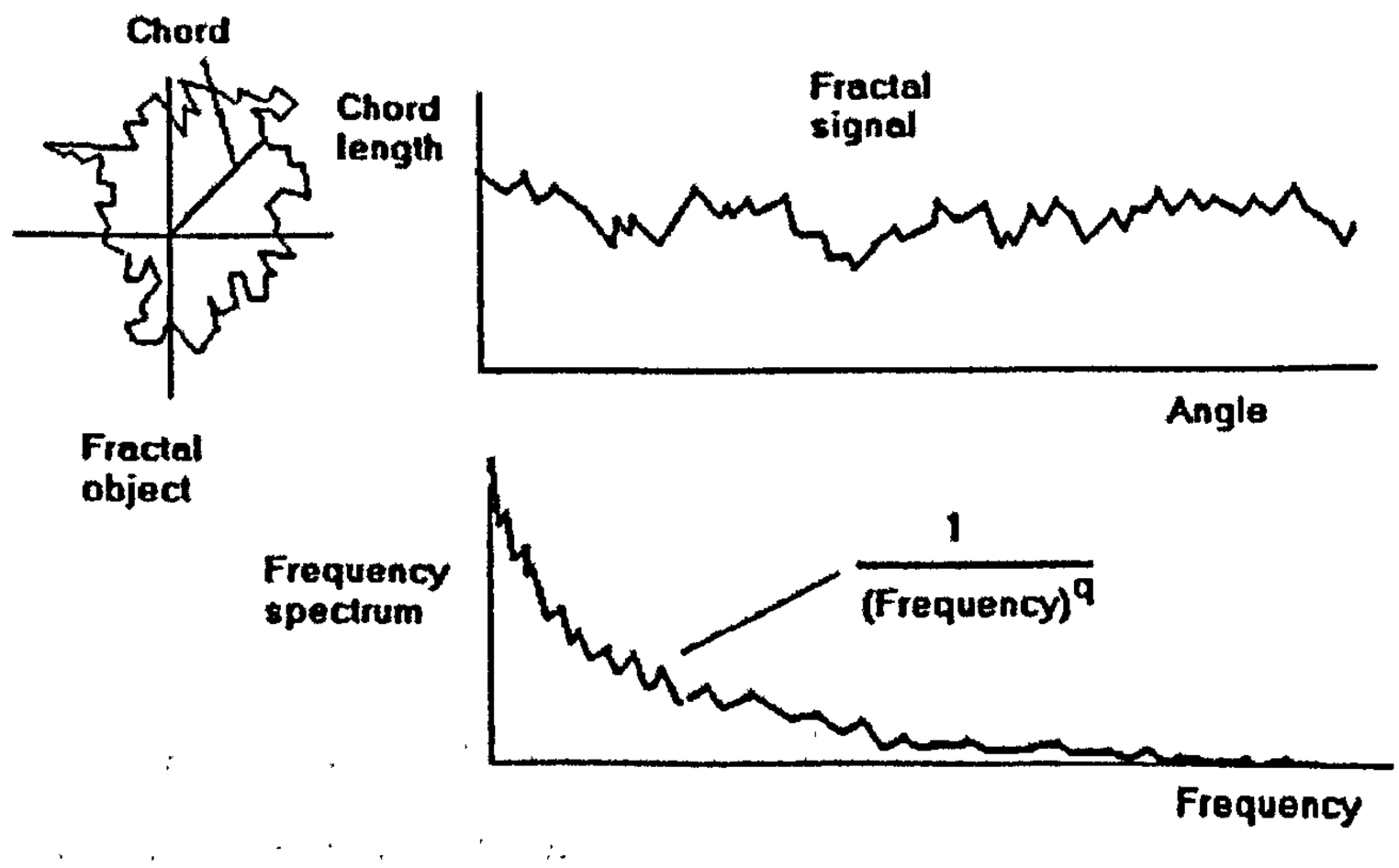



Figure 1.28: Fractal Objects and Fractal Signals After Parametrization

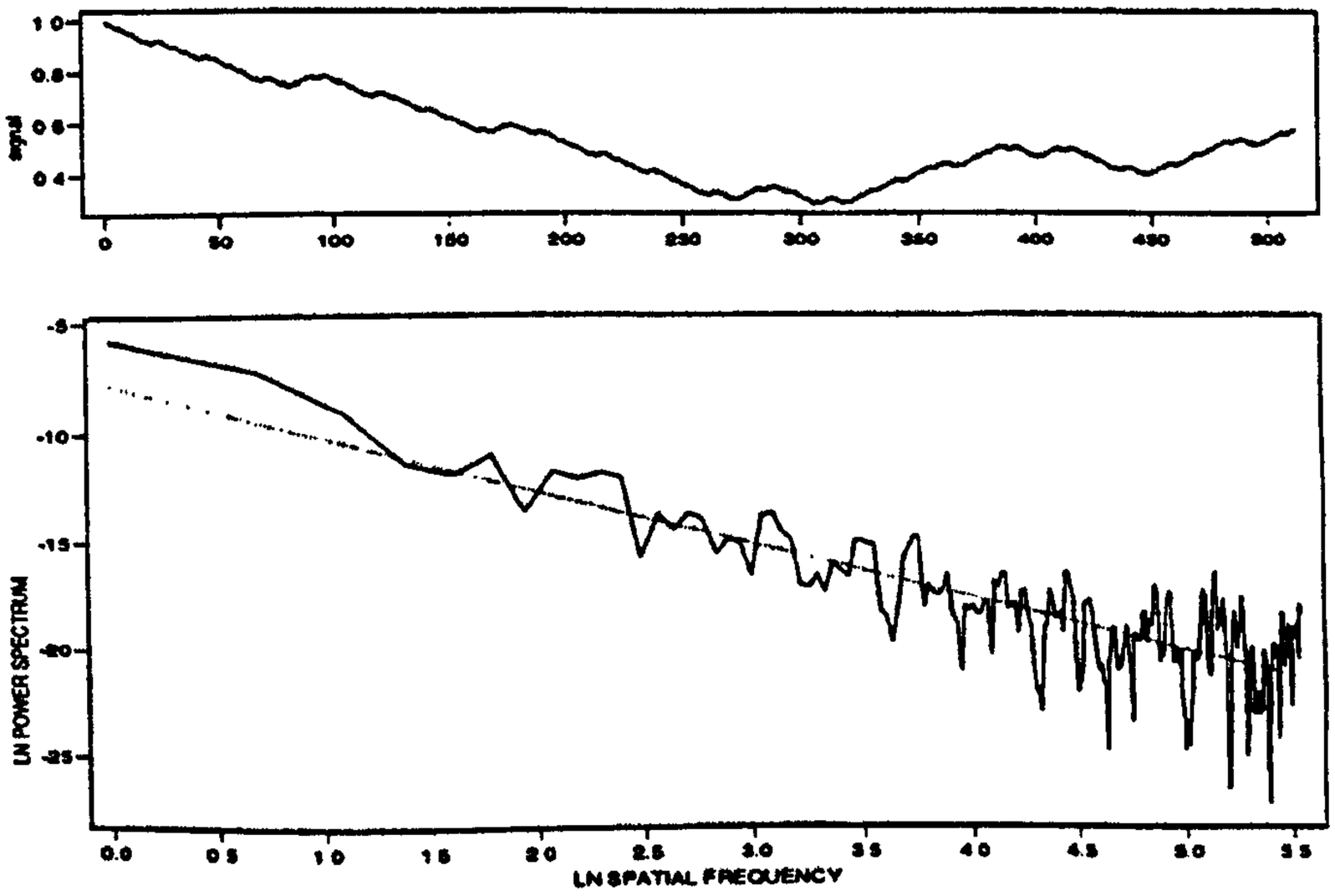



Figure 1.29: Fractal Signal Obtained From Koch Curve and its Log-Log Frequency Spectrum

this property: (i) Deterministic Fractals; (ii) Random Fractals. Deterministic fractals are objects which look identical at all scales. Each magnification reveals an ever finer structure which is an exact replica of the whole, i.e. they are exactly self-similar. Random fractals do not, in general, possess such deterministic self-similarity; such fractal sets are composed of $N$ distinct subsets, each of which is scaled down by a ratio $r$ from the original and is identical in all statistical respects to the scaled original - they are statistically self-similar. The scaling ratios need not be the same for all scaled down copies. Certain fractal sets are composed of the union of $N$ distinct subsets, each of which is scaled down by a ratio $r_i < 1, \quad 1 \leq i \leq N$ from the original in all co-ordinates. The similarity dimension is given by the generalisation,

$$\sum_{i=1}^{N} r_i^D = 1$$

A further generalisation leads to self-affine fractals sets which are scaled by different ratios in different co-ordinates. The equation

$$f(\lambda x) = \lambda^q f(x) \quad \forall \lambda > 0$$

where $\lambda$ is a scaling factor and $q$ is a scaling exponent implies that a scaling in the $x$ co-ordinate by $\lambda$ gives a scaling of the $f$ co-ordinate by a factor $\lambda^q$. A

special case of this equation occurs when $q = 1$; in this case, we have a scaling of $x$ by $\lambda$ producing a scaling of $f$ by $\lambda$, i.e. $f(x)$ is self-similar.

Naturally occurring fractals differ from strictly mathematically defined fractals in that they do not display statistical or exact self-similarity over all scales but exhibit fractal properties over a limited range of scales.

### 1.3.6 Random Fractals

There are many examples in the field of physics, chemistry and biology of random processes [89, 39]. Brownian motion is a relevant mathematical model for many such physical processes. These processes display properties which have now been shown to be best described by fractal processes.

In Brownian motion, the position of a particle at one time is not independent of the particles motion at a previous time. It is the increments of the position that are independent. Brownian motion in 1D is seen as a particle moving backwards and forwards on the x-axis for example. If we record the particles position on the x-axis at equally spaced time intervals, then we end up with a set of points on a line. Such a point-set is self-similar. On the other hand, if we include time as an extra co-ordinate and plot the particles position against time - called the record of the motion - we obtain a point- set that is self-affine.

To conclude this chapter let us consider some general themes. Fractals are objects in which one part resembles the whole. This is the principle of self-similarity. Many objects in nature are fractal. Fractal structures occur in systems undergoing phase transitions. Fractal structures are characterised by a frequency ($k$) distribution proportional to $k^{-q}$. Chaotic processes exhibit self-similar structures in an appropriate phase space. Regular or deterministic fractals are the same at smaller scales. Random fractals are statistically the same at smaller scales - they are statistically self-affine, i.e.

$$Pr[f(\lambda x)] = \lambda^q Pr[f(x)], \text{ where } q > 0$$

Functions with this property are governed by a spectral inverse $q$-power law. The spectral inverse $q$-power law is found throughout nature. This power law is as fundamental as many other laws of nature, e.g. the inverse square law for gravitational and electrostatic fields. Like gravitational fields for example, the

reasons why the geometry of nature is fractal is not as yet, understood. With regard to the basic law of fractals, it is therefore worth considering Newton's remark with respect to his universal law of gravitation ... *I have told you how it works, not why it works.*

## 1.4 Original Contribution

A number of problems are connected to the calculation of the Fractal dimension for two dimension objects or images. One of them is the computation of the power spectrum for images with a restricted resolution. The Fourier method for computing the fractal dimension works very well when the number of points exceeds 512 x 512 pixels for example. At smaller resolutions, the surface of image does not correspond to the sinusoidal form of a function, which is utilised in a convolution [55, 84]. For the solution of this problem with respect to the applications considered in this thesis, I utilize the triangular form of a function. Application of this solution provides an increase in the accuracy of the calculation for the frequency characteristics of a low resolution image including its fractal dimension.

The second innovation discussed in this work is connected with one of the steps used to search the location of an object edges [68]. The solution is directly related to the applications problem of cell recognition and the difference between normal and abnormal cells in medicine. After obtaining an image from a digital microscope some core features can have a different gradient rate on different sides of the object. This does not allow for the utility of existing methods for boundary detection. Further some images have diffused due to overlay and the addition of foreign ajdents (bacteria for example). The special filter designed to solve this problem provides a method of isolating and dividing the core cells of an image.

In image processing, there are a number of simple operations that can be applied such as stationary filtering and Euclidean geometric analysis of shape [44, 99, 82]. The result of considering existing image processing techniques (based on conventional filters) does not necesserily provide a universal solution for the purpose of selecting the boundary of an object and deriving the convex hull properties of the object. In this thesis, we offer a new approach to the solution of this problem. The method offered provides a completely general solution that can

be utilized in a range of other image processing and pattern recognition problems.

## 1.5 About this Thesis

The thesis is composed of seven chapters. Chapter 2 presents a review of fundamental methods for image formation and transformation. The chapter discussed in this thesis is based exclusively on the use of incoherent images (both grey level and colour) and is concerned with an evaluation of the 'physics' of image formation together with methods used for image filtering (including image restoration and reconstruction), some of which are used later on in the thesis for defining an image model and reconstructing image features (e.g. the Wiener filter). Chapter 3 focuses on image processing - edge detection, segmentation, object location and feature detection. Attention on the theory of scaling fractals in which the connection between fractal texture analysis and the mathematical description of fractal geometry is considered. This chapter includes a number of different methods for computing the fractal dimension that are used later on in the work. Chapter 4 discusses new approaches to the problem of image segmentation for cells detection, using fractal geometry known as fractal dimension for accurate textural characteristic and three two dimensional algorithms for object processing. Chapter 5 discusses the principal techniques for image analysis and recognition used in this thesis. It formulates the problem, introduces the technical solutions developed specifically for this thesis and gives a brief review of other competing recognition systems currently under investigation. Chapter 6 discusses the applications that have been considered for this thesis including the development of a surface inspection system for non-destructive evaluation, a cervical smear analysis system and finally the development of system prototype for the analysis of skin cancer. Finally, Chapter 7 provides a short conclusion to the thesis and provides some ideas for future work and further developments of the application systems designed to date. Some further technical aspects of the work including the software engineering associated with the systems produced are given in the Appendices.

# Chapter 2

# Image Formation and Basic Image Transformations

## 2.1 Introduction

This chapter introduces some image formation techniques and gives introducing to a reader about aspects of representing the graphic information. This material considers physical and mathematical basement of imaging, theoretical and practical approaches of transformation of a storage and transmission of colour graphics images.

## 2.2 Image Functions and Colour Models

The first view of a human eye provides an image similar to a continues surface without any distortion. However, with a simple lens system, it is possible for the eye to view details in the image plane through an array of points. In a human eye, these points are organized into a photosensitive matrix or array where each point can be enumerated in terms of coordinates $x$ and $y$. The value of each isolated point can be represented by value of a function $I$ with coordinates $x$ and $y$. Here, $x$ can be taken to represent the horizontal axis and $y$ the vertical axis. Video information is stored in the same way, i.e. in terms of the function $I(x, y)$ [87, 88]. The colour content(s) of an image is very important and contributes significantly to the image processing operations required and the object recognition methodologies applied [24]. In the case of medical imaging in general,

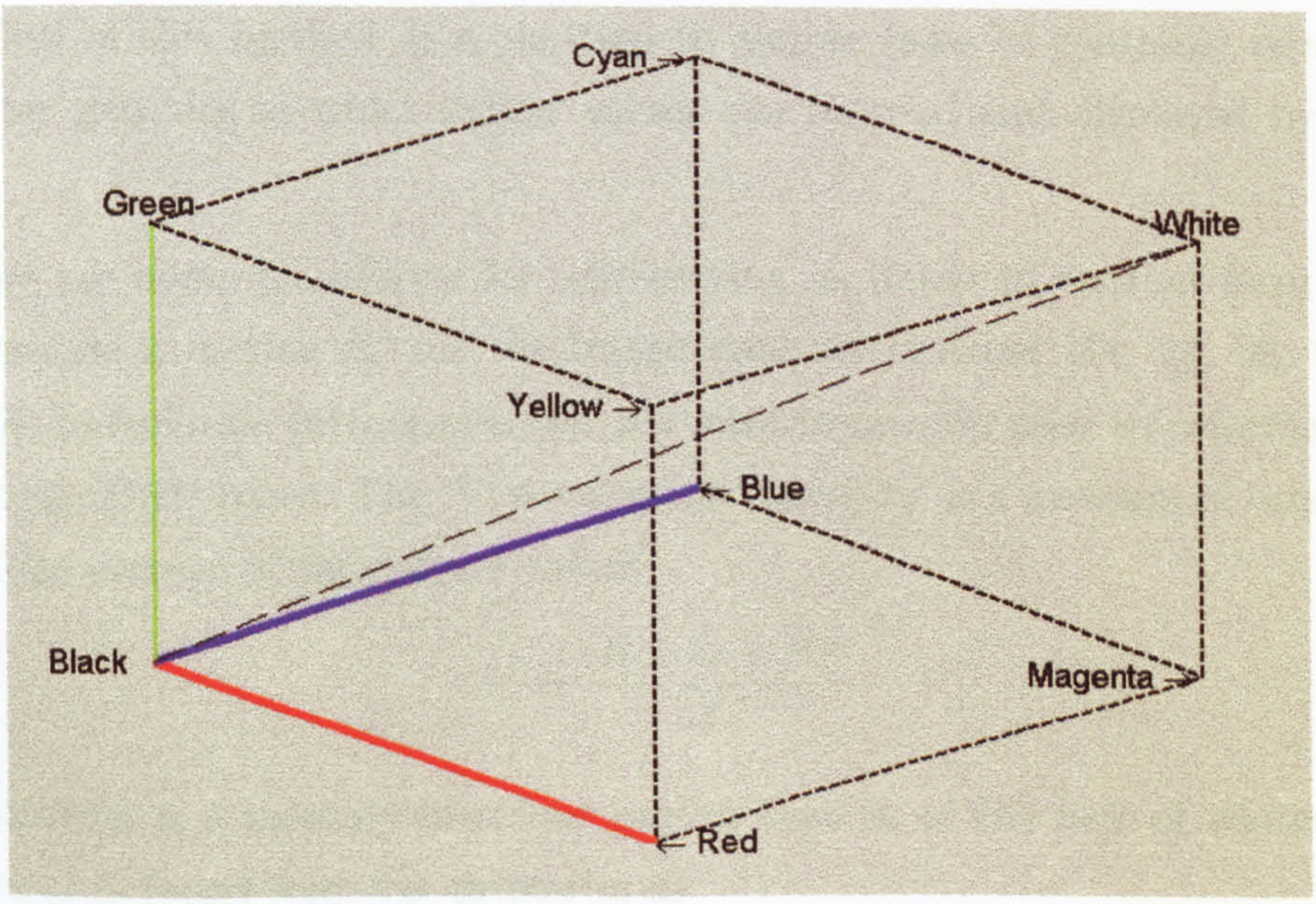



Figure 2.1: $RGB$ colour cube

colour processing and colour interpretation is critical to the diagnosis of many conditions and the interpretation of the information content of an image by man and machine. Colour image processing is become more and more important in object analysis and pattern recognition. This includes for example, a new class of edge detection filters which are discussed later on in this thesis which are based exclusively on the use of colour contrasts. For this reason, understanding image colour theory is a prerequisite and is therefore briefly considered as follows.

Let us consider the general image representation in terms of the field of machine vision field. A colour image $I_C(x, y)$ has the $RGB$ representation

$$I_C(x, y) = [I_R(x, y), I_G(x, y), I_B(x, y)]$$

where the colour primaries are Red (R), Green (G) and Blue (B) with $0 \leq R, G, B < I_{\max}$. It is a simple matter to obtain, from this representation, an image of only red, green, or blue regions. It is also straightforward to generate any other colour as a combination of these primaries. This can be visualised from the $RGB$ colour cube shown in Figure 2.1. For instance, the image containing all Yellow from Red and Green coloured regions has the form

$$I_{yellow}(x, y) = \frac{[R(x, y) + G(x, y)]}{2}$$

Most real life images display no uniformity of colouring, restricting the ap-

plicability of this method of separation to simple cases of synthetic images in computer graphics in which colour values can be read and displayed by exact values.

There are many transforms for representing an image in another form which may provide a 'better fit' for the image analysis problem [44, 24, 99, 8]. In practice, to facilitate further analysis, the most frequently used transformation is that to the $HSV$ bases. The 'V' is value for 'Intensity' is a measure of brightness and is the average of the colours values:

$$V = \frac{(R+G+B)}{3}$$

Saturation is a measure of colour purity, that is, of the lack of whiteness of colour and is found from the primaries as

$$S = \frac{1 - \min(R,G,B)}{I}$$

A saturation value of 0 indicates equal values of R,G and B, and corresponds to grey value. A saturation value of 1 indicates one or two colour values of zero and corresponds to a pure colour with no pastel characteristic.

Hue is proportional to the average wavelength of colour. It gives the gradation of colour as an image:

$$H = \cos^{-1}\left\{\frac{[(R-G)+(R-B)]}{2}\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}\right\}$$

In this polar coordinate system, red corresponds to $0^0$, green to $120^0$, and blue to $240^0$. Figure 2.2 shows a diagrammatic representation of this colour space. The $HSV$ space provides a representation in terms familiar colour features and thus is a natural basis for image feature analysis.

Related to the above mentioned problem of determining an image with objects of a single colour, the more pertinent problem in practice, is to generate an image with a single range of colours. For instance, to find all objects in the purple range, we would first define the purple range to lie some where between red and blue, say from $270^0$ to $330^0$. Then the purple image would consist of intensity value of all image pixels with hue within the specified range.

Conversely, it may be of interest to find all objects of a certain shape or size and to characterise them by a number of features, including colour features. In this

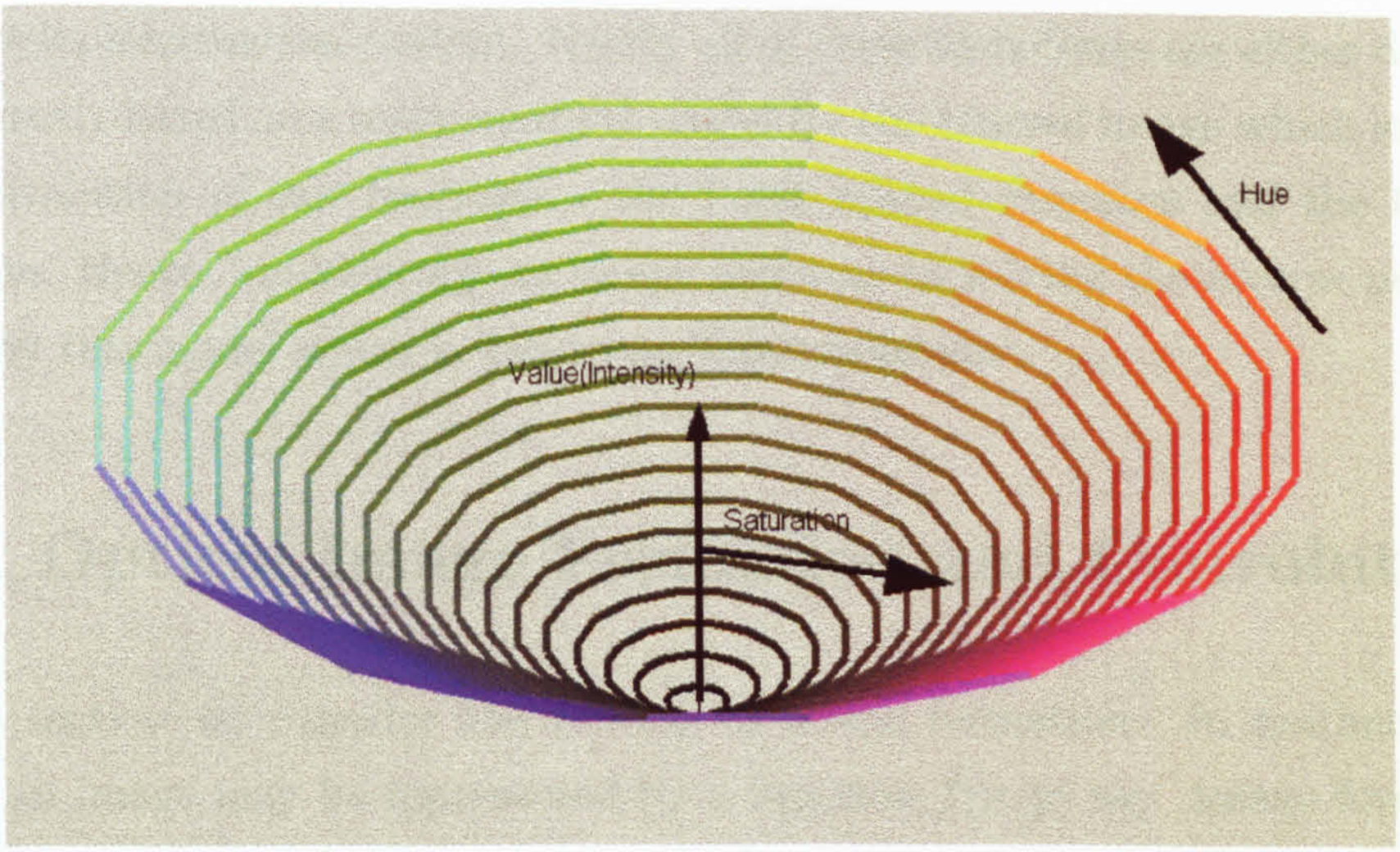



Figure 2.2: $HSV$ colour space

case, analysis to locate the object may be performed exclusively on the intensity image; colour attributes of the object, once located, may then be extracted from the other two images as the average hue and saturation within the bounds of the objects.

There are number of other transform to different basics. Although the $RGB$ to $HSV$ transformation is the most useful for image analysis, different applications may call for other colour basis. For example, in USA the National Television System Committee standard for television uses a $YIQ$ basis $Y$ specifies the intensity or brightness in same way as for $HSV$, $I$ is a red-cyan range, and $Q$ is a magenta- green range. This advantageous because black and white television sets are need extract only the black - white $Y$ basis. In addition, for colour video transmission, the bandwidth associated with $I$ and $Q$ can be limited to a degree without noticeable image degradation. This latter feature further reinforces the practice of using just the intensity information for locating an object and the two other colour primaries only to obtain colour feature information.

All image formation systems are inherently resolution limited. Moreover, many images are blurred due to a variety of physical effects such as motion in the object or image planes, the effects of turbulence and refraction and/or diffraction [76, 99, 79].

When an image is recorded that has been degraded in this manner, a number of digital image processing techniques can employed to 'de-blur' the image and

enhance its information content. Nearly all of these techniques are either directly or indirectly based on a mathematical model for the blurred image which involves the convolution of two functions - the Point Spread Function and the Object Function. Hence, 'de-blurring' an image amounts to solving the inverse problem posed by this model which is known as 'Deconvolution'.

## 2.3 Optical Image Formation and Convolution

In this section we consider the process of image formation. In mathematical terms, an image may be represented by a function $f(x, y)$ say. Image formation is based on the use of linear system which originate from the physics of waves interacting with matter [57].

### 2.3.1 Linear Systems

A 'system' may be defined as that which produces a set of output functions from a set of input functions. Physically, it may be an electrical circuit (inputs and outputs are voltages) or an optical system where the inputs and outputs are either complex amplitudes or intensities. From the point of view of 'linear systems theory', the physical nature of the system is unimportant.

A general system can be represented by the operator $\hat{L}$:

$$s(x, y) = \hat{L}[f(x, y)]$$

A linear system has the property that

$$\hat{L}[af_1(x, y) + bf_2(x, y)] = a\hat{L}[f_1(x, y)] + b\hat{L}[f_2(x, y)]$$

for all inputs $f_1$ and $f_2$ and all constants $a$ and $b$. Linearity implies that an output function can be broken down into elementary functions, each of which can be separately passed through the system; the total output is then the sum of the 'elementary' outputs.

The 'shifting property' of the delta function allows us to consider any input function to be a linear combination of weighted and displaced delta functions:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y')\delta(x - x')\delta(y - y')dx'dy'$$

giving an output

$$s(x,y) = \hat{L}[f(x,y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')\hat{L}[\delta(x-x')\delta(y-y')]dx'dy'$$

The system response at $(x,y)$ due to a delta function input at $(x',y')$ is called the Impulse Response Function (IRF)

$$p(x,y;x',y') = \hat{L}[\delta(x-x')\delta(y-y')]$$

In optical imaging systems, the quantity $p$ is called the Point Spread Function (PSF).

For a linear optical system:

$$s(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')p(x,y;x',y')dx'dy'$$

Note that the optical diffraction formulae are of the same form:

For Fraunhofer diffraction

$$p(x,y;x',y') = \frac{i}{\lambda}\frac{\exp(ikz)}{z}\exp\left(ik\frac{(x^2+y^2)}{2z}\right)\exp\left[-\frac{ik}{z}(xx'+yy')\right]$$

For Fresnel diffraction

$$p(x,y;x',y') = \frac{i}{\lambda}\frac{\exp(ikz)}{z}\exp\left(\frac{i\pi}{2z}[(x-x')^2+(y-y')^2]\right)$$

If the impulse response function of a linear system depends only on the coordinate differences $(x-x')$ and $(y-y')$, and not on each coordinate separately, i.e.

$$p(x,y;x',y') \equiv p(x-x',y-y')$$

then we obtain an expression for $g$ which involves the simple convolution relationship

$$s(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')p(x-x',y-y')dx'dy'$$

Such a system is called stationary. In optical imaging, a stationary optical systems is called isoplanatic. Isoplanicity requires that the point spread function is the same for all field angles and implies that the aberrations are independent of field angle. Many real imaging systems are (to a good approximation) both linear and isoplanatic [11, 36].

The convolution relationship between input and output suggests using Fourier transform techniques which, via the convolution theorem gives

$$S(u,v) = F(u,v)P(u,v)$$

where

$$P(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\exp[-2\pi i(ux+vy)]dxdy$$

i.e.

$$\text{FT of output} = (\text{FT of input}) \times (\text{FT of impulse response function})$$

The quantity $P$ is called the system transfer function. In optical imaging systems, $P$ is called the optical transfer function or OTF [43]. The OTF is just the 2D FT of the point spread function. Note that:

(i) The convolution relationship only applies to linear stationary optical systems.

(ii) There is no unique OTF for an optical system with field-dependent aberrations (non-stationary).

(iii) There is no unique OTF for an optical system when an object is illuminated by spatially partially coherent light (non-linear system in both complex amplitude and intensity).

### 2.3.2 Images of Lines and Edges

Suppose we know that a particular system is linear and stationary in some particular quantity where the 'quantity' is either:

(i) the complex amplitude for coherent illumination

(ii) the intensity (time averaged) for incoherent illumination

How are the images of lines and edges related to the point spread function?

**Lines**

The image of an infinitely narrow line object is called the line spread function A line input can be represented by

$$f(x,y) = \delta(x)$$

Hence

$$s(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x-x')p(x',y')dx'dy'$$

or

$$\ell(x) = \int_{-\infty}^{\infty} p(x,y')dy'$$

where $\ell$ is the line spread function. The line spread function (the image of an infinitely narrow line) is an integration over one variable of the point spread function. In general, the input may be at some arbitrary angle to the $y$ - axis and hence $\ell$ will also be a function of that angle.

For the special case in which the point spread function is rotationally symmetrical, i.e.

$$p(r) = p(x,y); \quad r^2 = x^2 + y^2$$

the line spread function is the Abel transform of the point spread function,

$$\ell(x) = \int_{-\infty}^{\infty} p(\sqrt{x^2+y'^2})dy' = 2\int_{x}^{\infty} p(r)(r^2-x^2)^{-\frac{1}{2}}rdr$$

**Edges**

Consider the image of an opaque edge laying along the $y$ axis

$$f(x) = \begin{cases} 0, & x < 0; \\ 1, & x \geq 0. \end{cases}$$

The result is called the edge spread function $e$ which is given by

$$e(x) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-x')p(x',y')dx'dy' = \int_{-\infty}^{\infty} f(x-x')\ell(x')dx'$$

or

$$e(x) = \int_{-\infty}^{x} \ell(x')dx'$$

The edge spread function is an integration over the line spread function. Therefore we can write

$$\ell(x) = \frac{d}{dx}e(x)$$

Clearly, the equation above indicates a way of measuring the line spread function from the image of an edge. The next section considers the optical transfer function as most imaging employs optical theory.

### 2.3.3 Optical Transfer Function

Optical Transfer Functions have been described in detail by Roberts, Heikkonen, Brammer [32, 79, 43]. This section provides some detail to enhance the understanding of the material.

Consider the response of a linear, stationary system to a 1D cosinusoidal input of the form

$$f(x,y) = a + b\cos(2\pi ux + \epsilon)$$

where $u$ is the spatial frequency and $\epsilon$ is the phase. The modulation of this input $M_{\text{in}}$ is defined as

$$M_{\text{in}} = \frac{f_{\max} - f_{\min}}{f_{\max} + f_{\min}} = \frac{b}{a}$$

The output $s$ is given by

$$s(x) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} [a + b\cos(2\pi u(x - x') + \epsilon)]p(x', y')dx'dy'$$

Integrating with respect to $y$ gives

$$s(x) = \int_{-\infty}^{\infty} \ell(x')[a + b\cos(2\pi u(x - x') + \epsilon)]dx'$$

where $\ell$ is the line spread function which, for convenience, can be considered to be normalized to unit area,

$$\int_{-\infty}^{\infty} \ell(x')dx' = 1$$

Using the result

$$\cos(A - B) = \cos A\cos B + \sin A\sin B$$

the output $s$ can be written in the form

$$s(x) = a + b\cos(2\pi ux + \epsilon)\int_{-\infty}^{\infty} \ell(x')\cos(2\pi ux')dx'$$

$$+b\sin(2\pi ux + \epsilon)\int_{-\infty}^{\infty} \ell(x')\sin(2\pi ux')dx'$$

or

$$s(x) = a + b\cos(2\pi ux + \epsilon)C(u) + b\sin(2\pi ux + \epsilon)S(u)$$

where

$$C(u) - iS(u) = P(u) = \int_{-\infty}^{\infty} \ell(x') \exp(-2\pi i u x') dx'$$

The function $P$ is the 1D OTF and it is equal to the Fourier transform of the line spread function $\ell$. Defining the modulus and phase of the OTF by

$$M(u) = \sqrt{C^2(u) + S^2(u)}$$

$$\phi(u) = \tan^{-1} \frac{-S(u)}{C(u)}$$

so that

$$C(u) = M(u) \cos \phi(u)$$

$$S(u) = -M(u) \sin \phi(u)$$

the output $s$ can be expressed as

$$s(x) = a + bM(u) \cos[2\pi u x + \epsilon + \phi(u)]$$

Like the input $f$, the output $s$ is also cosinusoidal with the same frequency $u$.

The output modulation is

$$M_{\text{out}} = \frac{g_{\max} - g_{\min}}{g_{\max} + g_{\min}} = M(u) \frac{b}{a}$$

so that the ratio of the output modulation to the input modulation is equal to the modulus of the OTF. The Modulation Transfer Function or MTF is given by

$$M(u) = \frac{M_{out}}{M_{\text{in}}}$$

The Phase Transfer Function $\phi$ describes the shift in phase (or position) of the output frequency $u$ relative to the input. The MTF can be found experimentally using sine wave objects or from the line spread function

$$M(u) = \left| \int \ell(x) \exp(-2\pi i u x) dx \right|$$

which itself could be found by differentiating the edge spread function.

### 2.3.4 Linearity of Optical Imaging Systems

Consider the case where the object plane is illuminated by a plane or spherical wave - by perfectly spatially coherent light. Let the complex amplitude immediately after the object by denoted by $U_{\text{in}}(x, y)$ and $U_{\text{out}}(x, y)$ be the complex

amplitude at the image plane. Also, let the complex amplitude at $(x, y)$ in the output due to a unit strength point at in the input be $p(x, y; x', y')$. The total amplitude at $(x, y)$ due to all such point in the object plane is then given by

$$U_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_{\text{in}}(x', y') p(x, y; x', y') dx' dy'$$

For an isoplanatic optical system, this reduces to

$$U_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_{\text{in}}(x', y') p(x - x', y - y') dx' dy'$$

**A spatially coherent optical system is linear in the complex amplitude**

Consider the case of narrowband light that is not perfectly spatially coherent. The general complex representation of the time-varying scalar field is called the analytic signal $V(\mathbf{r}, t)$; it is defined such that

$$\text{real scalar field} = \text{Re}[V(\mathbf{r}, t)]$$

For narrowband light, the analytic signal can be written as a product of a 'slowly varying' function - the time varying complex amplitude $U(\mathbf{r}, t)$ times $\exp(-i\omega t)$:

$$V(\mathbf{r}, t) = U(\mathbf{r}, t) \exp(-i\omega t)$$

The instantaneous intensity is defined as

$$I(\mathbf{r}, t) = \mid U(\mathbf{r}, t) \mid^2$$

whereas the time-averaged intensity $\bar{I}(\mathbf{r})$ (i.e. that observed by an optical detector), is equal to

$$\bar{I}(\mathbf{r}) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} I(\mathbf{r}, t) dt$$

In general, the time-varying complex amplitudes are related by

$$U_{\text{out}}(x, y, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_{\text{in}}(x', y', t) p(x, y; x' y') dx' dy'$$

Coherent illumination implies that $U(x, y, t) = U(x, y)$ - the field does not vary in time. The average intensity is therefore given by

$$\bar{I}_{\text{out}}(x, y) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} \mid U_{\text{out}}(x, y, t) \mid^2 dt$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y;x',y')p^*(x,y;x',y')$$

$$\times \left[\lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} U_{\text{in}}(x',y',t)U_{\text{in}}^*(x'',y'',t)dt\right] dx'dy'dx''dy''$$

In some sources is called the mutual intensity of the narrow band light

$$J_{\text{in}}(x',y';x'',y'') = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} U_{\text{in}}(x',y',t)U_{\text{in}}^*(x'',y'',t)dt$$

or

$$J_{in}(\mathbf{r}',\mathbf{r}'') = \langle U_{\text{in}}(\mathbf{r}',t)U_{\text{in}}^*(\mathbf{r}'',t)\rangle$$

Incoherent light is defined to be such that

$$J(\mathbf{r}',\mathbf{r}'') = \bar{I}(\mathbf{r}')\delta(\mathbf{r}'-\mathbf{r}'')$$

That is, two neighbouring points $\mathbf{r}'$ and $\mathbf{r}''$ have uncorrelated fields, for any $\mathbf{r}' \neq \mathbf{r}''$. Using the definition for incoherent light above, the expression for $\bar{I}_{\text{out}}$ becomes

$$\bar{I}_{\text{out}}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y;x',y')p^*(x,y;x',y')$$

$$\times \bar{I}_{\text{in}}(x',y')\delta(x'-x'')\delta(y'-y'')dx'dy'dx''dy''$$

or

$$\bar{I}_{\text{out}}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mid p(x,y;x',y') \mid^2 \bar{I}_{\text{in}}(x',y')dx'dy'$$

The quantity $\mid p(x,y;x',y') \mid^2$ is the intensity point spread function.

For an isoplanatic optical system, this result reduces to

$$\bar{I}_{\text{out}}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \bar{I}_{\text{in}}(x',y') \mid p(x-x',y-y') \mid^2 dx'dy'$$

The bar over $I$ is usually omitted when referring to the intensity -a time average is always assumed.

For perfectly incoherent illumination, an optical system is linear in intensity and, if isoplanicity holds, the output (image) intensity is equal to the input (object) intensity convolved with the intensity point spread function.

### 2.3.5 Coherent Image Formation

In coherent light, the complex amplitude of the image is equal to that at the object plane convolved with the amplitude point spread function (for an isoplanatic system), i.e.

$$U_{\text{out}}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_{\text{in}}(x',y')p(x-x',y-y')dx'dy'$$

where

$$p(x,y) = C\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v)\exp\left[-i\frac{2\pi}{\lambda z}(ux+vy)\right]dudv$$

and $P$ is the Pupil Function of the optical system, i.e. the complex amplitude in the exit pupil. The constant $C$ is usually chosen so that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)dxdy = P(0,0) = 1$$

The pupil function $P$ is, for a clear pupil, defined by

$$P(u,v) = \begin{cases} \exp[ikW(u,v)], & (u,v) \in \text{aperture;} \\ 0, & \text{otherwise.} \end{cases}$$

so that $P(0,0) = 1$ and therefore $C = 1$. The function $W$ is called the Wave Aberration Function.

A shaded or apodized pupil can be handled by introducing an absorption term $A$,

$$P(u,v) = A(u,v)\exp[ikW(u,v)]$$

Taking the Fourier transform of $U_{\text{out}}$ and using the convolution theorem we can write

$$\widetilde{U_{\text{out}}}(u,v) = \widetilde{U_{\text{in}}}(u,v)T(u,v)$$

where

$\widetilde{U_{\text{out}}}$ - spectrum of image amplitude

$\widetilde{U_{\text{in}}}$ - spectrum of object amplitude

$T$ - Coherent Optical Transfer Function (COTF)

Note that

$$T(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\exp[-2\pi i(ux+vy)]dxdy = P(\lambda zu, \lambda zv)$$

That is, the COTF at spatial frequency $(u, v)$ is simply equal to the pupil function at coordinates $(\lambda z u, \lambda z v)$.

### 2.3.6 Incoherent Image Formation

In incoherent illumination, there is a linear relationship between the input $I_{\text{in}}$ and output $I_{\text{out}}$ (time-averaged) intensities. For an isoplanatic optical system,

$$I_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_{\text{in}}(x', y') \mid p(x - x', y - y') \mid^2 dx' dy'$$

If $p$ is normalized to unit volume (it is, since $P(0, 0) = 1$), $\mid p \mid^2$ is not, so we normalize it by dividing by its infinite integral:

$$\frac{\mid p(x, y) \mid^2}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mid p(x, y) \mid^2 dx dy}$$

The Incoherent Optical Transfer Function (IOTF) $T(u, v)$ is the Fourier transform of the (normalized) point spread function; apply the autocorrelation theorem to the top line and Parseval's theorem to the bottom line,

$$T(u, v) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(\xi, \eta) P^*(\xi + \lambda z u, \eta + \lambda z v) d\xi d\eta}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mid P(\xi, \eta) \mid^2 d\xi d\eta}$$

where $P(\xi, \eta)$ is the inverse Fourier transform of $p(x, y)$.

$$P(\xi, \eta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \exp \left[ i \frac{2\pi}{\lambda z} (\xi x + \eta y) \right] dx dy$$

and is the pupil function.

The equation above for $T$ basically states that the IOTF is equal to the (normalized) spatial autocorrelation of the pupil function. Note that the IOTF relates the input and output intensity spectra,

$$\tilde{I}_{\text{out}}(u, v) = \tilde{I}_{\text{in}}(u, v) T(u, v)$$

The spatial frequencies are intensity frequencies and are not the same as the amplitude frequencies produced in a coherent optical system. This expression for

$T$ can be written in the form

$$T(u,v) = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P\left(\xi - \frac{\lambda z u}{2}, \eta - \frac{\lambda z v}{2}\right) P^*\left(\xi + \frac{\lambda z u}{2}, \eta + \frac{\lambda z v}{2}\right) d\xi d\eta}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mid P(\xi,\eta) \mid^2 d\xi d\eta}$$

From this result it follows that:

(i) $T(0,0) = 1$ (because of normalization)

(ii) $T(-u,-v) = T^*(u,v)$ (Fourier transform of a real quantity)

(iii) $T(u,v) \leq T(0,0)$ (using the Schwartz inequality)

As with the coherent OTF, the modulus $\mid T \mid$, or Modulation Transfer Function (MTF) describes the transfer or modulation of sinusoidal components of the object. The phase of $T$, describes spatial shifts of the sinusoidal components.

## 2.4 Deconvolution and Image Reconstruction

Deconvolution is a particularly important subject area in signal and image processing. In general, this problem is concerned with the restoration and/or reconstruction of information from known data and depends critically on *a priori* knowledge on the way in which the data (digital image for example) has been generated and recorded. Mathematically, the data obtained are usually related to some 'Object Function' via an integral transform. In this sense, deconvolution is concerned with inverting certain classes of integral equation - the convolution equation. In general, there is no exact or unique solution to the image restoration/ reconstruction problem - it is an ill-posed problem. We attempt to find a 'best estimate' based on some physically viable criterion subject to certain conditions [55, 84].

The fundamental imaging equation is given by

$$s = p \otimes\otimes f + n$$

where $s, p, f$ and $n$ are the image, the Point Spread Function (PSF), the Object Function and the noise respectively. The symbol $\otimes\otimes$ denotes 2D convolution. The imaging equation is a stationary model for the image $s$ in which the (blurring)

effect of the PSF at any location in the 'object plane' is the same. Using the convolution theorem we can write this equation in the form

$$S = PF + N$$

where $S, P, F$ and $N$ are the (2D) Fourier transforms of $s, p, f$ and $n$ respectively.

Assuming that $F$ is a broadband spectrum, then from position of classical point there are two cases we should consider:

(i) $P(k_x, k_y) \to 0$ as $(k_x, k_y) \to \infty$ where $k_x$ and $k_y$ are the spatial frequencies in the $x$ and $y$ directions respectively. The image restoration problem can then be stated as 'recover $F$ given $S$'.

(ii) $P(k_x, k_y)$ is bandlimited, i.e. $P(k_x, k_y) = 0$ for certain values of $k_x$ and/or $k_y$. The image reconstruction problem can then be stated as 'given $S$ reconstruct $F$'. This typically requires the frequency components to be 'synthesized' beyond the bandwidth of the data. This is a (spectral) extrapolation problem.

The image restoration problem can typically involve finding a solution for $f$ given that $s = p \otimes\otimes f + n$ where $p$ is a Gaussian PSF given by (ignoring scaling)

$$p(x, y) = \exp[-(x^2 + y^2)/\sigma^2]$$

($\sigma$ being the standard deviation) which has a spectrum of the form (ignoring scaling)

$$P(k_x, k_y) = \exp[-\sigma^2(k_x^2 + k_y^2)]$$

This PSF is a piecewise continuous function as is its spectrum.

An example of an image reconstruction problem is 'find $f$ given that $s = p \otimes\otimes f + n$' where $p$ is given by (ignoring scaling)

$$p(x, y) = \mathrm{sinc}(\alpha x)\,\mathrm{sinc}(\beta y)$$

This PSF has a spectrum of the form (ignoring scaling)

$$P(k_x, k_y) = H_\alpha(k_x) H_\beta(k_y)$$

where

$$H_\alpha(k_x) = \begin{cases} 1 & |\, k_x \,| \leq \alpha \\ 0 & |\, k_x \,| > \alpha \end{cases}$$

$$H_\beta(k_y) = \begin{cases} 1 & |\, k_y \,| \leq \beta \\ 0 & |\, k_y \,| > \beta \end{cases}$$

It is a piecewise continuous function but its spectrum is discontinuous, the bandwidth of $p \otimes\otimes f$ being given by $\alpha$ in the $x$ direction and $\beta$ in the $y$ direction.

## Restoration of Blurred Images

To put the problem in perspective, consider the discrete case, i.e.

$$s_{ij} = p_{ij} \otimes\otimes f_{ij} + n_{ij}$$

where $s_{ij}$ is a digital image. Suppose we neglect the term $n_{ij}$, then

$$s_{ij} = p_{ij} \otimes\otimes f_{ij}$$

or by the (discrete) convolution theorem

$$S_{ij} = P_{ij} F_{ij}$$

where $S_{ij}, P_{ij}$ and $F_{ij}$ and the DFTs of $s_{ij}, p_{ij}$ and $f_{ij}$ respectively. Clearly,

$$F_{ij} = \frac{S_{ij}}{P_{ij}}$$

and therefore

$$f_{ij} = \text{IDTF}\left(\frac{S_{ij}}{P_{ij}}\right)$$

Note that

$$\frac{1}{P_{ij}} = \frac{P_{ij}^*}{|\, P_{ij} \,|^2}$$

which is called the Inverse Filter.

Suppose, we were to implement this result on a digital computer; if $P_{ij}$ approached zero (in practice a very small number) for any value of $i$ and/or $j$ then depending on the compiler, the computer would respond with an output such as '... arithmetic fault ... divide by zero'. A simple solution would be to regularize the result, i.e. use

$$f_{ij} = \text{IDFT}\left(\frac{P_{ij}^* S_{ij}}{|\, P_{ij} \,|^2 + \text{constant}}\right)$$

and 'play around' with the value of the constant until 'something sensible' was obtained which in turn would depend on the *a priori* information available on the form and support of $f_{ij}$. The regularization of the inverse filter is the basis for some of the methods which are discussed in these notes. We start by considering the criterion associated with the inverse filter.

### 2.4.1 The Inverse Filter

The criterion for the inverse filter is that the mean square of the noise is a minimum. Since

$$s_{ij} = p_{ij} \otimes\otimes f_{ij} + n_{ij}$$

we can write

$$n_{ij} = s_{ij} - p_{ij} \otimes\otimes f_{ij}$$

and therefore

$$e = \| n_{ij} \|^2 = \| s_{ij} - p_{ij} \otimes\otimes f_{ij} \|^2$$

where

$$\| x_{ij} \| \equiv \left( \sum_i \sum_j x_{ij}^2 \right)^{\frac{1}{2}}$$

For the noise to be a minimum, we require

$$\frac{\partial e}{\partial f_{ij}} = 0$$

Differentiating (see Appendix A), we obtain

$$(s_{ij} - p_{ij} \otimes\otimes f_{ij}) \odot\odot p_{ij} = 0$$

Using the convolution and correlation theorems, in Fourier space, this equation becomes

$$(S_{ij} - P_{ij} F_{ij}) P_{ij}^* = 0$$

Hence, solving for $F_{ij}$ we obtain the result

$$F_{ij} = \frac{P_{ij}^*}{| P_{ij} |^2} S_{ij}$$

The inverse filter is therefore given by

$$\text{Inverse Filter} = \frac{P_{ij}^*}{| P_{ij} |^2}$$

In principle, the inverse filter provides an exact solution to the problem when the noise term $n_{ij}$ can be neglected. However, in practice, this solution is fraught with difficulties. First, the inverse filter is invariably a singular function. Equally bad, is the fact that even if the inverse filter is not singular, it is usually ill conditioned. This is where the magnitude of $P_{ij}$ goes to zero so quickly as $(i, j)$ increases, that $1/ | P_{ij} |^2$ rapidly acquires extremely large values. The effect of

this is to amplify the (usually) noisy high frequency components of $S_{ij}$. This can lead to a restoration $f_{ij}$ which is dominated by the noise in $s_{ij}$. The inverse filter can therefore only be used when:

(i) The filter is non-singular.

(ii) The SNR of the data is very large (i.e. $\|p_{ij} \otimes\otimes f_{ij}\| >> \|n_{ij}\|$).

Such conditions are rare. A notable exception occurs in Computed Tomography which is covered in Section 5 of these notes in which the inverse filter associated with the 'Back-Project and Deconvolution' algorithm is non-singular.

The computational problems that arise from implementing the inverse filter can be avoided by using a variety of different filters whose individual properties and characteristics are suited to certain types of data. One of the most commonly used filters for image restoration is the Wiener filter which is considered next.

### 2.4.2 The Wiener Filter

An algorithm has been derived [21] for deconvolving images that have been blurred by some lowpass filtering process and corrupted by additive noise. In mathematical terms, given the imaging equation

$$s_{ij} = p_{ij} \otimes\otimes f_{ij} + n_{ij} \tag{2.1}$$

the problem is to solve for $f_{ij}$ given $s_{ij}$, $p_{ij}$ and some knowledge of the SNR. This problem is solved using the least squares principle which provides a filter known as the Wiener filter.

The Wiener filter is based on considering an estimate $\hat{f}_{ij}$for $f_{ij}$ of the form

$$\hat{f}_{ij} = q_{ij} \otimes\otimes s_{ij} \tag{2.2}$$

Given this model, our problem is reduced to computing $q_{ij}$ or equivalently its Fourier transform $Q_{ij}$. To do this, we make use of the error

$$e = \|f_{ij} - \hat{f}_{ij}\|^2 \equiv \sum_i \sum_j (f_{ij} - \hat{f}_{ij})^2 \tag{2.3}$$

and find $q_{ij}$ such that $e$ is a minimum, i.e.

$$\frac{\partial e}{\partial q_{ij}} = 0$$

Substituting equation (2.2) into equation (2.3) and differentiating, we get

$$\frac{\partial e}{\partial q_{k\ell}} = -2\sum_i\sum_j\left(f_{ij} - \sum_n\sum_m s_{i-n,j-m}q_{nm}\right)\frac{\partial}{\partial q_{k\ell}}\sum_n\sum_m s_{i-n,j-m}q_{nm}$$

$$= -2\sum_i\sum_j\left(f_{ij} - \sum_n\sum_m s_{i-n,j-m}q_{nm}\right)s_{i-k,j-\ell} = 0$$

Rearranging, we have

$$\sum_i\sum_j f_{ij}s_{i-k,j-\ell} = \sum_i\sum_j\left(\sum_n\sum_m s_{i-n,j-m}q_{nm}\right)s_{i-k,j-\ell}$$

The left hand side of the above equation is a discrete correlation of $f_{ij}$ with $s_{ij}$ and the right hand side is a correlation of $s_{ij}$ with the convolution

$$\sum_n\sum_m s_{i-n,j-m}q_{nm}$$

Using operator notation, it is convenient to write this equation in the form

$$f_{ij} \odot\odot s_{ij} = (q_{ij} \otimes\otimes s_{ij}) \odot\odot s_{ij}$$

Moreover, using the correlation and convolution theorems, the equation above can be written in Fourier space as

$$F_{ij}S_{ij}^* = Q_{ij}S_{ij}S_{ij}^*$$

which, after rearranging gives

$$Q_{ij} = \frac{S_{ij}^*F_{ij}}{\mid S_{ij} \mid^2}$$

Now, in Fourier space, equation (2.1) becomes

$$S_{ij} = P_{ij}F_{ij} + N_{ij}$$

Using this result, we have

$$S_{ij}^*F_{ij} = (P_{ij}F_{ij} + N_{ij})^*F_{ij}$$

$$= P_{ij}^* \mid F_{ij} \mid^2 + N_{ij}^*F_{ij}$$

and

$$\mid S_{ij} \mid^2 = S_{ij}S_{ij}^* = (P_{ij}F_{ij} + N_{ij})(P_{ij}F_{ij} + N_{ij})^*$$

$$= \mid P_{ij} \mid^2 \mid F_{ij} \mid^2 + P_{ij}F_{ij}N_{ij}^* + N_{ij}P_{ij}^*F_{ij}^* + \mid N_{ij} \mid^2$$

Hence, the filter $Q_{ij}$ can be written in the form

$$Q_{ij} = \frac{P_{ij}^* \mid F_{ij} \mid^2 + N_{ij}^* F_{ij}}{\mid P_{ij} \mid^2 \mid F_{ij} \mid^2 + D_{ij} + \mid N_{ij} \mid^2}$$

where

$$D_{ij} = P_{ij}F_{ij}N_{ij}^* + N_{ij}P_{ij}^*F_{ij}^*$$

**Signal Independent Noise**

This result can be simplified further by imposing a condition which is physically valid in the large majority of cases. The condition is that $f_{ij}$ and $n_{ij}$ are uncorrelated, i.e.

$$f_{ij} \odot \odot n_{ij} = 0$$

and

$$n_{ij} \odot \odot f_{ij} = 0$$

In this case, the noise is said to be 'signal independent' and it follows from the correlation theorem that

$$F_{ij}N_{ij}^* = 0$$

and

$$N_{ij}F_{ij}^* = 0$$

This result allows us to cancel the cross terms present in the last expression for $Q_{ij}$ (i.e. set $D_{ij} = 0$ and $N_{ij}^* F_{ij} = 0$) leaving the formula

$$Q_{ij} = \frac{P_{ij}^* \mid F_{ij} \mid^2}{\mid P_{ij} \mid^2 \mid F_{ij} \mid^2 + \mid N_{ij} \mid^2}$$

Finally, rearranging, we obtain the expression for the least squares or Wiener filter,

$$Q_{ij} = \frac{P_{ij}^*}{\mid P_{ij} \mid^2 + \mid N_{ij} \mid^2 / \mid F_{ij} \mid^2}$$

**Estimation of the Noise-to-Signal Power Ratio $\mid F_{ij} \mid^2 / \mid N_{ij} \mid^2$**

From the algebraic form of the Wiener Filter derived above, it is clear that this particular filter depends on:

(i) the functional form of the PSF $P_{ij}$ that is used;

(ii) the functional form of $\mid N_{ij} \mid^2 / \mid F_{ij} \mid^2$.

The PSF of the system can usually be found by literally imaging a single point source which leaves us with the problem of estimating the noise-to-signal power ratio $| N_{ij} |^2 / | F_{ij} |^2$. This problem can be solved if one has access to two successive images recorded under identical conditions.

Consider two digital images denoted by $s_{ij}$ and $s'_{ij}$ of the same object function $f_{ij}$ recorded using the same PSF $p_{ij}$ (i.e. imaging system) but at different times and hence with different noise fields $n_{ij}$ and $n'_{ij}$. These images are given by

$$s_{ij} = p_{ij} \otimes\otimes f_{ij} + n_{ij}$$

and

$$s'_{ij} = p_{ij} \otimes\otimes f_{ij} + n'_{ij}$$

respectively where the noise functions are uncorrelated and signal independent, i.e.

$$n_{ij} \odot\odot n'_{ij} = 0 \tag{2.4}$$

$$f_{ij} \odot n_{ij} = n_{ij} \odot\odot f_{ij} = 0 \tag{2.5}$$

and

$$f_{ij} \odot\odot n'_{ij} = n'_{ij} \odot\odot f_{ij} = 0 \tag{2.6}$$

We now proceed to compute the autocorrelation function of $s_{ij}$ given by

$$c_{ij} = s_{ij} \odot\odot s_{ij}$$

Using the correlation theorem and employing equation (2.5) we get

$$C_{ij} = S_{ij}S^*_{ij} = (P_{ij}F_{ij} + N_{ij})(P_{ij}F_{ij} + N_{ij})^*$$

$$= | P_{ij} |^2 | F_{ij} |^2 + | N_{ij} |^2$$

where $C_{ij}$ is the DFT of $c_{ij}$. Next, we correlate $s_{ij}$ with $s'_{ij}$ giving the cross-correlation function

$$c'_{ij} = s_{ij} \odot\odot s'_{ij}$$

Using the correlation theorem again and this time, employing equations (2.4) and (2.6) we get

$$C'_{ij} = | P_{ij} |^2 | F_{ij} |^2 + P_{ij}F_{ij}N'^*_{ij}$$

$$+ N_{ij}P^*_{ij}F^*_{ij} + N_{ij}N'^*_{ij}$$

$$= | P_{ij} |^2 | F_{ij} |^2$$

The noise-to-signal ratio can now be obtained by dividing $C_{ij}$ by $C'_{ij}$ giving

$$\frac{C_{ij}}{C'_{ij}} = 1 + \frac{| N_{ij} |^2}{| P_{ij} |^2 | F_{ij} |^2}$$

Re-arranging, we obtain the result

$$\frac{| N_{ij} |^2}{| F_{ij} |^2} = \left( \frac{C_{ij}}{C'_{ij}} - 1 \right) | P_{ij} |^2$$

Note that both $C_{ij}$ and $C'_{ij}$ can be obtained from the available data $s_{ij}$ and $s'_{ij}$. Also, substituting this result into the formula for $Q_{ij}$, we obtain an expression for the Wiener filter in terms of $C_{ij}$ and $C'_{ij}$ given by

$$Q_{ij} = \frac{P^*_{ij}}{| P_{ij} |^2} \frac{C'_{ij}}{C_{ij}}$$

In those cases where the user has access to successive recordings, the method of computing the noise-to-signal power ratio described above can be employed. The problem is that in many practical cases, one does not have access to successive images and hence, the cross-correlation function $c'_{ij}$ cannot be computed. In this case, one is forced to make an approximation and consider a Wiener filter of the form

$$\text{Wiener Filter} = \frac{P^*_{ij}}{| P_{ij} |^2 + \text{constant}}$$

The constant ideally reflects any available information on the average signal-to-noise ratio of the image. Typically, we consider an expression of the form

$$\text{constant} = \frac{1}{(\text{SNR})^2}$$

where SNR stands for Signal-to-Noise Ratio. In practice, the exact value of this constant must be chosen by the user.

Before attempting to deconvolve an image the user must at least have some *a priori* knowledge on the functional form of the Point Spread Function. Absence of this information leads to a method of approach known as 'Blind Deconvolution'. A common technique is to assume that the Point Spread Function is Gaussian, i.e.

$$p_{ij} = \exp[-(i^2 + j^2)/2\sigma^2]$$

where $\sigma$ is the standard deviation which must be defined by the user. In this case, the user has control of two parameters:

(i) the standard deviation of the Gaussian PSF;

(ii) the SNR.

In practice, the user must adjust these parameters until a suitable 'user optimized' reconstruction is obtained. In other words, the Wiener filter must be 'tuned' to give a result which is acceptable based on the judgement and intuition of the user. This interactive approach to image restoration is just one of many practical problems associated with deconvolution which should ideally be executed in real time.

### 2.4.3 The Power Spectrum Equalization Filter

As the name implies, the Power Spectrum Equalization (PSE) [44, 17, 21] filter is based on finding an estimate $\hat{f}_{ij}$ whose power spectrum is equal to the power spectrum of the desired function $f_{ij}$. In other words, $\hat{f}_{ij}$ is obtained by employing the criterion

$$| F_{ij} |^2 = | \hat{F}_{ij} |^2$$

together with the linear convolution model

$$\hat{f}_{ij} = q_{ij} \otimes \otimes s_{ij}$$

Like the Wiener filter, the PSE filter also assumes that the noise is signal independent. Since

$$\hat{F}_{ij} = Q_{ij} S_{ij} = Q_{ij}(P_{ij} F_{ij} + N_{ij})$$

and given that $N^*_{ij} F_{ij} = 0$ and $F^*_{ij} N_{ij} = 0$, we have

$$| \hat{F}_{ij} |^2 = \hat{F}_{ij} \hat{F}^*_{ij} = | Q_{ij} |^2 \, (| P_{ij} |^2 | F_{ij} |^2 + | N_{ij} |^2)$$

Using the PSE criterion and solving for $| Q_{ij} |$, we obtain

$$| Q_{ij} | = \left( \frac{1}{| P_{ij} |^2 + | N_{ij} |^2 / | F_{ij} |^2} \right)^{\frac{1}{2}}$$

In the absence of accurate estimates for the noise to signal power ratio, we approximate the PSE filter by

$$\text{PSE filter} = \left( \frac{1}{| P_{ij} |^2 + \text{constant}} \right)^{\frac{1}{2}}$$

where

$$\text{constant} = \frac{1}{(\text{SNR})^2}$$

Note that the criterion used to derive this filter can be written in the form

$$\sum_i \sum_j (\mid F_{ij} \mid^2 - \mid \hat{F}_{ij} \mid^2) = 0$$

or using Parseval's theorem

$$\sum \sum (\mid f_{ij} \mid^2 - \mid \hat{f}_{ij} \mid^2) = 0$$

Compare this criterion with that use for the Wiener filter, i.e.

$$\text{Minimise} \sum_i \sum_j (f_{ij} - \hat{f}_{ij})^2$$

### 2.4.4 The Matched Filter

Matched filtering is based on correlating the image $s_{ij}$ with the complex conjugate of the PSF $p_{ij}$. The estimate $\hat{f}_{ij}$ of $f_{ij}$ can therefore be written as

$$\hat{f}_{ij} = p^*_{ij} \odot \odot s_{ij}$$

Assuming that $n_{ij} = 0$, so that

$$s_{ij} = p_{ij} \otimes \otimes f_{ij}$$

we have

$$\hat{f}_{ij} = p^*_{ij} \odot \odot p_{ij} \otimes \otimes f_{ij}$$

which in Fourier space is

$$\hat{F}_{ij} = \mid P_{ij} \mid^2 F_{ij}$$

Observe, that the amplitude spectrum of $\hat{F}_{ij}$ is given by $\mid P_{ij} \mid^2 \mid F_{ij} \mid$ and that the phase information is determined by $F_{ij}$ alone.

#### Criterion for the Matched Filter

The criterion for the matched filter is as follows. Given that

$$s_{ij} = p_{ij} \otimes \otimes f_{ij} + n_{ij}$$

the match filter provides an estimate for $f_{ij}$ of the form

$$\hat{f}_{ij} = q_{ij} \otimes \otimes s_{ij}$$

where $q_{ij}$ is chosen in such a way that the ratio

$$R = \frac{|\sum_i \sum_j Q_{ij} P_{ij}|^2}{\sum_i \sum_j |N_{ij}|^2 |Q_{ij}|^2}$$

is a maximum.

The matched filter $Q_{ij}$ is found by first writing

$$Q_{ij} P_{ij} = |N_{ij}| Q_{ij} \times \frac{P_{ij}}{|N_{ij}|}$$

and then using the inequality

$$\left|\sum_i \sum_j Q_{ij} P_{ij}\right|^2 = \left|\sum_i \sum_j |N_{ij}| Q_{ij} \frac{P_{ij}}{|N_{ij}|}\right|^2$$

$$\leq \sum_i \sum_j |N_{ij}|^2 |Q_{ij}|^2 \sum_i \sum_j \frac{|P_{ij}|^2}{|N_{ij}|^2}$$

From this result and the definition of $R$ given above we get

$$R \leq \sum_i \sum_j \frac{|P_{ij}|^2}{|N_{ij}|^2}$$

Now, recall that the criterion for the matched filter is that $R$ is a maximum. If this is the case, then

$$R = \sum_i \sum_j \frac{|P_{ij}|^2}{|N_{ij}|^2}$$

or

$$\left|\sum_i \sum_j |N_{ij}| Q_{ij} \frac{P_{ij}}{|N_{ij}|}\right|^2 = \left(\sum_i \sum_j |N_{ij}|^2 |Q_{ij}|^2\right) \left(\sum_i \sum_j \frac{|P_{ij}|^2}{|N_{ij}|^2}\right)$$

This is true, if and only if

$$|N_{ij}| Q_{ij} = \frac{P_{ij}^*}{|N_{ij}|}$$

because we then have

$$\left|\sum_i \sum_j \frac{|P_{ij}|^2}{|N_{ij}|^2}\right|^2 = \sum_i \sum_j \frac{|P_{ij}|^2}{|N_{ij}|^2} \sum_i \sum_j \frac{|P_{ij}|^2}{|N_{ij}|^2}$$

Thus, $R$ is a maximum when

$$Q_{ij} = \frac{P_{ij}^*}{|N_{ij}|^2}$$

## The Matched Filter for White Noise

If the noise $n_{ij}$ is white, then its power spectrum is can be assumed to be a constant, i.e.

$$\mid N_{ij} \mid^2 = N_0^2$$

In this case

$$Q_{ij} = \frac{P_{ij}^*}{N_0^2}$$

and

$$\hat{F}_{ij} = \frac{P_{ij}^*}{N_0^2} S_{ij}$$

Hence, for white noise, the match filter provides an estimate which may be written in the form

$$\hat{f}_{ij} = \frac{1}{N_0^2} p_{ij}^* \odot\odot s_{ij}$$

## Deconvolution of Linear Frequency Modulated PSFs

The matched filter is frequently used in coherent imaging systems whose PSF is characterized by a linear frequency modulated response. Two well known examples are Synthetic Aperture Radar and imaging systems that use (Fresnel) zone plates. In this section, we shall consider a separable linear FM PSF and also switch to a continuous noise free functional form which makes the analysis easier. Thus, consider the case when the PSF is given by

$$p(x,y) = \exp(i\alpha x^2)\exp(i\beta y^2); \quad \mid x \mid \leq X, \quad \mid y \mid \leq Y$$

where $\alpha$ and $\beta$ are constants and $X$ and $Y$ determine the spatial support of the PSF. The phase of this PSF (in the $x$-direction say) is $\alpha x^2$ and the instantaneous frequency is given by

$$\frac{d}{dx}(\alpha x^2) = 2\alpha x$$

which varies linearly with $x$. Hence, the frequency modulations (in both $x$ and $y$) are linear which is why the PSF is referred to as a linear FM PSF. In this case, the image that is obtained is given by (neglecting additive noise)

$$s(x,y) = \exp(i\alpha x^2)\exp(i\beta y^2) \otimes\otimes f(x,y); \quad \mid x \mid \leq X, \quad \mid y \mid \leq Y$$

Matched filtering, we get

$$\hat{f}(x,y) = \exp(-i\alpha x^2)\exp(-i\beta y^2) \odot\odot \exp(i\alpha x^2)\exp(i\beta y^2) \otimes\otimes f(x,y)$$

Now,

$$\exp(-i\alpha x^2) \odot \exp(i\alpha x^2) = \int_{-X/2}^{X/2} \exp[-i\alpha(z+x)^2] \exp(i\alpha z^2) dz$$

$$= \exp(-i\alpha x^2) \int_{-X/2}^{X/2} \exp(2i\alpha zx) dz$$

Evaluating the integral over $z$, we have

$$\exp(-i\alpha x^2) \odot \exp(i\alpha x^2) = X \exp(-i\alpha x^2) \operatorname{sinc}(\alpha X x)$$

Since the evaluation of the correlation integral over $y$ is identical, we can write

$$\hat{f}(x,y) = XY \exp(-i\alpha x^2) \exp(-i\beta y^2) \operatorname{sinc}(\alpha X x) \operatorname{sinc}(\beta Y y) \otimes\otimes f(x,y)$$

In many systems the spatial support of the linear FM PSF is relatively long. In this case,

$$\cos(\alpha x^2) \operatorname{sinc}(\alpha X x) \simeq \operatorname{sinc}(\alpha X x), \quad \cos(\beta y^2) \operatorname{sinc}(\beta Y y) \simeq \operatorname{sinc}(\beta Y y)$$

and

$$\sin(\alpha x^2) \operatorname{sinc}(\alpha X x) \simeq 0, \quad \sin(\beta y^2) \operatorname{sinc}(\beta Y y) \simeq 0$$

and so

$$\hat{f}(x,y) \simeq XY \operatorname{sinc}(\alpha X x) \operatorname{sinc}(\beta Y y) \otimes\otimes f(x,y)$$

In Fourier space, this last equation can be written as

$$\hat{F}(k_x, k_y) = \begin{cases} \frac{\pi^2}{\alpha\beta} F(k_x, k_y), & |k| \leq \alpha X, \quad |k| \leq \beta Y; \\ 0, & \text{otherwise.} \end{cases}$$

The estimate $\hat{f}$ is therefore a band limited estimate of $f$ whose bandwidth is determined by the product of the parameters $\alpha$ and $\beta$ with the spatial supports $X$ and $Y$ respectively. Note, that the larger the values of $\alpha X$ and $\beta Y$, the greater the bandwidth of the reconstruction.

### 2.4.5 Constrained Deconvolution

Constrained deconvolution provides a filter which gives the user additional control over the deconvolution process. This method is based on minimizing a linear operation on the object $f_{ij}$ of the form $g_{ij} \otimes\otimes f_{ij}$ subject to some other constraint. Using the least squares approach, we find an estimate for $f_{ij}$ by minimizing $\|g_{ij} \otimes\otimes f_{ij}\|^2$ subject to the constraint

$$\|s_{ij} - p_{ij} \otimes\otimes f_{ij}\|^2 = \|n_{ij}\|^2$$

where

$$\|x_{ij}\|^2 \equiv \sum_i \sum_j x_{ij}^2$$

Using this result, we can write

$$\|g_{ij} \otimes\otimes f_{ij}\|^2 = \|g_{ij} \otimes\otimes f_{ij}\|^2 + \lambda(\|s_{ij} - p_{ij} \otimes\otimes f_{ij}\|^2 - \|n_{ij}\|^2)$$

because the quantity inside the brackets on the right hand side is zero. The constant $\lambda$ is called the Lagrange multiplier. Using the orthogonality principle (see Appendix A), $\|g_{ij} \otimes\otimes f_{ij}\|^2$ is a minimum when

$$(g_{ij} \otimes\otimes f_{ij}) \odot\odot g_{ij} - \lambda(s_{ij} - p_{ij} \otimes\otimes f_{ij}) \odot\odot p_{ij} = 0$$

In Fourier space, this equation becomes

$$| G_{ij} |^2 F_{ij} - \lambda(S_{ij} P_{ij}^* - | P_{ij} |^2 F_{ij}) = 0$$

Solving for $F_{ij}$, we get

$$F_{ij} = \frac{S_{ij} P_{ij}^*}{| P_{ij} |^2 + \gamma | G_{ij} |^2}$$

where $\gamma$ is the reciprocal of the Lagrange multiplier ($= 1/\lambda$). Hence, the constrained least squares filter is given by

$$\text{Constrained Least Squares Filter} = \frac{P_{ij}^*}{| P_{ij} |^2 + \gamma | G_{ij} |^2}$$

The important point about this filter is that it allows the user to change $G_{ij}$ to suite a particular application. This filter can be thought of as a generalization of the other filters. For example, if $\gamma = 0$ then the inverse filter is obtained, if $\gamma = 1$ and $| G_{ij} |^2 = | N_{ij} |^2 / | F_{ij} |^2$ then the Wiener filter is obtained, and if $\gamma = 1$ and $| G_{ij} |^2 = | N_{ij} |^2 - | P_{ij} |^2$ then the matched filter is obtained.

Table 2.1: Filters list

| Name of Filter | Formula | Condition(s) |
|---|---|---|
| Inverse Filter | $Q_{ij} = P^*_{ij} / \mid P_{ij} \mid^2$ | Minimize$\|n_{ij}\|$ |
| Wiener Filter | $Q_{ij} = \frac{P^*_{ij}}{\|P_{ij}\|^2 + \|F_{ij}\|^2/\|N_{ij}\|^2}$ | Minimize $\|f_{ij} - q_{ij} \otimes\otimes s_{ij}\|^2$; $N^*_{ij}F_{ij} = 0, \quad F^*_{ij}N_{ij} = 0$ |
| PSE Filter | $Q_{ij} = \left(\frac{1}{\|P_{ij}\|^2 + \|F_{ij}\|^2/\|N_{ij}\|^2}\right)^{\frac{1}{2}}$ | $\mid F_{ij} \mid^2 = \mid Q_{ij}S_{ij} \mid^2$; $N^*_{ij}F_{ij} = 0, \quad F^*_{ij}N_{ij} = 0$ |
| Matched Filter | $Q_{ij} = P^*_{ij} / \mid N_{ij} \mid^2$ | Maximize $\frac{\|\sum_i \sum_j Q_{ij}P_{ij}\|^2}{\sum_i \sum_j \|N_{ij}\|^2 \|Q_{ij}\|^2}$ |
| Constrained Filter | $Q_{ij} = \frac{P^*_{ij}}{\|P_{ij}\|^2 + \gamma\|G_{ij}\|^2}$ | Minimize $\|g_{ij} \otimes\otimes f_{ij}\|^2$ |

The following table (2.1) lists the filters discussed so far. In each case, the filter $Q_{ij}$ provides a solution to the inversion of the following equation

$$s_{ij} = p_{ij} \otimes\otimes f_{ij} + n_{ij},$$

the solution for $f_{ij}$ being given by

$$f_{ij} = \text{IDFT}[Q_{ij}S_{ij}]$$

where IDFT stands for the 2D Discrete Inverse Fourier Transform and $S_{ij}$ is the DFT of the digital image $s_{ij}$. In all cases, the DFT and IDFT can be computed using a FFT.

### 2.4.6 Maximum Entropy Deconvolution

As before, we are interested in solving the imaging equation

$$s_{ij} = p_{ij} \otimes\otimes f_{ij} + n_{ij}$$

for the object $f_{ij}$. Instead of using a least squares error to constrain the solution for $f_{ij}$, we choose to find $f_{ij}$ such that the entropy $E$, given by

$$E = -\sum_i \sum_j f_{ij} \ln f_{ij}$$

is a maximum. Note, that because the ln function is used in defining the Entropy, the Maximum Entropy Method (MEM) must be restricted to cases where $f_{ij}$ is real and positive.

From the imaging equation above, we can write

$$s_{ij} - \sum_n \sum_m p_{i-n,j-m} f_{nm} = n_{ij}$$

where we have just written the convolution operation out in full. Squaring both sides and summing over $i$ and $j$ we can write

$$\sum_i \sum_j \left( s_{ij} - \sum_n \sum_m p_{i-n,j-m} f_{nm} \right)^2 - \sum_i \sum_j n_{ij}^2 = 0$$

But this equation is true for any constant $\lambda$ multiplying both terms on the left hand side. We can therefore write the equation for $E$ as

$$E = -\sum_i \sum_j f_{ij} \ln f_{ij}$$

$$+\lambda \left[ \sum_i \sum_j \left( s_{ij} - \sum_n \sum_m p_{n-i,m-j} f_{nm} \right)^2 - \sum_i \sum_j n_{ij}^2 \right]$$

because the second term on the right hand side is zero anyway (for all values of the Lagrange multiplier $\lambda$). Given this equation, our problem is to find $f_{ij}$ such that the entropy $E$ is a maximum, i.e.

$$\frac{\partial E}{\partial f_{ij}} = 0$$

Differentiating (an exercise which will be left to the reader), and switching to the notation for 2D convolution $\otimes\otimes$ and 2D correlation $\odot\odot$, we find that $E$ is a maximum when

$$1 + \ln f_{ij} - 2\lambda(s_{ij} \odot\odot p_{ij} - p_{ij} \otimes\otimes f_{ij} \odot\odot p_{ij}) = 0$$

or, after rearranging,

$$f_{ij} = \exp[-1 + 2\lambda(s_{ij} \odot\odot p_{ij} - p_{ij} \otimes\otimes f_{ij} \odot\odot p_{ij})]$$

This equation is transcendental in $f_{ij}$ and as such, requires that $f_{ij}$ is evaluated iteratively, i.e.

$$f_{ij}^{k+1} = \exp[-1 + 2\lambda(s_{ij} \odot\odot p_{ij} - p_{ij} \otimes\otimes f_{ij}^k \odot\odot p_{ij})]; \quad k = 0, 1, 2, ..., N$$

where $f_{ij}^0 = 0 \ \forall \ i, j$ say. The rate of convergence of this solution is determined by the value of the Lagrange multiplier that is used.

In general, the iterative nature of this nonlinear estimation method is undesirable, primarily because it is time consuming and may require many iterations before a solution is achieved with a desired tolerance.

We shall end this section by demonstrating a rather interesting result which is based on linearizing the MEM. This is achieved by retaining the first two terms (i.e. the linear terms) in the series representation of the exponential function leaving us with the following equation

$$f_{ij} = 2\lambda(s_{ij} \odot \odot p_{ij} - p_{ij} \otimes \otimes f_{ij} \odot \odot p_{ij})$$

Using the convolution and correlation theorems, in Fourier space, this equation becomes

$$F_{ij} = 2\lambda S_{ij} P^*_{ij} - 2\lambda \mid P_{ij} \mid^2 F_{ij}$$

Rearranging, we get

$$F_{ij} = \frac{S_{ij} P^*_{ij}}{\mid P_{ij} \mid^2 + 1/2\lambda}$$

Hence, we can define a linearized maximum entropy filter of the form

$$\frac{P^*_{ij}}{\mid P_{ij} \mid + 1/2\lambda}$$

Notice, that this filter is very similar to the Wiener filter. The only difference is that the Wiener filter is regularized by a constant determined by the SNR of the data whereas this filter is regularized by a constant determined by the Lagrange multiplier.

### 2.4.7 Bayesian Estimation

The processes discussed so far do not take into account the statistical nature of the noise inherent in a digital signal or image. To do this, another type of approach must be taken which is based on a result in probability theory called Bayes rule named after the English mathematician Thomas Bayes [13].

**The probability of an event**

Suppose we toss a coin, observe whether we get heads or tails and then repeat this process a number of times. As the number of trials increases, we expect that the number of times heads or tails occurs is half that of the number of trials. In other words, the probability of getting heads is 1/2 and the probability of

getting tails is also 1/2. Similarly, if a dice with six faces is thrown repeatedly, then the probability of it landing on any one particular face is 1/6. In general, if an experiment is repeated $N$ times and an event $A$ occurs $n$ times say, then the probability of this event $P(A)$ is defined as

$$P(A) = \lim_{N \to \infty} \left( \frac{n}{N} \right)$$

The probability is the relative frequency of an event as the number of trials tends to infinity. In practice, only a finite number of trials can be conducted and we therefore define the probability of an event $A$ as

$$P(A) \simeq \frac{n}{N}$$

where it is assumed that $N$ is large.

### The Joint Probability

Suppose we have two coins which we label $C_1$ and $C_2$. We toss both coins simultaneously $N$ times and record the number of times $C_1$ is heads, the number of times $C_2$ is heads and the number of times $C_1$ and $C_2$ are heads together. What is the probability that $C_1$ and $C_2$ are heads together? Clearly, if $m$ is the number of times out of $N$ trials that heads occurs simultaneously, then the probability of such an event must be given by

$$P(C_1 \text{ heads and } C_2 \text{ heads}) = \frac{m}{N}$$

This is known as the joint probability of $C_1$ being heads when $C_2$ is heads. In general, if two events $A$ and $B$ are possible and $m$ is the number of times both events occur simultaneously, then the joint probability is given by

$$P(A \text{ and } B) = \frac{m}{N}$$

### The Conditional Probability

Suppose we setup an experiment in which two events $A$ and $B$ can occur. We conduct $N$ trials and record the number of times $A$ occurs (which is $n$) and the number of times $A$ and $B$ occur simultaneously (which is $m$). In this case, the joint probability may written as

$$P(A \text{ and } B) = \frac{m}{N} = \frac{m}{n} \times \frac{n}{N}$$

Now, the quotient $n/N$ is the probability $P(A)$ that event $A$ occurs. The quotient $m/n$ is the probability that events $A$ and $B$ occur simultaneously given that event

$A$ has occurred. The latter probability is known as the conditional probability and is written as

$$P(B \mid A) = \frac{m}{n}$$

where the symbol $B \mid A$ means '$B$ given $A$'. Hence, the joint probability can be written as

$$P(A \text{ and } B) = P(A)P(B \mid A)$$

Suppose that we do a similar type of experiment but this time we record the number of times $p$ that event $B$ occurs and the number of times $q$ that event $A$ occurs simultaneously with event $B$. In this case, the joint probability of events $B$ and $A$ occurring together is given by

$$P(B \text{ and } A) = \frac{q}{N} = \frac{q}{p} \times \frac{p}{N}$$

The quotient $p/N$ is the probability $P(B)$ that event $B$ occurs and the quotient $q/p$ is the probability of getting events $B$ and $A$ occurring simultaneously given that event $B$ has occurred. The latter probability is just the probability of getting '$A$ given $B$', i.e.

$$P(A \mid B) = \frac{q}{p}$$

Hence, we have

$$P(B \text{ and } A) = P(B)P(A \mid B)$$

**Bayes Rule**

The probability of getting $A$ and $B$ occurring simultaneously is exactly the same as getting $B$ and $A$ occurring simultaneously, i.e.

$$P(A \text{ and } B) = P(B \text{ and } A)$$

Hence, by using the definition of these joint probabilities in terms of the conditional probabilities we arrive at the following formula

$$P(A)P(B \mid A) = P(B)P(A \mid B)$$

or alternatively

$$P(B \mid A) = \frac{P(B)P(A \mid B)}{P(A)}$$

This result is known as Bayes rule. It relates the conditional probability of '$B$ given $A$' to that of '$A$ given $B$'.

## Bayesian Estimation in Signal and Image Processing

In signal and image analysis Bayes rule is written in the form

$$P(f \mid s) = \frac{P(f)P(s \mid f)}{P(s)}$$

where $f$ is the object that we want to recover from the signal

$$s(x) = p(x) \otimes f(x) + n(x)$$

or image

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

This result is the basis for a class of restoration methods which are known collectively as Bayesian estimators.

Bayesian estimation attempts to recover $f$ in such a way that the probability of getting $f$ given $s$ is a maximum. In practice, this is done by assuming that $P(f)$ and $P(s \mid f)$ obey certain statistical distributions which are consistent with the experiment in which $s$ is measured. In other words, models are chosen for $P(f)$ and $P(s \mid f)$ and then $f$ is computed at the point where $P(f \mid s)$ reaches its maximum value. This occurs when

$$\frac{\partial}{\partial f} P(f \mid s) = 0$$

The function $P$ is the Probability Density Function (PDF). The PDF $P(f \mid s)$ is called the *a posteriori* PDF. Since the logarithm of a function varies monotonically with that function, the *a posteriori* PDF is also a maximum when

$$\frac{\partial}{\partial f} \ln P(f \mid s) = 0$$

Now, using Bayes rule, we can write this equation as

$$\frac{\partial}{\partial f} \ln P(s \mid f) + \frac{\partial}{\partial f} \ln P(f) = 0$$

Because the solution to this equation for $f$ maximizes the *a posteriori* PDF, this method is known as the Maximum *a Posteriori* or MAP method. To illustrate the principles of Bayesian estimation, we shall now present some simple examples of how this technique can be applied to data analysis.

### Bayesian Estimation - Example 1

Suppose that we measure a single sample $s$ (one real number) in an experiment where it is known *a priori* that

$$s = f + n$$

where $n$ is noise (a single random number). Suppose that it is also known *a priori* that the noise is determined by a Gaussian distribution of the form (ignoring scaling)

$$P(n) = \exp(-n^2/\sigma_n^2)$$

where $\sigma_n^2$ is the standard deviation of the noise. Now, the probability of measuring $s$ given $f$ - i.e. the conditional probability $P(s \mid f)$ - is determined by the noise since

$$n = s - f$$

We can therefore write

$$P(s \mid f) = \exp[-(s-f)^2/\sigma_n^2]$$

To find the MAP estimate, the PDF for $f$ must also be known. Suppose that $f$ also has a zero-mean Gaussian distribution of the form

$$P(f) = \exp(-f^2/\sigma_f^2)$$

Then,

$$\frac{\partial}{\partial f} \ln P(s \mid f) + \frac{\partial}{\partial f} \ln P(f) = \frac{2(s-f)}{\sigma_n^2} - \frac{2f}{\sigma_f^2} = 0$$

Solving this equation for $f$ gives

$$f = \frac{s\Gamma^2}{1+\Gamma^2}$$

where $\Gamma$ is the SNR defined by

$$\Gamma = \frac{\sigma_f}{\sigma_n}$$

Notice, that as $\sigma_n \to 0$, $f \to s$ which must be true since $s = f + n$ and $n$ has a zero-mean Gaussian distribution. Also, note that the solution we acquire for $f$ is entirely dependent on the *a priori* information we have on the PDF for $f$. A different PDF produces an entirely different solution. For example, suppose it is known or we have good reason to assume that $f$ obeys a Rayleigh distribution of the form

$$P(f) = f\exp(-f^2/\sigma_f^2), \quad f \geq 0$$

In this case,

$$\frac{\partial}{\partial f}\ln P(f) = \frac{1}{f} - \frac{2f}{\sigma_f^2}$$

and assuming that the noise obeys the same zero-mean Gaussian distribution

$$\frac{\partial}{\partial f}\ln P(s \mid f) + \frac{\partial}{\partial f}\ln P(f) = \frac{2(s-f)}{\sigma_n^2} + \frac{1}{f} - \frac{2f}{\sigma_f^2} = 0$$

This equation is quadratic in $f$. Solving it, we get

$$f = \frac{s\Gamma^2}{2(1+\Gamma^2)}\left(1 \pm \sqrt{1 + \frac{2\sigma_n^2}{s^2}\left(1 + \frac{1}{\Gamma^2}\right)}\right)$$

The solution for $f$ which maximizes the value of $P(f \mid s)$, can then be written in the form

$$f = \frac{s}{2a}\left(1 + \sqrt{1 + \frac{2a\sigma_n^2}{s^2}}\right)$$

where

$$a = 1 + \frac{1}{\Gamma^2}$$

This is a nonlinear estimate for $f$. If

$$\frac{\sigma_n\sqrt{2a}}{s} << 1$$

then

$$f \simeq \frac{s}{a}$$

In this case, $f$ is linearly related to $s$. In fact, this linearized estimate is identical to the MAP estimate obtained earlier where it was assumed that $f$ had a Gaussian distribution.

From the example given above, it should now be clear that Bayesian estimation (i.e. the MAP method) is only as good as the *a priori* information on the statistical behaviour of $f$ - the object for which we seek a solution. However, when $P(f)$ is broadly distributed compared with $P(s \mid f)$, the peak value of the *a posteriori* PDF will lie close to the peak value of $P(f)$. In particular, if $P(f)$ is roughly constant, then

$$\frac{\partial}{\partial f}\ln P(f)$$

is close to zero and therefore

$$\frac{\partial}{\partial f}\ln P(f \mid s) \simeq \frac{\partial}{\partial f}\ln P(s \mid f)$$

In this case, the *a posteriori* PDF is a maximum when

$$\frac{\partial}{\partial f} \ln P(s \mid f) = 0$$

The estimate for $f$ that is obtained by solving this equation for $f$ is called the Maximum Likelihood or ML estimate. To obtain this estimate, only *a priori* knowledge on the statistical fluctuations of the conditional probability is required. If, as in the previous example, we assume that the noise is a zero-mean Gaussian distribution, then the ML estimate is given by

$$f = s$$

Note that this is the same as the MAP estimate when the standard deviation of the noise is zero.

The basic and rather important difference between the MAP and ML estimates is that the ML estimate ignores *a priori* information about the statistical fluctuations of the object $f$. It only requires a model for the statistical fluctuations of the noise. For this reason, the ML estimate is usually easier to compute. It is also the estimate to use in cases where there is a complete lack of knowledge about the statistical behaviour of the object.

**Bayesian Estimation - Example 2**

To further illustrate the difference between the MAP and ML estimate and to show their use in signal analysis, consider the case where we measure $N$ samples of a real signal $s_i$ in the presence of additive noise $n_i$ which is the result of transmitting a known signal $f_i$ modified by a random amplitude factor $a$. The samples of the signal are given by

$$s_i = af_i + n_i, \quad i = 1, 2, ..., N$$

The problem is to find an estimate for $a$. To solve problems of this type using Bayesian estimation, we must introduce multidimensional probability theory. In this case, the PDF is a function of not just one number $s$ but a set of numbers $s_1, s_2, ..., s_N$. It is therefore a vector space. To emphasize this, we use the vector notation

$$P(\mathbf{s}) \equiv P(s_i) \equiv P(s_1, s_2, s_3, ..., s_N)$$

The ML estimate is given by solving the equation

$$\frac{\partial}{\partial a} \ln P(\mathbf{s} \mid a) = 0$$

for $a$. Let us again assume that the noise is described by a zero-mean Gaussian distribution of the form

$$P(\mathbf{n}) \equiv P(n_1, n_2, ..., n_N) = \exp\left(-\frac{1}{\sigma_n^2}\sum_{i=1}^{N} n_i^2\right)$$

The conditional probability is then given by

$$P(\mathbf{s} \mid a) = \exp\left(-\frac{1}{\sigma_n^2}\sum_{i=1}^{N}(s_i - af_i)^2\right)$$

and

$$\frac{\partial}{\partial a}\ln P(\mathbf{s} \mid a) = \frac{2}{\sigma_n^2}\sum_{i=1}^{N}(s_i - af_i)f_i = 0$$

Solving this last equation for $a$ we obtain the ML estimate

$$a = \frac{\sum_{i=1}^{N} s_i f_i}{\sum_{i=1}^{N} f_i^2}$$

The **MAP estimate** is obtained by solving the equation

$$\frac{\partial}{\partial a}\ln P(\mathbf{s} \mid a) + \frac{\partial}{\partial a}\ln P(a) = 0$$

for $a$. Using the same distribution for the conditional PDF, let us assume that $a$ has a zero-mean Gaussian distribution of the form

$$P(a) = \exp(-a^2/\sigma_a^2)$$

where $\sigma_a^2$ is the standard deviation. In this case,

$$\frac{\partial}{\partial a}\ln P(a) = -\frac{2a}{\sigma_a^2}$$

and hence, the MAP estimate is obtained by solving the equation

$$\frac{\partial}{\partial a}\ln P(\mathbf{s} \mid a) + \frac{\partial}{\partial a}\ln P(a)$$

$$= \frac{2}{\sigma_n^2}\sum_{i=1}^{N}(s_i - af_i)f_i - \frac{2a}{\sigma_a^2} = 0$$

for $a$. The solution to this equation is given by

$$a = \frac{\frac{\sigma_a^2}{\sigma_n^2}\sum_{i=1}^{N} s_i f_i}{1 + \frac{\sigma_a^2}{\sigma_n^2}\sum_{i=1}^{N} f_i^2}$$

Note, that if $\sigma_a >> \sigma_n$, then,

$$a \simeq \frac{\sum_{i=1}^{N} s_i f_i}{\sum_{i=1}^{N} f_i^2}$$

which is the same as the ML estimate.

## The Maximum Likelihood Filter

In the last section, the principles of Bayesian estimation were presented. We shall now use these principles to design deconvolution algorithms for digital images under the assumption that the statistics are Gaussian. The problem is as follows: Given the real digital image

$$s_{ij} = \sum_n \sum_m p_{i-n,j-m} f_{nm} + n_{ij}$$

find an estimate for $f_{ij}$ when $p_{ij}$ is known together with the statistics for $n_{ij}$. In this section, the ML estimate for $f_{ij}$ is determined by solving the equation

$$\frac{\partial}{\partial f_{ij}} \ln P(s_{ij} \mid f_{ij}) = 0$$

As before, the algebraic form of the estimate depends upon the model that is chosen for the PDF. Let us assume that the noise has a zero-mean Gaussian distribution. In this case, the conditional PDF is given by

$$P(s_{ij} \mid f_{ij}) = \exp\left[-\frac{1}{\sigma_n^2} \sum_i \sum_j \left(s_{ij} - \sum_n \sum_m p_{i-n,j-m} f_{nm}\right)^2\right]$$

where $\sigma_n^2$ is the standard deviation of the noise. Substituting this result into the previous equation and differentiating, we get

$$\frac{2}{\sigma_n^2} \sum_i \sum_j \left(s_{ij} - \sum_n \sum_m p_{i-n,j-m} f_{nm}\right) p_{i-k,j-\ell} = 0$$

or

$$\sum_i \sum_j s_{ij} p_{i-k,j-\ell} = \sum_i \sum_j \left(\sum_n \sum_m p_{i-n,j-m} f_{nm}\right) p_{i-k,j-\ell}$$

Using the appropriate symbols, we may write this equation in the form

$$s_{ij} \odot\odot p_{ij} = (p_{ij} \otimes\otimes f_{ij}) \odot\odot p_{ij}$$

where $\odot\odot$ and $\otimes\otimes$ denote the 2D correlation and convolution sums respectively. The ML estimate is obtained by solving the equation above for $f_{ij}$. This can be

done by transforming it into Fourier space. Using the correlation and convolution theorems, in Fourier space this equation becomes

$$S_{ij}P^*_{ij} = (P_{ij}F_{ij})P^*_{ij}$$

and thus

$$f_{ij} = \text{IDFT}(F_{ij}) = \text{IDFT}\left(\frac{S_{ij}P^*_{ij}}{|\ P_{ij}\ |^2}\right)$$

where IDFT is taken to denote the (2D) Inverse Discrete Fourier Transform. Hence for Gaussian statistics, the ML filter is given by

$$\text{ML Filter} = \frac{P^*_m}{|\ P_m\ |^2}$$

which is identical to the inverse filter.

**The Maximum a Posteriori Filter**

This filter is obtained by finding $f_{ij}$ such that

$$\frac{\partial}{\partial f_{k\ell}} \ln P(s_{ij} \mid f_{ij}) + \frac{\partial}{\partial f_{k\ell}} \ln P(f_{ij}) = 0$$

Consider the following models for the PDFs

(i) Gaussian statistics for the noise

$$P(s_{ij} \mid f_{ij}) = \exp\left[-\frac{1}{\sigma_n^2}\sum_i\sum_j\left(s_{ij} - \sum_n\sum_m p_{i-n,j-m}f_{nm}\right)^2\right]$$

(ii) Gaussian statistics for the object

$$P(f_{ij}) = \exp\left(-\frac{1}{\sigma_f^2}\sum_i\sum_j f_{ij}^2\right)$$

By substituting these expressions for $P(s_{ij} \mid f_{ij})$ and $P(f_{ij})$ into the equation above, we obtain

$$\frac{2}{\sigma_n^2}\sum_i\sum_j\left(s_{ij} - \sum_n\sum_m p_{i-n,j-m}f_{nm}\right)p_{i-k,j-\ell} - \frac{2}{\sigma_f^2}f_{k\ell} = 0$$

Rearranging, we may write this result in the form

$$s_{ij} \odot\odot p_{ij} = \frac{\sigma_n^2}{\sigma_f^2}f_{ij} + (p_{ij} \otimes\otimes f_{ij}) \odot p_{ij}$$

In Fourier space, this equation becomes

$$S_{ij}P^*_{ij} = \frac{1}{\Gamma^2}F_{ij} + \mid P_{ij} \mid^2 F_{ij}$$

where

$$\Gamma = \frac{\sigma_f}{\sigma_n}$$

The MAP filter for Gaussian statistics is therefore given by

$$\text{MAP Filter} = \frac{P^*_{ij}}{\mid P_{ij} \mid^2 + 1/\Gamma^2}$$

Note, that this filter is the same as the Wiener filter under the assumption that the power spectra of the noise and object are constant. Also, note that

$$\lim_{\sigma_n \to 0}(\text{MAP Filter}) = \text{ML Filter}$$

### 2.4.8 Reconstruction of Bandlimited Images

A bandlimited function is a function whose spectral bandwidth is finite. Most real signals and images are bandlimited functions. This leads one to consider the problem of how the bandwidth and hence the resolution of a bandlimited image, can be increased synthetically using digital processing techniques. In other words, how can we extrapolate the spectrum of a bandlimited function from an incomplete sample.

Solutions to this type of problem are important in image analysis where a resolution is needed that is not an intrinsic characteristic of the image provided and is difficult or even impossible to achieve experimentally. The type of resolution that is obtained by spectral extrapolation is referred to as super resolution.

Because sampled data are always insufficient to specify a unique solution and since no algorithm is able to reconstruct equally well all characteristics of an image, it is essential that the user is able to play a role in the design and execution of an algorithm and incorporate maximum knowledge of the expected features. This allows optimum use to be made of the available data and the users experience, judgement and intuition. Hence, an important aspect of practical solutions to the spectral extrapolation problem is the incorporation of *a priori* information on the structure of an object.

In this section, an algorithm is discussed which combines *a priori* information with the least squares principle to reconstruct a two dimensional function from

limited (i.e. incomplete) Fourier data. This algorithm is essentially a modified version of the Gerchberg-Papoulis algorithm to accommodate a user defined weighting function.

### The Gerchberg-Papoulis Method

Let us consider the case where we have an image $f(x, y)$ characterized by a discrete spectrum $F_{nm}$ [5, 90] which is composed of a finite number of samples:

$$-\frac{N}{2} \leq n \leq \frac{N}{2}$$

$$-\frac{M}{2} \leq m \leq \frac{M}{2}$$

These data are related to the image by the equation

$$F_{nm} = \int_{-X}^{X} \int_{-Y}^{Y} f(x, y) e^{-i(k_n x + k_m y)} dx dy$$

Here, $f$ is assumed to be of finite support $X$ and $Y$, i.e.,

$$| x | \leq X \quad \text{and} \quad | y | \leq Y$$

and $k_n, k_m$ are discrete spatial frequencies. With this data, we can define the BandLimited function

$$f_{BL}(x, y) = \sum_n \sum_m F_{nm} e^{i(k_n x + k_m y)}$$

which is related to $F_{nm}$ by a two-dimensional Fourier Series. Our problem is to reconstruct $f$ given $F_{nm}$ or equivalently, $f_{BL}$. In this section, a solution to this problem is presented using the least squares principle. First, we consider a model for an estimate $\hat{f}$ of $f$ given by

$$\hat{f}(x, y) = \sum_n \sum_m A_{nm} e^{i(k_n x + k_m y)} \tag{2.7}$$

This model is just a two-dimensional Fourier series representation of the object. Given this model, our problem is reduced to that of fining the coefficients $A_{nm}$. Using the least squares method, we compute $A_{nm}$ by minimizing the mean square error

$$e = \int_{-X}^{X} \int_{-Y}^{Y} | f(x, y) - \hat{f}(x, y) |^2 \, dx dy$$

This error is a minimum when

$$\frac{\partial e}{\partial A_{nm}} = 0$$

Differentiating, we obtain (see Appendix A)

$$\frac{\partial e}{\partial A_{pq}} = \frac{\partial}{\partial A_{pq}} \int_{-X}^{X} \int_{-Y}^{Y} \left| f(x,y) - \sum_n \sum_m A_{nm} e^{i(k_n x + k_m y)} \right|^2 dx dy$$

$$= \int_{-X}^{X} \int_{-Y}^{Y} \left( f(x,y) - \sum_n \sum_m A_{nm} e^{i(k_n x + k_m y)} \right) e^{-i(k_p x + k_q y)} dx dy$$

Thus, $e$ is a minimum when

$$\int_{-X}^{X} \int_{-Y}^{Y} f(x,y) e^{-i(k_p x + k_q y)} dx dy$$

$$= \sum_n \sum_m A_{nm} \int_{-X}^{X} \int_{-Y}^{Y} e^{-i(k_p - k_n)x} e^{-i(k_q - k_m)y} dx dy$$

The left hand side the above equation is just the Fourier data $F_{pq}$. Hence, after evaluating the integrals on he right hand side, we get

$$F_{pq} = 4XY \sum_n \sum_m A_{nm} \operatorname{sinc}[(k_p - k_n)X] \operatorname{sinc}[(k_q - k_m)Y] \qquad (2.8)$$

The estimate $\hat{f}(x,y)$ can be computed by solving the equation above for the coefficients $A_{nm}$. This is a two-dimensional version of the Gerchberg-Papoulis method and is a least squares approximation of $f(x,y)$.

## Incorporation of a Priori Information

Since we have considered an image $f$ of finite support, we can write equation 2.7 in the following 'closed form':

$$\hat{f}(x,y) = w(x,y) \sum_n \sum_m A_{nm} e^{i(k_n x + k_m y)} \qquad (2.9)$$

where

$$w(x,y) = \begin{cases} 1, & |x| \leq X, \ |y| \leq Y; \\ 0, & |x| > X, \ |y| > Y. \end{cases}$$

Writing it in this form, we observe that $w$ (i.e. essentially the values of $X$ and $Y$) represents a simple but crucial form of *a priori* information. This information

is required to compute the sinc functions given in equation 2.8 and hence the coefficients $A_{nm}$. Note, that the sinc functions (in particular the zero locations) are sensitive to the precise values of $X$ and $Y$ and hence small errors in $X$ and $Y$ can dramatically effect the computation of $A_{nm}$. In other words, equation 2.8 is ill-conditioned.

The algebraic form of equation 2.9 suggests incorporating further *a priori* information into the 'weighting function' $w$ in addition to the support of the object $f$. We therefore consider an estimate of the form

$$\hat{f}(x,y) = w(x,y) \sum_n \sum_m A_{nm} e^{i(k_n x + k_m y)}$$

where $w$ is now a generalized weighting function composed of limited *a priori* information on the structure of $f$. If we now employ a least squares method to find $A_{nm}$ based on the previous mean square error function, we obtain the following equation

$$\int_{-X}^{X} \int_{-Y}^{Y} f(x,y) w(x,y) e^{-i(k_p x + k_q y)} dx dy$$

$$= \sum_n \sum_m A_{nm} \int_{-X}^{X} \int_{-Y}^{Y} [w(x,y)]^2 e^{-i(k_p - k_n)x} e^{-i(k_q - k_m)y} dx dy$$

The problem with this result is that the data on the left hand side is not the same as the Fourier data provided $F_{pq}$. In other words, the result is not 'data consistent'. To overcome this problem we introduce a modified version of the least square method which involves minimizing the error

$$e = \int_{-X}^{X} \int_{-Y}^{Y} | f(x,y) - \hat{f}(x,y) |^2 \frac{1}{w(x,y)} dx dy \tag{2.10}$$

In this case, we find that $e$ is a minimum when

$$F_{pq} = \sum_n \sum_m A_{nm} W_{p-n,q-m} \tag{2.11}$$

where

$$W_{p-n,q-m} = \int_{-X}^{X} \int_{-Y}^{Y} w(x,y) e^{-i(k_p - k_n)x} e^{-i(k_q - k_m)y} dx dy$$

Equation 2.11 is data consistent, the right hand side of this equation being a discrete convolution of $A_{nm}$ with $W_{nm}$. Hence, using the notation for convolution, we may write this equation in the form

$$F_{nm} = A_{nm} \otimes \otimes W_{nm}$$

Using the convolution theorem, in real space, this equation becomes

$$f_{BL}(x,y) = a(x,y) w_{BL}(x,y)$$

where

$$f_{BL}(x,y) = \sum_n \sum_m F_{nm} e^{i(k_n x + k_m y)}$$

$$w_{BL}(x,y) = \sum_n \sum_m W_{nm} e^{i(k_n x + k_m y)}$$

and

$$a(x,y) = \sum_n \sum_m A_{nm} e^{i(k_n x + k_m y)}$$

Now, since

$$\hat{f}(x,y) = w(x,y) \sum_n \sum_m A_{nm} e^{i(k_n x + k_m y)} = w(x,y) a(x,y)$$

we obtain the simple algebraic result

$$\hat{f}(x,y) = \frac{w(x,y)}{w_{BL}(x,y)} f_{BL}(x,y)$$

Here $w_{BL}$ is a bandlimited weighting function, bandlimited by the same extent as $f_{BL}$.

The algorithm presented above is based on an inverse weighted least squares error [i.e. equation 2.10]. It is essentially an adaption of the Gerchberg-Papoulis method, modified to:

(i) accommodate a generalized weighting function $w(x,y)$;

(ii) provide data consistency [i.e. equation 2.11].

The weighting function $w(x,y)$ can be used to encode as much information as is available on the structural characteristics of $f(x,y)$. Since equation 2.10 involves $1/w(x,y)$, $w(x,y)$ must be confined to being a positive non-zero function. We can summarize this algorithm in the form

$$\text{reconstruction} = \frac{\text{bandlimited image} \times a \ priori \text{ information}}{\text{bandlimited } a \ priori \text{ information}}$$

Clearly, the success of this algorithm depends on the quality of the *a priori* information that is available, just as the performance of the Wiener filter or MEM depends upon *a priori* information on the functional form of the Point Spread Function.

### 2.4.9 Rotationally Symmetric Systems

An optical system is referred to as a rotationally symmetric system if the point spread function and OTF are rotationally symmetric, i.e.

$$p(r), \quad r = \sqrt{x^2 + y^2}$$

$$P(\omega), \quad \omega = \sqrt{u^2 + v^2}$$

For a rotationally symmetric system, the line spread function is the same for all angles of the line input, and is related to the point spread function by

$$\ell \leftarrow \text{Abel transform} \rightarrow p(r)$$

The OTF is the Fourier transform of the line spread function.

$$P(\omega) \leftarrow \text{1D FT} \rightarrow \ell(x)$$

The rotational symmetry of the 2D OTF and point spread function means that, since they are 2D Fourier transform pairs, they are 1D Hankel transform pairs

$$P(\omega) \leftarrow \text{Hankel transform} \rightarrow p(r)$$

### 2.4.10 Linearity of Optical Imaging Systems

Consider the case where the object plane is illuminated by a plane or spherical wave - by perfectly spatially coherent light. Let the complex amplitude immediately after the object by denoted by $U_{\text{in}}(x, y)$ and $U_{\text{out}}(x, y)$ be the complex amplitude at the image plane. Also, let the complex amplitude at $(x, y)$ in the output due to a unit strength point at in the input be $p(x, y; x', y')$. The total amplitude at $(x, y)$ due to all such point in the object plane is then given by

$$U_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_{\text{in}}(x', y') p(x, y; x', y') dx' dy'$$

For an isoplanatic optical system, this reduces to

$$U_{\text{out}}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_{\text{in}}(x',y')p(x-x',y-y')dx'dy'$$

**A spatially coherent optical system is linear in the complex amplitude**

Consider the case of narrowband light that is **not** perfectly spatially coherent. The general complex representation of the time-varying scalar field is called the analytic signal $V(\mathbf{r},t)$; it is defined such that

$$\text{real scalar field} = \text{Re}[V(\mathbf{r},t)]$$

For narrowband light, the analytic signal can be written as a product of a 'slowly varying' function - the time varying complex amplitude $U(\mathbf{r},t)$ times $\exp(-i\omega t)$:

$$V(\mathbf{r},t) = U(\mathbf{r},t)\exp(-i\omega t)$$

The instantaneous intensity is defined as

$$I(\mathbf{r},t) = \mid U(\mathbf{r},t) \mid^2$$

whereas the time-averaged intensity $\bar{I}(\mathbf{r})$ (i.e. that observed by an optical detector), is equal to

$$\bar{I}(\mathbf{r}) = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} I(\mathbf{r},t)dt$$

In general, the time-varying complex amplitudes are related by

$$U_{\text{out}}(x,y,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_{\text{in}}(x',y',t)p(x,y;x'y')dx'dy'$$

Coherent illumination implies that $U(x,y,t) = U(x,y)$ - the field does not vary in time. The average intensity is therefore given by

$$\bar{I_{\text{out}}}(x,y) = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} \mid U_{\text{out}}(x,y,t) \mid^2 dt$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y;x',y')p^*(x,y;x',y')$$

$$\times \left[ \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} U_{\text{in}}(x', y', t) U_{\text{in}}^{*}(x'', y'', t) dt \right] dx' dy' dx'' dy''$$

In some sources is called the **mutual intensity** of the narrow band light

$$J_{\text{in}}(x', y'; x'', y'') = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} U_{\text{in}}(x', y', t) U_{\text{in}}^{*}(x'', y'', t) dt$$

or

$$J_{in}(\mathbf{r}', \mathbf{r}'') = \langle U_{\text{in}}(\mathbf{r}', t) U_{\text{in}}^{*}(\mathbf{r}'', t) \rangle$$

**Incoherent** light is defined to be such that

$$J(\mathbf{r}', \mathbf{r}'') = \bar{I}(\mathbf{r}') \delta(\mathbf{r}' - \mathbf{r}'')$$

That is, two neighbouring points $\mathbf{r}'$ and $\mathbf{r}''$ have **uncorrelated** fields, for any $\mathbf{r}' \neq \mathbf{r}''$. Using the definition for incoherent light above, the expression for $\bar{I}_{\text{out}}$ becomes

$$\bar{I}_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y; x', y') p^{*}(x, y; x', y')$$

$$\times \bar{I}_{\text{in}}(x', y') \delta(x' - x'') \delta(y' - y'') dx' dy' dx'' dy''$$

or

$$\bar{I}_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} | p(x, y; x', y') |^2 \bar{I}_{\text{in}}(x', y') dx' dy'$$

The quantity $| p(x, y; x', y') |^2$ is the **intensity** point spread function.

For an isoplanatic optical system, this result reduces to

$$\bar{I}_{\text{out}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \bar{I}_{\text{in}}(x', y') | p(x - x', y - y') |^2 dx' dy'$$

The bar over $I$ is usually omitted when referring to the intensity -a time average is always assumed.

For perfectly **incoherent** illumination, an optical system is linear in **intensity** and, if isoplanicity holds, the output (image) intensity is equal to the input (object) intensity convolved with the intensity point spread function.

### 2.4.11 Coherent Image Formation

In coherent light, the complex amplitude of the image is equal to that at the object plane convolved with the amplitude point spread function (for an isoplanatic system), i.e.

$$U_{\text{out}}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_{\text{in}}(x',y')p(x-x',y-y')dx'dy'$$

where

$$p(x,y) = C\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v)\exp\left[-i\frac{2\pi}{\lambda z}(ux+vy)\right]dudv$$

and $P$ is the **Pupil Function** of the optical system, i.e. the complex amplitude in the exit pupil. The constant $C$ is usually chosen so that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)dxdy = P(0,0) = 1$$

The pupil function $P$ is, for a clear pupil, defined by

$$P(u,v) = \begin{cases} \exp[ikW(u,v)], & (u,v) \in \text{aperture;} \\ 0, & \text{otherwise.} \end{cases}$$

so that $P(0,0) = 1$ and therefore $C = 1$. The function $W$ is called the **Wave Aberration Function.**

A shaded or **apodized** pupil can be handled by introducing an absorption term $A$,

$$P(u,v) = A(u,v)\exp[ikW(u,v)]$$

Taking the Fourier transform of $U_{\text{out}}$ and using the convolution theorem we can write

$$\widetilde{U_{\text{out}}}(u,v) = \widetilde{U_{\text{in}}}(u,v)T(u,v)$$

where

$\widetilde{U_{\text{out}}}$ - spectrum of image amplitude

$\widetilde{U_{\text{in}}}$ - spectrum of object amplitude

$T$ - Coherent Optical Transfer Function (COTF)

Note that

$$T(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\exp[-2\pi i(ux+vy)]dxdy = P(\lambda zu, \lambda zv)$$

That is, the COTF at spatial frequency $(u, v)$ is simply **equal** to the pupil function at coordinates $\lambda z u, \lambda z v)$.

### Examples of Coherent Image Formation

**Example 1.** Aberration free, circular pupil of radius $a$.

$$T(\omega) = \begin{cases} 1, & \omega \leq a; \\ 0, & \omega > a. \end{cases}$$

where

$$\omega = \sqrt{u^2 + v^2}$$

or

$$P(u, v) = \begin{cases} 1, & \omega \leq \frac{a}{\lambda z}; \\ 0, & \omega > \frac{a}{\lambda z}. \end{cases}$$

Defining

$$\frac{a}{\lambda z} = \frac{1}{2\lambda F}$$

where $F$ is the 'F-number' we get

$$\widetilde{U_{\text{out}}} = \begin{cases} \widetilde{U_{\text{in}}}, & (u, v) \leq \frac{1}{2\lambda F}; \\ 0, & (u, v) > \frac{1}{2\lambda F}. \end{cases}$$

In this case, there is a specific frequency cutoff (at $\sqrt{u^2 + v^2} = \frac{1}{2\lambda F}$) beyond which no frequencies pass. Spatial features associated with spatial frequencies greater than $\frac{1}{2\lambda F}$ are not imaged. Thus, for example, a 1D amplitude sinusoid of frequency $\leq \frac{1}{2\lambda F}$ has an image amplitude exactly equal to the object amplitude but for a sinusoid of frequency $> \frac{1}{2\lambda F}$, the image amplitude is zero!

**Example 2.** Find the image intensity when an object of amplitude transmittance

$$U_{\text{in}}(x) = 1 + \cos(2\pi x u_0)$$

is imaged in an aberration free coherent optical system, with $u_0 < \frac{1}{2\lambda F}$, with and without a half-plane obstruction in the pupil

**Without** the half-plane obstruction, since $u_0 < \frac{1}{2\lambda F}$

$$\widetilde{U_{\text{out}}}(u, v) = \widetilde{U_{\text{in}}}(u, v)$$

so

$$U_{\text{out}}(x, y) = U_{\text{in}}(x, y) = 1 + \cos(2\pi x u_0)$$

and the intensity is given by

$$I_{\text{out}}(x,y) = \mid 1 + \cos(2\pi x u_0) \mid^2$$

$$= [2\cos^2(\pi x u_0)]^2$$

i.e. a periodic image of frequency $u_0$ equal to the original amplitude frequency.

**With** the obstruction

$$\widetilde{U_{\text{out}}}(u,v) = \begin{cases} \widetilde{U_{\text{in}}}(u,v), & u > 0; \\ 0, & u \leq 0. \end{cases}$$

Now

$$\widetilde{U_{\text{in}}} = \delta(u,v) + \frac{1}{2}\delta(u-u_0) + \frac{1}{2}\delta(u+u_0)$$

and therefore

$$\widetilde{U_{\text{out}}}(u,v) = \frac{1}{2}\delta(u-u_0)$$

$$U_{\text{out}}(x,y) = \frac{1}{2}\exp(-2\pi i u x_0)$$

and

$$I_{\text{out}}(x,y) = \frac{1}{4} \quad \text{uniform intensity}$$

**Example 3.** Effect of aberrations (for a clear pupil)

The COTF is given by

$$T(u,v) = \begin{cases} \exp[ikW(u,v)], & (u,v) \in \text{aperture}; \\ 0, & \text{otherwise}. \end{cases}$$

$\mid T \mid$ is unaffected by aberrations - each sine wave is transmitted without attenuation, provided $(u,v) < \frac{1}{2\lambda F}$. However, aberrations do give a **phase** contribution to the COTF, yielding a shift of spatial frequency components in the image. This shift can markedly affect the intensity distribution of the image. Consider a single cosinusoidal object with $u_0 < \frac{1}{2\lambda F}$ given by

$$U_{\text{in}}(x,y) = 2\cos(2\pi u_0 x)$$

Then

$$\widetilde{U_{\text{in}}}(u,v) = [\delta(u-u_0) + \delta(u+u_0)]\delta(v)$$

$$\widetilde{U_{\text{out}}}(u,v) = \exp[ikW(u,v)]\widetilde{U_{\text{in}}}(u,v)$$

$$= \exp[ikW(u_0,0)]\delta(u-u_0) + \exp[ikW(-u_0,0)]\delta(u+u_0)$$

For an **even distribution** $W(u_0,v) = W(-u_0,v)$. Thus

$$\widetilde{U_{\text{out}}}(u,v) = \exp[ikW(u_0,0)][\delta(u-u_0) + \delta(u+u_0)]$$

$$U_{\text{out}}(x,y) = \exp[ikW(u_0,0)]2\cos(2\pi u_0 x)$$

and

$$I_{\text{out}} = [2\cos(2\pi u_0 x)]^2$$

Although the aberration does affect the image amplitude (as a phase term) there is no visible effect on the image (for this **single** sinusoid). For an **odd** aberration $W(u_0,v) = -W(-u_0,v)$

$$U_{\text{out}}(x,y) = \exp[ikW(u_0,0)]\exp(-2\pi i x u_0)$$

$$+ \exp[-ikW(u_0,0)]\exp(2\pi i x u_0)$$

$$= 2\cos[2\pi x u_0 - kW(u_0,0)]$$

and

$$I_{\text{out}}(x,y) = \{2\cos[2\pi x u_0 - kW(u_0,0)]\}^2$$

In this case, the periodic images are **shifted** by $kW(u_0,0)$ - each frequency is shifted differently.

### 2.4.12 Phase Contrast Imaging

Imaging in coherent light is linear in the complex amplitude. In the special case of a weak phase object:

$$U_{\text{in}}(x,y) = \exp[i\psi(x,y)], \quad \psi(x,y) << 1 \simeq 1 + i\psi(x,y)$$

A linearity exists between the **phase** of the object and the **intensity** of the image, as the following analysis shows.

$$I_{\text{out}}(x,y) = \mid U_{\text{out}}(x,y) \mid^2 = \mid U_{\text{in}}(x,y) \otimes\otimes p(x,y) \mid^2$$

where $\otimes\otimes$ denotes the 2D convolution integral. Therefore, using the correlation theorem (where $\odot\odot$ denotes the 2D **correlation** integral)

$$\widetilde{I}_{\text{out}}(u,v) = [\widetilde{U_{\text{in}}}(u,v)T(u,v)] \odot\odot [\widetilde{U_{\text{in}}}(u,v)T(u,v)]^*$$

or

$$\widetilde{I}_{\text{out}}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[\delta(u',v') - i\widetilde{\psi}^*(u',v')T^*(u',v')]$$

$$\times[\delta(u'+u,v'+v) + i\widetilde{\psi}(u'+u,v'+v)T(u'+u,v'+v)]du'dv'$$

$$\simeq \delta(u,v) + i\widetilde{\psi}(u,v)T(u,v) - i\widetilde{\psi}^*(-u,-v)T^*(-u,-v)$$

Now, $\psi$ is real, therefore

$$\widetilde{\psi}^*(-u,-v) = \psi(u,v)$$

and thus, defining $T_p$ as

$$T_p(u,v) = T(u,v) - T^*(-u,-v)$$

we get

$$\widetilde{I}_{\text{out}}(u,v) \simeq [\delta(u,v) + i\widetilde{\psi}(u,v)]T_p(u,v)$$

i.e.

$$\text{FT of output intensity} = \text{FT of input phase} \times T_p$$

The function $T_p$ is called the Phase Contrast Transfer Function (PCTF).

The PCTF can be written as

$$T_p(u,v) = [T(u,v) - T^*(-u,-v)] = [P(\lambda zu,\lambda zv) - P^*(\lambda zu,\lambda zv)]$$

where the pupil function $P$ is given by

$$P(\xi,\eta) = \begin{cases} \exp[ikW(\xi,\eta)], & (\xi,\eta) \in \text{ aperture;} \\ 0, & \text{otherwise.} \end{cases}$$

and $W$ is the aberration function. Note that if there are no aberrations then $W = 0$ and $T_p = 0$! In other words, some kind of aberration, or 'phase plate' is required to see the object phase.

Suppose that a $\frac{\pi}{2}$ phase plate is placed in the pupil function so that

$$P = \exp(i\pi/2)$$

Then

$$T_p(u,v) = \exp(i\pi/2) - \exp(-i\pi/2)$$

$$= 2i\left(\frac{\exp(i\pi/2) - \exp(-i\pi/2)}{2i}\right) = 2i\sin(\pi/2) = 2i$$

except at $(u,v) = (0,0)$ where $T_p(0,0) = 1$.

$$\begin{aligned}\widetilde{I}_{\text{out}}(u,v) &= [\delta(u,v) + i\widetilde{\psi}(u,v)]T_p(u,v) \\ &= \delta(u,v) - 2\widetilde{\psi}(u,v)\end{aligned}$$

so that

$$I_{\text{out}}(x,y) = 1 - 2\psi(x,y), \quad \psi << 1$$

This result is used in he **phase contrast microscope** - the intensity fluctuation is equal to twice the phase fluctuation, with the appropriate change in polarity. The phase contrast principle is also used in high resolution electron microscopy. In this application it is technically impossible to make a special phase retarder, and the phase contrast is made possible by using **even aberrations** (defocus, spherical) of the electron lens.

$$T_p(u,v) = (\exp[ikW(\lambda zu, \lambda zv)] - \exp[-ikW(\lambda zu, \lambda zv)])$$

for even aberrations $W(\xi,\eta) = W(-\xi,-\eta)$ and therefore

$$T_p = 2i\sin[kW(\lambda zu, \lambda zv)]$$

As before if there are no aberrations and $W = 0$ then $T_p = 0$.

### 2.4.13 Incoherent Image Formation

In incoherent illumination, there is a linear relationship between the input $I_{\text{in}}$ and output $I_{\text{out}}$ (time-averaged) **intensities.** For an isoplanatic optical system,

$$I_{\text{out}}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I_{\text{in}}(x',y') \mid p(x-x', y-y') \mid^2 dx'dy'$$

If $p$ is normalized to unit volume (it is, since $P(0,0) = 1$), $\mid p \mid^2$ is not, so we normalize it by dividing by its infinite integral:

$$\frac{\mid p(x,y) \mid^2}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mid p(x,y) \mid^2 dxdy}$$

The **Incoherent Optical Transfer Function** (IOTF) $T(u,v)$ is the Fourier transform of the (normalized) point spread function; apply the autocorrelation

theorem to the top line and Parseval's theorem to the bottom line,

$$T(u,v) = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(\xi,\eta)P^*(\xi+\lambda zu,\eta+\lambda zv)d\xi d\eta}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |P(\xi,\eta)|^2 d\xi d\eta}$$

where $P(\xi,\eta)$ is the inverse Fourier transform of $p(x,y)$.

$$P(\xi,\eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\exp\left[i\frac{2\pi}{\lambda z}(\xi x+\eta y)\right]dxdy$$

and is the **pupil function.**

The equation above for $T$ basically states that the IOTF is equal to the (normalized) spatial **autocorrelation** of the pupil function. Note that the IOTF relates the input and output **intensity** spectra,

$$\tilde{I}_{\text{out}}(u,v) = \tilde{I}_{\text{in}}(u,v)T(u,v)$$

The spatial frequencies are intensity frequencies and are **not** the same as the amplitude frequencies produced in a coherent optical system. This expression for $T$ can be written in the form

$$T(u,v) = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P\left(\xi-\frac{\lambda zu}{2},\eta-\frac{\lambda zv}{2}\right)P^*\left(\xi+\frac{\lambda zu}{2},\eta+\frac{\lambda zv}{2}\right)d\xi d\eta}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |P(\xi,\eta)|^2 d\xi d\eta}$$

From this result it follows that:

(i) $T(0,0) = 1$ (because of normalization)

(ii) $T(-u,-v) = T^*(u,v)$ (Fourier transform of a real quantity)

(iii) $T(u,v) \leq T(0,0)$ (using the Schwartz inequality)

As with the coherent OTF, the modulus $|\,T\,|$, or Modulation Transfer Function (MTF) describes the transfer or **modulation** of sinusoidal components of the object. The phase of $T$, describes spatial shifts of the sinusoidal components.

## 2.5 Summary

In this chapter, we have considered the basic methods and mathematical models for image formation. This provides the background to the types of images

that require appropriate processing. Knowledge of how the image is generated is required to characterize objects at the parameter extraction stage of a pattern recognition process. This preliminary information can be expressed as adjusting factors which in turn corrects the accuracy of the whole object recognition processes. For example, the restoration of an image is a resolution limited process and can not be implemented without information on the noise characteristics of the image that is to be restored together with accurate estimates of the optical function of the imaging system.

In the following chapter we consider the processing operations that are most suitable for application to the optical image obtained, the decision paths required for implementing these processes and their performance.

# Chapter 3

# Image Processing and Decision Making

## 3.1 Introduction to Image Processing

This chapter introduces some of the basic image processing methods and the procedures required in order to initiate the process of image analysis and pattern recognition research for this work. The discussion considers images that have already been obtained from suitable sensors, e.g. RGB/Black &White video/photo camera, microscope, line scan camera, microwave/laser radar set and etc. In particular, we consider the background theory associated with the image processing operations undertaken for image pre-processing and object recognition, the background to the theory of fractal geometry used for texture segmentation and finally, a short overview of fuzzy logic which is used for designing an expert system that makes decisions based on a set of Euclidean and Fractal parameters. In each case, this material forms the background to the methods used to design and construct the system reported in this thesis [19, 68, 9, 66, 98].

A typical colour image consists of mixed $RGB$ signals. A grey-tone image appears as normal 'black and white' photograph. However, on closer inspection it may be seen that it is composed or a large number of individual picture cells or pixels. In fact, a digital image $P$ is an $[r \times c]$ array of pixels. One can get better feel for the 'digital limitations' of such a digitized image by zooming into a section of the picture that has been enlarged so that the pixels can be examined individually. It is then easy to appreciate that each pixel contains $z$ gray level

digitalization. This level will include a certain amount of noise and so it is seldom worth digitizing more accurately than 8 bits of information per pixel. The number of these levels depends on the signal-to-noise of the image capture devise and the analogue-to-digital converter. Modern digital cameras can store up to 10 bits per pixel. Note, that if the human eye can see an object in a digitized image of practical spatial and gray-scale resolution, then, it is in principle possible to devise a computer algorithm to do the same thing.

### 3.1.1 Image Pre-processing

Image pre-processing is used for adjusting the artifacts after the operation of an image acquisition system. We consider the following sub-tasks.

*Low brightness and contrast* The correction of brightness and contrast is usual a pre-processing procedure, after which, the image looks better and more precise. Nevertheless, it is necessary to note, that such a correction does not provide any additional information of value to procedures such as feature extraction or boundary detection for example). The existence or otherwise of spatial frequencies is indifferent to whether the map is contrast stretched or not. In this thesis, the brightness and contrast of the images used is sufficiently good for the system to exclude this pre-processing procedure although in other application of the system, this may be a necessary requirement.

*Image graininess* Some types of images can have a grainy structure - often due to the nature or features of the image acquisition system. It is a typical problem in those cases where it is necessary to acquire an image with maximal resolution. The main problem with processing coarse-grained maps is related to the in-practicality of detecting the boundaries, i.e. boundaries are detected that are associated with grains instead of the contours of objects. A typical solution consists of smoothing the image using minor diffusion in which the boundaries of the grains become fuzzy and diffused with each other, while the contours of object remain (albeit over a larger spatial extent). A similar effect can be obtained using the median filter. However, use of the median filter includes an inevitable loss of information characterized by shallow details (i.e. low gray level variability). In this thesis, the Wiener filter is used which is computational efficient, robust and optimal with regard to grain diffusion and information preservation. This filter eliminates high-frequency noise and thus does not distort the edge of objects.

Other solutions include preliminary de-zooming for the purpose decreasing grit size up to and including the size of a separate pixel. Such a method involves loss of shallow details however, and thus, the size of the map (and accordingly, the processing time) decreases. The other advantage of such a method is concerns hardware implementation, e.g. application of a nozzle to an optical system. In situations where the methods described here are unacceptable, it is necessary to use a more complex quality detector for boundary estimation which is discussed below.

*Geometrical distortions* In practice, the most important geometrical distortions are directly related to character of an image acquisition. In the majority of cases is possible to use a standard video camera as the image sensor. However, the majority of industrial production specifications for video systems use an interlaced scan technique for image capture. This leads to 'captured lines' in the image of both even and odd types which leads to a time delay between neighbouring lines (equal to half the acquisition time frame). If there is a moving object in the field of view, then its position on even and odd lines will be different - the picture of the object will be 'washed' in a horizontal direction. This is a particularly important problem in the extraction of edges. In this case, it is impossible to bleed the verticals. The elementary solution to this problem is to simply skip the even or odd frames (preferably the even frames as the odd frames consist of later information). Other way is to handle even and odd frames separately providing the processing speed allows for practical implementation. If this is not possible, it is necessary to use a video system with non interlaced scanning. Over the past few years, with the development of digital video and engineering the capability has emerged to use digital video cameras with high resolution. A singular advantage of this is the uniformity of the picture without the distortions discussed above. However, the video RGB of matrices need to be analyzed to avoid inter-colour distortions. These distortions are connected to the geometrical distribution of the RGB cells on the surface of a CCD matrix and can be seen when increases in the size of the digital are introduced. Special filters need to be designed that can be used in the prevention this kind of distortion.

## 3.2 Basic Image Processing Operators

One of the simplest image processing operators is that of clearing an image or setting the contents of a given image space to a constant level. The following Matlab routine provides an example if its implementation.

```
for j = 1:c
   for i = 1:r
      P( i, j) = a;
   end
end
```

where $a \in (0 \ldots 255)$. A more interesting operator is that of inverting the image, as in the process of converting a photograph from a negative to a positive. This operator $a$ say, is of the form:

$$a = 255 - P(i, j)$$

Some other simple operators that are of value involves application of a mask which windows the neighbouring location in an original image. It is necessary to allocate the neighbouring pixels in some convenient way. A commonly used scheme is:

$$\begin{bmatrix} P_4 & P_3 & P_2 \\ P_5 & P_0 & P_1 \\ P_6 & P_7 & P_8 \end{bmatrix}$$

If a new image if formed after processing to yield $H$ say, then it is easy to express a left shift of an image for example in terms of

$$H_0 = P_1$$

Similarly, a shift down to the bottom right can be expressed thus:

$$H_0 = P_4$$

Conversion of a grey-scale image into binary image can be utilized using a threshold. The threshold operator is

$$A_0 = \ if\ P_0 > TR\ then\ 1\ else\ 0$$

where $TR \in 1...255$ which can be implemented automatically. Let us use the first letter A from the alphabet to start the name of binary image spaces, and let the letters (P,Q,H,R,G,B) be used to denote gray-scale images or part of an RGB signal. For adjusting the whole range of an image which may be more satisfactory for the human viewer, we can add a constant intensity that makes the image brighter or darker:

$$H_0 = P_0 + b$$

Another interesting operation is to stretch the contrast of a dull image:

$$H_0 = P_0 g + b$$

where $g > 1$ and where the output is taken to be in the normal range.

In many practical applications use is made of a transfer function. One of these functions provides 2D filtering which is consistent with applying a process of convolution. Convolution is powerful and widely used in image processing and it occurs in many applications. The definition of the convolution of two functions $f(x)$ and $g(x)$ is in terms of the integral:

$$f(x) \otimes g(x) = \int_{-\infty}^{\infty} f(u)g(x-u)du$$

The action of this integral is normally described as the result of applying a point spread function $g(x)$ to all points of a function $f(x)$ and accumulating the contributions at every point. It is significant that if the point spread function is very narrow (i.e. approaches a delta function - which can be viewed hypothetically as being infinity at one point and zero elsewhere while having an integral of unity) then the convolution is identical to the original function $f(x)$. This makes it natural to think of the function $f(x)$ as having been spread out under the influence of $g(x)$. This argument may give the impression that convolution necessarily blurs the original function but this is not always so.
When convolution is applied to digital images there two changes required:

1. Double integrals must be used in two dimensions;
2. the integration must be changed to discrete summation;

Convolution in digital form is:

$$F(x,y) = f(x,y) \otimes m(x,y) = \sum_i \sum_j f(i,j)m(x-i,y-j)$$

where $m$ is a spatial convolution mask. When convolution is performed on a whole images, it is usual to restrict the size of the mask as far as possible in order to save computation. The convolution masks are typically 3x3 and not often larger than 15 pixels square. A convolution can be calculated using the result

$$F(x,y) = \sum_i \sum_j f(x+i, y+j)m(i,j)$$

The mask $M$ with $3 \times 3$ neighborhood coefficients takes the form

$$\begin{bmatrix} m4 & m3 & m2 \\ m5 & m0 & m1 \\ m6 & m7 & m8 \end{bmatrix}$$

The machine operator for this process is

$$H_0 = P_0{*}m0{+}P_1{*}m1{+}P_2{*}m2{+}P_3{*}m3{+}P_4{*}m4{+}P_5{*}m5{+}P_6{*}m6{+}P_7{*}m7{+}P_8{*}m8$$

The application of convolution techniques is related to filtering operations. For example after acquisition, an image needs to be filtered from noise. Typically, noise suppression is accomplished by averaging not over corresponding pixels of different images, but over nearby pixels of the same image. A simple way of achieving this is to use following mask

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Another example is the $5 \times 5$ pixels mask for a Gaussian filter with standard deviation 0.5 which takes the form

$$\begin{bmatrix} 0 & 0 & 0.0002 & 0 & 0 \\ 0 & 0.0113 & 0.0837 & 0.0113 & 0 \\ 0.0002 & 0.0837 & 0.6187 & 0.0837 & 0.0002 \\ 0 & 0.0113 & 0.0837 & 0.0113 & 0 \\ 0 & 0 & 0.0002 & 0 & 0 \end{bmatrix}$$

These are examples of simple liner operators. In fact, they are the most general spatially invariant linear operators that can be applied to a signal such as an image. Note, that using estimation results after filtering one can introduce nonlinear filtering. Further, such processes (linear or nonlinear) are non- recursive.

There are many standard digital filters available for this process. Taking into account that in many images, high frequency noise (white noise) is usually present, we consider an appropriate adaptive filtering strategy. A well known adaptive filter is the Wiener filter which can be applied to an image adaptively, tailoring itself to the local image variance. The mathematical background to this filter has been considered in Section 2.4.2. When the variance is large, the Wiener filter performs little smoothing; when the variance is small, it performs more smoothing. This approach often produces better results than linear filtering. The adaptive filter is more selective than a comparable linear filter, preserving edges and other high frequency parts of an image. The Wiener filter does require more computational time than linear filtering. However, the Wiener filter performs better when the noise is constant-power or 'white' additive noise, such as Gaussian noise which is one of the conditions required to simplify the result of applying a least squares criterion (see Section 2.4.2).

The Wiener filter algorithm used in this work uses a pixel-wise adaptive filtering procedure with neighborhoods of size m-by-n to estimate the local image mean and standard deviation. It estimates the local mean and variance around each pixel given respectively by

$$\mu = \frac{1}{nm} \sum_{r,c \in \eta} I_s(r,c) \text{ - } \textit{mean of the brightness of the image}$$

and

$$\sigma^2 = \frac{1}{nm} \sum_{r,c \in \eta} (I_s^2(r,c) - \mu^2) \text{ - } \textit{dispersion}$$

where the sum is taken over the $n$-by-$m$ local neighborhood of each pixel in the image $I$. The algorithm then creates a pixel-wise Wiener filter using the following estimates

$$I_D(r,c) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (I_s(r,c) - \mu)$$

where $\nu^2$ is the noise variance. If the noise variance is not given, the Wiener filter uses the average of all the local estimated variances. In this thesis, the Wiener filter is used as a first step to processing the image. The result is shown in Figure 3.1 and 3.2. Figure 3.1 shows the original image and Figure 3.2 is the result of applying the Wiener filter described above.

We now consider an image processing technique involving a Shrink and Expand mask. These masks are used in thresholding techniques where the threshold is

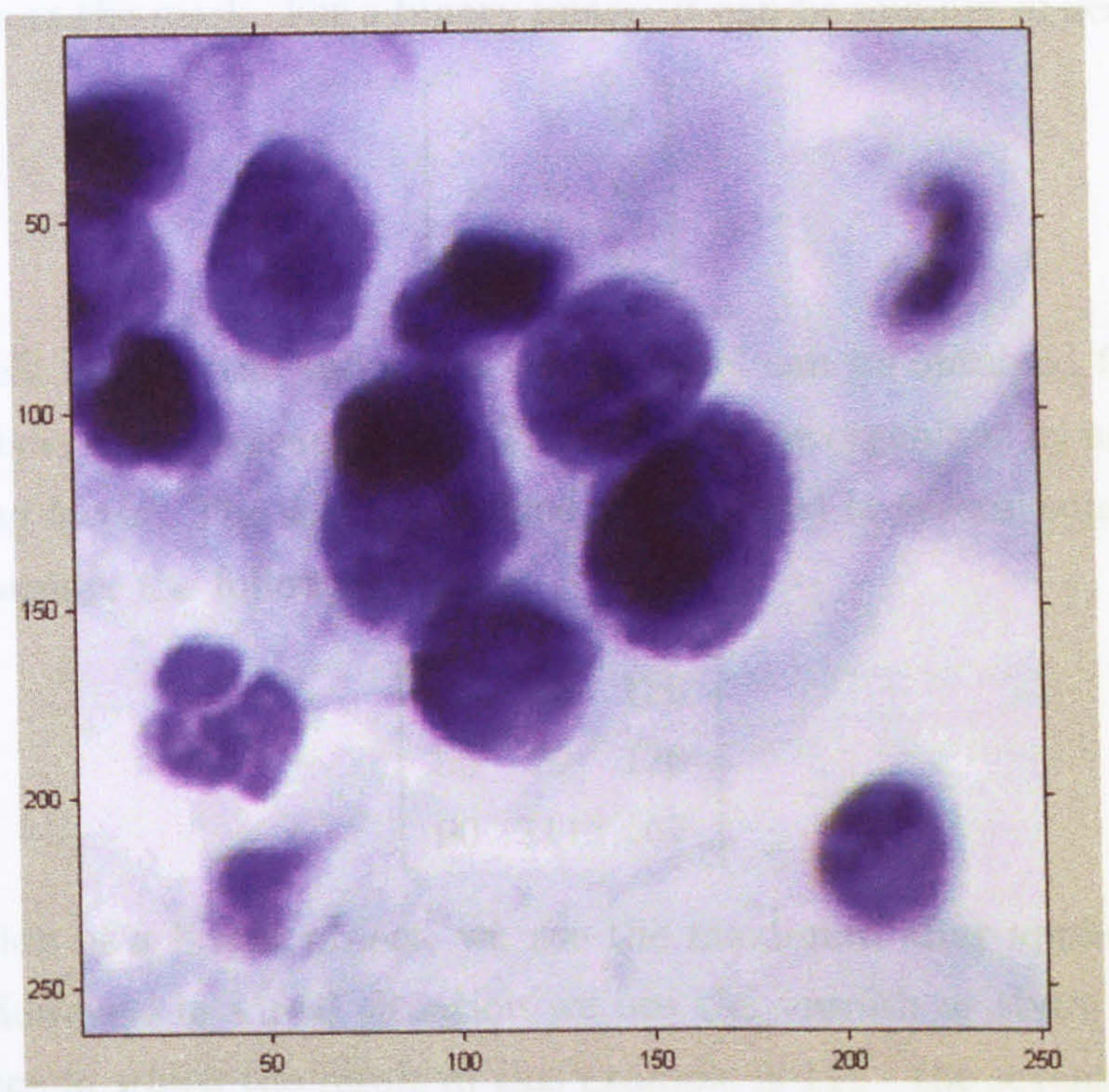



Figure 3.1: The original image

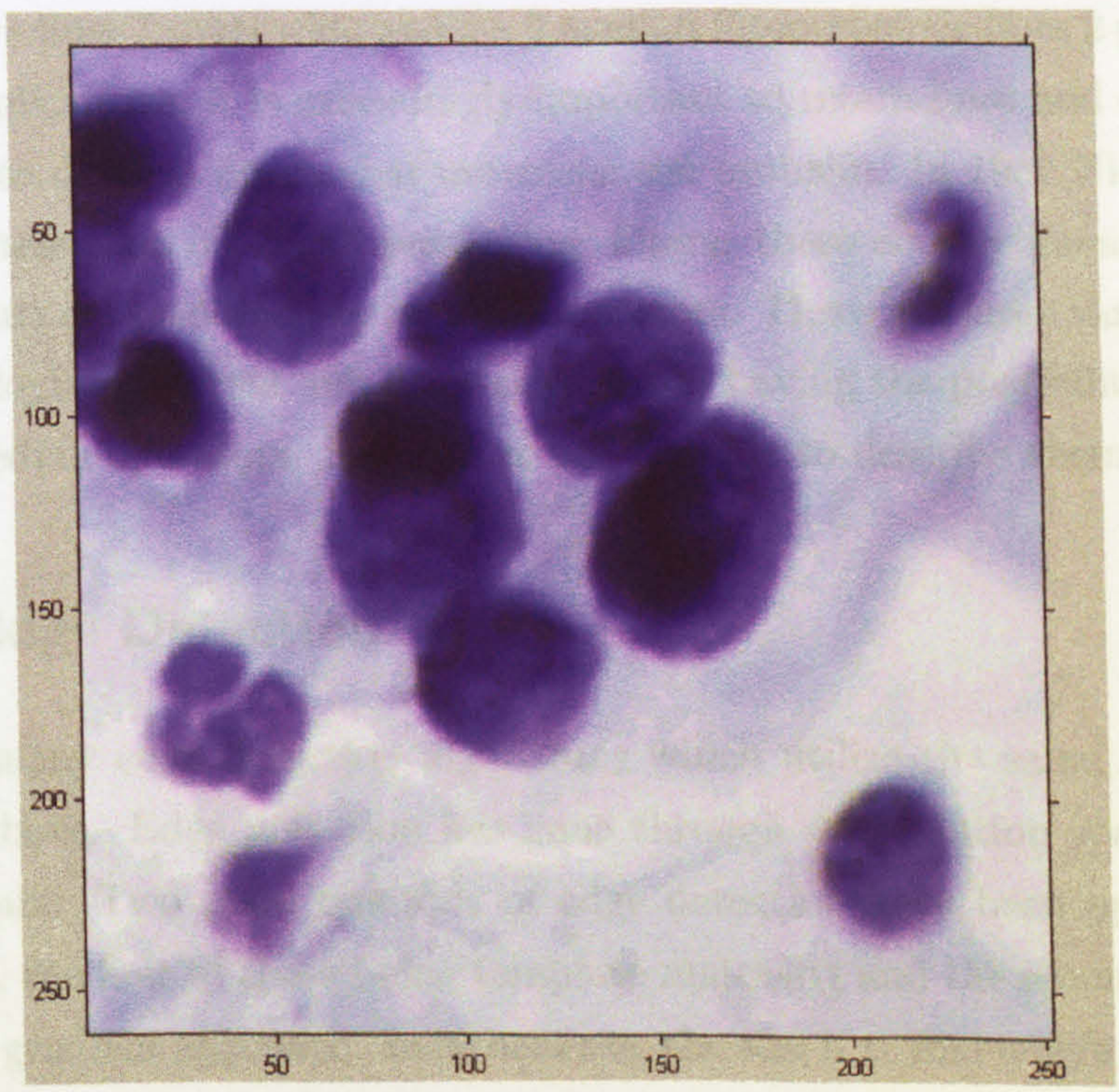



Figure 3.2: The Wiener filtered image

calculated from the mask. For a binary image, it can be number of nearby pixels:

$$\begin{bmatrix} \times & \vee & \vee \\ \times & \times & \vee \\ \times & \times & \vee \end{bmatrix}$$

For grey-scale images, the value of a central pixel can be obtained from sorting all the numbers of the current mask. The sort may be organized by an increasing or decreasing factor. The sort algorithms are explored in several sources [4]. For example, consider the following image

$$\begin{bmatrix} 34 & 45 & 126 \\ 55 & 79 & 130 \\ 90 & 111 & 167 \end{bmatrix}$$

For expansion of a bright object, we use the maximum after applied a sort by increase. However, in a real situation we use the seventh or sixth number of a sort sequence in which the result in this example is 111 - the seventh.

The median filter is of significant value in image processing. This filter preserves edges in digital images but can also remove lines. For example $3 \times 3$ median filters remove lines 1 pixel wide and $5 \times 5$ median filters remove lines 2 pixels wide. In many applications, it is exceedingly important to retain lines and efforts have been made to develop filters that overcome the problem. In 1987 Nieminen [58] reported a new class of 'detail preserving' filters: these employ linear sub-filters whose outputs are combined by median operators. There is a large variety of such filters, employing different sub-filter shapes and having the possibility of several layers of median operators. Hence it is not possible to describe them fully here.

### 3.2.1 Edge Detection

There are many edge detection algorithms which utilize the same approach as discussed above. Edge detection has gone through an evolution spanning more then 20 years. Two main methods of edge detection have been apparent over this period, the first of these being template matching and the second, being the differential gradient approach. In either case, the aim is to find where the intensity gradient magnitude g is sufficiently large to be taken as a reliable indicator of the edge of an object. Then, g can be thresholded in a similar way to that in which

the intensity is thresholded in binary image estimation. Both of these methods differ mainly in that they proceed to estimate g locally. However, there are also important differences in how they determine local edge orientation, which is an important variable in certain object detection schemes.

Each operator estimates the local intensity gradients with the aid of suitable convolution masks. In a template matching case, it is usual to employ up to 12 convolution masks capable of estimating local components of the gradient in the different directions. Common edge operators used are due to Sobel [34], Roberts [43], Kirsch [76], Marr and Hildreth [46], Haralick [82, 83], Nalwa and Binford [56] and Abdou and Pratt [1]. In the approach considered here, the local edge gradient magnitude or the edge magnitude is approximated by taking the maximum of the responses for the component mask:

$$g = \max(g_i : i = 1 ton)$$

where $n$ is usually 8 or 12. The orientation of the boundary is evaluated in terms of the number of a mask giving maximal value of amplitude of a gradient.

The most popularly used $3 \times 3$ edge operators are the Prewitt and Sobel [87]. The Roberts operator is not so sensitive because it uses a $2 \times 2$ pixels mask. Davies [14] has shown that of the $3 \times 3$ edge detectors, the Sobel is the most accurate. The commonality between these two operators is that they compute the edge magnitude and edge direction. A local neighborhood edge operator which is direction-invariant is the Laplacian. These operators are illustrated below:

a) Mask for the Roberts 2x2 operator:

$$For\ X \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \ and\ Y \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

b) Mask for Sobel $3 \times 3$ operator:

$$For\ X \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \ and\ Y \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

c) Mask for Prewitt $3 \times 3$ 'smoothed gradient' operator:

$$For\ X \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \ and\ Y \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

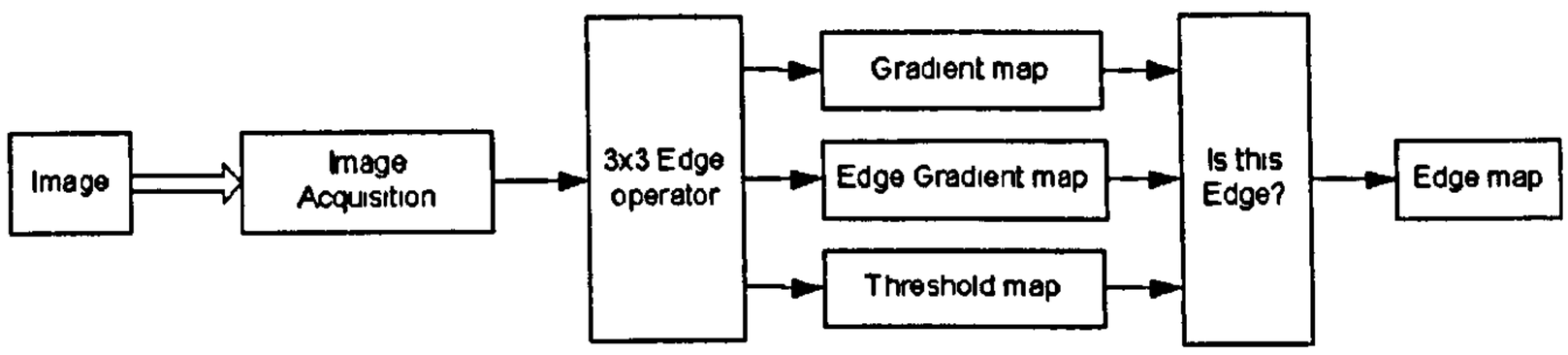



Figure 3.3: Edge Detection Scheme

d) Laplacian operator:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The Laplacian is less popular because it responds strongly to noise and corners [87] and because it gives a response on both sides of an edge. Because of the weighting of the central pixel, while the Laplacian will respond to edge points, it will enhance more strongly isolated points, corners, lines line ends as if they have a higher contrast than the surroundings. Furthermore, the Laplacian returns a signed image and for an 8-bit image, it may return a 10-bit result. Using such $3 \times 3$ gradient operators, a possible edge detection scheme has been suggested in [85]. This method is illustrated below:

The Sobel operation gives the x and y components of edge magnitude and direction. In the case of a differential gradient type operator, only two such masks are required, for $x$ and $y$ directions.

The detector, on the basis of calculating a gradient, as against the previous case, requires only two masks - for calculation of a derivative in the $x$ and $y$ directions. The amplitude of a local gradient can be calculated by vectorial addition, i.e.

$$g = \sqrt{g_x^2 + g_y^2}$$

In order to save computation, it is common practice [60] to approximate this formula by one of the simple forms:

$$g = |g_x| + |g_y|$$

or

$$g = \max(|g_x|, |g_y|)$$

which are, on average, equally accurate [22]. Accordingly, the orientation of a boundary can be retrieved as follows:

$$\theta = atan\left(\frac{g_y}{g_x}\right)$$

The value of a function of an arctangent can be approximately counted with use of the two dimensional table of values generated from the function $T[g_x, g_y] = atan(g_y/g_x)$. The problems associated with the development of boundaries detectors is surveyed in the works of [34, 88, 73]. The standard solution widely used in practice consists of calculating a gradient through the Sobel operator [73] of size $3 \times 3$. It provides enough split-hair accuracy (approximately 3 degrees for the orientation of a boundary) and requires a rather small level of computation. The important property of this operator is its symmetry in relation to a central point. The operators of larger size ($5 \times 5$) require 8 times of more operations and the accuracy increases by up to 1 degree; however, in most cases this accuracy is not required. The operators of the smaller size (Roberts $2 \times 2$ operator) are too sensitive to the noise and grains of a map, and frequently give accident-sensitive results.

Further accuracy in edge detection requires the Marr-Hildreth or Canny algorithm. The advantages of Marr-Hildreth's method are

(1) the Gaussian smoothing reduces the effect of noise,

(2) the second derivative detects localised edge contours, the loci of which are the gradient maxima; also looking for zero-crossing avoids arbitrary thresholding and thinning,

(3) it guarantees closed curves (this is very important for many segmentation tasks), and

(4) it does not give a response in areas where the intensity is changing smoothly.

The disadvantage is that it is computationally expensive because very large convolution masks are required. Wiejak, H. Buxton, and B.F. Buxton discuss methods for improving the computational performance [36, 23]. Such a method requires that 2D convolutions are separated into two 1D convolutions. For example, con-

sider the Sobel kernel:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \equiv \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$$

In this example, the gain is slight, but could be significant for larger kernels such as the Gaussian.

$$\exp\left[-\left(\frac{r^2}{2\sigma^2}\right)\right] = \exp\left[-\left(\frac{x^2}{2\sigma^2}\right)\right] \exp\left[-\left(\frac{y^2}{2\sigma^2}\right)\right]$$

Canny [11] developed an edge operator which extracts not only step edges but also ridge and roof edges. A ridge, is a closely spaced step edge. These are often too small to be dealt with effectively by even the narrowest of edge operators. A roof edge is a concave junction of two planar surfaces of polyhedral objects. Canny derives an edge operator which is approximated by the first derivative of a Gaussian $G(x)$

$$G(x) = \exp\left[-\left(\frac{x^2}{2\sigma^2}\right)\right]$$

The impulse response of the first derivative filter is

$$f(x) = -\frac{x}{X^2} \exp\left[-\left(\frac{x^2}{2\sigma^2}\right)\right]$$

Canny notes that his edge operator is almost identical to the 1D Marr-Hildreth operator. While Marr-Hildreth needs no thresholding, an adaptive thresholding technique can be advantageous for a first derivative operator. However, first derivative operators respond to smooth shading and therefore, both Canny and Nevatia-Babu [57] detect false edges on smoothly shaded surfaces. Also, this filter produces false results at points connected by three or more edges.

The noise characteristics of an operator are determined by its size; bigger operators have the effect of greater noise smoothing. However, a large operator can overlap several features and hence, it reduces the resolution and the detectability and localisation of high-curvature edges. Directional operators introduce averaging which is along the edge; however, isotropic operators, such as the Marr-Hildreth, smooth across the edge. Many of these algorithms have been evaluated by Chen for edge detection [12] on brain images obtained from various sources such as Computer Tomography (CT), Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET). These results indicate that

none of the edge detectors mentioned are universally applicable. Despite these numerous efforts, the problem of edge detection remains largely unsolved. Some work better than others for particular examples. Hence, the user must choose an operator which best suits his or her particular application.

### 3.2.2 Object Location

When there is more then one texture in an image, the procedure to extract regions covered with the same textures or to detect edges between different textures is called object location or image segmentation. Object location is actually the first step of object recognition, because each region usually corresponds to one individual object, or each edge to the boundary between different objects. In this section we present the algorithms for object location.

In the design of an optical recognition device to estimate the level of a disease in the cervical smear test for example, the cells may touch each other, such as happens in the case of a medical test or coloring. A problem then arises of segmenting the bounding cells so as to apply the subsequent classification process to individual cells. Even when the cells do not touch each other, the problem of windowing exists, i.e. of locating the individual cells in the window of a microscope for example and restricting the information sent to the classifier at one time to that concerning the current cell.

The segmentation problem obviously becomes much more difficult in a system intended to search for a mole on the skin for example, where there can be significant noise-type artifacts together with different types of personal skin characteristics. Another type of segmentation problem arises where the image to be analyzed does not divide up in a natural way into discrete objects. This is the case, for example, in scene analysis and in the interpretation of many types of medical images. In a given field of view there may be sufficient resolution to detect individual trees, yet the purpose of the given interpretation may be satisfied by detecting a 'forest'.

As result of edge detection we can find pixels with non-zero values by derivative from brightness. However, attention must be payed to the fact that there may be a large number of pixels and not all of them will be units of contours. It can be the fragments of texture, shallow details and noise (the operation of derivation

always underlines shallow details, speckle and noise with high spatial frequencies) and also the points near to real contours (because natural boundaries are always fuzzy). Such points must be by necessity removed.

**Binarization** The first task is to remove points with small amplitude of a local gradient of brightness with the purpose to separate points of a contour from textures, shallow details and noise. There are two cases to the segmentation algorithm:

(i) Pixels similarities based approach

(i) Surface discontinuities based approach

The first way is to select some value of a threshold binarization $TR$ and to remove points with amplitude of a gradient $|g| < TR$. Some of thresholding processing need to be considered *a priori.*

The main problem in defining the value of a threshold $TR$ say, is that it should be different for different images. Moreover, if the objects on the image have different brightness, the value $TR$ should be different for different areas of the image. The solution is usually employed through a method of adaptive binarization, based on calculating the value of a threshold $TR$ for small areas of the image (size 88.. 1212) - so-called block binarization. For each area, the average value $I_0$ of brightness amplitude is evaluated and then the value of a threshold is calculated as follows:

$$TR = k * I_0$$

where k = 1.2.. 1.8 - binarization coefficient. Change in the value of $k$ are invalid when considerable changes in the quality of an extracted contour occur (as against the level of a threshold $TR$), and the value $k = 1.5$ can be adopted for the overwhelming majority of the images.

Another modification of this method involves the calculation of the level of a threshold separately for different boundary orientations. This prevents the deletion of important details close to brighter objects. However, this method requires more computing cost (as it is necessary to compute the local histograms but allows for an increased value of a threshold $TR$ without loss of essential details and, as a corollary, reduction in the quantity of false points.

In addition to the thresholding methods one can employ multi-region based segmentation. Regions, which are contiguous, are simply connected clusters of pixels which are mutually exclusive and exhaustive (i.e. a pixel can only belong to a single region and all pixels have to belong to some region). A region may support a set of predicates; however, an adjacent region cannot support the same set of predicates. The advantages of using a region-based segmentation are: (1) there are far fewer regions than pixels in an image, thus allowing data compression; and (2) regions are connected and unique. The disadvantages of the method are: (1) assumptions are made about the uniformity of image features; (2) a region could be erroneously considered to be a single surface; (3) surface properties or such as reflection can produce regions of noise.

There are two principal approaches to region-based segmentation which are discussed below

1. **Region growing:** Initially each pixel can be considered to be a separate region. Adjacent regions are merged if they have similar properties (such as gray-level). This merging process continues until no two adjacent regions are similar. The similarity between two regions is often based upon simple statistics such as the variance measure or the range of gray-levels within the regions. Region-based segmentations are described in [9, 98, 19, 10].

2. **Region splitting:** Initially the image is regarded as being a single region. Each region is recursively subdivided into subregions if the region is not homogeneous. The measure for homogeneity is similar to that for region growing. Robertson et al. subdivided a region either horizontally or vertically if the pixel variance in the region is large [74]. Others - e.g. [34] - use the bimodality of histograms to split regions. This is referred to as the *mode method* and has been extended [66, 72] to use multiple thresholds from the histogram of the region. However, as discussed above, a histogram gives global information about a region. Using this method, some pixels can be assigned a wrong label and therefore pre- and post-filtering is required [66].

Growing regions is a more difficult task then region splitting. However, region splitting, using the method described in [74, 34], can lead to the generation of more regions than required. Some region merging at the end of the splitting phase is required. A large group of segmentation techniques is used texture analysis. In

the previous section, we described methods for image segmentation based on the grey-level properties of objects. These methods generally work well for man-made objects which usually have a smooth grey-level surface. We observe a textured region as being *homogeneous*, although the intensity across the region may be non-uniform. This leads to the intensity-based segmentation methods to produce results which do not match with our perception of the scene.

Texture is important not only for distinguishing different objects but also because the texture gradient contains information describing the objects depth and orientation. Texture can be described by its statistical or structural properties [6]. A texture surface having no definite pattern is said to be *stochastic*, while texture with a definite array of sub-patterns is said to be *deterministic*. These textured surfaces can then be described by some placement rule for the pattern primitives. In reality, deterministic texture is corrupted by noise so that it is no longer ideal; this is referred to as the *observable* texture. If the pattern making up the deterministic texture itself has subpatterns, then these are called *micro-textures* and the larger patterns are called *macrotextures*. One of most powerful texture measures originates from the work of Blackledge [47] in the field of fractal geometry. The main idea is that fractal properties can be used like individual features of an image or part of an image.

Segmentation should be invariant to indexing algorithms. The indexing phase has until recently been almost entirely ignored. In practice, emphasis of the recognition procedure has been placed on producing reliable correspondence algorithms [25]. The results of all these methods combined is not enough to automatically segmentation such complex image structure as medical images such as a cervical smear. Using local estimation and a special suite of algorithm developed for this thesis allows us to produce a successful result.

## 3.3 Fractal Analysis in Digital Imaging

This section discusses the use of fractal geometry for segmenting digital images. A method for texture segmentation is introduced which uses the Fractal Dimension to measure image texture. Using this approach, variations in texture across an image can be characterised in terms of variations in the fractal dimension. By imaging the spatial fluctuations in fractal dimension obtained using a conven-

tional moving window approach, a digital image can be texture segmented - this is the principle of Fractal Dimension Segmentation.

An overview of the methods for computing the fractal dimension is presented focusing on an approach which makes use of the characteristic Power Spectral Density Function (PSDF) of a Random Scaling Fractal Signal. A more general model for the PSDF of a stochastic signal is introduced and discussed with reference to texture segmentation of digital signals and images. Here, a new approach is considered to precisely obtain the PSDF representation of images with limited resolution.

### 3.3.1 Introduction - What is Texture?

Texture is a word that is commonly used in a variety of contexts but is at best a qualitative description of a sensation. Visual texture can be associated with a wide range of scenes and images but the term cannot be taken to quantify any particular characteristic. How then can we quantify texture mathematically - is there a specific and unique definition for texture? To begin with, we can state that textural information is not well defined in terms of Euclidean geometry. Most images can be divided into regions containing either:

(i) deterministic information (where Euclidean geometry is usually applicable);

(ii) textural information (which is not easily described by Euclidean geometry).

The latter case requires suitable methods of texture segmentation to be researched. Typically one or more 'measures' for texture can be defined and a moving window (usually square) passed over the image. For each window position, each of the texture measures are computed. If only one measure is defined then one 'measure image' is produced on which conventional image processing techniques can then be applied. If more than one measure is defined then some sort of clustering algorithm is needed to reduce the several measure images into one final image. In addition, scaling can be applied whereby the window size is gradually decreased, giving rise to yet more measure- images at different scales.

What are these measures for texture? It is worth stating at the outset that there is no universally accepted mathematical definition of texture. Mandelbrot [8] states the following:

(i) 'texture is an elusive notion which mathematicians and scientists tend to avoid because they cannot grasp it' and...

(ii) 'much of fractal geometry could pass as an implicit study of texture'.

It is essential in the field of computer vision that an unsupervised, automatic means of extracting textural features, that agrees with human sensory perception, can be found. This is particularly important in the area of remote sensing where analysis of both coherent and incoherent images often involves having to differentiate between texture contrast. More and more data is becoming available from an increasingly diverse range of imaging systems, the rate of data acquisition being vastly greater than the speed at which conventional analysis and interpretation can be performed. In the area of medical imaging for example, an increasing amount of research is being undertaken into the analysis of on-line data compared with the design of new instrumentation for example. Most of this effort is concerned with either visualization of the data or on the design of recognition systems in which features of particular medical significance can be detected automatically.

Although this thesis is basically concerned with the application of fractal geometry to texture segmentation in which the fractal dimension is taken to be an appropriate measure, it is worth discussing other measures which have been considered. In [28], Tamura et al. define six mathematical measures which are considered necessary to classify a given texture. These measures are as follows:

(i) Coarseness; course versus fine.

Coarseness is the most fundamental textural feature and has been investigated thoroughly since early studies [38]. Often, in a narrow sense, courseness is synonymous with texture. When two patterns differ only in scale, the magnified one is coarser. For patterns with different structures, the bigger its element size, and/or the less its elements are repeated, the coarser it is felt to be.

(ii) Contrast; high contrast versus low contrast

The simplest method of varying image contrast is by stretching or shrinking its gray sale [87]. By changing the contrast of an image we alter the image quality, not the image structure. When two patterns differ only in

gray-level distribution, the difference in their contrast can be measured. However, more factors are supposed to influence the contrast difference between two texture patterns with different structures. The following four factors are considered for a definition of contrast: (a) dynamic range of gray levels; (b) polarization of the distribution of dark-field and bright-field regions in the gray- level histogram or the ratio of dark and bright areas; sharpness of edges (images with sharp edges have higher contrast); (c) period of repeating patterns.

(iii) Directionality; directional versus non-directional.

This is a global property over the region. Directionality involves both element shape and placement rule. Bajcsy [77], divides directionality into two groups, monodirectional and bidirectional. If only the total degree of directionality is considered then orientation of the texture pattern does not matter; i.e. patterns which differ only in orientation should have the same degree of directionality.

(iv) Line-likeness; line-like versus blob-like

This concept is concerned with the shape of a texture element. It is expected that this feature supplements the major ones previously mentioned, especially when two patterns cannot be distinguished by directionality.

(v) Regularity; regular versus irregular

This is a property for variations of a placement rule. However, it can be supposed that variation of elements, especially in the case of natural textures, reduces the regularity on the whole. Additionally, a fine structure tends to be perceived as regular.

(vi) Roughness; rough versus smooth.

This description was originally intended for tactile textures, not for visual textures. However, when we observe natural textures, we are usually able to compare them in terms of rough or smooth. It is still a matter for debate as to whether this subjective judgement is due to the total energy of changes in gray-level or due to our imaginary tactile sense.

Tamura et al.[28] gives mathematical definitions of all these measures together with results of experiments to find the correlation between these measures and

human perception. The work reported in this paper is an alternative approach to the problem of image discrimination and classification through texture to the other principal approaches, namely:

(i) the statistical co-occurrence approach

(ii) the method of mathematical morphology

(iii) scale space filtering

It should be noted, however, that these approaches are not mutually exclusive. For example, some recent papers on morphology also include fractal ideas [70] and in [71] the performance of four classes of textural features are compared with the conclusion that *'...the results show that co-occurrence features perform best followed by fractal features' and ... 'However, there is no universally best subset of features. The feature selection task has to be performed for each specific problem to decide which feature of which type one should use'.*

In this thesis, the use of the fractal dimension as a measure for texture is discussed. Different methods of computing this measure for signals and images are introduced and the theoretical basis for this approach to texture segmentation presented. This includes a new texture model which is based on a generalization of the Power Spectral Density Function (PSDF) of a Random Scaling Fractal (RSF) which incorporates stochastic processes such as the Ornstein-Uhlenbeck process and the Bermann process.

### 3.3.2 Fractal Geometry

Central to fractal geometry is the concept of self-similarity in which an object appears to look similar at different scales - an obvious concept when observing naturally occurring features, but one that has only relatively recently started to be developed mathematically and applied to various branches of science and engineering. This concept can be applied to systems of varying physical size depending on the complexity and diversity of the fractal model that is considered. Ultimately, it is of philosophical interest to view the Universe itself as a single fractal, the self-similar parts of which have yet to be fully categorized; those naturally occurring objects for which fractal models abound, being smaller subsets of

a larger whole. This view is closely related to the concept of a chaotic Universe in which the dynamical behaviour of a system cannot necessarily be pre-determined. Such systems exhibit self-similarity when visualised and analysed in an appropriate way (i.e. an appropriate phase space). In this sense, the geometry of a chaotic system may be considered to be fractal.

Self-similarity is a very general term. There are two distinct types of self-similarity:

(i) Deterministic self-similarity in which the fractal is composed of distinct features which resemble each other in some way at different scales (feature scale invarience)

(ii) Statistical self-similarity in which the features of the fractal may change at different scales but whose statistical properties at all scales are the same (statistical scale invarience).

Deterministic fractals associated with (i) above are usually generated through some Iterated Function System and are remarkable for the complexity that can be derived through the simplest of these iterated systems. The way in which the output from these systems is viewed graphically and interpreted geometrically changes substantially from one fractal to another but the overall principal remains the same.

Statistically self-similar fractals are those used to model a variety of naturally occurring objects (clouds, landscapes, coastlines etc.). They can be generated through a variety of different stochastic modelling techniques. The measure most commonly associated with a self-similar object is its fractal (or similarity) dimension. If we consider a bounded set $A$ in a Euclidean $n$-dimensional space, then the set $A$ is said to be self-similar if $A$ is the union of $N$ distinct (non-overlapping) copies of itself, each of which has been scaled down by a ratio $r < 1$ in all coordinates. The fractal is described by the relationship 1.3 ([29] for example) where $D$ is the fractal dimension. The ranges in the value of $D$ characterise the type of fractal 1.1.

In each case, the fractal may be deterministic or random. In the latter case, the fractal is taken to be composed of $N$ distinct subsets each of which is scaled down by a ratio $r < 1$ from the original and is the same in all statistical respects

to the scaled original. The fractal dimension in this case is also given by equation 1.3.

The scaling ratios need not be the same for all the scaled down copies. Certain fractal sets are composed of the union of $N$ distinct subsets each of which is scaled down by a ratio $r_i$, $1 \leq i \leq N$ from the original in all coordinates. The fractal dimension is given by a generalization of equation 1.3, namely

$$\sum_{i=1}^{N} r_i^D = 1$$

Finally, there are self-affine fractal sets which are scaled by different ratios in different coordinates. For example, consider the curve $f(x)$ which satisfies

$$f(\lambda x) = \lambda^{\alpha} f(x) \quad \forall \quad \lambda > 0$$

where $\lambda$ is a scaling factor and $\alpha$ is the scaling exponent. This equation implies that a scaling of the $x$-coordinate by $\lambda$ gives a scaling of the $f$-coordinate by a factor $\lambda^{\alpha}$ which is an example of a self-affinity. A special case occurs when $\alpha = 1$ when we have a scaling of $x$ by $\lambda$ producing a scaling of $f$ by $\lambda$ which is an example of self-similarity. Random fractal signals and images are, in general, examples of self-affine records.

Naturally occurring fractals also differ from the strictly mathematically defined fractals in that they do not display statistical or exact self- similarity over all scales. Rather, they display fractal properties over a limited range of scales. Yokoya et al [65] have developed an algorithm for computing the upper and lower scale limits.

### 3.3.3 Fractals and Images

Given that natural surfaces can be approximated as fractals over a range of scales, it is necessary to examine how the imaging process maps fractal surfaces into grey-level surfaces. Kube and Pentland [69] show that the image of a fractal surface is itself a fractal surface under the following conditions:

(i) the surface is Lambertian,

(ii) the surface is illuminated by (possibly several) distant point sources,

(iii) the surface is not self-shadowing.

This thesis is concerned with the segmentation of images through the computation of fractal dimensions which give a measure of the 'roughness' or image texture. A high value fractal dimension indicates a rough surface whereas a low value indicates a smooth surface. A variety of methods have been developed to calculate the fractal dimension of signals and images. The following section provides an overview of these techniques as applied to fractal signals and/or images.

### 3.3.4 Methods of Computing the Fractal Dimension

As with many other techniques of digital signal and image processing, the computation of the fractal dimension can be undertaken in 'real space' (processing the data directly) or in 'transform space' (processing the data after taking an appropriate integral transform). In the later case, use can be made of the Fourier transform as the PSDF of a fractal signal or image has an expected form. This important relationship between a random fractal and its PSDF is introduced later, where it forms the basis for a more general discussion on stochastic modelling using PSDF models.

In general, there is no unique and general rule for computing the fractal dimension. A large number of algorithms have been developed over the past fifteen years to compute the fractal dimension which can be broadly categorized into two families:

(i) Size-measure relationships, based on recursive length or area measurements of a curve or surface using different measuring scales.

(ii) Application of relationships based on approximating or fitting a curve or surface to a known fractal function or statistical property such as the variance.

By way of a brief introduction to the methods associated with (i) above, consider a simple Euclidean straight $\ell$ of length $L(\ell)$ over which we 'walk' a shorter 'ruler' of length $\delta$. The number of steps taken to cover the line $N[L(\ell), \delta]$ is then $L/\delta$ which is not always an integer for arbitrary $L$ ad $\delta$. Hence, we should compute $N(L, \delta)$ and $N(L, \delta)+1$, but as $\delta \rightarrow 0$ the errors become smaller. Clearly

$$N[L(\ell), \delta] = \frac{L(\ell)}{\delta} = L(\ell)\delta^{-1}$$

$$\Rightarrow 1 = \frac{\ln L(\ell) - \ln N[L(\ell), \delta]}{\ln \delta} = -\frac{\partial \ln N[L(\ell), \delta]}{\partial \ln \delta}$$

which expresses the topological dimension $D_T = 1$ of the line. In this case, $L(\ell)$ is the Lebesgue measure of the line and if we normalise by setting $L(\ell) = 1$, the latter equation can then be written as

$$1 = -\lim_{\delta \to 0} \left[ \frac{\ln N(\delta)}{\ln \delta} \right]$$

since the smaller is $\delta$, the less is the error in counting $N(\delta)$. We also then have $N(\delta) = \delta^{-1}$. For extension to a fractal curve $f$, the essential point is that the various fractal dimensions satisfy an equation of the form

$$N[F(f), \delta] = F(f)\delta^{-D}$$

where $N[F(f), \delta]$ is read as the number of rulers of size $\delta$ needed to cover a fractal set $f$ whose measure is $F(f)$ which can be any valid suitable measure of the curve. Again we may normalize, which amounts to defining a new measure $F'$ as some constant times the old measure and get

$$D = -\lim_{\delta \to 0} \left[ \frac{\ln N(\delta)}{\ln \delta} \right]$$

where $N(\delta)$ is taken to be $N[F'(f), \delta]$ for notational convenience. In practice, if we are dealing with a digital signal (a sampled curve) rather than an abstract analogue mathematical curve which has precise fractal properties over all scales, then

$$D = -\left\langle \frac{\partial \ln N(\delta)}{\partial \ln \delta} \right\rangle \tag{3.1}$$

where we choose values $\delta_1$ and $\delta_2$ satisfying $\delta_1 < \delta < \delta_2$ over which we do some sort of averaging processes denoted by $\langle\ \rangle$. The most common approach is to look at the bilogarithmic graph of $\ln N(\delta)$ against $\ln \delta$, choose values $\delta_1$ and $\delta_2$ over which the graph appears to be straight and then apply a least squares fit to the straight line within these limits.

### The Least Squares Approximation

All algorithms discussed in this section use a least squares approach to computing the fractal dimension. It is therefore worth briefly reviewing this technique.

Let $f_i, i = 1, 2, ..., N$ be a real digital function consisting on $N$ elements and let $\hat{f}_i$ be an approximation to this function. We assume that $\hat{f}_i$ is the expected

form of the data $f_i$. The least squares error e is then defined as

$$e = \sum_i (f_i - \hat{f}_i)^2$$

In most cases, algorithms for computing the fractal dimension use logarithmic or semi-logarithmic plots to fit the results of a given algorithm to a line. In these cases, we are interested in finding the slope $\beta$ and in certain cases the constant $c$ of the line

$$\hat{f}_i = \beta x_i + c$$

To find the best fit, we are required to minimise the error $e$ which is taken to be a function of $\beta$ and $c$. This is achieved by finding the solutions to the equations

$$\frac{\partial e}{\partial \beta} = 0 \text{ and } \frac{\partial e}{\partial c} = 0$$

Differentiating with respect to $\beta$ and $c$ gives

$$\sum_i x_i (f_i - \beta x_i - c) = 0$$

and

$$\sum_i (f_i - \beta x_i - c) = 0$$

Solving for $\beta$ ad $c$ we obtain

$$\beta = \frac{N \sum_i f_i x_i - \left(\sum_i f_i\right)\left(\sum_i x_i\right)}{N \sum x_i^2 - \left(\sum_i x_i\right)^2}$$

and

$$c = \frac{\sum_i f_i - \beta \sum_i x_i}{N}$$

This approach can also be used when the data is two-dimensional (a digital image or grey level surface) where we are required to approximate the data $f_{ij}$ by a function

$$\hat{f}_{ij} = \beta x_{ij} + c$$

The result (i.e. the expression for $\beta$ and $c$) is the same as above except that the summation is over $i$ and $j$. In the following sections, algorithms for computing the data used to calculate the fractal dimension with the least square method are discussed. Some of these algorithms are based on the following relationship

$$Length = c \ \ Step^{\beta}$$

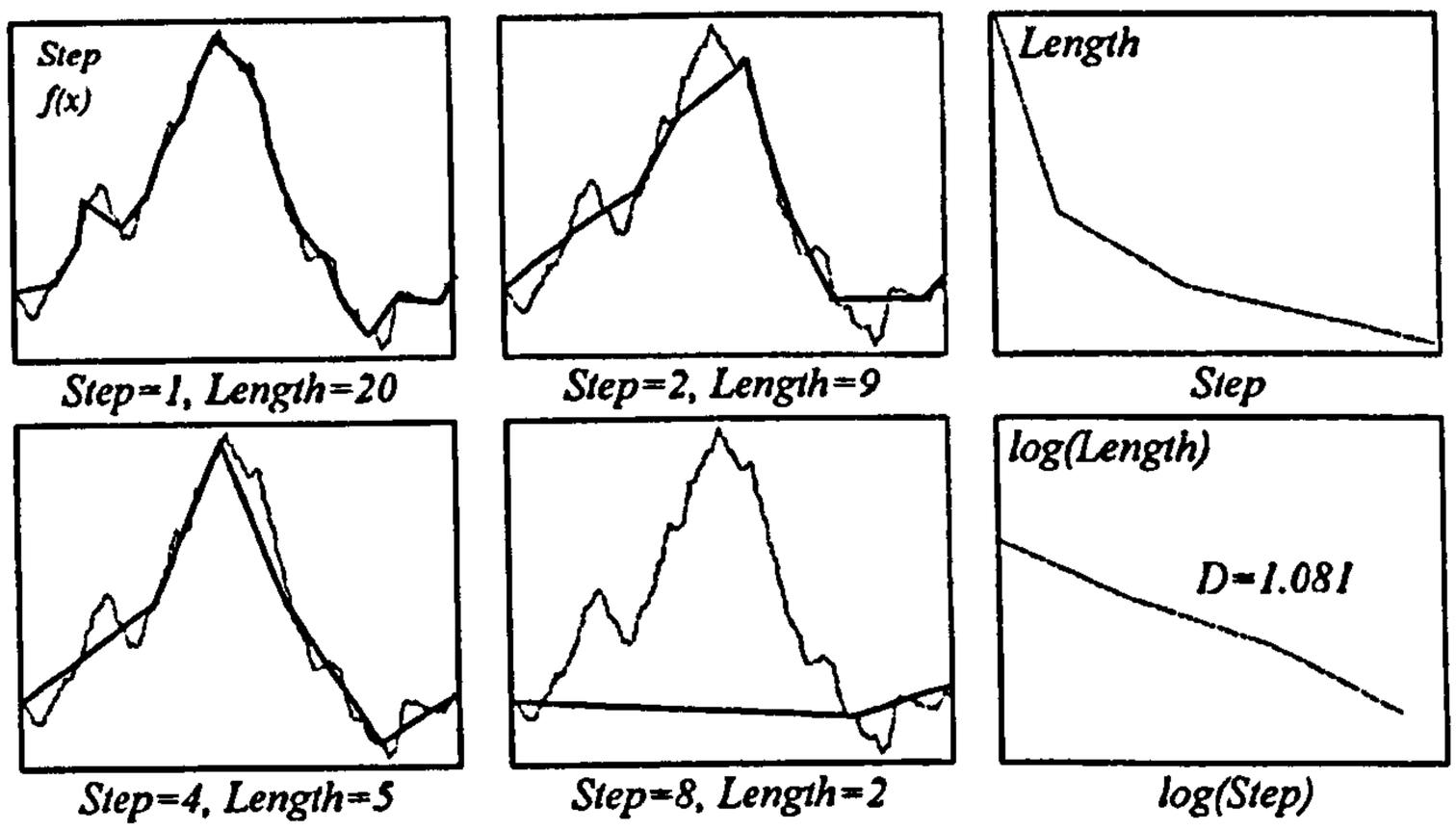



Figure 3.4: Illustration of the walking-divider method for computing the fractal dimension $D$ of a signal showing four iterations and the least squares fit

which can be linearised thus

$$\ln(Length) = \ln(c) + \beta \ln(Step)$$

Here, length represents the measurement of the curve or surface using a 'ruler' of size *Step*. $\beta$ is the slope of the log-log plot which has a simple algebraic relationship with the fractal dimension D, depending on the algorithm used.

### The Walking-Divider method

Introduced by Shelberg [53], this method uses a chord length (*Step*) and measures the number of chord lengths (*Length*) needed to cover a fractal curve. The technique is based on the principle of taking smaller and smaller rulers of size *Step* to cover the curve and counting the number of rulers *Length* required in each case. This approach is based on a direct interpretation of equation 3.1 where $N(\delta) \equiv Length$ and $\delta \equiv Step$ are estimated in a systematic fashion. It is a recursive process in which the *Step* is decreased (typically halved) and the new *Length* calculated. Here, the input signals are taken to be of size $N$ where $N$ is a power of 2 because of the recursive nature of the method. A least squares fit to the bilogarithmic plot of *Length* against *Step* gives $\beta$ where $D = -\beta$. This part of the calculation essentially provides and estimate of the average gradient in equation 3.1 as illustrated in Figure 3.4.

The Walking-Divider method suffers from a number of problems. First of all, the initial *Step* must be carefully chosen. Shelberg [53] and further papers [39],

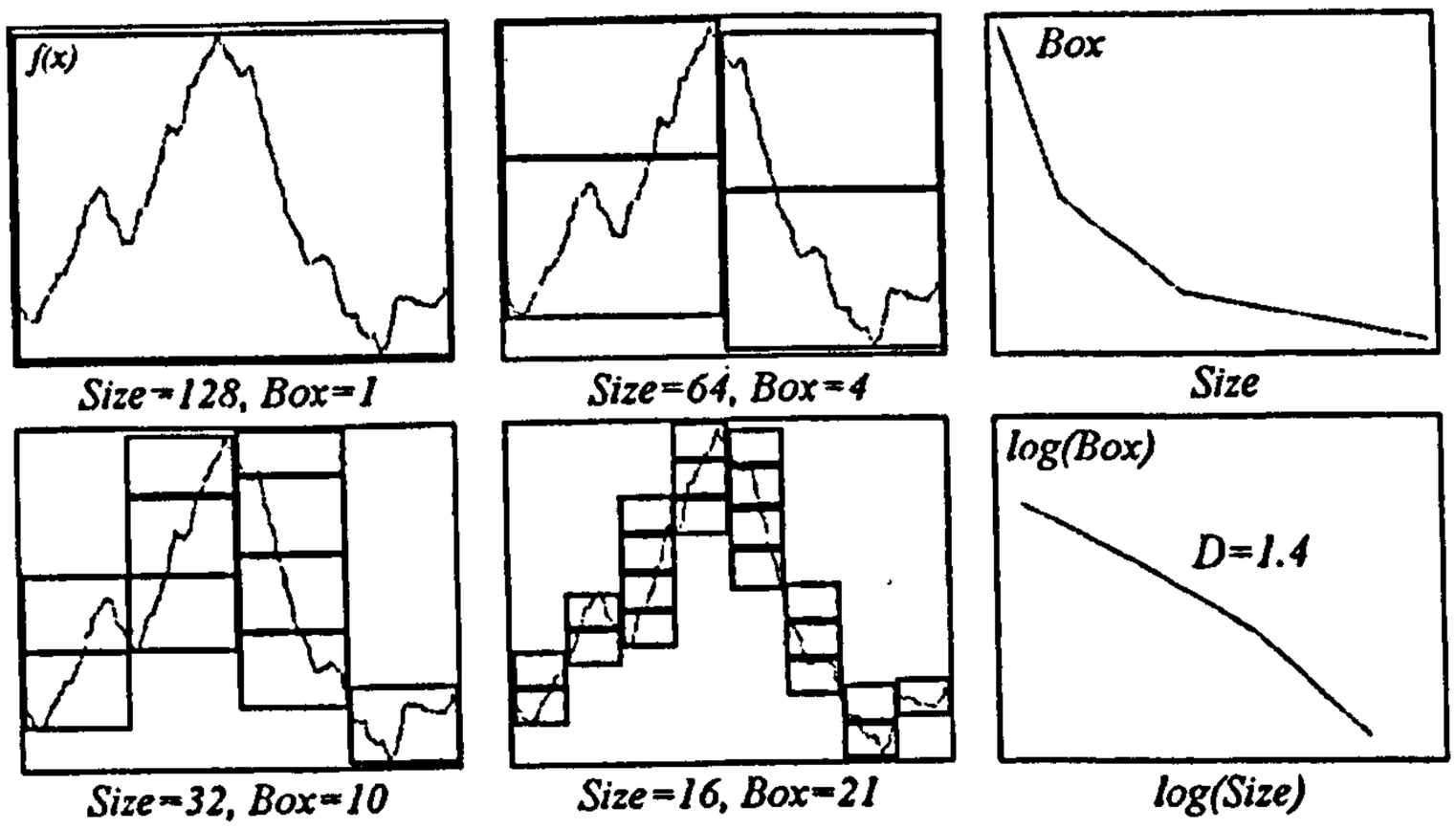



Figure 3.5: Illustration of the box counting method for computing the fractal dimension $D$ of a signal showing 4 iterations and the least squares fit

[59] describe an appropriate starting value as half of the average distance between the points. Clearly, the computation of the initial value, and the procedure required to count the number of *Steps*, makes this algorithm time consuming.

### The Box Counting Method

One of the most popular algorithms for computing the fractal dimension of fractal signals and images is the box counting method originally developed by Voss [89] but modified by others to develop a reasonably fast and accurate algorithm. The numerous algorithms published which are based on this theme, follow the same basic principle.

Box Counting in general involves covering a fractal with a grid of $n$-dimensional boxes or hyper-cubes with side length $\delta$ and counting the number of non-empty boxes $N(\delta)$. For signals, the grid is one of squares and for images, a grid of cubes. Boxes of recursively different sizes are used to cover the fractal. Here again, an input signal with $N$ elements or an image of size $N \times N$ is used as input where $N$ is a power of 2. The slope $\beta$ obtained in a bilogarithmic plot of the number of boxes used against their size then gives the fractal dimension (also known as the Box or Minkowski dimension) where $D = -\beta$. The principle is illustrated in Figure 3.5

Successive divisions by a factor of two are used for the box *Size* to give a regular spacing in the bilogarithmic plot and least squares fit. In practice, a

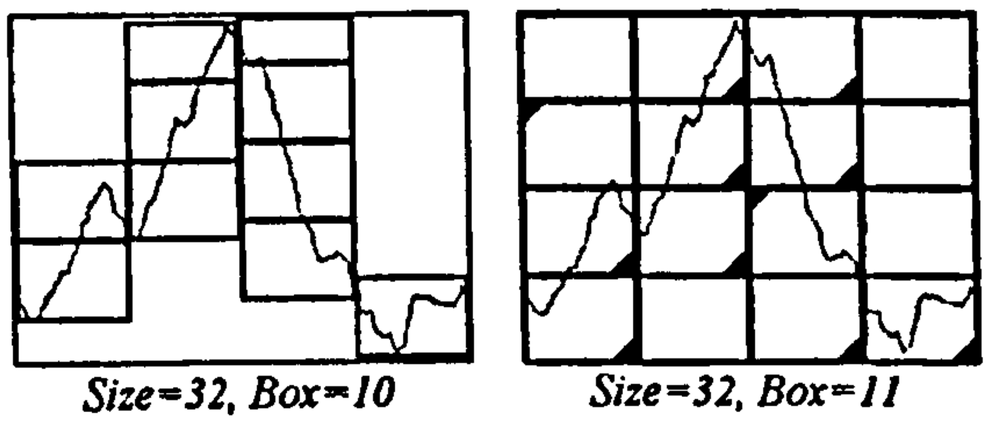



Figure 3.6: Illustration of irregular and regular grids used for the box counting algorithm

regular grid is usually applied (see Figure 3.6) to the data and the non-empty boxes counted.

The behaviour of this algorithm is such that the greater the number of points used for the least squares fit, the better the estimate of the fractal dimension. In a two-dimensional version of this algorithm, Sarkar and Chaudhuri [63], [62] have considered the problem of optimizing the number of boxes for a given size required to compute an accurate fractal dimension. The solution is to map the entire image with boxes (cubes) and then identify the lower box and higher box covering the surface.

In general, Box Counting algorithms behave well and produce accurate estimates for fractal dimensions between 1 and 1.5 for digital signals and between 2 and 2.5 for digital images and are easy to code and fast to compute. Outside this range, (i.e. for higher fractal dimensions), they tend to give less accurate results; underestimating in most cases and saturating near 0.6 above the topological dimension as confirmed by Keller and Chen [33].

### The Prism Method

Clarke [40] defines an algorithm based on the idea of Box Counting, in which instead of counting the number boxes in a region for a given size, the area based on four triangles defined by the corner points is computed and summed over a grey level surface. The triangles define a prism based on the elevated corners and a central point computed in terms of the average of the four corners. A bilogarithmic plot of the sum of the prisms' areas for a given base area gives a fit to a line whose slope is $\beta$ in which $D = 2 - \beta$ The basic engine for this algorithm is similar to the Box Counting method, but is slower due to the number of multiplications implied by the calculation of the areas.

### Hybrid Methods

Hybrid methods calculate the fractal dimension of 2D surfaces using 1D methods. This approach is based on a result by Goodchild [49] who confirms a simple relationship between the fractal dimensions of a surface's contours (1D fractal curves) and the fractal dimension of the surface itself, namely

$$D2 = 1 + D1$$

where $D1$ is the average of the fractal dimensions of each contour line and $D2$ is the fractal dimension of the surface. In principle, this result holds for any algorithm used to compute $D1$.

## 3.3.5 Contour Lines

As a direct consequence of this fundamental result, Shelberg et al. [52] have developed an algorithm based on extracting $N$ lines of the same elevation in a gey-level surface, computing their fractal dimensions $D1_n$ using the Walking-Divider technique and then finding the general 2D fractal dimension $D$ via the formula

$$D = 1 + \frac{1}{N}\sum_{n=1}^{N} D1_n$$

This method has been used with some success by Lam [60] to measure fractal dimensions of Landsat TM images. The results are generally close to those computed using the Box Counting method for fractal dimensions between 2.1 and 2.4. However, to use this approach requires a certain amount of pre-processing [59], [91].

### Robust Fractal Estimator

This term was introduced by Clarke and Schweizer [39] as an alternative algorithm using the 1D-2D method. Instead of considering elevation slices, profiles in the north-south and east-west directions are taken to compute the general fractal dimension. Each vertical intersection is processed via the Walking-Divider method, and a new map of fractal dimensions created, where each point is defined by the average fractal dimension of the two profiles intersecting at that position.

### Vertical slice averaging

This approach considers the fractal dimension of the surface to be the normal average of all the vertical slices in the $x$ and $y$ directions plus 1. To add flexibility, this technique can be implemented with the possibility of computing either or both directions (rows and/or columns) and to consider only a limited number of slices. Another choice can be made with regard to 1D algorithm used to compute the fractal dimensions.

### 3.3.6 Closed Fractal Curves

For non-fractal closed curves in the plane, the perimeter $\ell$ is related to the enclosed area $A$ by

$$\ell = c\sqrt{A}$$

where $c$ is a constant for a given type of shape (e.g. for squares $c = 1$ and for circles $c = 2\sqrt{\pi}$. In [8] Mandelbrot generalizes this equation for the case of closed fractal curves to give

$$\ell = c\left(\sqrt{A}\right)^D, \quad 1 < D < 2$$

De Cola [42] uses an algorithm based on this generalized perimeter-area relationship to classify different types of land-usage areas from Landsat Thematic Mapper SAR images. He also proposes a complete image segmentation scheme using parallel processing.

### 3.3.7 Other Fractal Properties, Dimensions and Higher Order Fractals

Most authors have made the comment that the fractal dimension alone is not sufficient to quantify a given texture. Many quite different fractals can have the same value of fractal dimension since

$$D = -\frac{\ln N}{\ln r} = -\frac{\ln N^a}{\ln r^a}$$

for all values of $a$ provided that $r^a < 1$. For this reason, other fractal measures have been proposed to supplement fractal dimension in an attempt to uniquely define 'texture'.

**The Fractal Signature**

Many methods of computing the fractal dimension depend on the use of Richardson's equation for some measured property $M$ which is a function of scale $\epsilon$,

$$M(\epsilon) = c\epsilon^{D_T - D}$$

where $c$ is a constant and $D_T$ is the topological dimension. A single value of $D$ can then be computed using a bilogarithmic least squares fit. If however, we use the fact that $M(1) = c$ we can compute $D$ for $\epsilon = 2, 3, \ldots$ A plot of $D$ against $\epsilon$ then gives the fractal signature.

Peli et al [95] give examples of the fractal signatures for different sample imagery. Natural imagery, such as a dense tree background gives a slowly changing, essentially constant signature. A synthetic image consisting of a single edge gives a low $D$ at small $\epsilon$ rising to $D = 3$ at $\epsilon = 4$

### The Correlation Dimension and Signature

Each pixel in a grey-level image can be regarded as a point in a three dimensional space $X_k = [i, j, g_{ij}]$ where $i$ and $j$ are the spatial coordinates of a pixel and $g_{ij}$ is the grey level at these coordinates. For each pixel, a cube of size $2\epsilon + 1$ is constructed centred at the pixel. The number of points $X_\ell$ that fall inside this cube is counted for various values of $\epsilon$. The probability $C(\epsilon)$, that at least one point lies within the cube can then be obtained by dividing the number of points by the cube volume,

$$C(\epsilon) = \frac{1}{N(2\epsilon+1)^3} \sum_{k=1}^{N} \sum_{\substack{\ell=1 \\ \ell \neq k}} H(\epsilon - | X_k - X_\ell |)$$

where $N$ is the number of pixels in the image (if we are considering the whole image or the number of pixels in a moving window - for the purpose of segmentation) and

$$H(\xi) = \begin{cases} 1, & \text{if } \xi \geq 0, \\ 0, & \text{if } \xi < 0. \end{cases}$$

$C(\epsilon)$ obeys the Richardson law with a dimension $D_C$, the correlation dimension, given by

$$C(\epsilon) = c(2\epsilon + 1)^{3 - D_C}$$

Here, a single value of $D_C$ can be computed by the normal bilogarithmic least squares fit or we may compute $C(0) = c$ and then compute a value of $C(\epsilon)$ for

$\epsilon = 1, 2, ....$ In the latter case, we obtain the correlation signature as a plot of $D_C$ against $\epsilon$. For an image, the values of $D_C$ range from two to three. A highly correlated surface gives a correlation dimension close to two whereas a highly uncorrelated surface gives a value close to three. Peli et al. [95] give examples of correlation signatures for a selection of real and synthetic images. In [54], Vepsäläinen and Ma use the correlation dimension to search for cracks in the ice in NOAA images made by the Finish Meteorological Institute. They also give results of their work applied to 3D microscopy using a confocal light microscope.

**Lacunarity**

The term Lacunarity is another due to Mandelbrot, coming from the Latin Lacuna meaning gap. The most straightforward illustration comes from considering the class of Cantor dusts for which the fractal dimension (in this case, the similarity dimension) is given by $D = -\ln N / \ln r$ where $N$ is the number of copies of the real line interval $[0, 1]$ and $r < 1$ is the scaling factor. Clearly, there are an infinite set of different $[N, r]$ which give the same $D$. For example the classic triadic Cantor set $[2, 1/3]$ gives the same $D$ as $[4, 1/9]$, $[8, 1/27]$, etc. The appearance of these point-sets will however be quite different. The difference lies in the way the gaps are distributed.

In [8], Mandelbrot first defines suitable mathematical measures for the lacunarity of deterministic fractal sets such as the Cantor set discussed above. Such definitions are however unsuitable for random fractals and he continues to give suitable candidate definitions of lacunarity for the class of random fractals. These definitions are based on the idea of mass distribution. Consider, for example, a curve $f(x)$: (i) if the curve has negative values, then translate so that it is non-negative; (ii) consider the values of $f$ as representing mass so that we regard $f$ as a mass distribution over the support of $f$. Mandelbrot proposes definitions such as

$$\Lambda = \left\langle \left( \frac{f}{\langle f \rangle} - 1 \right)^2 \right\rangle = \frac{\langle f^2 \rangle}{\langle f \rangle^2} - 1$$

and

$$\Lambda = \left\langle \left| \frac{f}{\langle f \rangle} - 1 \right| \right\rangle$$

for the lacunarity $\Lambda$ where $\langle\ \rangle$ denotes the mean. Note that the first definition can be expressed as the variance divided by the square of the mean. The definitions are not dissimilar to generalised statistical moments and this provides the key to

the multifractals or the higher order fractal approach.

### Higher Order Fractal and Dimensions

In principle, there are an unlimited number of fractals or generalised dimensions giving rise to the growing use of multifractals or higher order fractals in the mathematical and physical sciences. It is stated in the work of many authors on fractals, that the fractal dimension alone is not sufficient to characterise a fractal set. The fractal dimension discussed in this work is only one of a number of generalised dimensions. However, these generalised dimensions are defined via a measure theoretic analysis in 'real space'. The use of transforms and in particular, the Fourier transform, applies principally to the similarity dimension. This approach differentiates between regions in an image with different fractal dimension, but not regions with the same fractal dimension but differing correlation dimension, information dimension and so on.

A good example of the inefficiency of the fractal dimension follows from the result that the fractal dimension $D$ of the sum of two fractal curves $C_1$ and $C_2$, with fractal dimensions $D_1$ and $D_2$ is given by ([41] for example)

$$D = \max[D_1, D_2]$$

Suppose, for example, that $D_1$ is significantly larger than $D_2$; $C_2$ could be a square wave for example, in which cases $D_2 = 1$. Whilst it is clear to the eye that the merged curve is quite different from $C_1$ (or $C_2$), the same value of fractal dimension is found for the merged curve and $C_1$

The need for multi-fractal measures can easily be grasped by considering the definition and calculation of the box dimension. Consider the case of a point-set with a large but finite number $N$ of points embedded in fractal dimensional space where we cover the set with a uniform grid of hypercubes of size $\delta$ and count the number $M(\delta)$ of non-empty boxes. Such a strategy does not include information concerning the distribution of the number of points in the non-empty boxes. Suppose there are $N_k$ points in the $k^{\text{th}}$ hypercube and let $p_k = N_k/N$

The similarity dimension (which takes account only of the total of no-empty boxes $M(\delta)$ and not the $N_k$) is then given by

$$D = -\lim_{\delta \to 0} \lim_{N \to \infty} \frac{\ln M(\delta)}{\ln \delta}$$

The information (or Renyi) dimension $\sigma$ is defined by

$$\sigma = -\lim_{\delta \to 0} \lim_{N \to \infty} \frac{S(\delta)}{\ln \delta}$$

where

$$S(\delta) = -\sum_{k=1}^{M(\delta)} p_k \ln p_k$$

The correlation dimension $\nu$ is

$$\nu = \lim_{\delta \to 0} \lim_{N \to \infty} \frac{\ln C(\delta)}{\ln \delta}$$

where

$$C(\delta) = \frac{1}{N^2} \sum_{i \neq j} H(\delta - \mid X_i - X_j \mid)$$

and $H$ is the Heaviside step function. $C(\delta)$ counts the number of points whose distance $\mid X_i - X_j \mid$ is less than $\delta$. In [27], Hentschel and Procaccia show that $D, \sigma$ and $\nu$ are the first three numbers of a hierarchy of generalised dimensions $D_q$ for $q \geq 0$, i.e.

$$D = \lim_{q \to 0} D_q, \quad \sigma = \lim_{q \to 1} D_q, \quad \nu = \lim_{q \to 2} D_q$$

and that for $q = 3, 4, ..., n$ we have correlation dimensions associated with triplets, quadruplets and $n$-tuplets of points. The same authors show that $D_q$ form a non-increasing series.

$$D_q > D_{q'} \text{ for any } q' > q$$

with the inequality replaced by an equality if and only if the fractal is homogeneous. Hence, we see that the various dimensions, previously introduced quite independently form a general series. Moreover, a general expression for $D_q$ can be obtained given by [41]

$$D_q = \frac{1}{(q-1)} \lim_{\delta \to 0} \frac{\ln \left( \sum_i p_i^q \right)}{\ln \delta}$$

A valuable summary of the uses of both the fractal and chaos models (where often correlation dimensions are computed) in imaging science is given in [79].

## 3.4 Overview of Fuzzy Logic

### 3.4.1 Introduction

Pattern recognition methods occur in many different fields of knowledge which are by their very nature too composite and variational for use of only one precise solution, i.e. a well defined models and corresponding algorithm. Many concepts and classifications are fuzzy in nature and require special mathematical descriptions. For example:

(i) human thinking;

(ii) the approximate character of conclusions;

(iii) linguistic descriptions.

The origins of Fuzzy logic theory can be attributed to Dr Zadeh [100]. In applications involving recognition tasks, we utilize methods of interpretation and analysis that are less formalized with linguistic prior information being used for fuzzy logic theory representations. The tasks contained in a statement an are frequently characterized by the presence of essential (not stochastic) uncertainties such as:

(i) uncertainty of the purposes (multi criteria) 'a maximum of the income at a minimum of costs'

(ii) natural uncertainty

(iii) uncertainty of operations of "opponents"

The application areas of fuzzy logic algorithms are varied within the context of building a expert system, including:

a) non-linear control of processes (production);

b) self-learning systems (or classifiers), research of risk and critical situations;

c) pattern recognition;

d) financial analysis (markets of the financial credit instruments);

e) research of the data (corporate storehouses);

f) increasing the strategies of control and coordination of operations, e.g. complex industrial production.

The power and intuitive simplicity of fuzzy logic as a methodology to solving problems guarantees its successful usage in built-in monitoring systems and information analysis. Thus, there is a connection between human intuition and operator experience. Compared to conventional mathematics, requiring at each step the simulation of a precise and one valued formulations, fuzzy logic offers a completely different level of thinking from which the creative process of simulation takes place at the best level of abstraction and where the minimum set of dependence is postulated only. Numerical data are obtained from 'non-precise measurements' or less formalised areas which in many respects, are similar to distributions in probability theory, but are free from disadvantages that are intrinsic to the latter, e.g. small numbers being fit to distribution functions, suitability of the analysis, necessity of forced normalization, keeping the requirements of additivity, difficulty in the substantiation and adequacy of a mathematical form for describing the behaviour of real data. In the limit, as we increase the accuracy, the fuzzy logic tends to standard Boolean arithmetic. In comparison with random methods, this indistinct method allows as to sharply reduce the volume of computations, which, in turn, results in an increase in the efficiency of implementing a decision making process.

The disadvantages of fuzzy systems are:

a) absence of a standard technique for designing indistinct systems;

b) exist hardware deals with $\{0, 1\}$ and fuzzy chips, still use there rather than [0,1];

c) application of the fuzzy logic approach in comparison to random methods does not result in an increase in the accuracy of calculations.

In this thesis, an attempt has been made to formulate a theory of indistinct sets. This parallels existing theories relating to ill-conditioned tasks and optimal complexity.

### 3.4.2 Fuzzy Arithmetic

In classical predicate logic, a simple proposition $P$ is a linguistic or declarative statement contained within a universe of elements, say $X$, which can be identified as being a collection of elements in $X$ that are strictly true or strictly false. Hence, a proposition $P$ is a collection of elements, that is, a set, where the truth values for all elements in the set are either all true or all false. The veracity (truth) of an element in the proposition $P$ can be assigned a binary truth value, just as an element in a universe is assigned a binary quantity to measure its membership in a particular set. For binary ('Boolean') predicate logic, we assign a value of 1 ('truth') or 0 ('false'). If $U$ is the universe of all propositions, then $T$ is a mapping of the elements $U$ in these propositions (sets) to the binary quantities $(0, 1)$.

A fuzzy logic proposition, $P$, is a statement involving some concept without clearly defined boundaries. Linguistic statements that tend to express subjective ideas and that can be interpreted slightly differently by various individuals, typically involve fuzzy propositions. The most natural language is fuzzy, in that it involves vague and imprecise terms. Statements describing a person's height or weight or assessments of people's preferences about colours or menus can be used as examples of fuzzy propositions. The truth value assigned to $P$ can be any value on the interval $[0, 1]$. The assignment of the truth value to a proposition is actually a mapping from the interval $[0, 1]$ to the universe $U$ of truth values, $T$, as described by

$$T : u \in U \rightarrow \{0, 1\}$$

As in classical binary logic, we assign a logical proposition to a set in the universe of discourse. Fuzzy propositions are assigned to fuzzy sets. Suppose proposition $P$ is assigned to fuzzy set $A$; then the truth value of a proposition, denoted T(P), is given by

$$T(P) = \mu_A(x) \text{ where } 0 \leq \mu_A \leq 1 \tag{3.2}$$

Equation (3.2) indicates that the degree of truth for the proposition, is equal to the membership grade of $x$ in the fuzzy set $A$.

#### Logical connectives

The logical connectives of negation, disjunction, conjunction, and implication are defined for a fuzzy logic. These connectives are given in equation (3.3–3.6) for

two simple propositions: proposition $P$ defined on fuzzy set $A$ and proposition $Q$ defined on fuzzy set $B$.

Negation

$$T\left(\overline{P}\right) = 1 - T\left(P\right) \tag{3.3}$$

Disjunction

$$P \vee Q : x \text{ is A or B } T\left(P \vee Q\right) = \max\left(T\left(P\right), T\left(Q\right)\right) \tag{3.4}$$

Conjunction

$$P \wedge Q : x \text{ is A or B} T\left(P \wedge Q\right) = \min\left(T\left(P\right), T\left(Q\right)\right) \tag{3.5}$$

Implication [45] $P \rightarrow Q : x$ is $A$, then $x$ is $B$

$$T\left(P \rightarrow Q\right) = T\left(\overline{P} \vee Q\right) = \max\left(T\left(\overline{P}\right), T\left(Q\right)\right) \tag{3.6}$$

The logical connective *implication*, i.e. $P \rightarrow Q$ ($P$ implies $Q$) presented here is also known as the classical implication, to distinguish it from an alternative form devised is the 1930s by Lukasiewicz, a Polish mathematician, who was the first to be credited with exploring logics other than Aristotelian (classical or binary logic) [61].

In binary logic, the implication connective can be modeled in rule-based form: $P \rightarrow Q$ is, IF $x$ is $A$, THEN $y$ is $B$ and is equivalent to the following fuzzy relation,

$$R = \left(A \times B\right) \cup \left(\overline{A} \times Y\right) \equiv \text{IF A, THEN B}$$

$$\text{IF } x \in A \text{ where } x \in X \text{ and } A \subset X$$

$$\text{THEN } y \in B \text{ where } y \in Y \text{ and } B \subset Y$$

The membership function of $R$ is expressed by the following formula:

$$\mu_R\left(x, y\right) = \max[\left(\mu_A\left(x\right) \wedge \mu_B\left(y\right)\right), \left(1 - \mu_A\left(x\right)\right)] \tag{3.7}$$

or

$$\mu_R\left(x, y\right) = \max\left\{\min\left[\mu_A\left(x\right), \mu_B\left(y\right)\right], 1 - \mu_A\left(x\right)\right\}$$

When the logical conditional implication is of the compound from

$$\text{IF } x \text{ is } A, \text{ THEN } y \text{ is } B, \text{ ELSE } y \text{ is } C$$

then the equivalent fuzzy relation, $R$, is expressed as, $R = (A \times B) \cup (\overline{A} \times C)$, whose membership function is expressed by the following formula:

$$\mu_R(x,y) = \max[(\mu_A(x) \wedge \mu_B(y)), ((1 - \mu_A(x)) \wedge \mu_C(y))] \tag{3.8}$$

There is a number other techniques for obtaining the Fuzzy relation $R$ based on the IF $A$, THEN $B$, or $R = A \rightarrow B$. They are known as fuzzy implication operations, and are valid for all values of $x \in X$ and $y \in Y$. The following forms of the implication operator shows different techniques for obtaining the membership function values or fuzzy relation $R$ defined on the Cartesian product space $X \times Y$:

$$\mu_R(x,y) = \max\{\mu_B(y), 1 - \mu_A(x)\} \tag{3.9}$$

$$\mu_R(x,y) = \min[\mu_A(x), \mu_B(y)] \tag{3.10}$$

$$\mu_R(x,y) = \min\{1, [1 - \mu_A(x) + \mu_B(y)]\} \tag{3.11}$$

$$\mu_R(x,y) = \min\{1, [\mu_A(x) + \mu_B(y)]\} \tag{3.12}$$

$$\mu_R(x,y) = \min\left\{1, \left[\frac{\mu_B(y)}{\mu_A(x)}\right]\right\}, \quad \mu_A(x) > 0 \tag{3.13}$$

$$\mu_R(x,y) = \max\{\mu_A(x) \cdot \mu_B(y), [1 - \mu_A(x)]\} \tag{3.14}$$

$$\mu_R(x,y) = \mu_A(x) \cdot \mu_B(y) \tag{3.15}$$

$$\mu_R(x,y) = \begin{cases} 1, & \text{for} \mu_A(x) \leq \mu_B(y) \\ \mu_B(y), & \text{otherwise} \end{cases} \tag{3.16}$$

$$\mu_R(x,y) = \begin{cases} 1, & \text{for } \mu_A(x) \leq \mu_B(y) \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

In situations where the universes are represented by discrete elements, the fuzzy relation $R$ is a matrix.

The classical implication (3.7) was developed earlier by Zadeh [45]. Equation (3.9) is equivalent to Equation (3.7) for $\mu_B(y) \leq \mu_A(x)$. Equation (3.10) has been given various terms in the literature: it has been referred to as *correlation-minimum and as Mamdani's implication*, after the British mathematician Mamdani into his work in the area of systems control [16]. This formulation for the implication is also equivalent to the fuzzy cross product of fuzzy sets $A$ and $B$, i.e., $R = A \times B$. For $\mu_A(x) \geq 0.5$ and $\mu_B(y) \geq 0.5$ classical implication reduces to Mamdani's implication. The implication of Eq. (3.11) is known as *Lukasiewicz' implication*, after the Polish logician Jan Lukasiewicz [61]. The fuzzy implication relation defined by Eq. (3.12) is commonly referred to as the *bounded sum implication.* Equation (3.13) is due to Goguen [31]. Equations (3.14) and (3.15) describe two forms of the *correlation-product implication* and are based on the notions of conditioning and reinforcement. Both of these product forms tend to dilute the influence of joint membership values that are small and, as such, are related to Hebbian-type learning algorithms in neuropsychology when used in artificial neural network computations. Equation (3.14) is a recently suggested form by Vadiee [64] and is equally valid for crisp and fuzzy cases. Equation (3.16) is sometimes called the *Brouwerian* implication and is discussed in Sanchez [18]. Equation (3.17) gives a very simple form of Eq.(3.16) that has been termed in the literature as the $R - SEQ$ (*standard sequence logic*) implication [81]. Although the classical implication continues to be the most popular and is valid for fuzzy and crisp applications, these other methods have been introduced for their computationally effectiveness under certain conditions of the membership values, $\mu_A(x)$ and $\mu_B(y)$.

### Composition Operations

Let $R$ be a relation that relates, or maps, elements from universe $X$ to universe $Y$, and let $S$ be a relation that relates, or maps, elements from universe $Y$ to universe $Z$. The question is to find the answer to whether we can find a relation $T$ that relates the same elements in universe $X$ that $R$ contains to the same elements in universe $Z$ that $S$ contains. It turn out we can find such a relation using an operation known as a *composition.*

There are two common forms of the composition operation; one is called the max-min composition and the other the max-product composition. Many other techniques are mentioned in the literature. Each method of composition of fuzzy

Table 3.1: Composition operation

| Name of method | Operation |
|---|---|
| max–min | $\mu_B(y) = \max_{x \in X} \{\min[\mu_A(x), \mu_R(x,y)]\}$ |
| max–product | $\mu_B(y) = \max_{x \in X} [\mu_A(x) \cdot \mu_B(x,y)]$ |
| min–max | $\mu_B(y) = \min_{x \in X} \{\max[\mu_A(x), \mu_B(x,y)]\}$ |
| max–max | $\mu_B(y) = \max_{x \in X} \{\max[\mu_A(x), \mu_R(x,y)]\}$ |
| min–min | $\mu_B(y) = \min_{x \in X} \{\min[\mu_A(x), \mu_R(x,y)]\}$ |
| max–average | $\mu_B(y) = \frac{1}{2} \max_{x \in X} [\mu_A(x) + \mu_R(x,y)]$ |
| sum–product | $\mu_B(y) = f\left\{\sum_{x \in X} [\mu_A(x) \cdot \mu_R(x,y)]\right\}$ |

relations reflects a special inference machine and has its own significance and applications. The max-min method is the one used by Zadeh in his original paper on approximate reasoning using natural language IF-THEN rules. Many have claimed, since Zadeh's introduction, that this method of composition effectively expresses the approximate and interpolative reasoning used by humans when they employ linguistic proposition for deductive reasoning [64].

The following common methods are among those proposed in the literature for the composition operation $B = A \circ R$, where $A$ is the input, or antecedent defined on the universe $X$, $B$ is the output, or consequent defined on universe $Y$, and $R$ is a fuzzy relation characterizing the relationship between specific inputs ($x$) and specific outputs($y$). The list of existing compositions is presented in Table (3.1).

In the table 3.1 $f(\cdot)$ is a logistic function that limits the value of the function within the interval $[0, 1]$. This composition method is commonly used in applications of artificial neural networks for mapping between parallel layers in a multi-layer network. The appropriate choice of operators are consider in Chapter 5 'Object recognition' for which typical examples are given.

## 3.5 Summary

In this chapter, we have considered solutions to the problem of processing images to obtain data that has been optimized in terms of its suitability for submission

to a pattern recognition algorithm.

The given materials explain the connection between an optical image and methods of its representation and processing on modern computing hardware. Understanding the processes of digital imaging is a basis for the representation of objects in the image. The Fractal Geometry in the given work shows that it is the most suitable tool for characterizing structures. Eventually, when there is a digital object representation, the system should make a decision to which class the object belongs. The Fuzzy Logic theory introduces concepts on the mechanism of decision-making. All these stages of image processing are united in the image recognition system which is discussed in Chapter 5.

For practical realization involving the design of real time recognition algorithms, it is necessary to understand the complexity associated with an application area. Thus, not all existing algorithms are applicable for the realisation of compact and fast calculations in real time. In the following Chapter 4, we investigate new solutions to different stages of the pattern recognition problem associated with the different classes of images (i.e. medical images) considered in this work.

# Chapter 4

# New Image Processing Algorithms

In this chapter, a new approach is presented to the recognition problem concerning textured images. The common approach to the task of recognition has been outlined in Chapter 1 and the necessary background knowledge has been given in Chapter 3. The novel algorithms have been especially developed for fast object recognition and classification which are presented in this chapter.

## 4.1 Filter design for the detection of cell boundaries

### 4.1.1 Specific edge detection problem

The edges of the images we are concerned with are associated with the boundaries of features such as the nuclei of cells in a region of interest. Edge detection is used to identify the edges in an image which are those areas that correspond to object boundaries. To find these edges, an algorithm is designed that looks for places in the image where the intensity changes rapidly; this is typically based on using one of two principal criteria:

(i) areas where the first derivative of the intensity is larger in magnitude than some threshold;

(ii) regions where the second derivative of the intensity has a zero crossing.

Edge detection methods are based on a number of derivative estimators, each of which implements one of the definitions above. For some of these estimators, it is possible to specify whether the operation should be sensitive to horizontal or vertical edges, or both. In each case, the aim is to return a binary image - an array containing elements which are either 0 or 1 where 1 represents an element of an edge 0 represents an empty from an edge space. A study of the advantages and disadvantages of applying different edge detection algorithms to images is described in Chapter 3. Moreover, within the context of the overall approach, it is assumed that different edge detectors will yield minimal effect on the final result. Thus, the Canny (1986)[11] method is chosen which comparably provides a valuable first step. The Canny edge detector is based on a functional analysis to derive an optimal function for edge detection, starting with three optimisation criteria, namely, good detection, good localization, and only one response per edge under white noise conditions. The 1D 'Canny function' is accurately approximated by the derivative of a Gaussian function which is then combined with a Gaussian of identical standard deviation in the perpendicular direction, truncated at 0.001 of its peak value, and split into suitable masks. Underlying this method, is the idea of locating edges at local maximum of gradient magnitude of a Gaussian-smoothed image. In addition, the Canny implementation employs a hysteresis operation on edge magnitude in order to make edges reasonably connected. Finally, a multiple-scale method is employed to analyse the output of the edge detector.

The result of applying a Canny filter to Figure 3.2 is given in Figure 4.1. This result typically illustrates that it is not possible to uniquely tell where the edge of a cell or nuclei occurs, especially when there is a connection between one edge with another gradient, where Canny edge detection introduces errors. For this purpose, it is necessary to design a new filter as discussed in the following section.

### 4.1.2 Space-oriented filter

In some cases, the nuclei can occur very close to, or be in touch with a foreign object. In this case, an extra filter must be used to obtain a contour boundary. For this purpose, the space-oriented filter for the detection of 'holes' has developed. The nuclei represent a 'hole' if the image is visualised in terms of

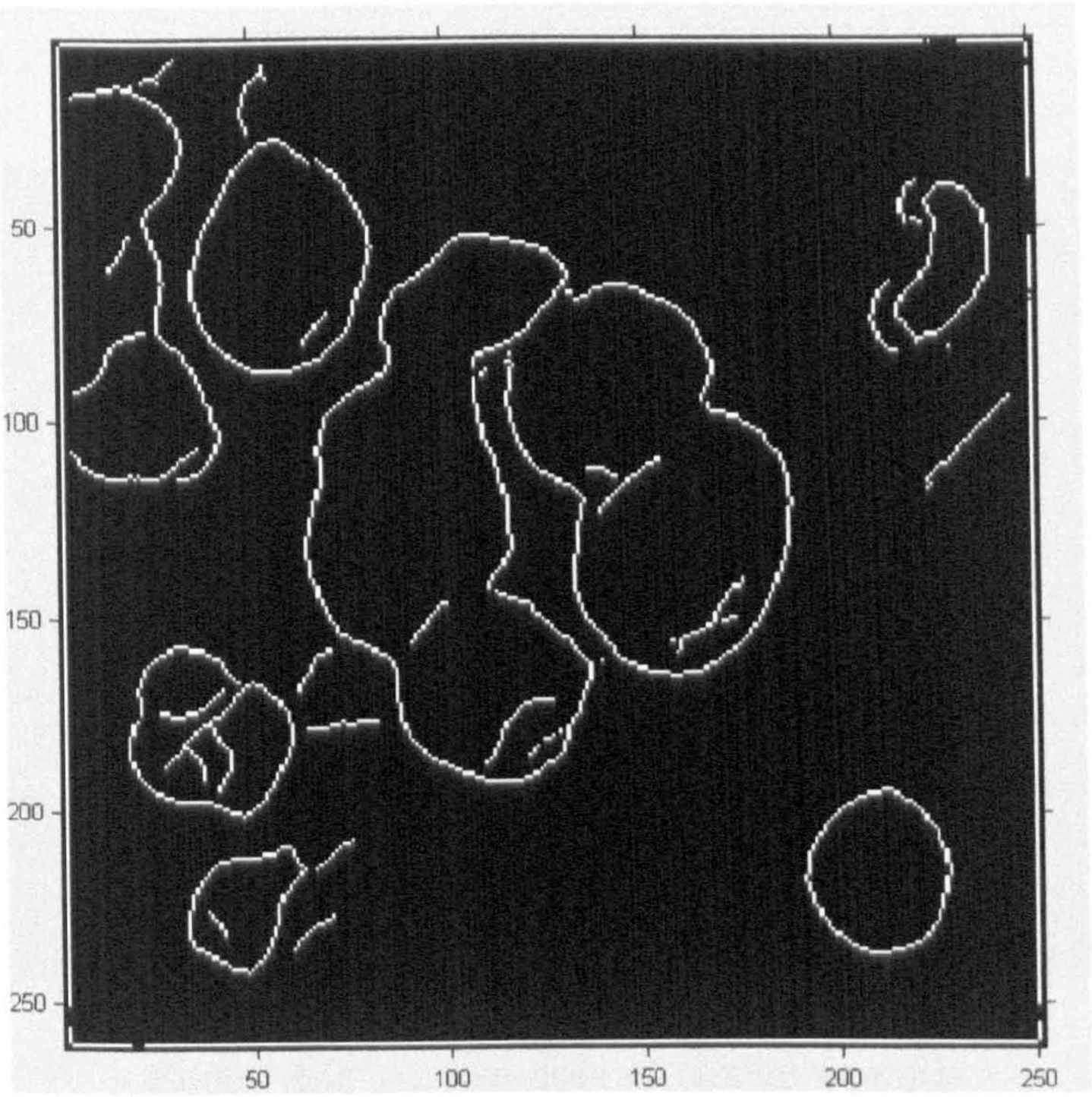



Figure 4.1: Application of a Canny Filter to Figure 3.2

a surface in which the nuclei are regions of lower intensity. The filter has been designed to take account of the following: (i) objects should be of quasi spherical form; (ii) the search space should include objects with lower intensity (i.e. which have darker colour); (iii) it is necessary to find only the surface of a cell without a hysteresis zone. An example of a profile that is characteristic of a nucleus is given in Figure 4.2. The same principle can of course be used for other objects.

The solution to this problem is compounded in the algorithm that is now described. The procedure is illustrated in Figure 4.3 and described below.

To start with, we estimate the brightness of the central area (using a window of 9x9 pixels) and a circle (a layer consisting of 2 pixels). If the center is dark, then suppose that it is part of the nuclei. Then compare the intensity along the white line in Figure 4.3 with the central zone. If the profile along this line has a maximum and minimum gradient, we consider the angle between them. If the angle lies in the range $79^0$ to $248^0$ then we assume that we are near to the border of a nucleus. This angle can be estimated automatically or established as a constant. The next step is to apply the hole detection method (red and brown line in Figure 4.3). This hole detection algorithm is extended in a procedure to decide

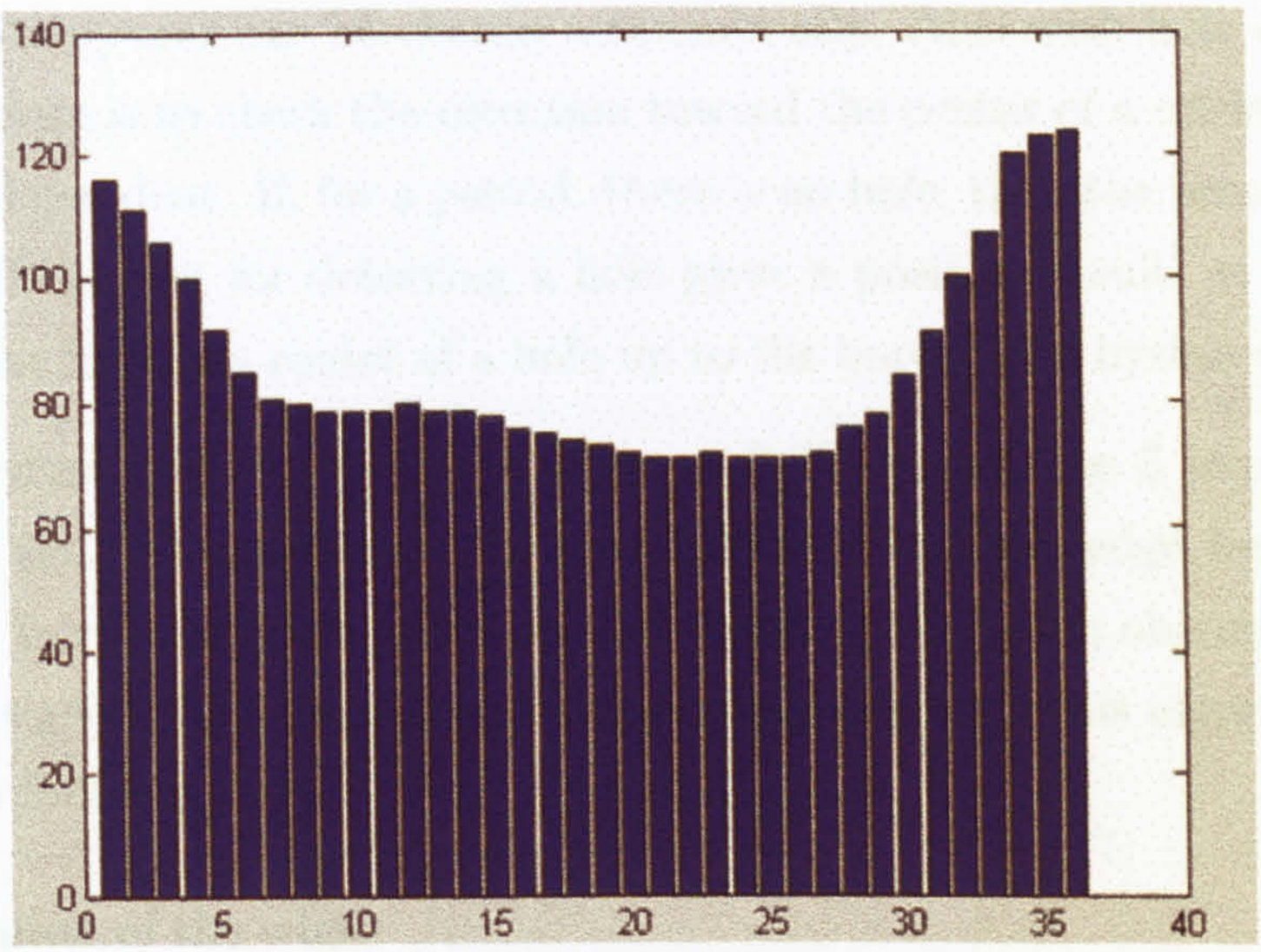



Figure 4.2: An example of Nuclei's profile

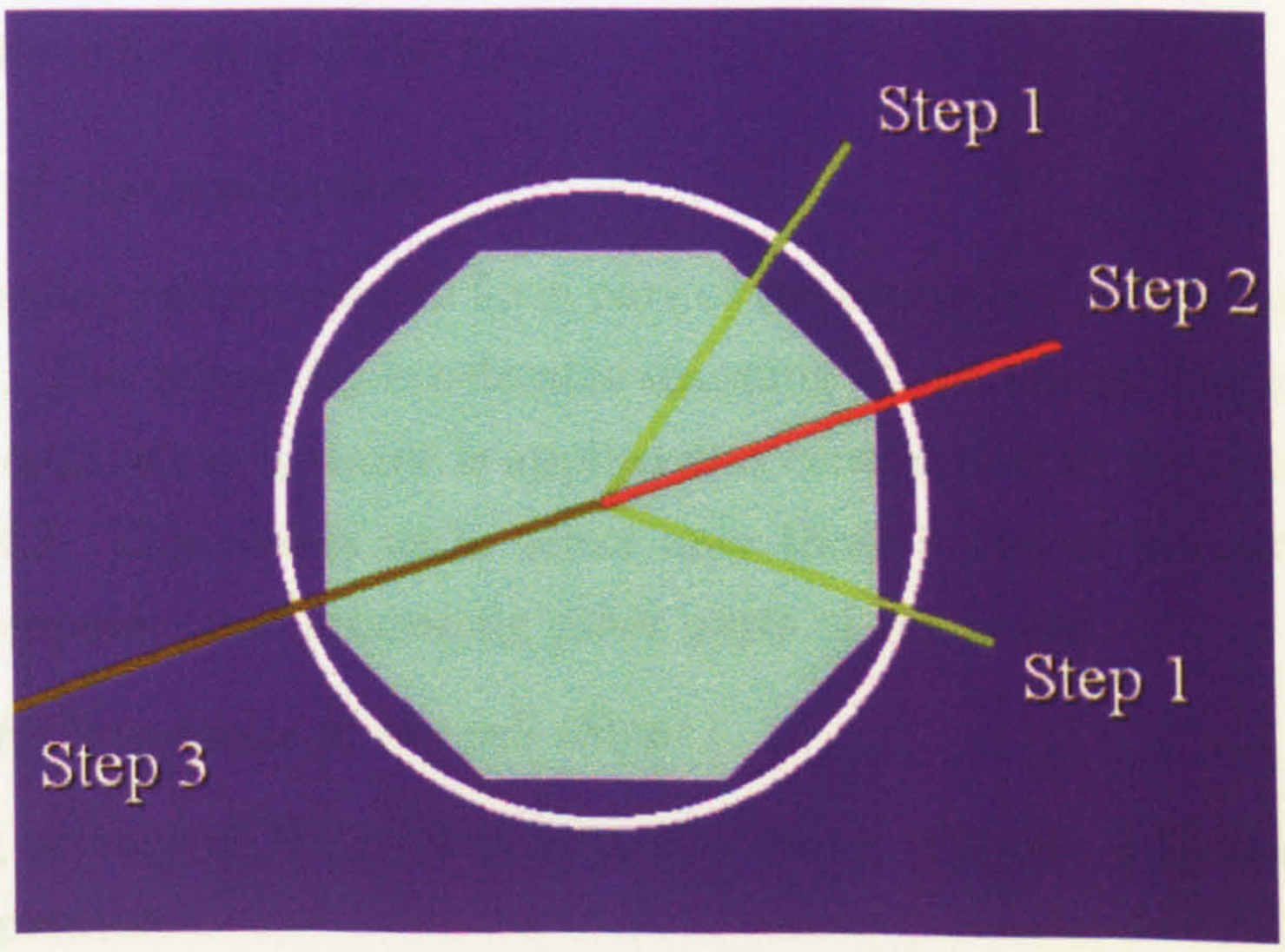



Figure 4.3: Mask of Oriented Filter

whether the area under investigation is a nuclei or otherwise. In Figure 4.3, the maximum length of the brown line is approximately 70 pixels (it depends on the image resolution) and can be chosen automatically. Also with hole detection, a useful procedure is to check the direction toward the center of a nuclei but this is application dependent. If, for a period, there is no hole, then the present position is ignored. If the test for detecting a hole gives a positive result, as in an index figure, the line from the center of a hole up to the border of a hysteresis is drawn.

In the central part of the image (Figure 3.2) one can see 5 joint kernels in the form of the "C" character. To automatically find the edge between all of these nuclei is very difficult and a special algorithm for such objects separation has been designed. The sequence of steps associated with this algorithm can be divided into following list:

1) estimation of the edge;

2) search the boundaries of the cell;

3) calculate the direction to the center of a core;

4) search the opposite edge of a core;

5) calculate the centers of the kernels;

6) save the index map of the figure.

*Estimation of edge expectation*

Pre-processing can be used to form part of the estimated performance for edge expectation. This allows for accelerates scanning of the image. For this purpose a structure estimation operator is applied at a central part of the mask as shown in Figure 4.3. This allows us to select only those nuclei of interest and avoid spending computer time processing other parts of the image.

*Searching the boundaries of the cell (Step1)*

The ring around of the central part of a mask (Figure 4.3) is decomposed using the operator:

$$R = [x_1, x_2, ...x_n]$$

. In the following analysis we evaluated the gradient sequence:

$$g_{1..n} = \frac{dR}{dn}$$

. Upon demarcation of a core and after the derivation, the gradient window will contain two maxima - positive and negative. The polar angle then gives the direction of the nuclear center $\theta_1$.

*Calculation of the direction of the center (Step2)*

At this point, the direction from the expected direction to the center is updated by means of a check on the position of the angle on a plane between two the maxima values obtained from previous step. In general, for purpose of recognition, a point on the binary map uses a convolution technique with series of masks for searching the exact point on the object edge. The sequence of masks is as follows :

$$M = \left\{ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \right\}$$

The appropriate mask is applied in the direction of a local gradient rate and give a maximal convolution between both the points obtained from the previous step. From the definition of the angle $\theta_2$, utilizing prior results of the derivation we form the ratio:

$$\theta = \frac{\theta_1 + \theta_2}{2}$$

The logical conformity of the mask and adjacent points of the binary map is further evaluated and the binary representation of object is determined via:

$$I_B(r,c) = \begin{cases} 1, & \text{if } M \notin I_g; \\ 0, & \text{if } M \in I_g. \end{cases}$$

The profile information (gradient and amplitude) is memorized for Step 3. The dimension $I_B(r,c)$ completely corresponds to the dimension and starting map $I_g(r,c)$.

*Search for the opposite edge of a core (Step3)*

The opposite gradient is searched for by finding of center of a nuclei. Also, the gradient on the opposite end serves as final confirmation for the coordinates

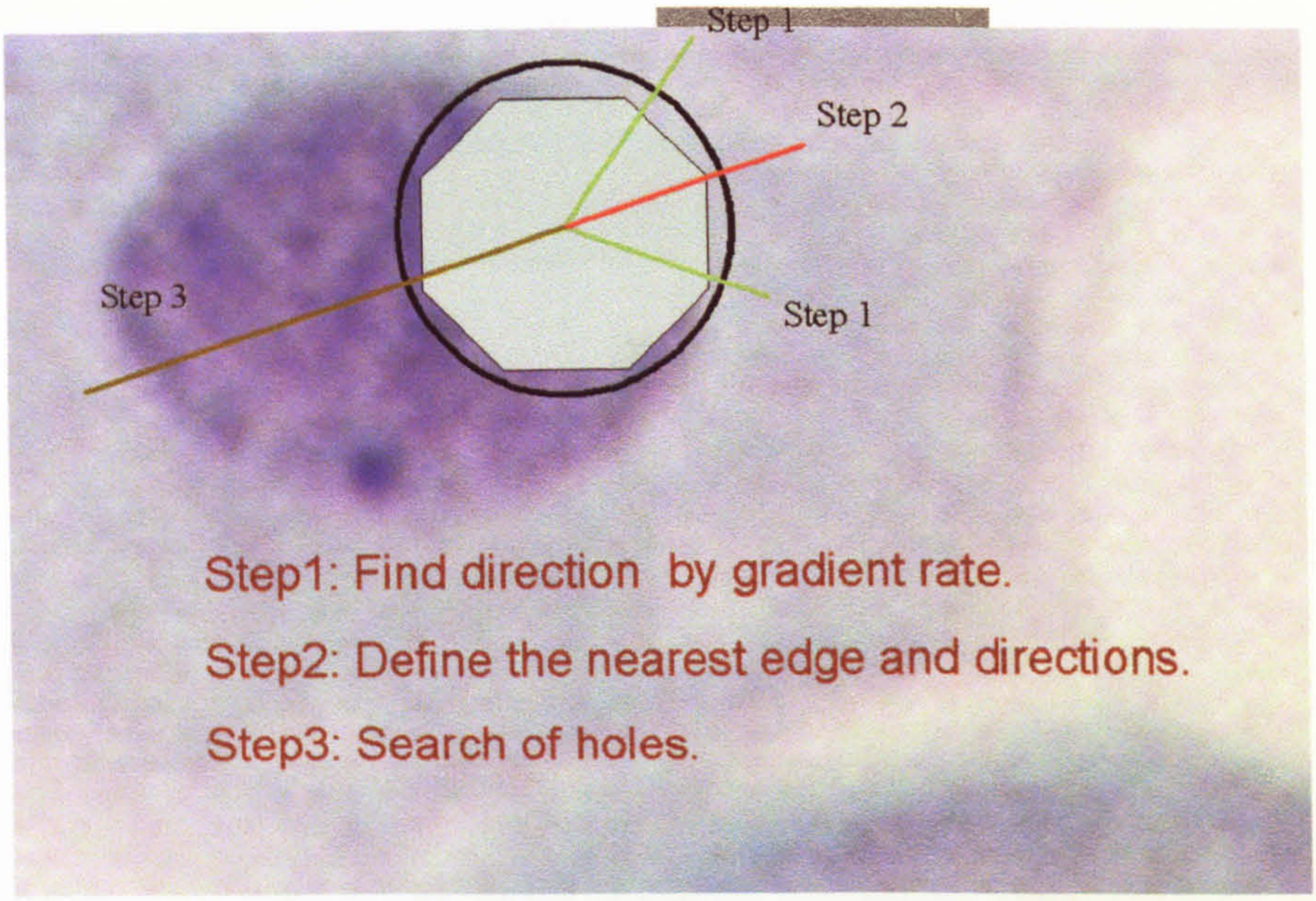



Figure 4.4: Mask of Oriented Filter with an image

of object. In a figure 4.4 these lines are selected by brown colour . The opposite profile has to have the same properties as at Step 2. This prevents any wrong detection through irregularities in the image. If the opposite profile is found, then a green line is 'painted' on the index binary image from the center to the boundary of the nucleus Figure 4.5.

*Calculation of the central of kernel*

The center calculation algorithm is based on the weighted mean from the total number of bars detected in the previous steps. The calculation depends on the kind of implementation used to design the engine. If the calculations are implemented in a programmed logic, the data are better stored in an index space. For a PC, the data are stored as array of coordinates. The result is shown in Figure 4.6.

*Saving the index map (Figure 4.7)*

After application of this algorithm, a connected area can be detected which serves as an index for further processing. An example of an index image is given in Figure 4.7 which includes the application of erosion and dilation for the subdivision of close located objects.

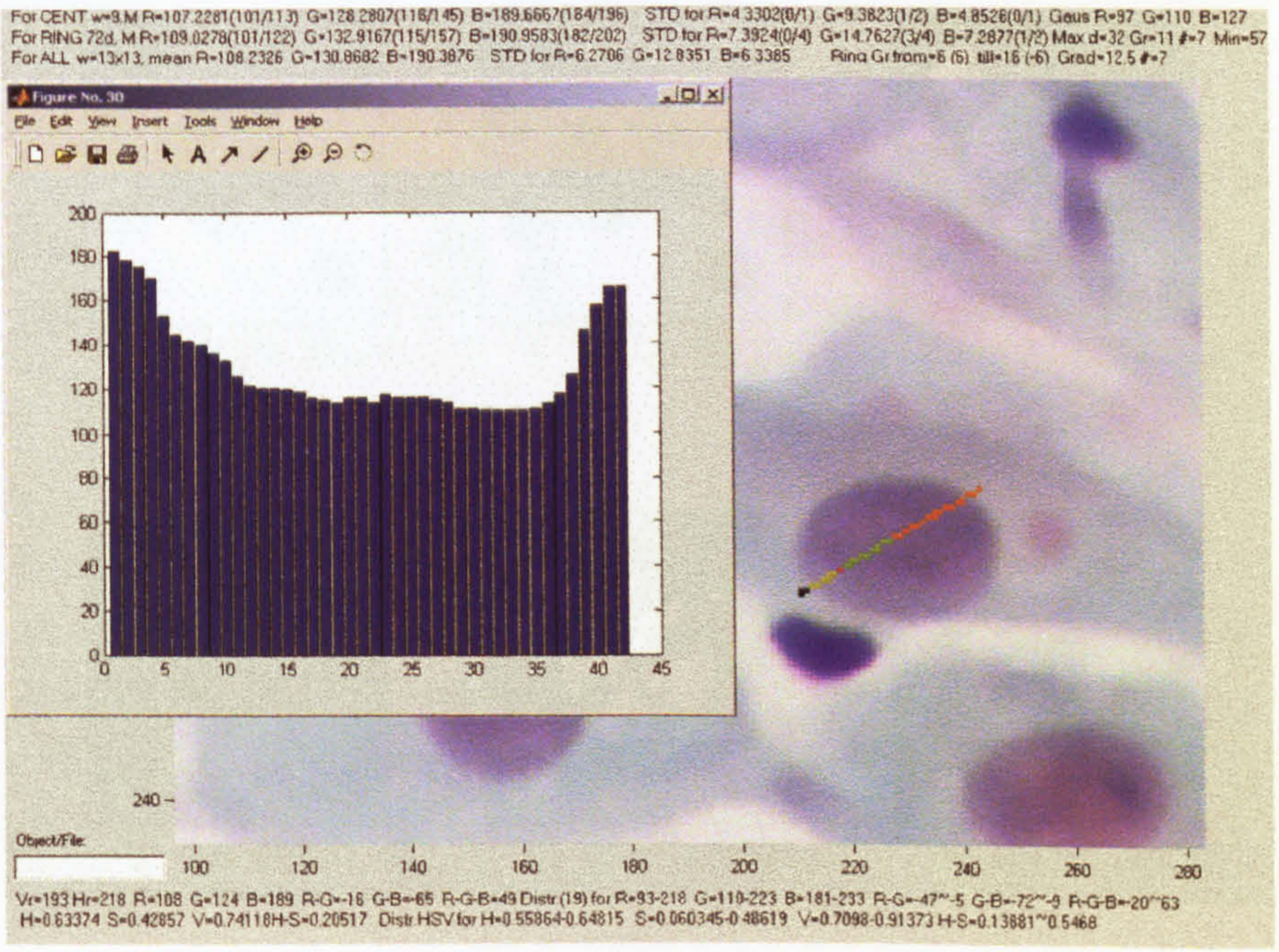



Figure 4.5: Result application Oriented Filter to an image

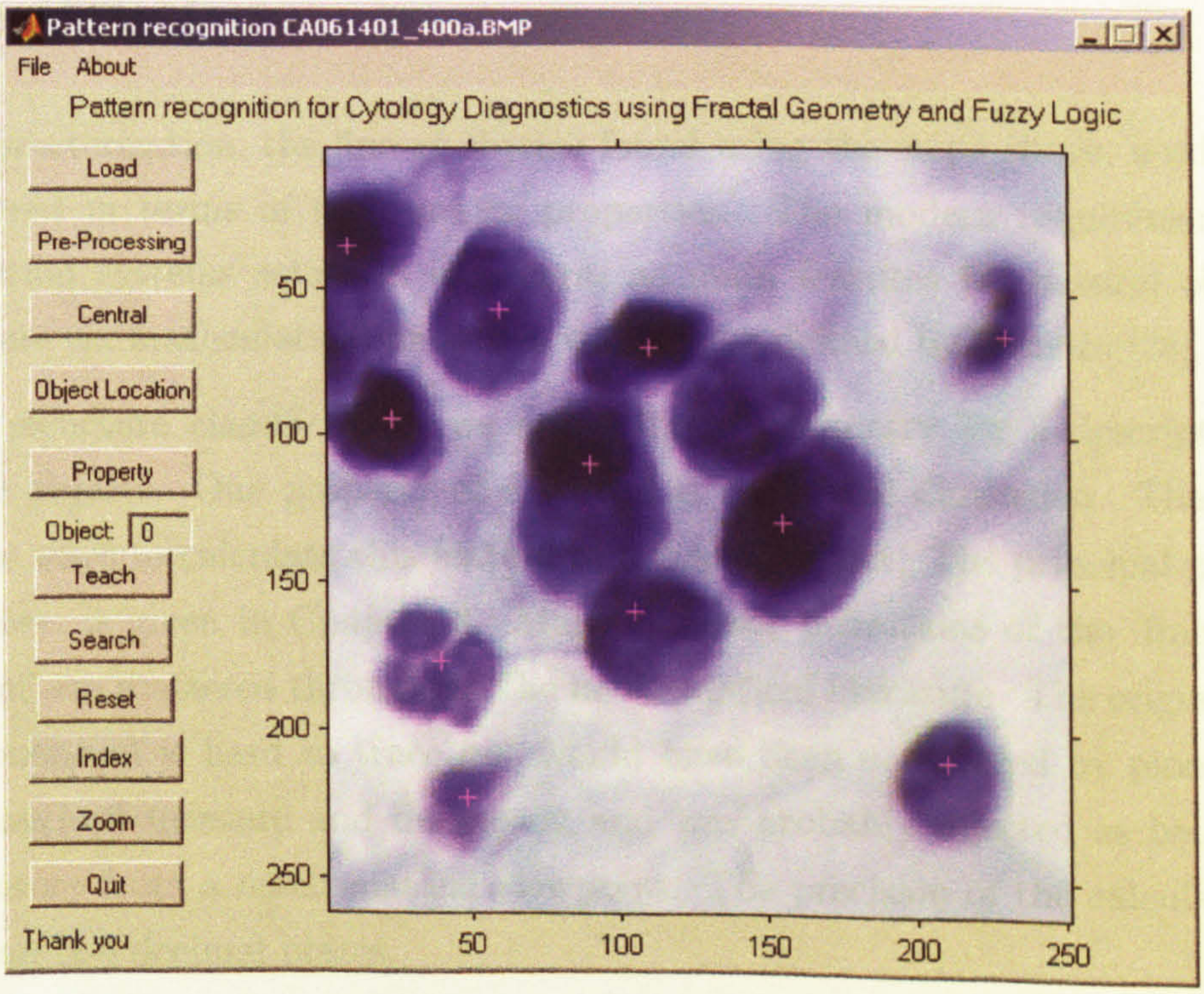



Figure 4.6: Result center calculation algorithm

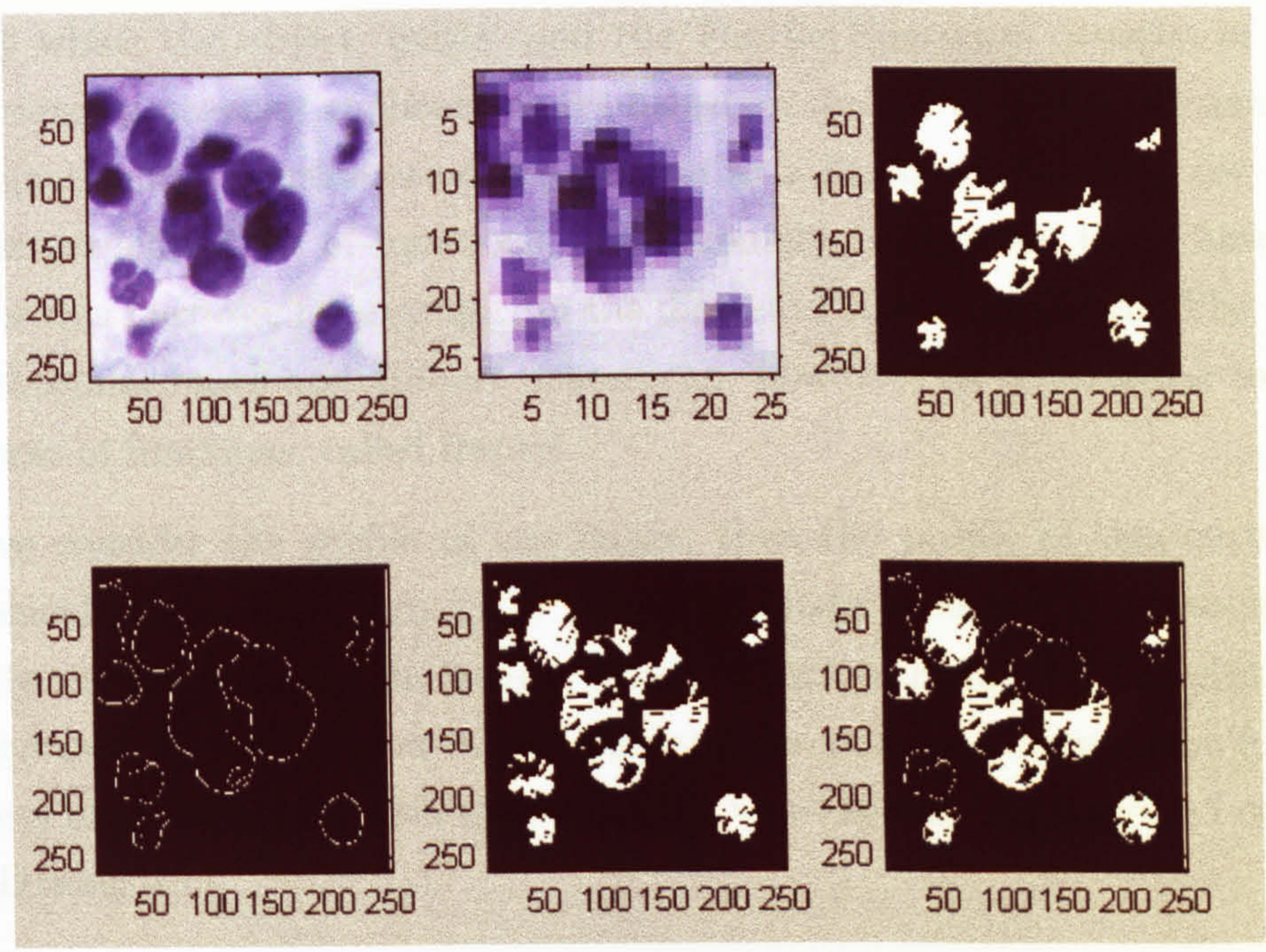



Figure 4.7: Segmentation of nuclei (Index Image)

## 4.2 Precision calculations of the measure of structure

For characterization, the line of objects found using the steps above, need to be considered in terms of their major properties. The modern requirements for recognition systems establish structures as main features for natural objects. There are six mathematical measures which were defined by Tamura [28].

For structure classification, we apply fractal geometry for a description of natural objects. One property of a fractal is its fractal dimension. There is a number ways to calculate this feature of a fractal object. The principal method used here, is given in Chapter 3. Many different definitions of the 'fractal dimension' are scattered throughout the mathematical literature. The origin of the box dimension is hard to trace but would have been considered by pioneers of the Hausdorff measure and dimension and was probably rejected as being less satisfactory from a mathematical viewpoint. The precision of the calculation is less than two decimal places.

The calculation of the Fourier dimension provides a much better result [41]. However, in our case, we have to estimate an image with a lower resolution than

that at which the object 'exists' and the Fourier Spectrum consists of noise. Many signal processing applications are based on the use of different transforms. The signals under consideration are written as a linear combination (or series) of some predefined set of functions. Traditionally, orthogonal bases have been used for this purpose, for example, in the discrete Fourier transform. The theory for orthogonal bases and Hilbert spaces can, however, be generalized to other sequences of functions, called frames.

If we consider the profile of the image, then the profile of this curve does not coincide a sine-wave signal. To obtain adequate accuracy, it is necessary to magnify the resolution of the image, which in turn introduces a new distortion. For increased accuracy on low-resolution data, we consider a convolution function of a form more consistent with the profile of a video signal. For a signal $I$ consider the following 4.1.

$$\begin{aligned} F(k) &= \sum_{n=1}^{N} I(n) \arccos\left(\cos\left(\frac{2\pi(\mathrm{k}-1)(\mathrm{n}-1)}{N} - \pi/2\right)\right) - \pi/2 \\ &\quad -i \arcsin\left(\cos\left(\frac{2\pi(\mathrm{k}-1)(\mathrm{n}-1)}{N}\right)\right) \end{aligned} \tag{4.1}$$

For image $I$ with resolution $m \times n$, the spectrum is calculated in terms of

$$\begin{aligned} F(p,q) &= \sum_{m=1}^{M}\sum_{n=1}^{N} I(m,n) \\ &\quad \left(\left(\arccos\left(\cos\left(\frac{2\pi(p-1)(m-1)}{M} - \pi/2\right)\right) - \pi/2\right)\right. \\ &\quad \left.\left(\arccos\left(\cos\left(\frac{2\pi(k-1)(n-1)}{N} - \pi/2\right)\right) - \pi/2\right)\right) \\ &\quad -i\left(\arcsin\left(\arccos\left(\frac{2\pi(k-1)(p-1)}{M}\right)\right)\right. \\ &\quad \left.\arcsin\left(\cos\left(\frac{2\pi(k-1)(n-1)}{N}\right)\right)\right) \end{aligned} \tag{4.2}$$

Application of the power spectrum method involves the use of the expression given by formula 4.2 to compute the power spectrum $P_i$. We then consider the power spectrum of an ideal fractal signal given by $P_i = |k_i|^{-\beta}$, where $c$ is a constant and $\beta$ is the spectral exponent which is related to the Fourier dimension

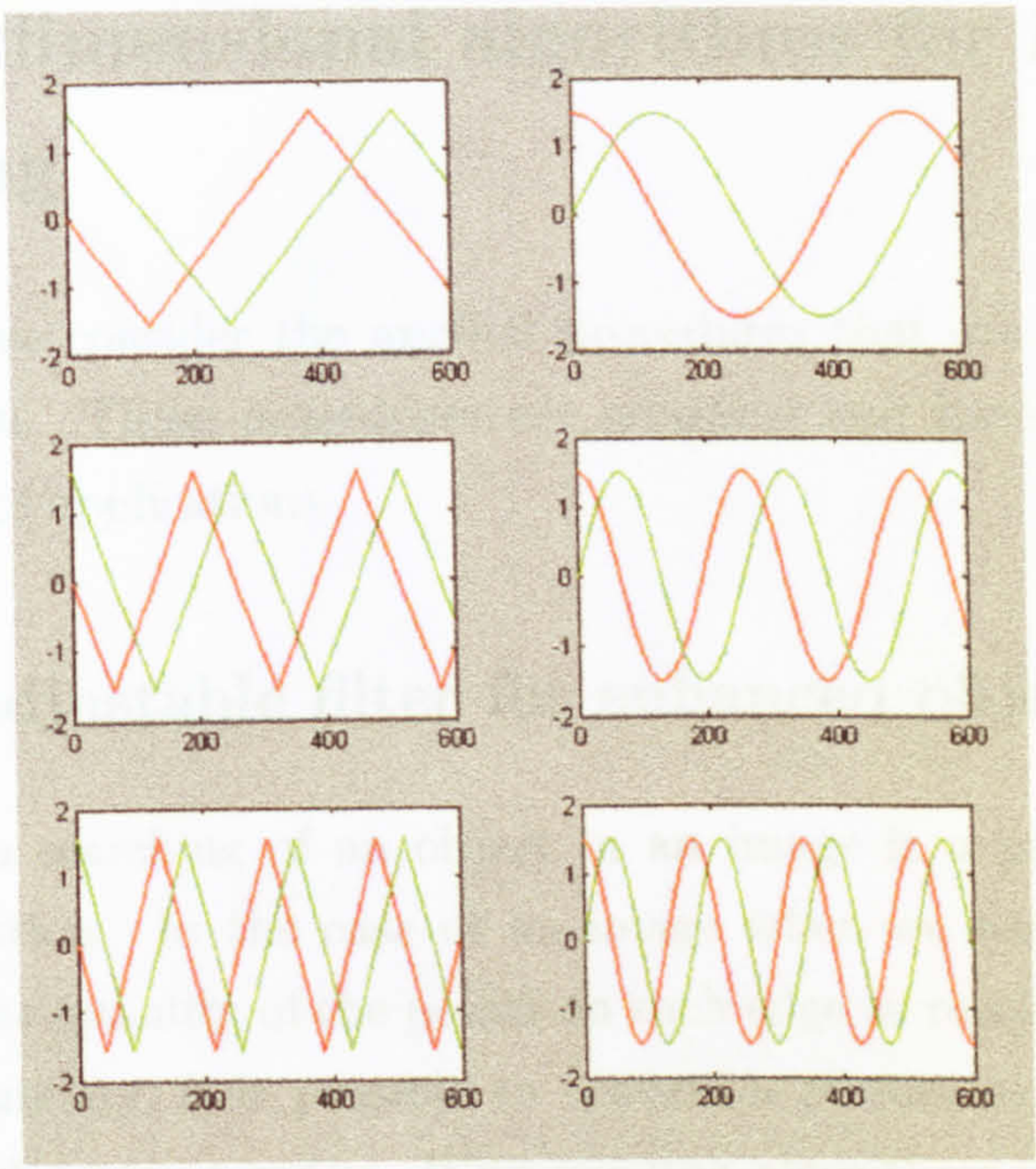



Figure 4.8: Orthogonal basis for triangle and sinusoid signals

$D_F$. Using the least squares approach, the value of $\beta$ and hence $D_F$ can be found for the input signal.

In two dimensions, the power spectrum is given by $P(k_x, k_y) = |k_i|^{-\beta}$, where $|k| = \sqrt{(k_x^2 + k_y^2)}$ and $c$ is a constant. This result is derived in Appendix A. The least squares method, for discrete data, yields the following:

$$\beta = \frac{N \sum_i \sum_j (\ln P_{ij})(\ln |k_{ij}|) - \left(\sum_i \sum_j \ln |k_{ij}|\right)\left(\sum_i \sum_j \ln P_{ij}\right)}{N \sum_i \sum_j (\ln |k_{ij}|)^2 - \left(\sum_i \sum_j \ln |k_{ij}|\right)^2} \tag{4.3}$$

and

$$C = \frac{\sum_i \sum_j \ln P_{ij} + \beta \sum_i \sum_j \ln |k_{ij}|}{N} \tag{4.4}$$

where $C = \ln c$ This solution allows us to drop the limits on the recognition of small objects since application of the FFT works well (in terms of computational accuracy) only for large data sets, i.e. arrays from 256 - 512 points to 128-256 points. The orthogonal basis for triangle and sinusoid signals is present in figure 4.8. The tests show that usage of the given function increases accuracy of the calculation by 5 percents.

## 4.3 Two dimensional algorithms for image processing

In this section, we consider the applied procedures that are necessary during object recognition. These procedures are adaptive and have no binding to a particular range of applications.

### 4.3.1 Self adjustable filter for enhanced object sharpness

The task of edge searching of an object in an image is a part of the process of object recognition. In the case of an image when we have no preliminary information on the quantity of the points on each edge or resolution of the image or particular boundary, it is possible to convert a picture to an auxiliary map with an increased contrast range. With existing algorithms, an image contrast increment does not give sufficiently good results to cope with an unknown level of difference between objects. Typically, noise appears causing an increase in the level of transformation parameters and at a low level there is no detection of the objects edges.

An image $I$, is represented in a computer memory in terms of an array $r \times c$ of points and the value of a particular point is determined as $I(r,c)$. One of the approaches to applying a filter or transformation to a two dimensional representation of the information is in terms of a sequence of masks $M$ over $m \times n$ points and subsequent calculation of a value for a central pixel depending on its environment.

Let us now consider an algorithm for calculating the value of a central point in a moving window $M$ with $m \times n$ points. The algorithm is applied sequentially and not recursively to all points of an image. For example, let us consider the image of cytology cells (Figure 4.9). The characteristic property of the given image is that during preparation of a sample, a cell can be fixed at a given angle, and as a result, during image processing it can have a different gradient rate on different boundaries. The mask sizes $m$ and $n$ are selected according to the proportional sizes of the object to the image. The method is compounded in the following stages:

1. The first step is to sort out the array $M[m \times n]$ in terms of increasing values.

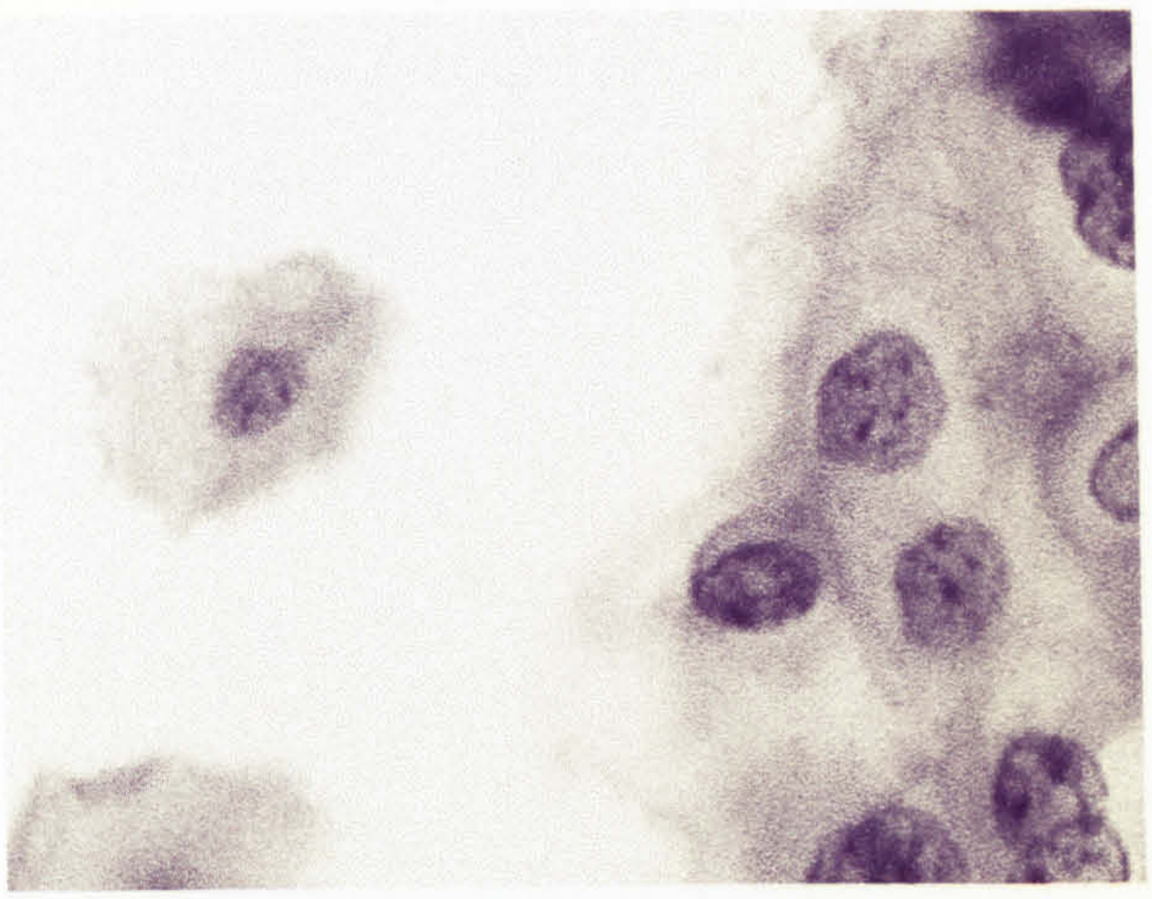


Figure 4.9: Cytology cells

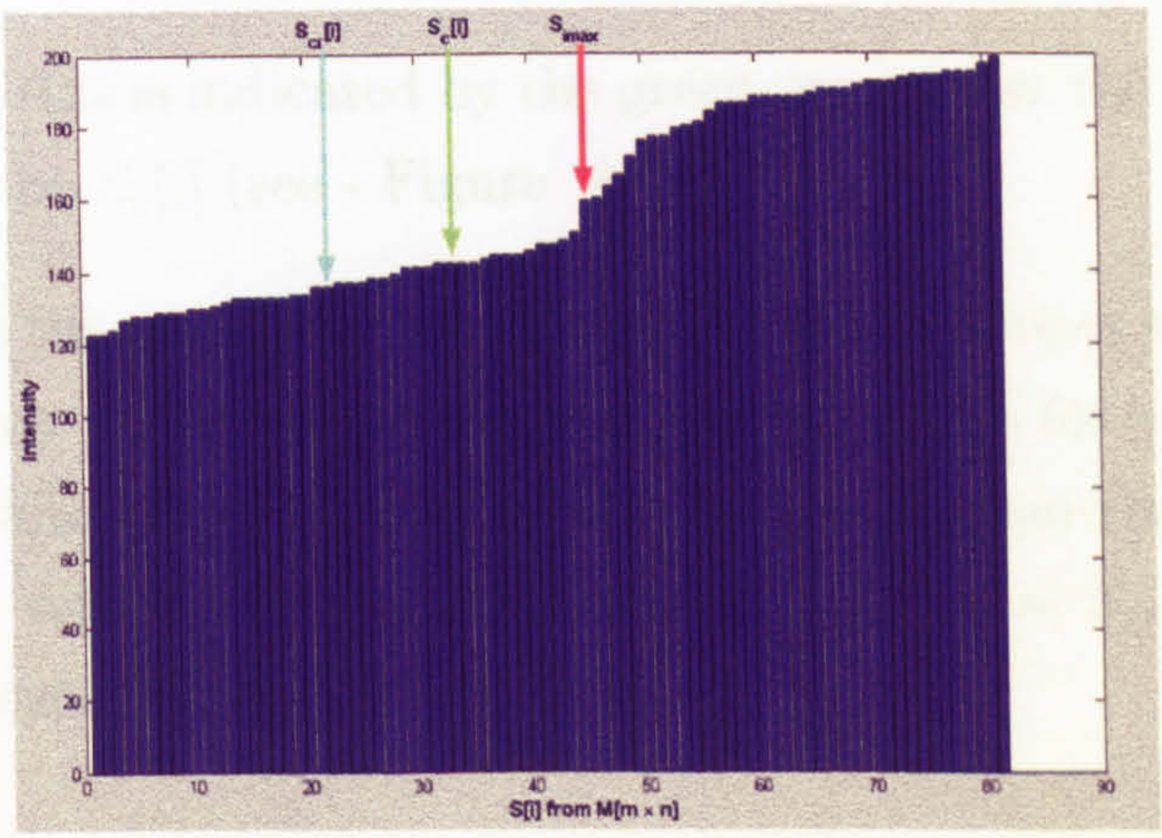



Figure 4.10: Sorted bar

The result of applying this operation gives an information represented in terms of a one-dimensional array $S[i]$ (Figure 4.10).

2. We define an index $i$ as a point with the greatest value of a gradient rate ($S_{imax}$). Otherwise, we determine a maximal gradient rate such that the given position of the window $M$ does not correspond to a boundary of the object. It is then possible to apply general filtering methods, e.g. to calculate the average value or to take the value of a point with a predetermined index and with this value assign to a central point. For example, in Figure 4.10 $S_{imax}$ is the point that shown by the red arrow.

3. We estimate in which part of the sorted array $S[i]$ from a mask $M$ there exists a value of the original central mask point $I_c(r, c)$. For example, in

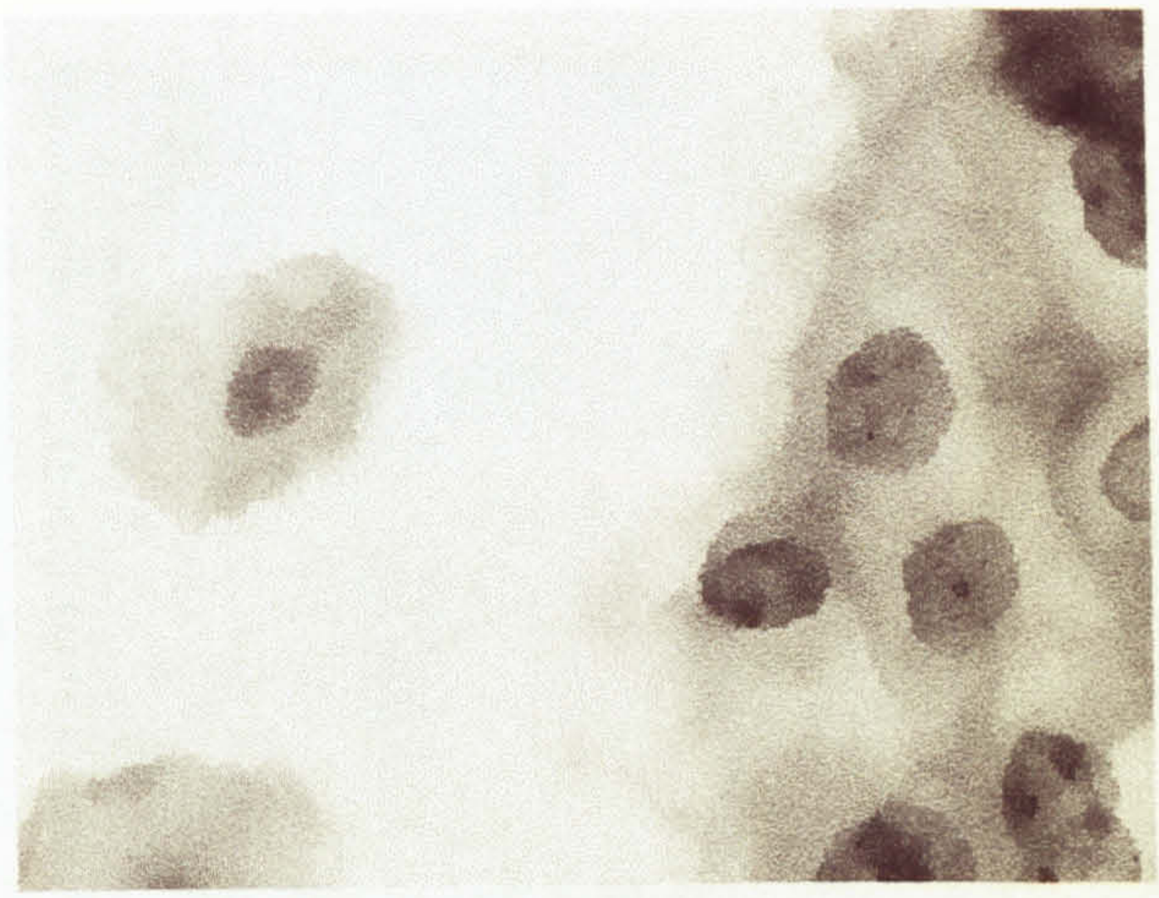


Figure 4.11: Filtered image

Figure 4.10 this is indicated by the green arrow. Let the name of this part of the array be $S_c[i]$ (see - Figure 4.10).

4. We estimate the parameter established by the developer which sets a factor on a boundary excretion - in percentage terms, 50% for example - and then define the value of point $S_{cr}[i]$ of the array $S_c[i]$ from the beginning of the array. This value is the resultant solution $I_c(r, c) = S_{cr}[i]$ displayed cyan arrow in Figure 4.10.

The result of the above algorithm is shown in Figure 4.11. Application of this filter allows us to observe very precisely the evolution of cell boundaries during the operation of the object recognition system. The full code of this procedure has been written using C++.

### 4.3.2 Detour over an object contour

After application of the edge detection algorithm and subsequent binarization, the two dimensional binary representation of the object is the index map $I_{bin}(r, c)$. This map has the same dimension as the initial image $I(r, c)$ where '1' corresponds to the object edge and '0' displays where is a background to the image.

At particular stages of image processing, it is necessary to receive a serial list of boundary coordinates of particular objects or coordinates in a figure, in which inscribed object is set. It is possible to use a special detour algorithm on an object contour to get this list of coordinates. The polar start (particular point

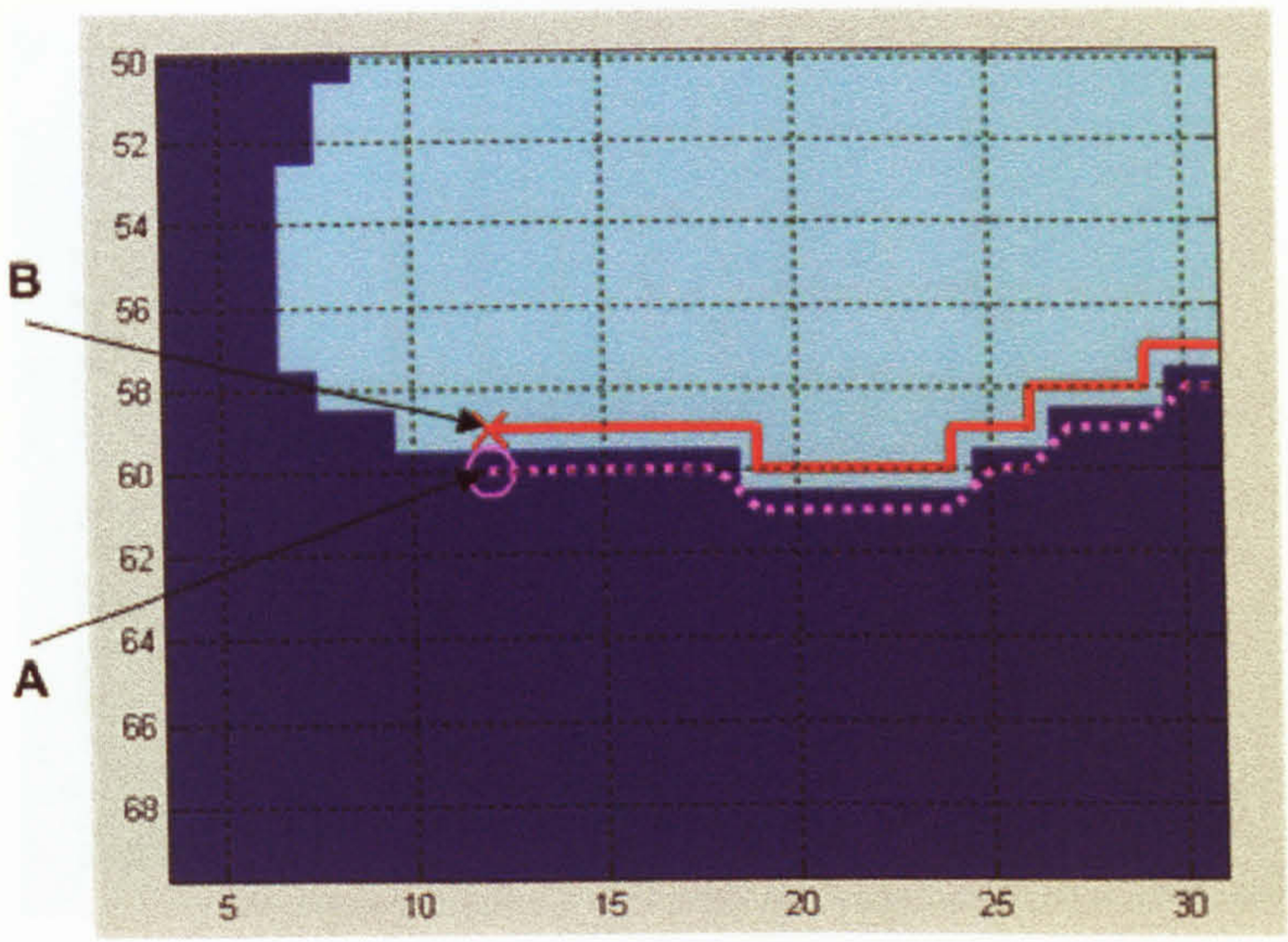



Figure 4.12: Detour by object contour

'A' in Figure 4.12) is not significant and if necessary, can be determined from previous stages of the processing.

This algorithm is efficient and accurate. Let us consider the image in Figure 4.12. For simplicity, let the detour algorithm for object contour be named 'Sprocket wheel' because this virtual sprocket is rolled on a virtual contour. Let us now zoom in on the image and find out how this sprocket wheel looks together with the binary map of an object as in Figure 4.13. Let us take the minimum radius of a wheel equivalent to the distance between two points on the image. Then, the surface of the wheel will correspond to a circle of 3 by 3 points and length 8 points. Let point 'A' corresponds to a wheel axle, with the dashed-line curve showing its track. Then one of the points of the wheel will certainly be connected to the objects edge at point 'B' (Figure 4.12). From the initial conditions, the coordinates of arbitrary points 'A' and 'B' is known. These coordinates can be recovered from preliminary phases of the processing or can be found by scanning for a nearest transferring $0-1$. Thus, the coordinate of a point 0 will correspond to the coordinate of a point 'A', and 1 according to a point 'B'.

The direction of movement has no value in the example above so let us consider a counter-clockwise motion. The full listing of this procedure is written in the programming language C++. The motion of the virtual sprocket is continued along the boundary while the current position of an axis meets the initial conditions. We assume that the object does not involve the 'walls' of an image

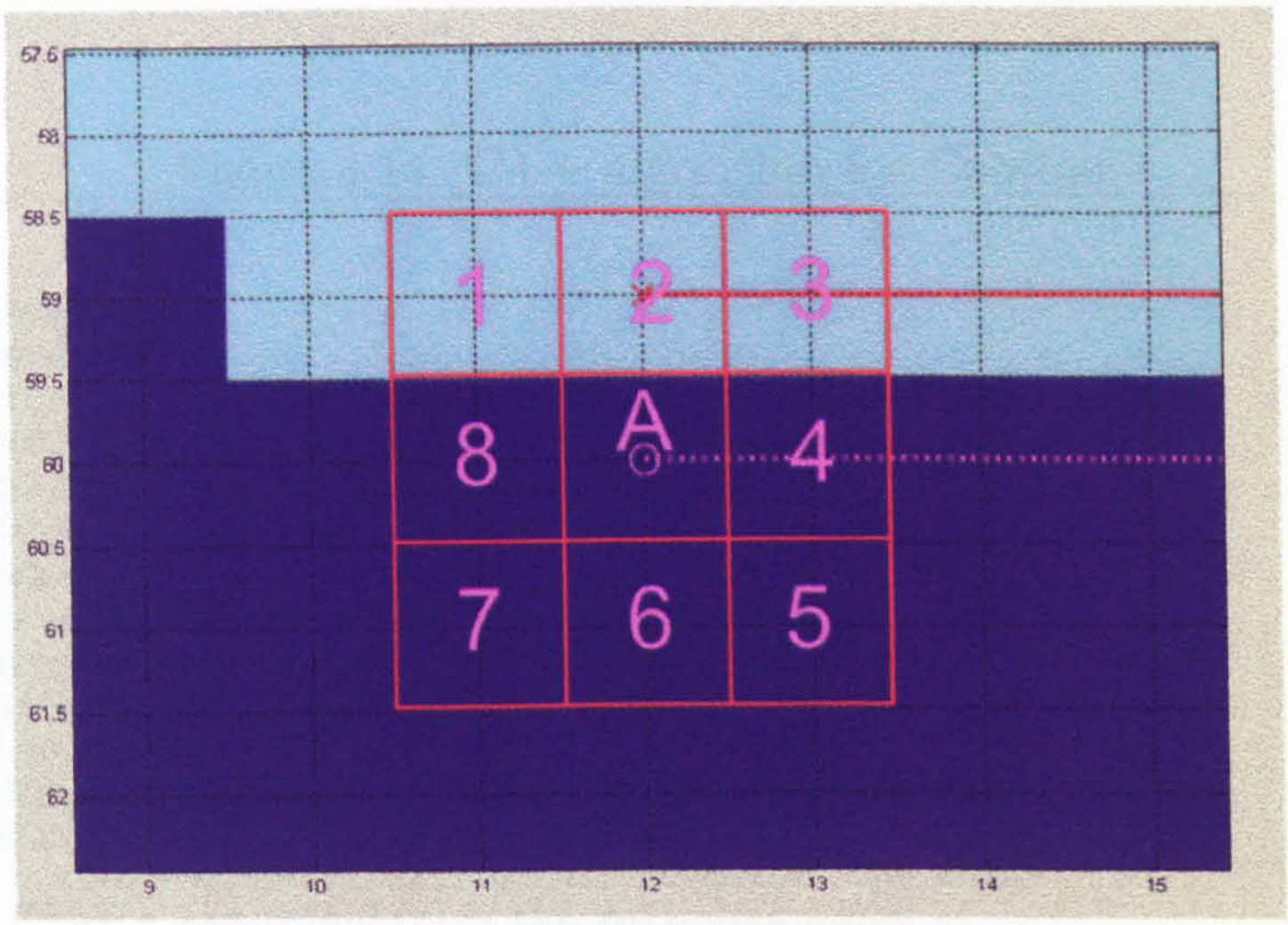



Figure 4.13: Structure of sprocket wheel

otherwise it is required to follow-up and zero tracing the image or make additional checks.

The matching algorithm for the edge coordinates is presented in Figure 4.14. The purpose of this example is to obtain the list of coordinates of the edge points of the object.

The result of the alghorithm described above is seen in Figure 4.15 together with the object in Figure 4.18. The red line of connected points in the figure shows the edge of the object. Their coordinates are contained in arrays $ListDotsX[0...ks]$ and $ListDotsY[0...ks]$ for $X$ and $Y$, accordingly.

The advantage of this algorithm is that the movement of axial coordinate occurs less often than the change of edge points; therefore, the computational costs are reduced average by of 2 - 3 times and depends only on the complexity of the figure.

### 4.3.3 Convex hull algorithm 'Spider'

As in the previous algorithm described, after binarization of image $I(r, c)$ we acquire a two dimensional binary representation of an object on the index map $I_{bin}(r, c)$ which has the same dimension as the initial image (1 corresponds to the presence of an edge or body of the object and 0 corresponds to the background of the image). Let us consider the task of obtaining the coordinates of a convex

Figure 4.14: C++ algorithm for Contour

```
int ks=0;
ListDotsX[0]=StartX;
ListDotsY[0]=StartY;
long DotX[9]={0,-1,0,1,1,1,0,-1,-1};//Extension for wheel
long DotY[9]={0,-1,-1,-1,0,1,1,1,0};//surface from axle.
int Ox=StartX;// Sets position of points 'A' and 'B'.
int Oy=StartY+1;
int ht,HaveToch=2;
do {  // Cycle while not return to initial coordinates
  for (nl=1;nl<=7;nl++){ // Cycle by surface of wheel
    ht=HaveToch+nl;
    if (ht>8) ht=ht-8;
    x1=Ox+DotX[ht]; //Calculation the following coordinates
    y1=Oy+DotY[ht]; // for surface of wheel
    if (*(pp + x1*h + y1)==0){ //If 0 then movements the wheel
      Ox=x1; // axle and calculation the following point of
      Oy=y1; // tangency for wheel surface with object edge
      if((ht==1)||(ht==3)||(ht==5)||(ht==7)) HaveToch=ht+5;
      if((ht==2)||(ht==4)||(ht==6)||(ht==8)) HaveToch=ht+6;
      if (HaveToch>8) HaveToch=HaveToch-8;
      break;
    }
    if (*(pp + x1*h + y1)==1){ //If 1 than checking the
      if ((x1==StartX)&&(y1==StartY)) break;// initial
      ks=ks++;
      ListDotsX[ks]=x1;// conditions and saving the edge
      ListDotsY[ks]=y1;// coordinate of object
    }
  }
} while ((x1!=StartX)||(y1!=StartY));
```

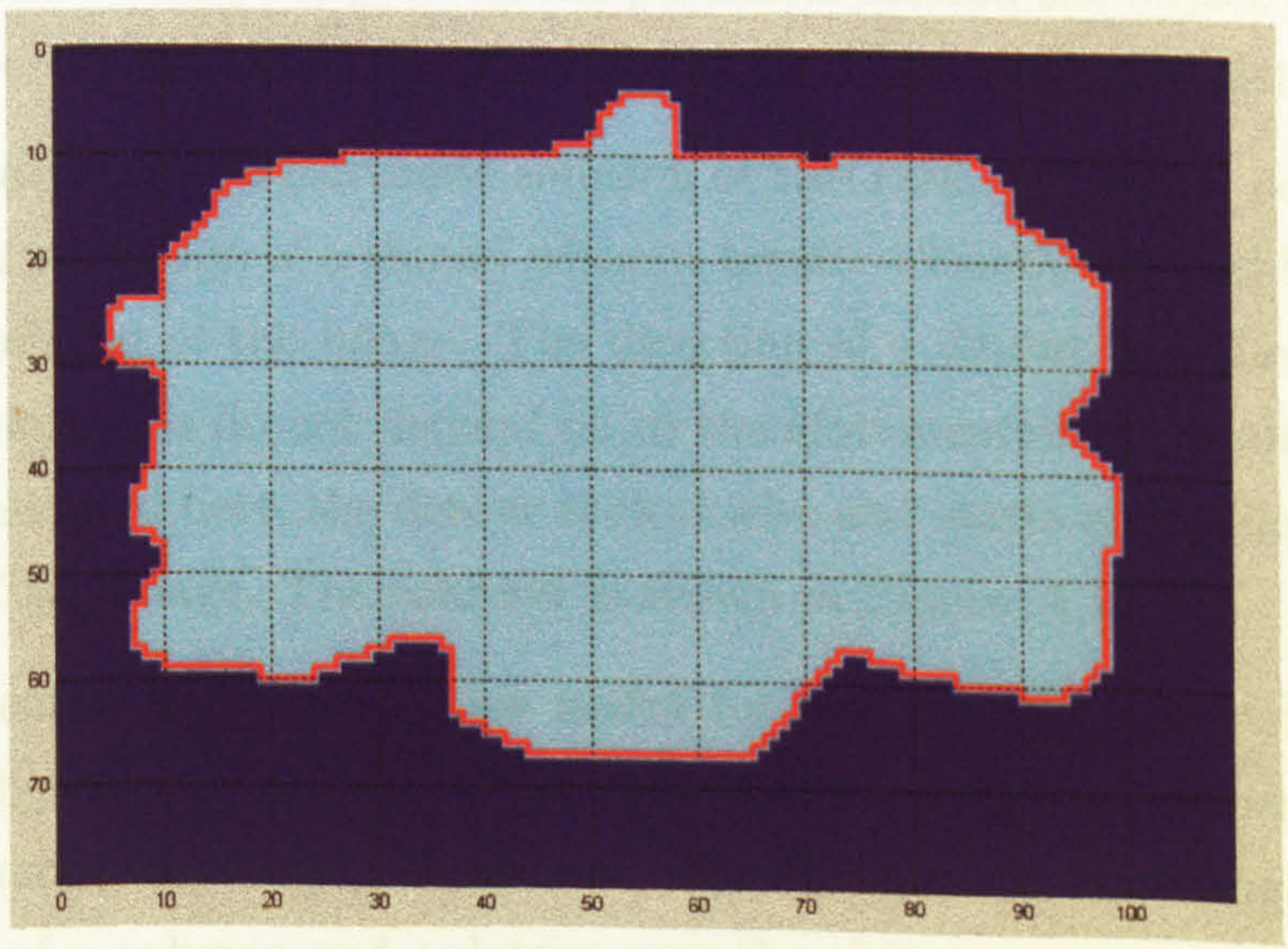



Figure 4.15: Result of Contour Walk Algorithm

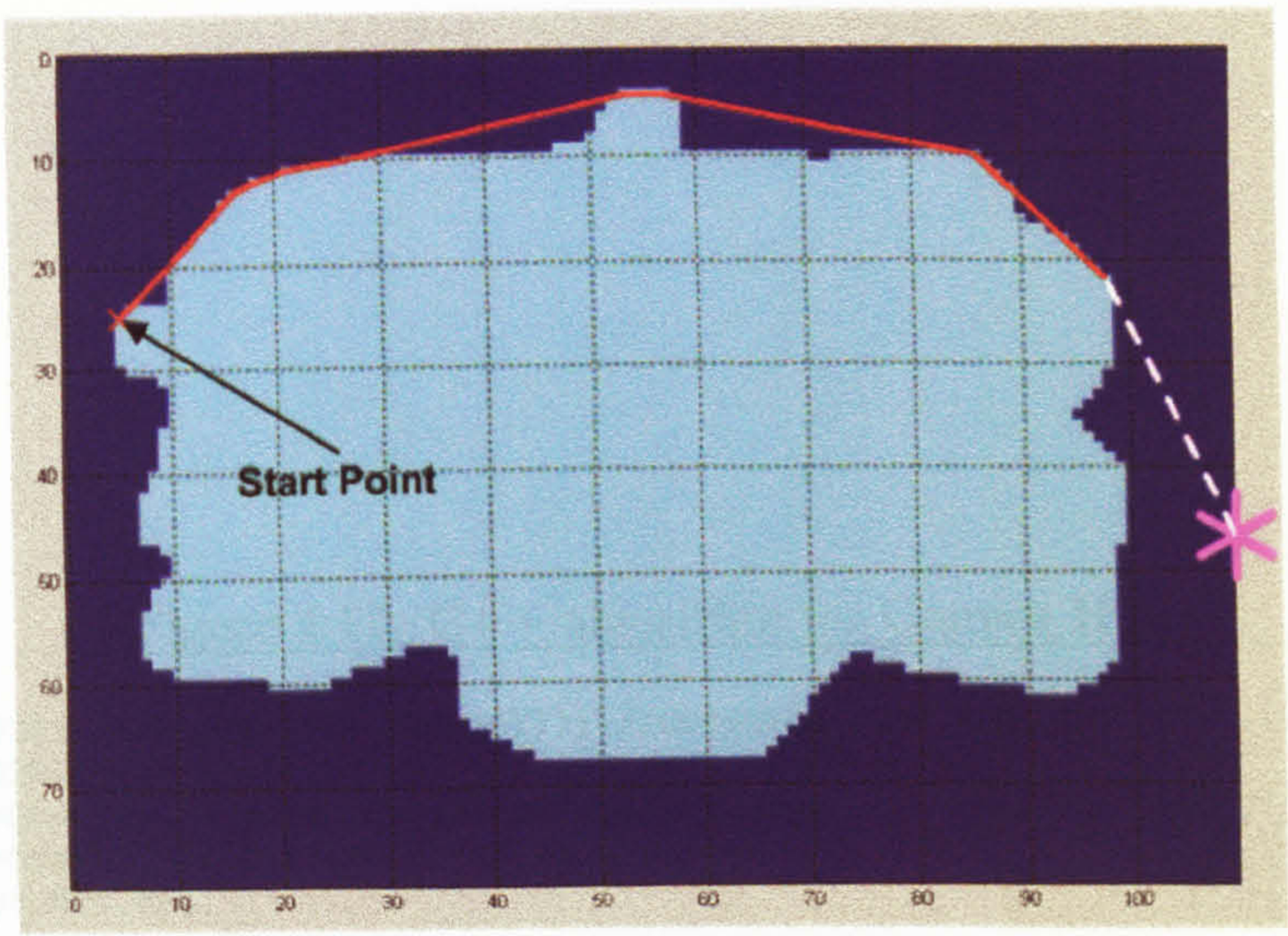



Figure 4.16: Obtaining coordinates for Convex hull

polygon. This task is given in the MATLAB function 'Qhull'. The algorithm applied in this thesis differs from that of MathWorks Inc one in terms of its simplicity, reliability and fast computation. The reason is that a number of cycles performed is limited and equal to the total border length of the object.

The main idea can be presented in terms of a 'Spider', which creeps on a wall of the map and pulls behind itself a thread. This thread is attached to the object. At the 'point of curvature' the thread stores the coordinates of the outer polygonal point. Thus, the path on the perimeter around the object provides the coordinates of all the outer polygonal points as shown in Figure 4.16.

Let us consider the algorithmic solution of this task. For initial conditions we should select a position of a thread without bends. Clearly, this will be along one of four boundaries of the image. The direction of a detour and the selection of the initial conditions do not depend on to the aforementioned conditions. In the example considered here, the detour is clockwise and starts along the left vertical boundary of the image. The solution is shown in Figure 4.17.

The presented algorithm is also useful for defining the geometrical location of separated points or objects. The algorithm successfully has been using the developed computer recognition system. The example of computing of the outer coordinates of a polygon and a detour over the object contours are presented in Figure 4.18.

Figure 4.17: C++ algorithm for Convex Hull

```
{NListDotsX[02*((maxX-minX)+(maxY-minY))] //Creation list dots
 NListDotsY[02*((maxX-minX)+(maxY-minY))]}//of object bound
ListDotsX[0]=StartX; // Sets the initial coordinates
ListDotsY[0]=StartY; // for other end of a thread
int nc=0,x4,y4,Mx4,My4;
double fi,cs,sn,step,r,RR,bz,sz;
for(nt=0;nt<(2*((maxX-minX)+(maxY-minY)));nt++){//Begin creeps
  fi=atan2(NListDotsY[nt]-StartY,...          // around object
  NListDotsX[nt]-StartX);
    RR=sqrt(pow((NListDotsX[nt]-StartX),2)+...
                          +pow((NListDotsY[nt]-StartY),2));
    cs=cos(fi);
    sn=sin(fi);
    if (fabs(sn)>fabs(cs)){ //Calculation the step length
      bz=fabs(sn);
      sz=fabs(cs);
    }else{
      bz=fabs(cs);
      sz=fabs(sn);
    }
    step=sqrt(pow(((sz*(1-bz))/bz),2)+pow((1-bz),2))+1;
    for (r=0;r<=RR;r+=step){ // Searching for all objects
      x4=round((double)StartX + r*cs);//in way of thread
      y4=round((double)StartY + r*sn);
      if (*(ppg + x4*h + y4) == 1){
        Mx4=x4; // saving last coordinate
        My4=y4; // in temporary variables
      }
    }
    if (((Mx4!=StartX)&&(My4!=StartY)) || //Last dot check
 ((Mx4==StartX)&&(Mx4==NListDotsX[nt])&&(Mx4!=NListDotsX[nt+1]))||
 ((My4==StartY)&&(My4==NListDotsY[nt])&&(My4!=NListDotsY[nt+1]))){
      StartX=Mx4; // Assign new start coordinates
      StartY=My4;
      nc=nc++;
      ListDotsX[nc]=StartX; // Saving list tops coordinates
      ListDotsY[nc]=StartY; // of polygon
    }
}
```

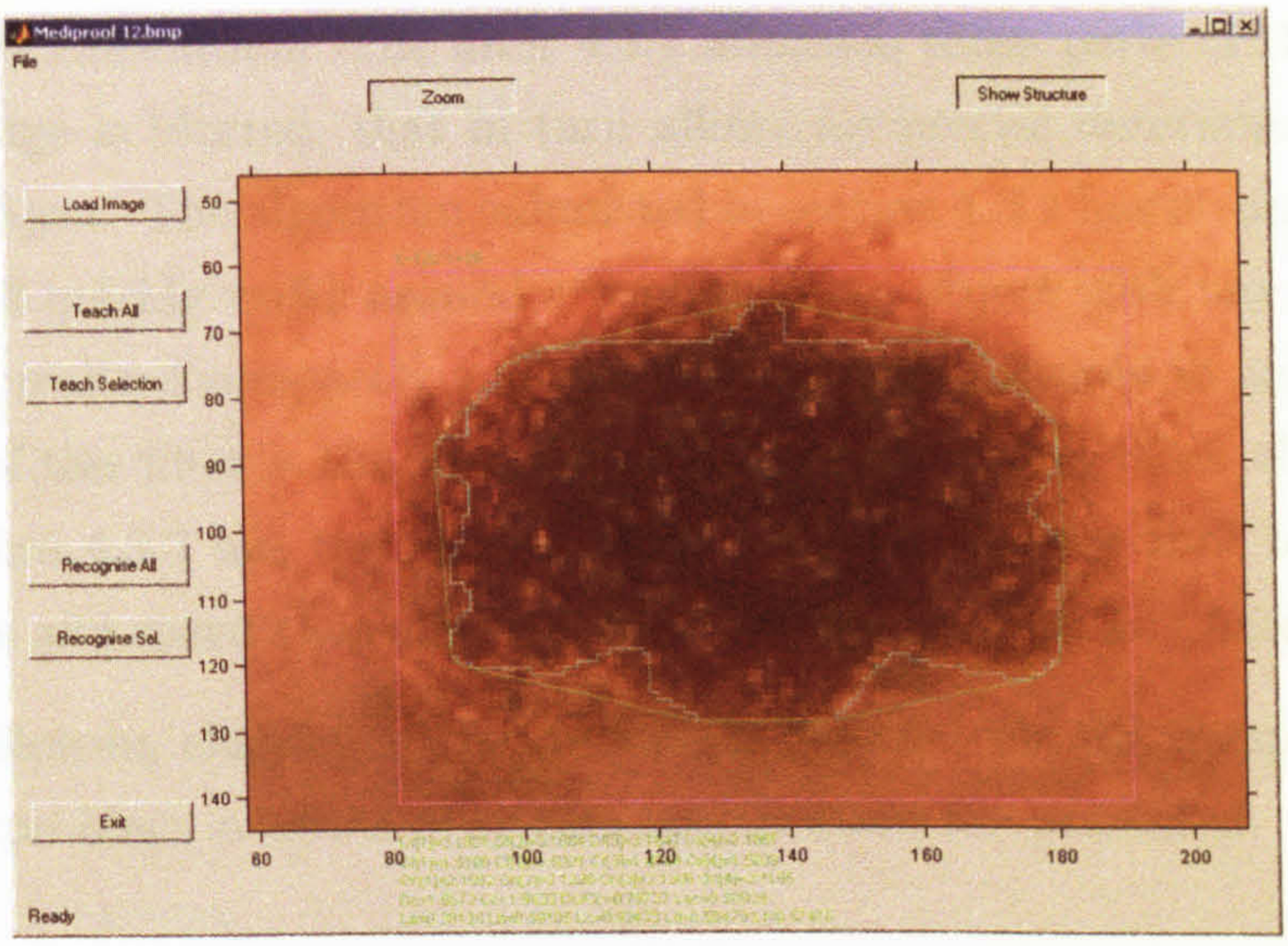



Figure 4.18: Object with Contour and Convex Hull

## 4.4 Summary

This chapter is concerned with the development of algorithms for solving specific pattern recognition problems associated with the class of images considered in this thesis. These algorithms have been tested using MatLab and C++ environments for prototyping and numerical optimization respectively.

The developed methods of recognition processing at each stage of the calculation have been compared to procedures realized in a modern mathematical package MatLab, publications in journals and the Internet. The comparative characteristic shows computing completeness, speed and compactness of these calculations. The latter consideration is very important in the realisation of a real time system.

The algorithms considered in sections 4.1 4.2 represent a new algorithmic approach for application in real time which can be optimised depending on equipment requirements. The algorithms considered in section 4.3 can be compiled on any PC platform under direct requirements of hardware.

In complex structure medical tests, the filter 4.1 allows us to allocate zones of interest (segmentation) especially in case when there is partial contact, one cell to another. The problem is very complex and there are no unified methods. The operating tests show good results. Applying a technique of calculation Fractal Dimension using the approach discussed in Section 4.2 provides better accuracy

for parameter calculation. The filter 4.3.1 allocates those parts of the object in which the edge is blurred, that in turn allows for precise determination of the edge of an object. The algorithm submitted in section 4.3.2 has no analogues and correctly and quickly walks around a contour of an object with any complexity. The filter is realised compactly and can be easy manufactured in any hardware. The result of this filter is the consecutive list of coordinates of an object's edge. The algorithm 4.3.3 has the general basis with the Convex Hull but differs is compactness and speed that allows it to be implemented in any hardware.

In the following chapter all the processes required for the complete pattern recognition problem in terms of system specification are presented.

# Chapter 5

# Process of Object Recognition

The image processing tasks are not of high for the human. For machine vision systems, the processing of a video sequence in real time still remains an unsolved problem. Although computers can 'see', they do not have a fully compatible model for explaining the processes of viusal image comprehension.

Suppose we have an image which is given by a set of brightness values $I[x, y]$ and it contains some object described by a set $S = \{S1, S2, , Sn\}$. It is then necessary to define a sample, which is somewhat close to this object. This task can be reduced to the construction of some function determining a degree of proximity of the object and a sample. Recognition is the process of comparing individual features against some pre-established template subject to a set of conditions and tolerances. The process of recognition commonly takes place in four definable stages:

(i) image acquisition and filtering;

(ii) object location (with edge detection);

(iii) measurement of object parameters;

(iv) object class estimation.

We now consider the common aspects of each step, which details are discussed in the following sections providing design features and their implementation together with their advantages, disadvantages and proposals for final solution.

Image acquisition depends on the technology that is best suited for integration with the particular application. For pattern recognition in histopathology,

high fidelity digital images are required for image analysis their resolution is determined by the microscope used for human inspection. Line scan cameras are suitable with surface inspection system. The colour images are, in general, relatively noise free they are digitized using a standard CCD camera and stored in a file.

## 5.1 Dualism in the localization tasks and object identification of an image

The majority of tasks needed for the analysis of video information include a search for some samples (one or several) on an image (task of localization) or search for a sample, most close to a given fragment of the image (task of identification), i.e. the task of image recognition. The task of localization and identification are indissolubly connected. For recognition of an object it is necessary at first to find out an appropriate image field and then to find out the object, which must be defined in terms of a set of features belonging to required object. In most simple cases the total task can be reduced only to identification and localization. For example, by applying a procedure of identification to each area of the image sequentially it is possible to find out the position of an object which provides an identification region. By sequential search all possible samples on the image it is possible to find out what samples are present on the image. However in practice it is generally not possible to use similar approaches. Object analysts typically use transformations of pattern that provide more convenient data for driving a 'search engine'.

Let us consider two types of transformation which are invariant and parametric. The invariance transforms the image and converts it to some set of properties, which are invariant to a position of object. The task of localization in this case disappears. The parametric property transfers the data from a 'dot map' to its representation in a space of 'object parameters'. In this space, each object of the map will be given by a unique point, which coordinates are the object parameters values. As the task of searching and identification of a point in a discrete space is trivial, the task of recognition is reduced to the selection of an appropriate transformation.

## 5.2 Methods of the Object Recognition

### 5.2.1 Straight Line Method

The straight line method consists of calculating some function of a pointwise coincidence between the map of the object and the image together with search for maximum of this function correspond they to the closest object pair or sample. In terms of 'similarity function', this method can be represented as a sum of square deviations, a sum of the modulus of deviations or as a pair sum of multiplications of values of brightness (function of the greatest transparency) and others. The first two similarity functions compute the 'smallness' of a functional pair; instead of searching for a maximum it is necessary to search for a minimum.

Since not all fragments of an object are equally important for recognition, a broadly distributed functional evaluation matched with weighted coefficients can be undertaken on separate parts. Appropriate similarity functions can be used as a sum of the weighted squares of deviations, a sum of the weighted modules of deviations and the sum of the weighted multiplication of pairs of brightness values. The correct selection of weight coefficients are important in the field of identification. They can be calculated from a given set of samples. The common application for weighted comparisons occurs in the field of artificial neural networks. The advantage of usage of neural networks, bearing in mind their high efficiency, lies in the capability of introducing a flexible set of weights during operation (training system). This property becomes especially important, if a set is based on a non-stationary model, which varies in time while it is extended and updated.

The main disadvantage of point wise matching relates to the tasks considered here in this chapter. If the required object does not coincide completely with the image, then it is required to localize the object and to determine its coordinates, orientation, scale, brightness and so on, accurately separating it from the background, shadows, from overlap by other objects and to take into account invisible fragments of the object. Further more, all this must be accomplished using an acceptable pixel accuracy.

### 5.2.2 Correlation Methods

There are several ways of overcoming the complexities discussed above. The first one is related to the evaluation of coincidence function for all probable status of an object. The most popular method is the so-called correlative - extreme approach. This approach utilises the property of cross-correlation function in terms of the sample brightness, where the function has a maximum at the point of coincidence between the object and sample. The cross-correlation function is calculated for all permissible positions of the object in the image plane and a search is undertaken for the maximum value of this function. The value of the maximum gives the information on the range in similarity between the object and the sample, and also the location the cross-correlation of maximum provides the position of the object in the image plane. The advantage of the correlative - extreme approach is in its simplicity and high reliability. For this reason this method has found a wide range of applications, where it can be considered as the most reliable method for registering the position of a control object in received map of a physical field (e.g. visible images, distribution of altitudes and depth, intensity of a gravitational field etc.). The disadvantages of this method are related with thier excessive computational complexity. The main reason is connected with a cross-correlation function especially when it does not allow us to define the position of a maximum analytically. The method is not very efficient when required to specify the orientation and scale of an object, and also due to its sensitivity to changes in the brightness and reflectivity of an object. The outline and different applications of the correlative - extreme method is surveyed in [37, 96].

### 5.2.3 Application of the Invariant Property

Another approach to recognition is based on excluding the use of pointwise matching processes and transforming the characteristics of an object to a more convenient form before application of an appropriate matching process. This must ideally include a set of properties that are invariant to any geometrical transformations. Such invariant properties allow us to cancel any misalignment between an object and sample that may be related by different brightness and contrast, to overcome any uncertainty between position and orientation of an object and

to cancel influence of the background and other objects.

Invariant feature extraction methods are divided into two classes: analytical and geometrical. Analytical methods are related to an analytical function from which the brightness values of isolated points of the image may be extracted. The statistical moments of the first and second orders are examples of elementary analytic invariant solutions to the displacement problem for example. It is easy to conceive this in terms of the expectation that the brightness of points (mean brightness) does not depend on the position of an object in the image plane. The same is true for dispersion, which plays an important role in image contrast. Further, an object can be characterized by a set of autocorrelation coefficients, which are also invariant to displacement. We can also apply rotational and scale invariant techniques based on computing second order moments after transformation an image to the polar plane. Analytical invariant methods have found applications in problems concerning pattern recognition in neural networks for which are considered in [80, 2] for example.

Geometrical invariance is usually attributed to relationships between geometrical characteristics of an object that do not change the positions of objects of a particular class. Elementary examples of geometric invariance to shifting and rotation include the area and perimeter of an object. The selection of geometrical invariant properties depends on the type of object considered. For example, if it is required to find out on an image a flat detail having two 'holes' for features that invariant to position, it is convenient to take the central distance between the holes. Wide applications in the task of analyzing video scenes involve such invariance in terms of the ratio of the linear dimensions of projections which are invariant to the full group of affine transforms; this is particularly important in the analysis of projections of three-dimensional scenes in an image plane [35]. For example in the use of geometrical invariance in technical vision is considered .

In general, it is much easier to compute analytical invariance. However, their is an open problem in terms of the influence on a background. For geometrical invariance, the problem of influence on a background with composite texture is less clear. Geometrical invariance requires the computation of object representation in terms of a set of parameters characterizing geometrical primitives. An addition problem concerns with the sensitivity of geometrical invariant parameters to noise.

### 5.2.4 Geometrical Primitives

An approach that deserves special attention is concerned with the preliminary transformation of an image into a set of geometrical primitives - lines, circles, bars, angles, figures and other specific features. The image analysis is in two principal steps. The first step is concerned with transformation of a dot map into a set of interdependent geometrical figures (vectors). An analysis of the selected geometrical properties is then made for example via a correlative - extreme method.

A geometrical representation gives some essential advantages. A geometrical description provides a more convenient form of analysis compared with a set of points. It typically requires significantly less processing time which is important for real time applications. Geometrical objects can be passed in terms of image contours and therefore depend only on the form of a contour and the difference of colour inside and outside a given figure and do not depend on absolute luminance. This helps to eliminate the problem of background influence. A description of an image in terms of geometrical figures allows the restoration of hidden object segments which are otherwise distorted due to overlap by neighboring features or shading. The principal complexity of this approach consists of selection of the geometrical representation. It should completely describe the form of an object at any probable operational conditions. On the other hand, it should be simple enough to implement in real-time mode. There is no general purpose solution of the problem discussed above. In majority of practical cases, the use is made of 'binding' a geometrical form to any particular type of object within some context that an application permits. However, for many applications involving objects of a fuzzy form, such an approach is unacceptable.

The wide applications that involve the analysis of a set of geometrical primitives can often be solved using a method based on the generalized Hough transformation [15]. The main idea of such a transformation consists in representing elementary fragments of a contour in terms of some point in a space of parameters of an affine transformation group. The main task is then to search for this point (i.e. search for the maximum in the array of possible values). The disadvantage of this method is the same as discussed earlier. It is necessary to have *a priori* information on the shape of the object. This approach traditionally focuses on linear object features - sections and straight lines. Such features are universal

and occur in many natural images and are simple and immediate to extract. The use of linear feature recognition for motion definition, for example, along with other applications are addressed in [32, 78]. There are two principal problems with this approach. First, the design of an algorithms for line extraction requires high possess efficiency and adequate accuracy. Second, such algorithms used for the comparison of sections require very high output accuracy.

## 5.3 Implemented Approach to Object Recognition

The present solution detects objects by computer analysis from physical objects. The procedure of object detection is performed at the segmentation stage and should be adjusted for the each area of application. The recognition algorithm then makes a decision using a knowledge database and outputs a result by subscribing objects. The 'expert data' in the application field creates the knowledge database by using the supervised training system with a number of model objects.

The recognition process is illustrated in Figure 5.1, a process that includes the following steps:

1. Image Acquisition and Filtering
   A physical object is digitized and transferred into memory, e.g. the current image acquisition hardware available on the market for cervical disease which is compared in Table 6.3. The image is filtered to reduce noise and to remove unnecessary features such as light flecks.

2. Special Transform or Edge Detection
   The image $f_{m,n}$ is transformed into $\tilde{f}_{m,n}$ to identify regions of interests and provide an input dataset for the segmentation and feature detection operations. This transform avoids the use of edge detection filters which proved to be highly unreliable in the present application.

3. Segmentation
   The image $\{f_{m,n}\}$ is segmented into individual objects $\{f^1_{m,n}\}, \{f^2_{m,n}\}, \ldots$ to perform a separate analysis of each region. The phase includes such

operations as thresholding, morphological analysis, edge or contour tracing(section 4.3.2) and the convex hull method(section 4.3.3).

4. Feature Detection

   Feature vectors $\{x_k^1\}, \{x_k^2\}, \ldots$ are computed from the object images $\{f_{m,n}^1\}, \{f_{m,n}^2\}, \ldots$ and corresponding $\{\tilde{f}_{m,n}^1\}, \{\tilde{f}_{m,n}^2\}, \ldots$. The features are numeric parameters that characterize the object. The vector consists of a number of geometrical, fractal and statistical features in 1 and 2 dimensions. 1D features correspond to the border of object (cell, nucleus or cytoplasm) whereas 2D features relate to the surface within or/and around the object.

5. Decision Making

   This involves assigning a probability to a predefined set of classes (e.g. for cervical diseases Tables 6.1 and 6.2). Probability theory and Fuzzy Logic have been applied to estimate the class probability vectors $\{p_j^1\}, \{p_j^2\}, \ldots$ from object feature vectors $\{x_k^1\}, \{x_k^2\}, \ldots$. The fundamental problem of the project was to establish a quantitative relationship between features and class probabilities:

$$\{p_j\} = \mathcal{F}(\{x_k\}).$$

   A 'decision' is the estimated class of object coupled with the probabilistic accuracy.

### 5.3.1 Special Transform

A special transform (filter) is applied to the image before the segmentation and feature detection processes. This transform provides a good response to the features of particular interest i.e. cells and cell clusters. It can be considered to be a local lacunarity (section 3.3.4) computed using a moving window. In one dimension $x$, the transform of degree $k > 0$ is given by

$$\tilde{f}^k(x) = \left[ \int_\infty^\infty \left( \left| \frac{f(x')}{\int_\infty^\infty f(x'')g(x''-x',\sigma)\,dx''} \right| - 1 \right)^k g(x'-x,\sigma)\,dx' \right]^{\frac{1}{k}}, \quad (5.1)$$

where $g(x,\sigma)$ is a smoothing function (defining the moving window) such as

$$g(x,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}.$$

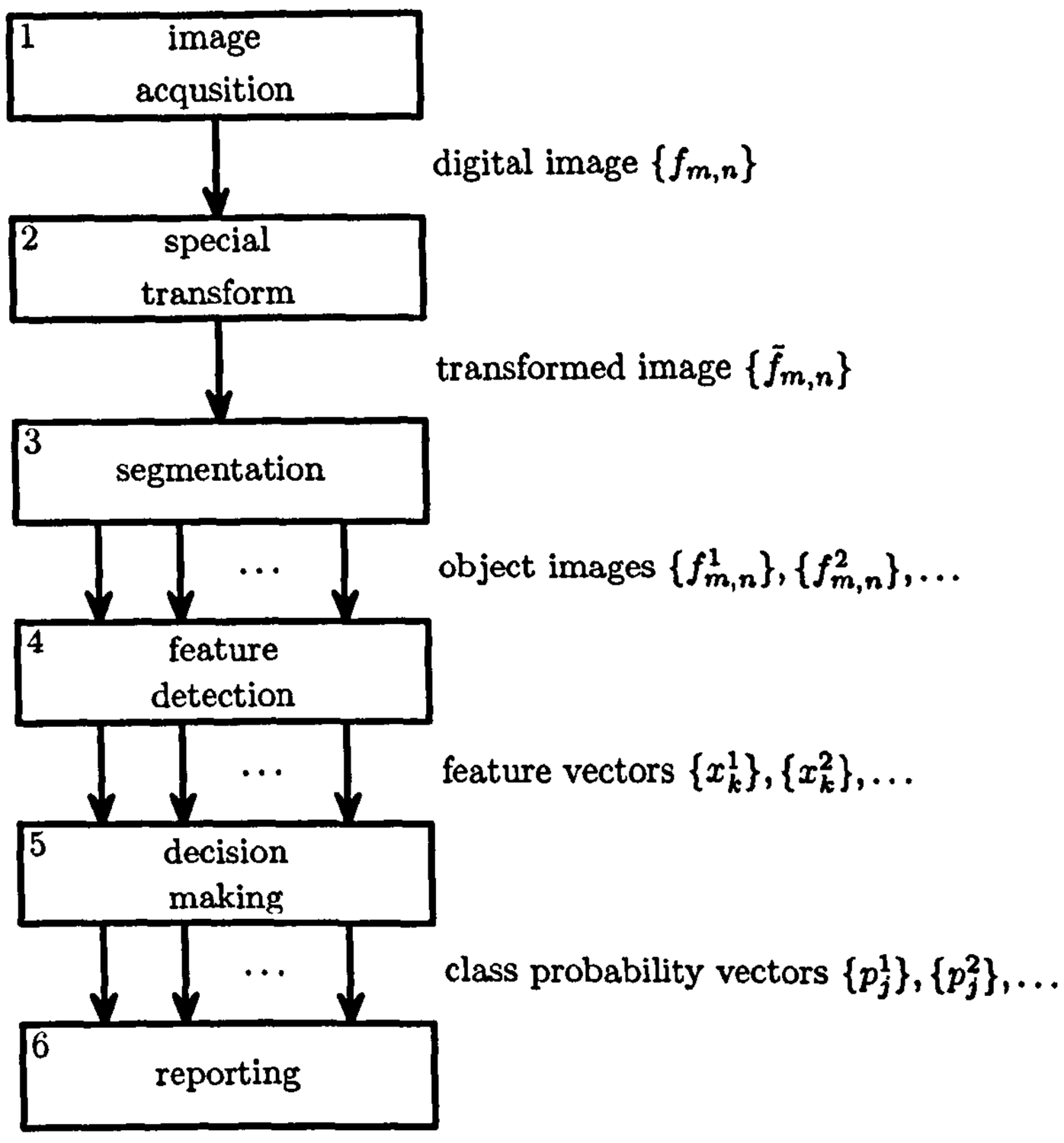



Figure 5.1: Recognition process

Degree $k$ defines the filter sensitivity to irregularities and typically lies between 1 and 3. For two dimensions, we can write

$$\mathbf{z} = (x, y), \quad \mathbf{z}' = (x', y'), \quad |\mathbf{z} - \mathbf{z}'| = \sqrt{(x - x')^2 + (y - y')^2},$$

and

$$\tilde{f}^k(\mathbf{z}) = \left[ \iint_{\infty}^{\infty} \left( \left| \frac{f(\mathbf{z}')}{\iint_{\infty}^{\infty} f(\mathbf{z}'') g(|\mathbf{z}'' - \mathbf{z}'|, \sigma)\, d\mathbf{z}''} \right| - 1 \right)^k g(|\mathbf{z}' - \mathbf{z}|, \sigma)\, d\mathbf{z}' \right]^{\frac{1}{k}}. \tag{5.2}$$

In the discrete case, the transform can be approximated as

$$\tilde{f}^k_{m,n} = \left[ \frac{1}{N} \sum_{m'=0,n'=0}^{m'^2+n'^2 \le R^2} \sum \left( \left| \frac{f_{m',n'}}{\frac{1}{N} \sum \sum_{m''=0,n''=0}^{m'^2+n'^2 \le R^2} f_{m'',n''} g_{m''-m',n''-n'}} \right| - 1 \right)^k g_{m'-m,n'-n} \right]^{\frac{1}{k}}, \tag{5.3}$$

where $R$ is the finite window radius and $N = \pi R^2$ is the number of points in window. The typical value of $R$ for $40\times$ magnification slides is $20 \div 50$.

### 5.3.2 Segmentation

Segmentation is implemented by adaptive thresholding and morphological analysis. The adaptive image threshold is given by

$$T_x = \frac{1}{2} \left( \min_y (\max_x f(x, y)) - \langle \max_x f(x, y) \rangle_y \right) + \langle \max_x f(x, y) \rangle_y,$$

$$T_y = \frac{1}{2} \left( \min_x (\max_y f(x, y)) - \langle \max_y f(x, y) \rangle_x \right) + \langle \max_y f(x, y) \rangle_x,$$

$$T = \begin{cases} T_x, & T_x \ge T_y, \\ T_y, & \text{otherwise}, \end{cases} \tag{5.4}$$

where $\langle \cdot \rangle_x$ and $\langle \cdot \rangle_y$ are the means within column $x$ and row $y$, respectively. This approach is an original solution for the most significant image object. So if an object covers the large space on the image then this filter is the fastest compact solution. For example, in the skin cancer application, there is preliminary information based on the fact that there is just one object on the image (Figure 6.9). In order to get a clear boundary, the morphological analysis selects objects with a predefine area.

### 5.3.3 List of Computed Features

Features describe the object state on an image and provides the input for the decision making engine (Figure 5.1). They are computed in the spatial domains of the original image $\{f_{m,n}\}$ and transformed image $\{\tilde{f}_{m,n}\}$. Also, features are extracted from different colour channels such as red (R), green(G) and blue (B) captured by the CCD array.

The following features or their derivatives are used in the recognition system:

**Average Lacunarity** $\Lambda^k$ determines the roughness of the object area. A typical value of lacunarity degree $k$ is 2. The average lacunarity is obtained directly from the transformed image:

$$\Lambda_k = \left\langle \left| \frac{f_{m,n}}{\langle p_{m,n} \rangle} - 1 \right|^k \right\rangle^{\frac{1}{k}},$$

where

$$\langle p_{(m,n)} \rangle = \frac{1}{N} \sum\sum_{(m,n)\in S} p_{m,n}$$

is the average inside the $N$-pixel area of the object $S$.

**Average Gradient** $G$ describes how the intensity changes when scanning from the object center to the border. The object gradient is obtained with the least squares method in polar coordinates:

$$g = \frac{N \sum\sum_{(m,n)\in S} r_{m,n} \tilde{f}_{m,n} - \sum\sum_{(m,n)\in S} r_{m,n} \sum\sum_{(m,n)\in S} \tilde{f}_{m,n}}{N \sum\sum_{(m,n)\in S} r_{m,n}^2 - \left( \sum\sum_{(m,n)\in S} r_{m,n} \right)^2},$$

where $N$ is the number of object pixels and $r_{m,n}$ is the distance between $(m, n)$ and the center $(m', n')$:

$$r_{m,n} = \sqrt{(m - m')^2 + (n - n')^2}.$$

The centers $(m', n')$ correspond to local maximums of $\tilde{f}_{m,n}$ within the cluster. The cluster gradient is the average of object gradients:

$$G = \langle g_i \rangle_{i\in I},$$

where $i \in I$ is the object index.

**Colour Composites $\Upsilon$ and $\Upsilon^D$** characterise the relationship between R, G and B layers of the transformed image. The triangle incircle formula

$$r_{\text{incircle}}(a,b,c) = \sqrt{\frac{(s-a)(s-b)(s-c)}{s}}, \quad s = \frac{1}{2}(a+b+c)$$

is applied to 'colour triangle' R-G-B so the pixel colour composite is obtained

$$\upsilon_{m,n} = r_{\text{incircle}}(a,b,c)$$

with

$$a = \tilde{f}^R_{m,n},$$

$$b = \tilde{f}^G_{m,n},$$

$$c = \tilde{f}^B_{m,n},$$

and $\upsilon^D = r_{\text{incircle}}(a,b,c)$ with

$$a = |\tilde{f}^R_{m,n} - \tilde{f}^G_{m,n}|,$$

$$b = |\tilde{f}^G_{m,n} - \tilde{f}^B_{m,n}|,$$

$$c = |\tilde{f}^R_{m,n} - \tilde{f}^B_{m,n}|.$$

The average colour composites are

$$\Upsilon = \langle \upsilon_{m,n} \rangle_{(m,n)\in S}, \Upsilon^D = \langle \upsilon^D_{m,n} \rangle_{(m,n)\in S}.$$

**Fourier Dimension $\beta$** determines the roughness and frequency characteristics of the object. It represents a measure for textures 4.2. The Fourier dimension describes a fractal image signal with the power spectrum of the form

$$P^2(k_x,k_y) = c|k|^{-\beta},$$

where $|k| = \sqrt{(k_x^2 + k_y^2)}$ is the spatial frequency and $c$ is a constant.

**Lacunarity (Gap Dimension)** $\Lambda_k$ characterizes the way the 'gaps' are distributed in the signal 3.3.4. The gap dimension is, roughly speaking, the number of light or dark spots in the image. It is defined for the given degree $k$ by

$$\Lambda_k = \left\langle \left| \frac{f_{m,n}}{\langle f_{m,n} \rangle} - 1 \right|^k \right\rangle^{\frac{1}{k}},$$

where $\langle f_{m,n} \rangle = \frac{1}{N} \sum f_{m,n}$ denotes the mean value. In the present version, an average of local lacunarities of the degree $k = 2$ is measured in the spatial and frequency domains.

**Symmetry Features** $S_n$ **and** $M$ are estimated by morphological analysis in 3D space ($X, Y$ — spatial coordinates, $Z$ — intensity). A symmetry feature $S_n$ is measured for a given degree of symmetry $n$ (currently $n = \{2, 4\}$). This value shows the deviation from the perfectly symmetric object, i.e. $S_n$ is close to zero when the object is symmetric and $S_n > 0$ otherwise. Feature $M$ describes the fluctuation of mass centres of points with different intensities; $M = 0$ for symmetric objects and $M > 0$ otherwise.

**Structure** $\gamma$ provides an estimation of the 2D curvature of the object:

$\gamma < 0$, if the mole bulging is less than a threshold,

$\gamma = 0$, if the mole has the standard bulging,

$\gamma > 0$, if the mole has a higher level of bulging.

**Geometrical Features** include minimum $R_{\min}$ and maximum radius $R_{\max}$ of the object (or ratio $R_{\max}/R_{\min}$), object area $S$, object perimeter $P$ (or ratio $S/P^2$), coefficient of infill $S/S_R$, where $S_R$ is the area of bounding circle.

### 5.3.4 Object Recognition Process

In order to characterize an object, the system has to know its mathematical representation. In this section the list of object features is considered. This set of features has been used to create an image of the object in the 'electronic

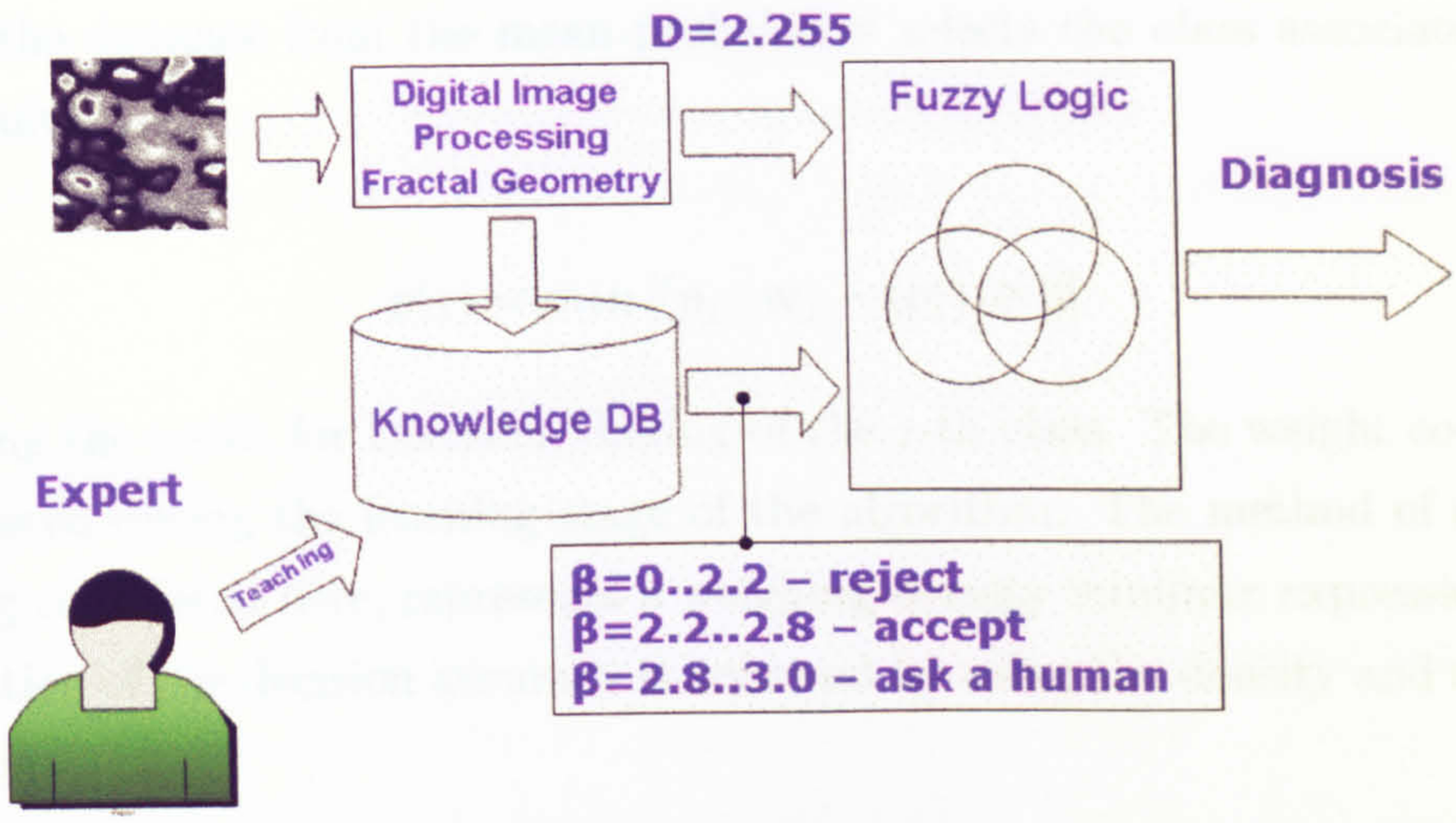



Figure 5.2: Architecture of Diagnostic Computational System

mind'. The basis for this area of application is the textural feature but for some objects, the Euclidian or morphological measure is also very important. In the case of a general application, all objects are represented by a list of parameters for implementation of supervised learning (section 5.3.5) in which the Fuzzy Logic system automatically adjusts the weight coefficients for the remaining features.

The methods developed represent the principal contribution of this Thesis to pattern recognition based on Fractal Geometry, Fuzzy Logic and the implementation of a fully automatic recognition scheme. The recognition procedure uses the decision making rule from Fuzzy Logic theory 3.4.

**Decision Making**

The class probability vector $\mathbf{p} = \{p_j\}$ is estimated from the object feature vector $\mathbf{x} = \{x_i\} \in \mathbf{X}$ and membership functions $m_j(\mathbf{x})$ defined in the knowledge database. If $m_j(\mathbf{x})$ is a membership function, the following equation defines the probability for each $j$-th class and $i$-th feature:

$$p_j(\mathbf{x}_i) = max\left[\frac{\sigma_j}{|\mathbf{x}_i - \mathbf{x}_{j,i}|} \cdot m_j(\mathbf{x}_{j,i})\right]$$

and the weight coefficient matrix gives:$w_j = w_{j,i}$ where $\sigma_j$ is the distribution density of value $\mathbf{x}_j$ in the point $\mathbf{x}_i$ of the membership function. The next step is to compute the mean class probability:

$$\langle p \rangle = \frac{1}{j}\sum \mathbf{w}_j p_j$$

where the distance from the mean probability selects the class associated with the result

$$p(j) = min\left[(p_j \cdot \mathbf{w}_j - \langle p \rangle) \geq 0\right] \tag{5.5}$$

giving the result for Decision Making of the $j$-th class. The weight coefficient is adjusted during the learning stage of the algorithm. The method of decision making considered here, represents a weighing-density minimax expression. The estimation of the decision accuracy is achieved by using the density and distance function:

$$d_i = |\mathbf{x}_{\sigma max} - \mathbf{x}_i|^3 + (\sigma_{max}(\mathbf{x}_{\sigma max}) - p_j(\mathbf{x}_i))^3$$

with an accuracy determined by

$$P = \mathbf{w}_j p_j - \mathbf{w}_j p_j \frac{2}{\pi} \sum_{i=1}^{N} d_i \tag{5.6}$$

### 5.3.5 Supervised Learning Process

The supervised learning procedure is the most important part of the system for operation in automatic recognition mode. The training set of sample objects should cover all ranges of class characteristics and with a uniform distribution along the universe of the membership function. This rule should be taken into account for all classes participating in training.

The expert is then in a corresponding field which defines the class and accuracy for each model object. The accuracy is the level of self confidence that this object belongs to the given class. The graphical user interface of the training procedure is shown in Figure 6.11. During this procedure the system computes and memorizes into the knowledge database, a vector of values of parameters $\mathbf{x} = \{x_i\} \in \mathbf{X}$ which forms the membership function $m_j(\mathbf{x})$. The matrix of weight factors $w_{j,i}$ is formed at this stage accordingly for the $i$-th parameter and $j$-th class using the following expression:

$$w_{i,j} = \left| 1 - \sum_{k=1}^{N} \left( p_{i,j}(\mathbf{x}_{i,j}^k) - \langle p_{i,j}(\mathbf{x}_{i,j}) \rangle \right) p_{i,j}(\mathbf{x}_{i,j}^k) \right| \tag{5.7}$$

The result of the weight matching procedure is that all parameters which have been computed and have not made any contribution into characteristic set of an object should be removed from the decision making algorithm by putting $w_{j,i}$ to null.

## 5.4 Summary

The methods discussed in the previous sections are typical of classical algorithms. They are rather prodigal from the computing point of view and using them to create a system working in real time is not usually possible. The applications considered in this thesis have required a symbiosis of the parametric representation of an object and its geometrical invariant properties. In comparison with existing methods, the approach adopted has the following advantages:

**Speed of operation.** The developed approach uses a restricted set of parameters on a visible object instead of a representation using a large set of points. This provides a considerably higher operational speed in comparison with existing schemes, especially with composite tasks, where the large majority of methods discussed previously require object separation. Note that, the approach considered here requires time computation of the object parameters from an initial image.

**Accuracy.** The methods constructed for the analysis of sets of geometrical primitives are, in general, more precise. Because the parameters are material values, which are not connected to an orthogonal grid, it is possible to design different transformations (shifts, rotational displacements and scaling) without any significant loss of accuracy compared with a set of pixels. On the other hand, the overall accuracy of a method will be influenced considerably by the accuracy of a procedure to extract the required geometrical tags. Generally, the accuracy of a method will be always lower, than, for example, classical correlative - extreme methods because of a padding error arising during the extraction of a parameter set. However, by using precise parametrization structures based on fractals, remarkably good results are obtained.

**Reliability.** The proposed approach relies first and foremost on the reliability of the extraction procedure used to establish the geometrical and parametric properties of objects, which, in turn, depends on the quality of the image; princi-

pally in terms of the quality of their contours. It should be noted, that the image quality is a common problem in any visual system and that in conditions of poor visibility all systems fail.

An additional feature of the developed approach considered in this thesis, is that the sub-products of the image processes can be utilised for the tasks that are related to image analysis such as a search for objects in a field of view, objects identification, maintaining an object in a view field, optical correction of a view point and many others. There are a number of tasks involving the relative motion of an object with respect to another object or with respect to background. The developed method can be also applied for collisions avoidance task.

Among the characteristic disadvantages of the considered approach, should be noted:

(i) The method requires a considerable number of different calculations to be performed and appropriate hardware requirements are therefore mandatory.

(ii) The accuracy of the method is intimately connected with the required computing speed. An increase in accuracy can be achieved but may be inadequate to the computing costs.

It is a difficult problem to acquire a template of samples in real life conditions with a uniform distribution along the membership functions. Even with the presence of a large number of training objects which are non-uniformly distributed on the membership function, it is impossible to reach an accurate recognition.

The original approach to the decision making problem proposed in this section includes the following important steps:

(i) The estimation of the density distribution is given accurately for the original samples in the membership function of the supervised learning process at the recognition mode which improves the quality of the recognition in real life conditions.

(ii) Special pre-filtering procedures provide a good response to the required features of the object without generating noise.

(iii) Special segmentation procedures efficiently select only those objects required.

(iv) Computation of the average Lacunarity helps to characterize some 'difficult' features (in terms of their classification) associated with the object.

The proposed integrated approach provides a complete suite of tools for pattern recognition an combination with supervised learning and fuzzy logic criteria. In the following chapter, we consider the implementation of our approach for applications that include, surface detection for evaluation and quality control, the detection of skin cancer and an early applications system for the evaluation of cervical smear images.

# Chapter 6

# Applications of the Developed Diagnostic System

## 6.1 Surface Inspection System

New technology for the production of high quality metals needs surface inspection quality control. A quality control system is required for several tasks such as: screening defected products, monitoring the manufactures process, sorting information for different applications and product certification for end customers. The system discussed here was developed for the Novolipetck Iron and Still Corporation in Russia. The program model was tested with images captured at rolling mill 2000.

### 6.1.1 Data Description in Surface Inspection

Digital images were taken from real metal surface in standard conditions using the same resolution as derived from a moving roll using line scan cameras. This data were saved into a data base in standard grey-scale format with 8bit dynamic range. The data base was generated using a Microsoft SQL Server for several users.

The block - schema on picture 6.1 shows the configuration of the system. This schema presents the whole process of the surface inspection system. The line scan camera works like a CCD scanner and acquires an image of the metal

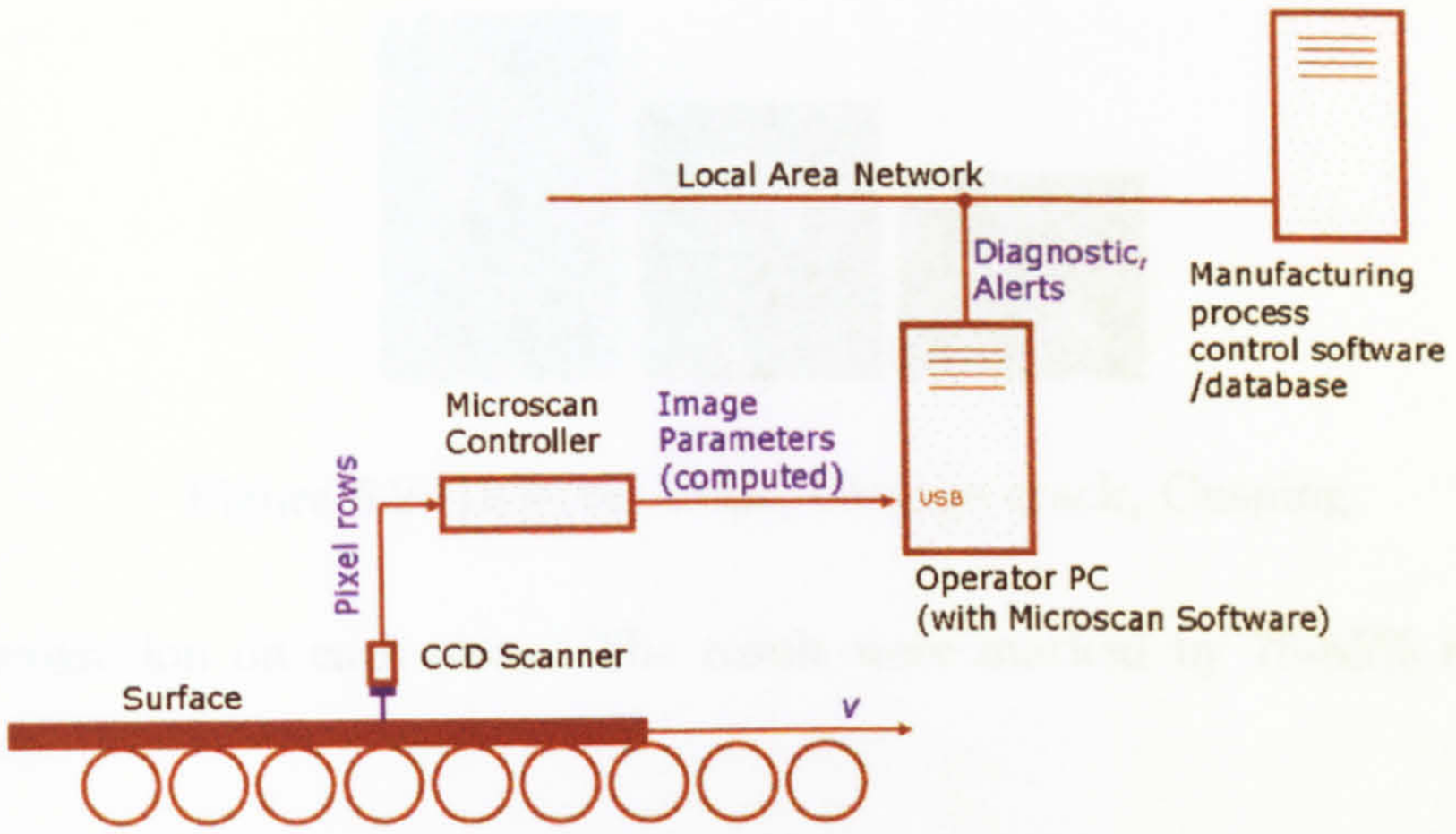



Figure 6.1: Configuration of surface inspection system

surface on a line-by-line basis. The light source in not shown because it depends on camera requirements. The hardware 'Microscan' controller is composed of a small frame and makes the decision in real time. The image and its parameters are transmitted to the main PC which makes macro decisions for large objects. Eventually, the picture of a defect goes to the data base where information on a current product is stored.

After recognition (section 5.3.4), the result can be stored using the same Database table. The user can check the location of a defect on a roll and inspect the location dimensions and recognition result. The system was designed for six classes of defect:

1) Not metallic pats

2) Shrunken leaf

3) Cusping

4) File mark

5) Clevage crack

6) Tear

Supervised training (section 5.3.5) of the system utilizes upto 17 samples of each class. Approximately 15-20 objects were used for estimating the results

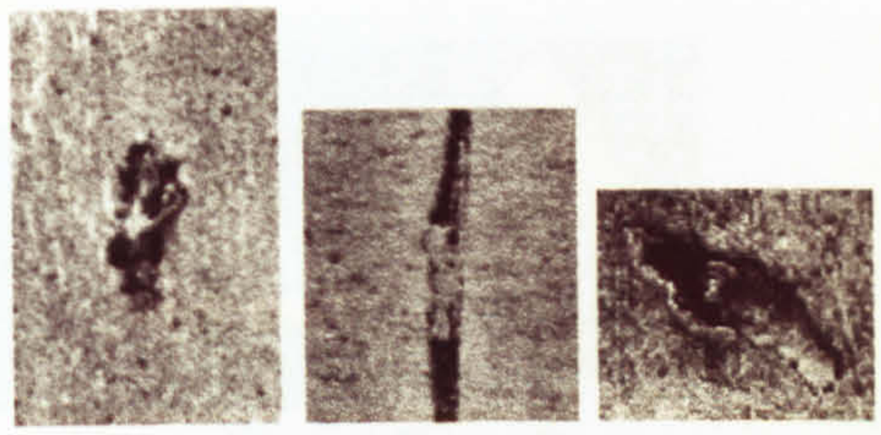


Figure 6.2: Defects: Scale, Clevage crack, Cusping

of a recognition on each class. The result were marked by 75-85% recognition accuracy.

### 6.1.2 Example

We consider a typical task associated with the recognition system. For visualization, we take two parameters and three simples defects, through which training of the system is based, an overview of the process using the knowledge database with defect properties being given. Let is consider the defect images of Scale, Clevage crack and Cusping which are shown in a Figure 6.2. It is experimental program has been developed on MatLab Figure 6.6 use the real pictures in off-line mode.

The Wiener filter is applied to the image after loading. The edge detection algorithm is then applied for the boundaries determination of defects. In this application in order to save commutation time I used simple Sobel filter which has been shown in chapter 3. Then Object location algorithm has been applied. At this stage is used simple distance function algorithm. The list of computation parameters surveyed in chapter 5.

For visual exposition on the output of the recognition engine, two parameters will be used: the fractal dimension and the convexity factor. The system is then trained, the supervised learning process being described previously 5.3.5. In the learning process, the system stores the membership function. The membership functions, for both parameters and the three defects considered are given in Figure 6.3 and 6.4.

The horizontal axis displays the universe values. On the vertical axis are displayed values relating to the accuracy of the recognition established by the expert (in this cases, a metallurgist) during the training session on the basis of

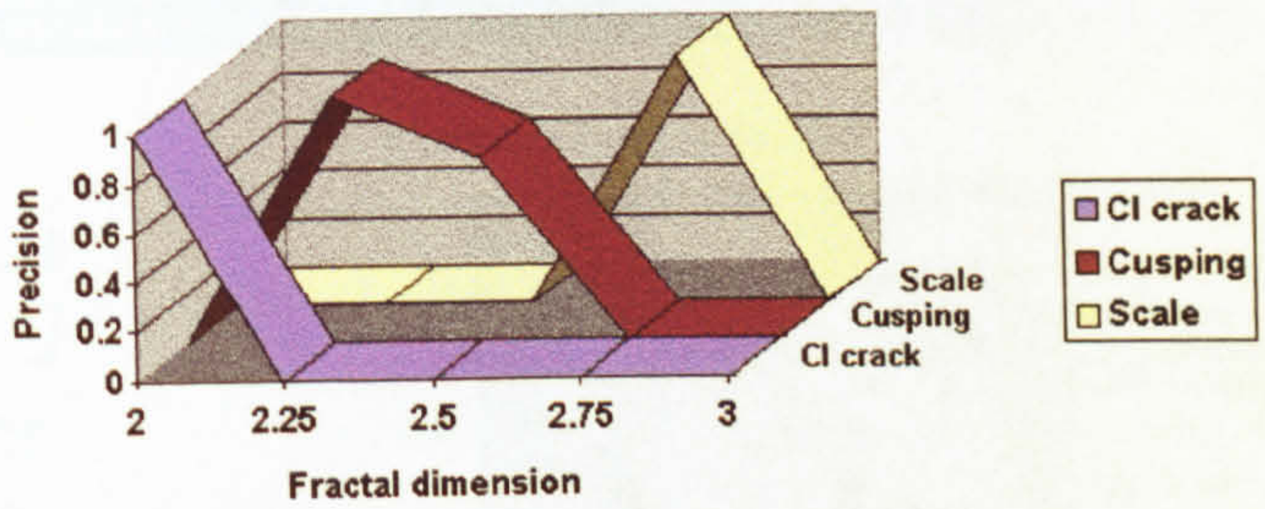



Figure 6.3: Fractal dimension

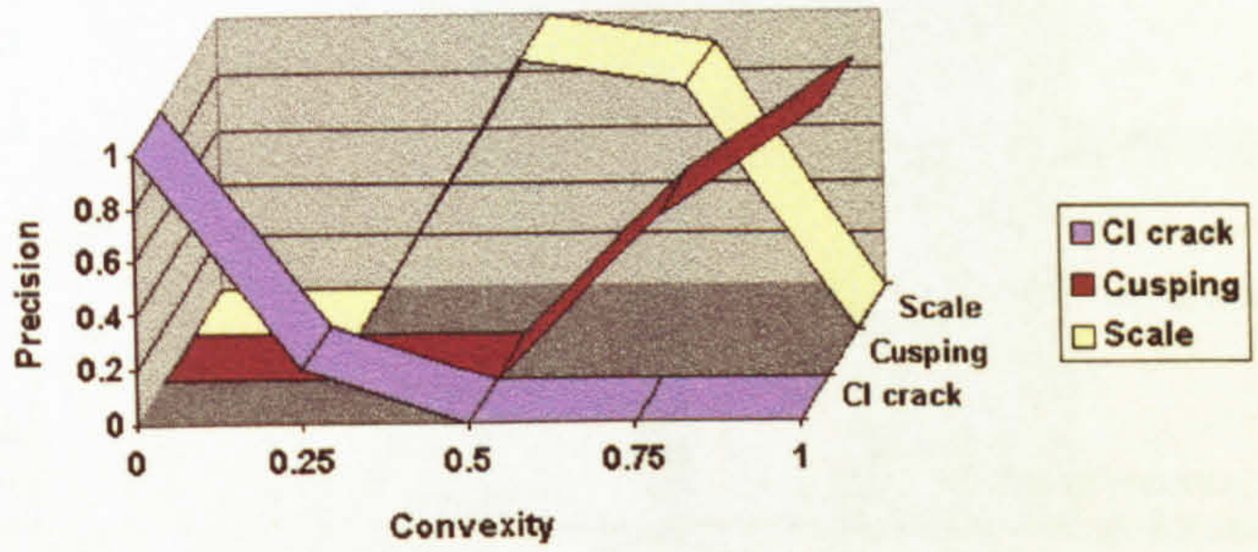



Figure 6.4: Coefficient of convexity

his/her knowledge on the nature of defect. The final stage concerns the decision making process discussed in Section 2.5.

We now consider a new sample, belonging to one of three groups: Scale, Clevage crack or Cusping and undertake the same operations as those made during the training session. In summary, it is necessary to activate the "Search" button. The system then finds the object, computes its fractal dimension which in this case is 2.58 and the convexity factor (0.69). The degree of confidence determined by all the parameters functions is displayed in Figure 6.5.

We compute the degree of confidence for each class:

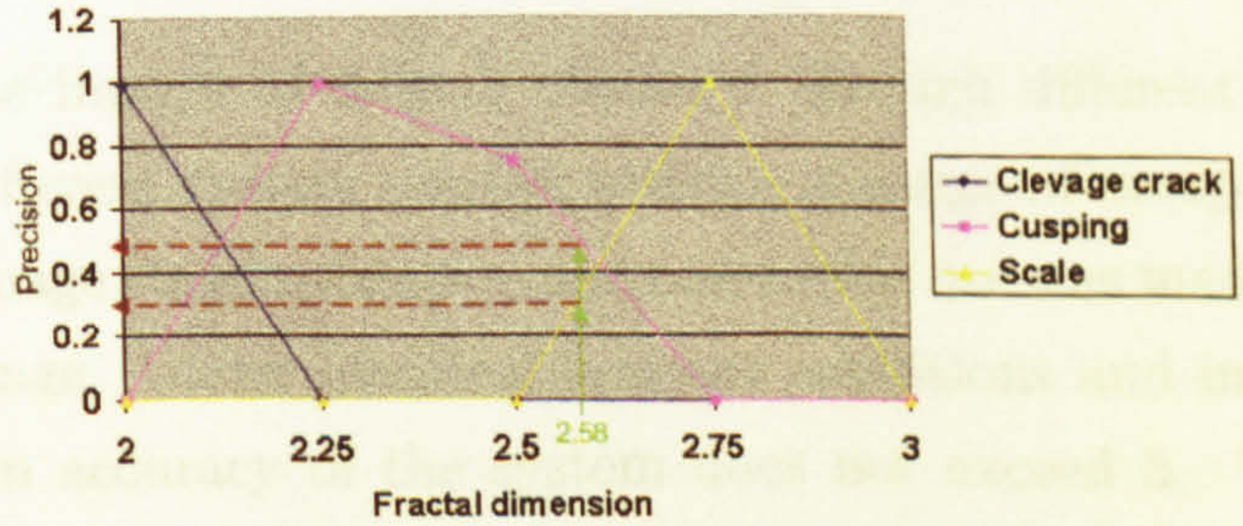



Figure 6.5: Precision definition

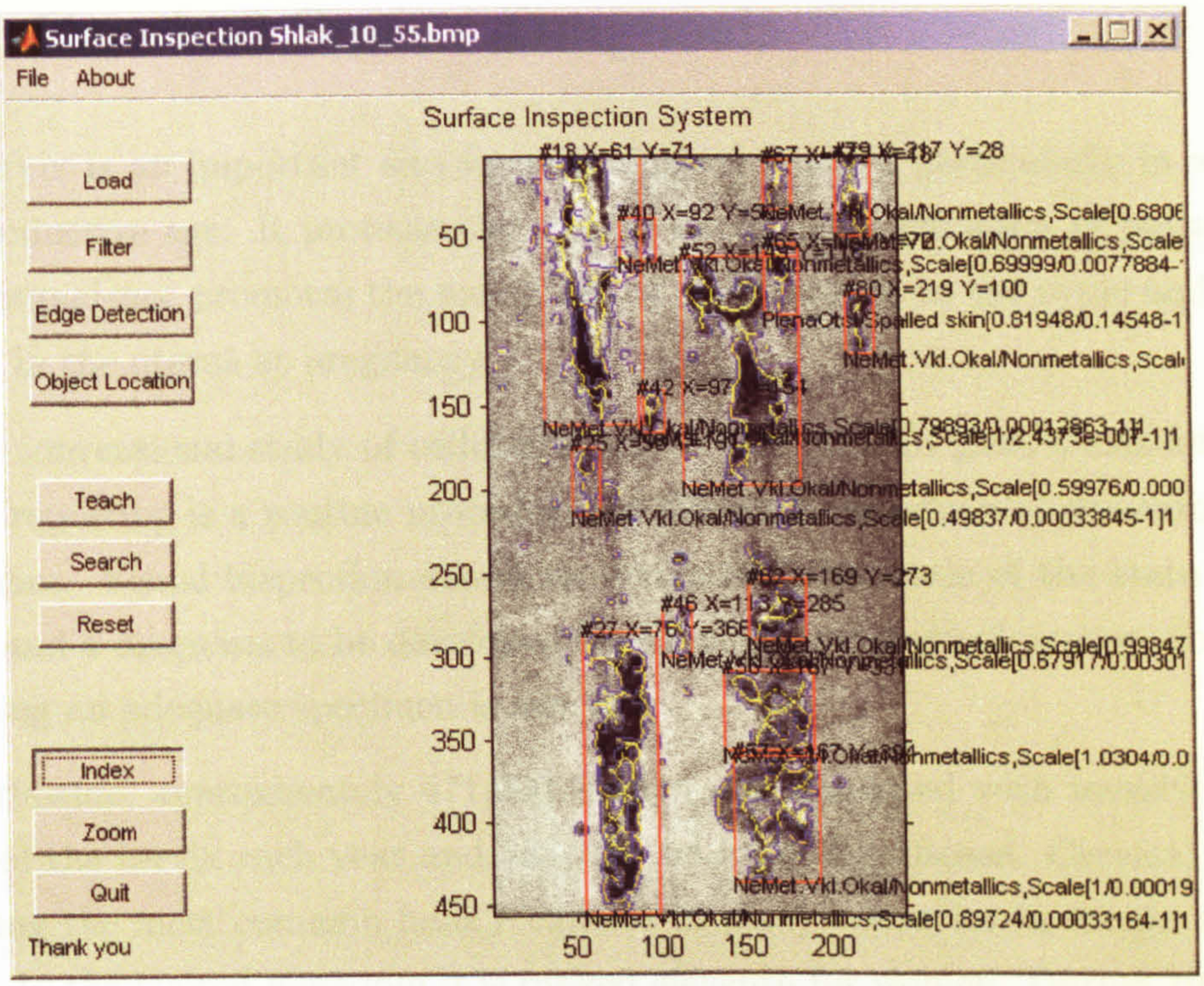



Figure 6.6: Result of object recognition system with the surface inspection

(Scale)=0.27+0.87=1.14

(Cusping)=0.46+0.42=0.88

(Clevage crack)=0+0=0

The maximum of these values characterizes that class, to which the given image corresponds. In the example given, the output is "Scale".

Industrial systems with many reference classes, if it is possible to apply why utillise scaling factors for each of the computed parameters in conformity with a measure of influence (weight coefficient) on a parameter for a class definition.

Observing the images of defects obtained through different frame grabbers and with the different system display utilities a range of acceptable results are obtained. For image standardization and correction use was made of Adobe Photoshop 5.5 package. Under identical systems conditions and image acquisition, the dispersion in accuracy of the system does not exceed 5. This is a unique system and at present, there are no automatic recognition system of this type on the market.

## 6.2 Cervical Smear Diagnosis

The cervix is an important site for pathological studies, particularly in women of reproductive age. It protects the uterine cavity from intrusion of pathogenic micro-organisms, promotes the movement of spermatozoa to the ovule and holds a fetus in the uterus at pregnancy.

The conventional study of cellular structures on stained glass slides for cytological reporting is a routine procedure for the early detection of pre-carcinoma conditions. Visual inspection allows an estimate to be made of the state of the cervix and a diagnosis to be developed based on the cytological pattern observed providing an adequate specimen is available.

Worldwide, approximately 471,000 women are diagnosed with invasive carcinoma of the cervix each year and 233,000 die from the disease. Cervical cancer is among the most common female cancers in many countries in the developing world. In the United Kingdom it is ranked eleventh for women. During 1998 just over 3,200 new cases of invasive cervical cancer were diagnosed in the UK, and in 2000 the disease caused 1,250 deaths. Mortality from cervical cancer continues to decrease, while incidence in the UK has decreased since the late 1980s. Sexually transmitted infection by certain strains of the human papillomavirus is the major cause of cervical cancer; smoking has also been linked to the disease (see Cancer Research Statistical Report attached).

Cervical cancer is preceded by a precancerous condition called CIN, cervical intraepithelial neoplasia, which can be easily treated if detected. Therefore it is important to identify CINs by screening women. The screening test is called a cervical smear. A nurse or doctor takes a small sample of cells from the surface of your cervix and spreads them onto a glass slide. The material is fixed in alcohol. When it reaches the lab, the slide is stained by papanicolou method and then put under a microscope. The cells are examined and any abnormal ones reported.

Four million cervical smears are taken annually in the UK and fifty million in USA. One of the diagnostic problems is that on about one fifth of the borderline preparations actually show severe disease on referral and biopsy. Overall there is a 50% 'failure' rate of detecting significant disease within borderline cases. In addition there is a 50% 'failure' of detecting significant decease within negative cases. The reasons vary from the skill in taking the smear, the actual prepara-

tion of the slide and then the sequential reading of the slide in the diagnostic laboratory.

At present, the actual reading of the slide is performed manually. It typically takes 8-10 minutes for a cytothologist to screen a slide and involves up to 300 movements of a microscope over the slide. This approach not only takes time but also inevitably cannot guarantee consistent and accurate results. Many borderline results are generated. This stage might be significantly improved by employing image analysis and object recognition techniques. Unlike a subjective assessment of the cytothologist, this approach will provide a reliable, consistent and quantitative estimation of CINs and other abnormalities.

**Liquid Based Cytology**

The NHS screening programme is to bring in a new way of preserving the cells taken in smear tests. It is called liquid based cytology (LBC). The nurse collects the cells from the cervix in the same way as PAP smear, but using a very small brush instead of a spatula. The head of the brush is broken off directly into a small pot of liquid, instead of putting the cells straight onto a slide. This is better at preserving the cervical cells and so the results of the smear test are more reliable. At the moment, about 1 in 12 PAP smears have to be done again because they can't be read properly. This causes a lot of anxiety for women. With LBC, far fewer smears will have to be repeated.

There are 3 pilot hospitals already using LBC in England, in Bristol, Newcastle and Norwich. It could take up to 5 years for LBC to be brought in throughout the UK because all the lab technicians and all those taking smears will have to be trained to use it.

The present recognition system is being designed to read LBC slides.

### 6.2.1 Classes of Cervical Cells

There are two main types of cervical cancer

- Squamous cell cancer
- Adenocarcinoma

Table 6.1: Classification of Squamous Cells

| **Medesine** | **Bathesda 2001** |
|---|---|
| Normal Sq | Normal squamous cells |
| Normal Sq | Atypical squamous cells – 'undetermined significance' (ASC-US) |
| Normal Sq | Atypical squamous cells – 'cannot exclude high grade disease' (ASC-H) |
| LSIL | Low grade squamous intra-epithelial lesion (LSIL) |
| HSIL | High grade squamous intra-epithelial lesion (HSIL) – CIN2 |
| HSIL | High grade squamous intra-epithelial lesion (HSIL) – CIN3 |
| Invasive Sq | Invasive squamous carcenoma |

They are named after the type of cell that becomes cancerous. Squamous cells are the flat skin-like cells that cover the surface of the cervix. Squamous cell cancer is the most common type of cervical cancer.

Adenocarcinoma cells are glandular cells that produce mucus. The cervix has these glandular cells along the inside of the passageway that runs from the cervix to the womb (the endocervical canal). Adenocarcinoma is a cancer of these cell types. It is less common than squamous cell cancer, but has become more commonly recognised in recent years. Only about one in five to one in ten cases of cervical cancer are adenocarcinoma. Adenocarcinoma is associated with a similar precancerous phase. It is treated in the same way as squamous cell cancer of the cervix. For more information see *CancerStats Cervical cancer UK* attached.

Tables 6.1 and 6.2 explains the relationship between Medesine and Bathesda 2001 classifications. The first class represents normal cells and the last one are malignant (cancerous) cells. Intermediate classes represent different degrees of abnormalities; it is important to detect these as well. Medesine classification is simplified because, unlike Bathesda 2001, it provides a *fuzzy* estimation of class membership, which gives a better description of the cell state.

An additional class Exudate is defined to described irrelevant structures on slides.

Table 6.2: Classification of Glandular Cells

| **Medesine** | **Bathesda 2001** |
|---|---|
| Normal Gl | Normal glandular cells |
| Normal Gl | Atypical glandular cells (AGC) – endocx/endom/not specified |
| Normal Gl | Atypical glandular cells (AGC) – favour neoplasia |
| AIS | Adenocarcinoma in situ (AIS) |
| Invasive Adeno | Adenocarcinoma |

With current techniques, all cervical smear tests are examined by 'screeners' who have only a few minutes per slide. This means that the screening is done at low magnification and high speed so it is not surprising that mistakes can be made. The 'screeners' look for abnormal variations in the ratio of the size of the nucleus relative to the size of the cell, as well as other markers of diseased tissue. When they identify suspect areas of the slide they mark these with a felt tip pen and pass them on for further inspection. These slides are then looked at by 'checkers' who have more experience and examine the slide more carefully and at higher magnification. If they are not satisfied that 'all is well', then they pass the suspect slides to a cytopathologist for further, more detailed analysis and diagnosis. Even at this final stage, mistakes can be made as each slide is prepared differently and it is common for cells to overlie each other, compounding the problem of accurate diagnosis further. New techniques that using cytocentrifuge preparations may overcome this last problem but have yet to be introduced in general.

One of the major criteria of assessing whether a cell is pre-malignant or malignant is the ratio of the size of the nucleus of the cell compared with that of the whole cytoplasm - the nuclear/cytoplasmic ratio. The rapid identification of variations in these ratios enables 'checkers' to quickly and more accurately determine if there are abnormalities by examining cells that are located in a small area. To estimate the condition of the cells, the cytologist typically makes upto 300 slide movements over a period of 8-10 minutes on a desk microscope and may consequently miss many important features. This approach not only takes time but inevitably can not guarantee consistent and accurate estimates of the condition of the cells. With an increasing number of screening projects taking place together with the variability of different preparations, diagnostic errors can lead

to a number of fatalities due to false negatives and lack of appropriate treatment in the early stages of cervical cancer.

At present, there are no commercial or experimental systems available for the automatic identification and classification of tissue cells without human participation. Obtaining results from cytology diagnostics in real time with a robust least error criterion is a widespread and important problem for screening the cervix uteri. The automatic coloring (staining) and scanning of the material creates preconditions in designing an algorithm and technical devices for the automatic identification and classification in cytopathology. A key point is to identify and classify the condition of the cell nuclei using a suitable recognition process.

There are about 3000 new cases of cervical cancer diagnosed each year in England and Wales, leading to about 1,200 deaths. The number of deaths from cervical cancer has been decreasing over the last 20 years, due in part to the introduction of the NHS Cervical Screening Programme in 1987.

About 8 out of 10 (84%) of eligible women have been screened at some time within the last five years and about 7 out of 10 (67%) at some time within the last three years. About four million women are screened each year in England and Wales.

The current screening programme is based on the 'Pap' smear test. During a smear, a sample of cells is collected from the woman's cervix (neck of the womb) using a simple disposable spatula device. The smear usually takes place at the GP surgery or a community clinic. The sample collected is spread on to a glass slide and sprayed with, or dipped in, a liquid to fix the sample onto the slide. The slide is then sent to a hospital laboratory for examination. The hospital laboratory examines the sample with a microscope to look for cells that show signs of pre-cancerous changes. If such cell-types are found, the patient may be called back for further tests. Cervical smear tests are not a test for cancer but a way of looking for signs of the disease, which may turn into cancer if left untreated.

### 6.2.2 Comparison with other approaches

Two available automatic techniques involve the examination of slides using integrated optical densitometry. Cytotec uses thin preparation slides where the cells

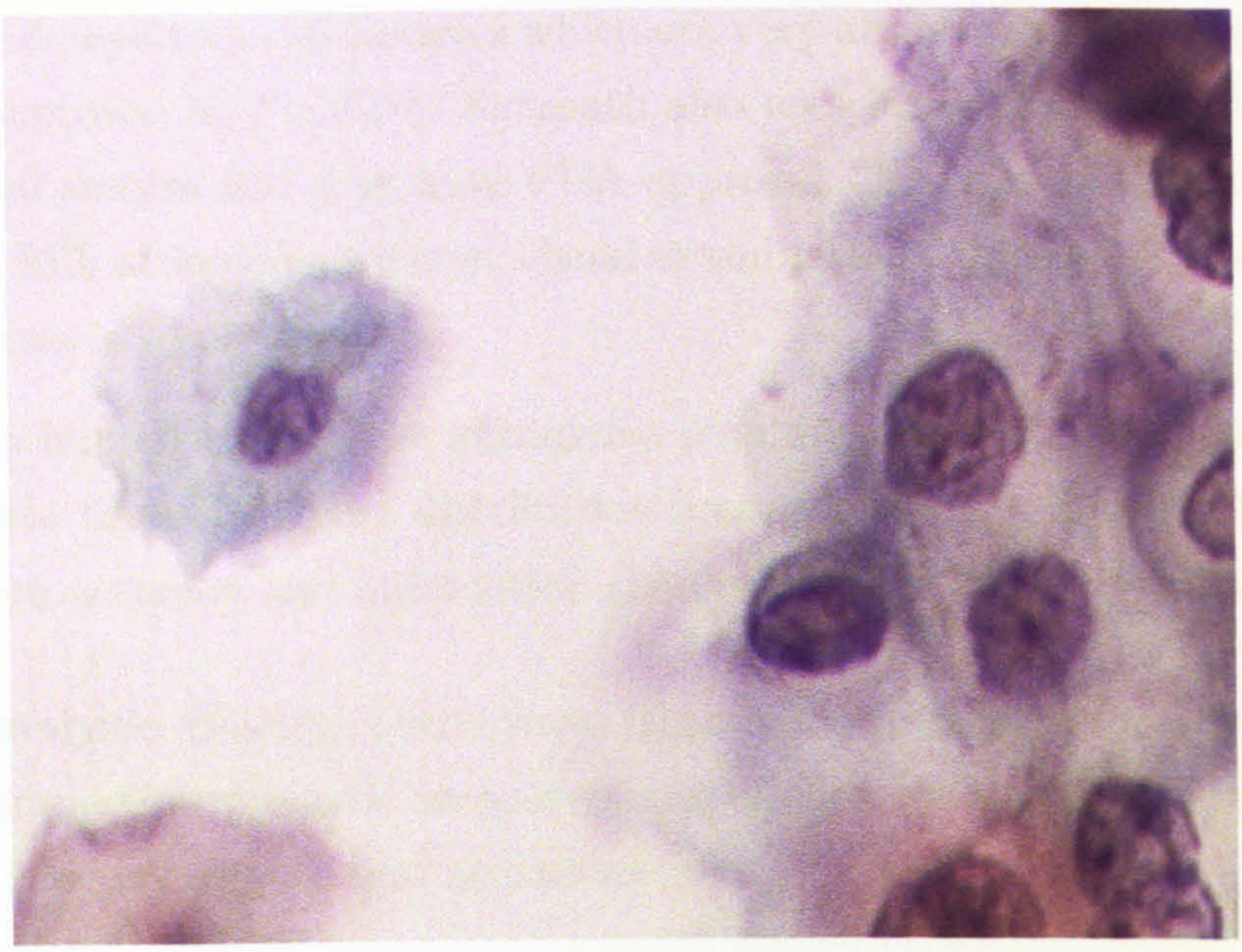


Figure 6.7: Cervical cells prepared by Papanicolaou method

Table 6.3: Imaging acquisition hardware

| Model and Supplier | Advantages | Shortcomings |
|---|---|---|
| Nikon Coolscope (Nikon Instruments Europe BV) | Available on the market<br>Magnification 40x<br>Complete solution with a slide feeder | Very slow (several hours/slide)<br>Small focus depth / automatic focus does not find the optimal $z$-position<br>Dynamic range to be adjusted<br>Tiling scan |
| Aperio Scanscope (Aperio Technologies / DakoCytomation) | High scanning speed (20 min/slide)<br>Magnification 40x<br>Non-tiling scan<br>Better focus<br>Better dynamic range | Not fully developed<br>May be a problem to achieve 60x |
| Nikon Eclipse E8000 + JVC 3-CCD KY-F55B (St. Georges Hospital) | Variable resolution 4X-80X<br>Manual focus<br>Manual brightness | Manual image capturing<br>Can be used only for testing |

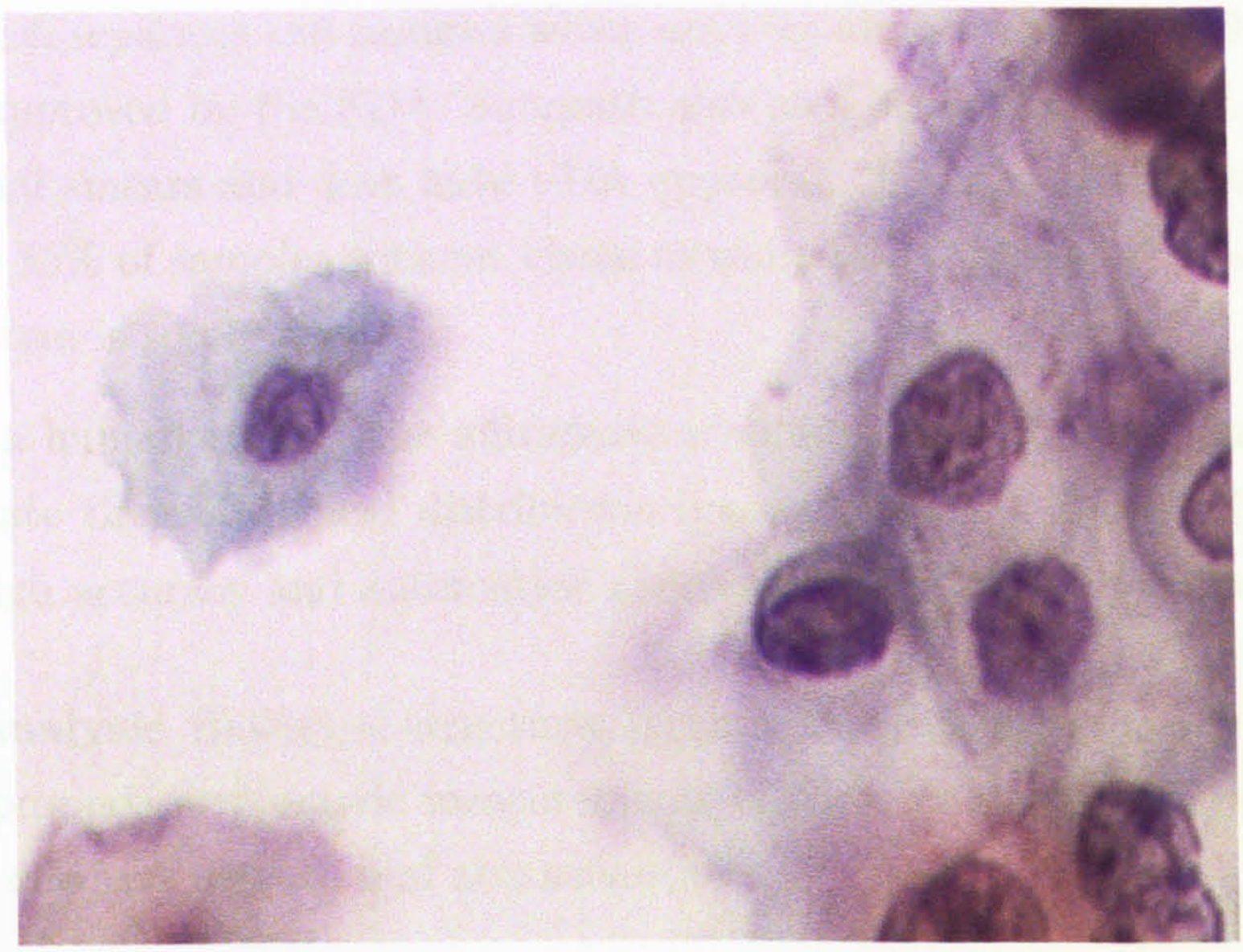


Figure 6.7: Cervical cells prepared by Papanicolaou method

Table 6.3: Imaging acquisition hardware

| Model and Supplier | Advantages | Shortcomings |
|---|---|---|
| Nikon Coolscope (Nikon Instruments Europe BV) | Available on the market<br>Magnification 40x<br>Complete solution with a slide feeder | Very slow (several hours/slide)<br>Small focus depth / automatic focus does not find the optimal $z$-position<br>Dynamic range to be adjusted<br>Tiling scan |
| Aperio Scanscope (Aperio Technologies / DakoCytomation) | High scanning speed (20 min/slide)<br>Magnification 40x<br>Non-tiling scan<br>Better focus<br>Better dynamic range | Not fully developed<br>May be a problem to achieve 60x |
| Nikon Eclipse E8000 + JVC 3-CCD KY-F55B (St. Georges Hospital) | Variable resolution 4X-80X<br>Manual focus<br>Manual brightness | Manual image capturing<br>Can be used only for testing |

are centrifuged onto a bonded filter in a single final layer. An automatic scanner is used which separates out samples which are very abnormal. This technique has not been approved by the FDA. Surepath also uses integrated optical density of conventional smears and does have FDA approval. The aim of this technique is to exclude 25% of samples without visual examination. Data on the failure rates of this system is sorely needed.

Unlike a human expert, the automatic scanning method can count the cells and estimate their statistical distribution among classes or states. The system delivers high accuracy and automation thanks to the following innovations:

**Fractal analysis** Biological structures (such as body tissues) have natural fractal properties. Numeric measurements of these properties enables efficient and effective detection of abnormalities.

**Extended set of detectable features** High accuracy is achieved when multiple features are measured together and combined into a result

**Advanced fuzzy logic engine** The knowledge-based recognition scheme enables highly accurate diagnosis.

### 6.2.3 Program Description

It is proposed that such a system could assist cytopathologists in reducing their workload by eliminating in a secure manner a percentage of normal smears thus allowing more time for the evaluation of the abnormal cases.

Software solution detects abnormalities in organic structures such as cells by digital image analysis. Cancer experts create the knowledge database by training the system (section 5.3.5) with a number of pictures. The recognition algorithm (section 5.3.4) comprises the following steps:

**Filtering** The image is filtered to reduce noise and remove unnecessary features (bacteria, broken cells).

**Segmentation** The image is segmented to perform a separate analysis of each object (cell or mole). In order to separate connected objects the new algorithm 4.1.2 has been applied. The objects which have been found are represented in Figure 6.8 in the rectangular boxes.

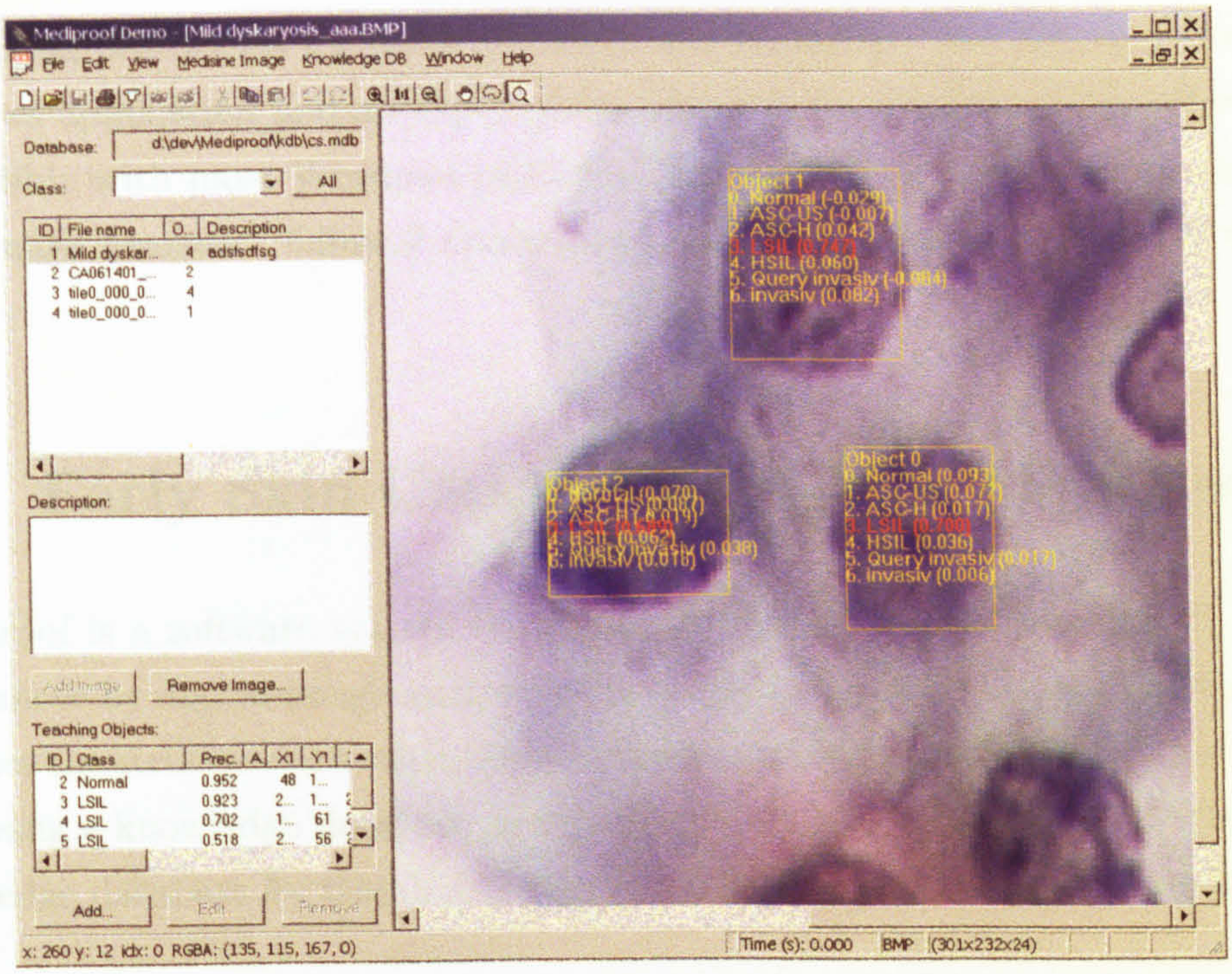



Figure 6.8: Recognition of cells

**Feature Detection** For each object, a set of recognition features are detected. The features are numeric parameters that describe the object. The list of computed features is given in Chapter 5. The system captures a variety of geometrical, fractal 4.2 and statistical features in 1 and 2 dimensions. 1D features correspond to the border of object, whereas 2D features relate to the surface within and around object. In order to get 2D feature the new algorithms from section 4.3 has been applied. The algorithm for feature selection has been considered in chapter 5.

**Decision Making** The system uses fuzzy logic to combine features into a decision (section 5.3.5). A decision is the estimated class of object and accuracy probability. The basic version has two classes of moles: normal and abnormal. In-between states are determined by the probability. For example, 35% normal is equivalent to 65% abnormal and suggests careful analysis by cancer specialists. In the extended training version for cervical cancer, there will be 7 or more classes (CINs) depending on the classification system. Generally system designed for up to 10 classes and number of classes depends only on available samples for learning procedure.

Working prototypes for cervical cancer are available. The system includes a Windows'application and an experimental knowledge database. The software is compatible with most Windows platforms and can import/export data in common image formats. Clinical trials have started with several hospitals in the UK.

## 6.3 Early Skin Cancer Diagnosis - Mediproof

Mediproof is a software system developed for this thesis and designed to detect skin cancer by digital image analysis of suspected cancerous moles. The software analyses the structure of a mole, detects cancer-identifying features, makes a decision using a knowledge database and outputs a result. Cancer experts create the knowledge database by training Mediproof with a number of case study images.

### 6.3.1 Recognition Technology

Mediproof makes use of this thesis' achievements in pattern recognition based on the applications of fractal geometry and fuzzy logic and allows full automatic recognition of tissue (cell) abnormalities. The recognition procedure comprises the following steps:

1. Filtering
   The image is filtered to reduce noise and remove unnecessary features such as hair and light flecks.

2. Segmentation
   The image is segmented to perform a separate analysis of each object such a mole or cell (Figure 6.9). Two segmentation modes are then available:

   - Automatic Mode
     The software identifies a mole as the largest and darkest object in the image. This mode is applicable in most cases.
   - Manual Mode
     The area of interested is manually selected by the user. This is most useful in cases of multiple moles and/or foreign objects on the image.

3. Feature Detection

   For each object, a set of recognition features are detected. This part has the same stages like in the second application and explained in chapter 5. Additionally, Mediproof can analyse the mole *structure* i.e. its growth stages.

4. Decision Making

   The system uses fuzzy logic to combine features into a decision. A decision is the estimated class of the object and accuracy probability. Mediproof has two classes of moles: Abnormal (Class 1) and Normal (Class 2). The system outputs the highest probability class (Figure 6.10). Low probability suggests that the case is not covered in the knowledge base and further analysis is required to be carried by cancer specialists. In cervical cancer, there are 7 or more classes (CINs) depending on the classification system.

### 6.3.2 Knowledge Database

Mediproof is a knowledge-based system and requires extensive training before clinical operation. The training process includes a review and probabilistic classification of appropriate images by cancer experts (Figure 6.11). The minimal number of training images depends on the number of classes and the diversity of objects within each class. For skin cancer, we recommend that at least 50 pictures of each class (Normal and Abnormal) are used.

The default knowledge database is loaded from

`C:\Program␣Files\Mediproof␣Demo\bin\def.kdb`.

### 6.3.3 Installing

1. Insert the Mediproof Demo CD into your CD-ROM drive.
2. Installation software should start automatically. If it does not happen, run `setup.exe` from the root folder of the CD.
3. Follow instructions on screen.

**Launching Mediproof** Locate **Mediproof Demo** group in your **Program** menu and click **Mediproof Demo** icon.

### 6.3.4 Recognition

1. Click **Load Image** and select an image of a mole. Samples can be found in folder `Pictures`, by default in

   `C:\Program␣Files\Mediproof␣Demo\Pictures\`.

2. Press **Recognise All**. If the mole is not found automatically click **Recognise Selection** and select the area of interest.

3. The recognition result appear in 10–30 seconds (Figure 6.10).

### 6.3.5 Teaching

1. The default knowledge database is loaded from

   `C:\Program␣Files\Mediproof␣Demo\bin\def.kdb`.

   To create a new database, select **New knowledge DB** from in the **File** menu (Figure 6.12).

2. Click **Load Image** and select a picture of a mole.

3. Click **Teach All**. If the mole is not found automatically click **Teach Selection** and select the area of interest.

4. Mediproof analyses the mole for 10–30 seconds whereupon the Teaching Dialog (Figure 6.11) pops up. Enter your estimation:

   (a) Class: number 1 (for Abnormal) or 2 (for Normal),

   (b) Probability: a number between 0.0 and 1.0. 1.0 means you are absolutely sure, whereas zero should not be normally used. Typical values are 0.90–0.95.

5. Repeat Steps 1-4 to process all training images.

6. Select **Save knowledge DB...** from **File** (Figure 6.12) and enter a file name of the knowledge database.

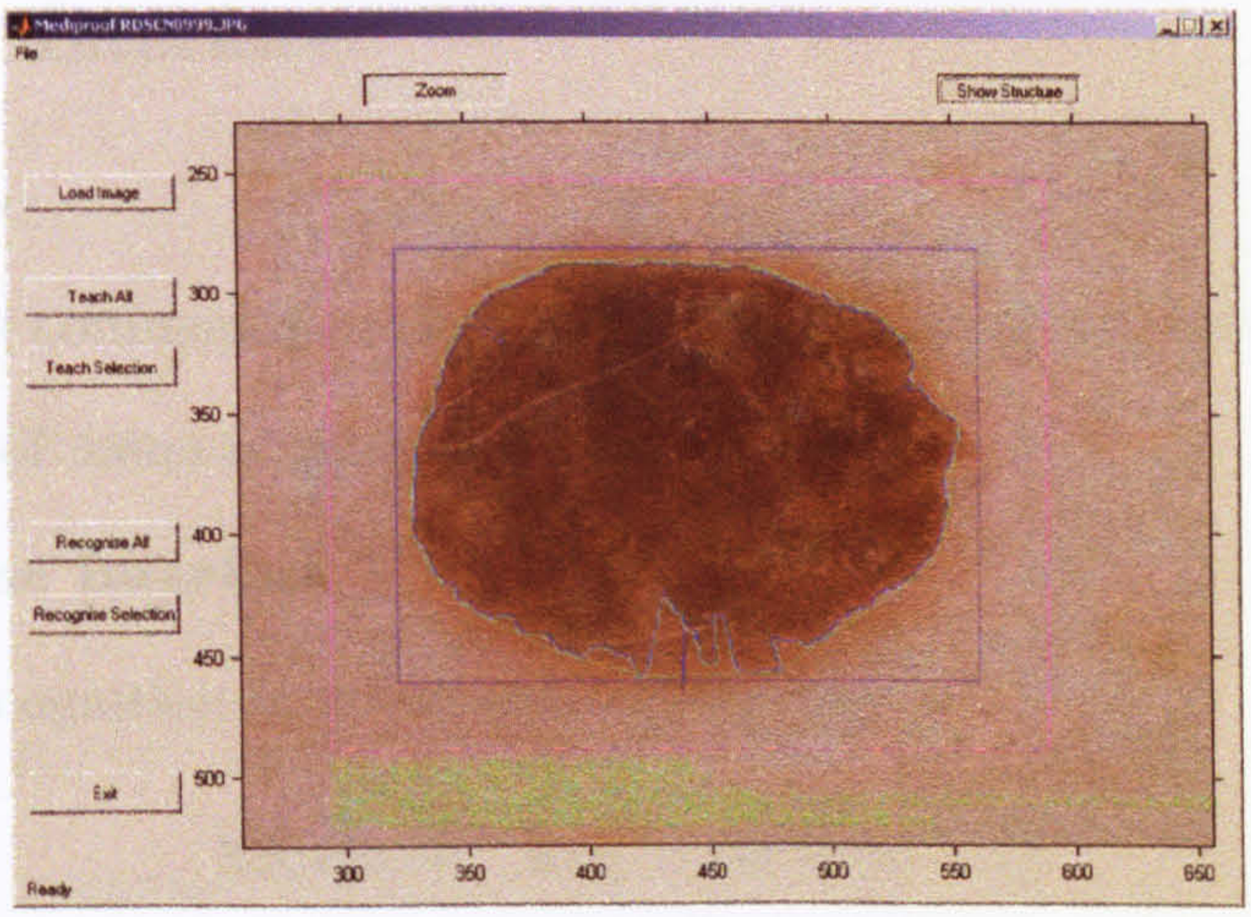



Figure 6.9: Mediproof main frame

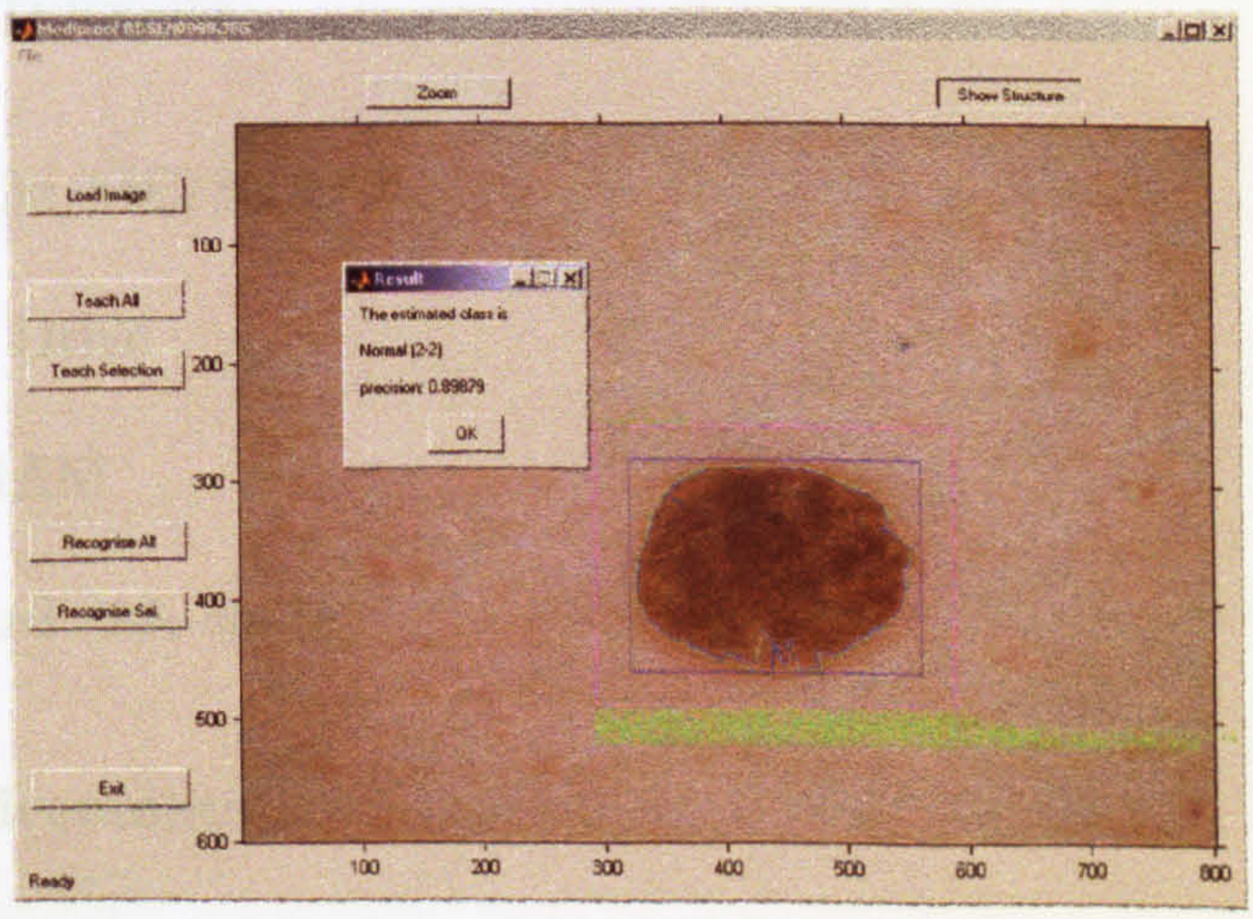



Figure 6.10: Recognition result

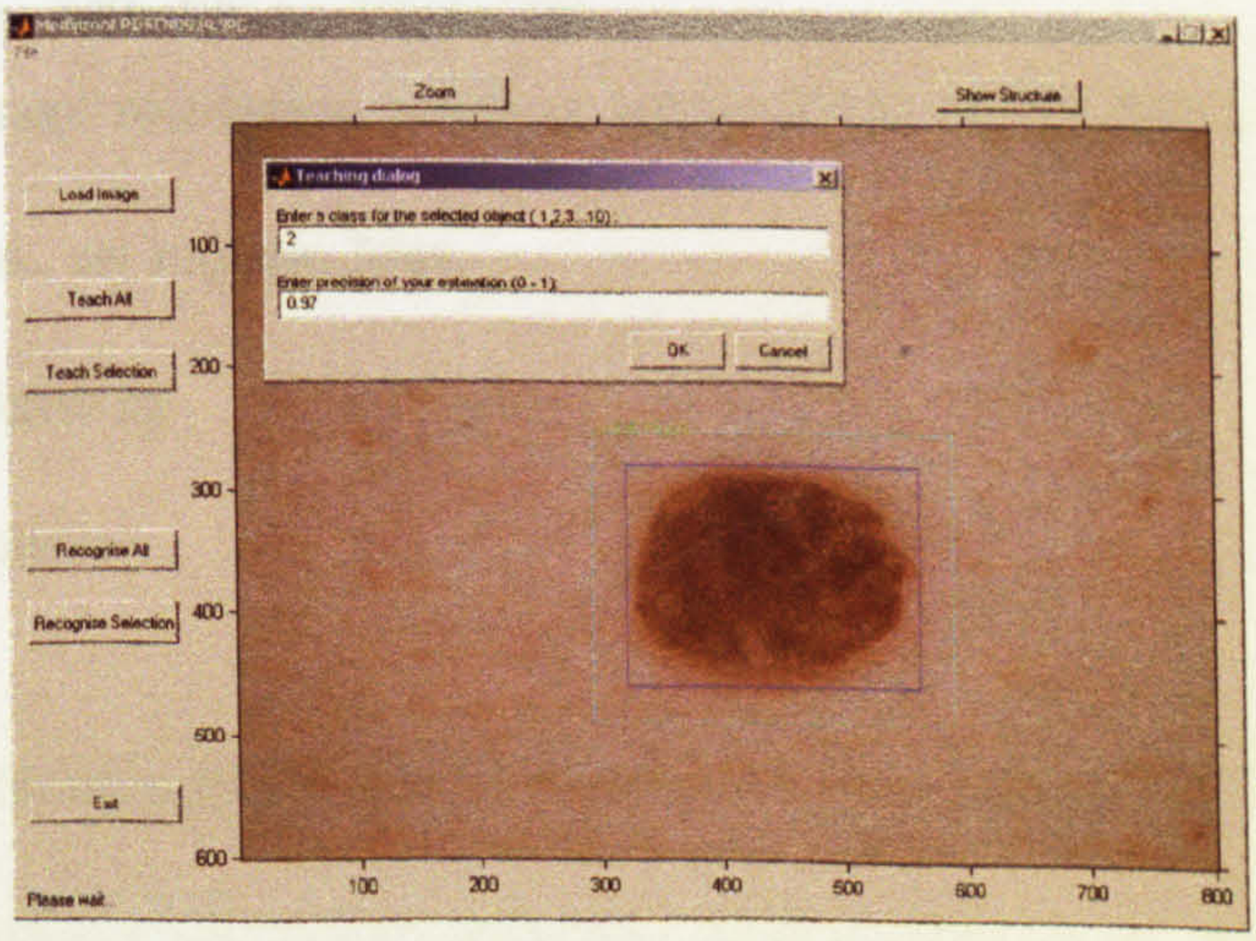



Figure 6.11: Teaching dialog

### 6.3.6 User Interface

#### Main Window

The following commands are available from the main window (Figure 6.9):

Menu Mediproof menu is given in Figure 6.12:

#### Command Line Interface

To launch recognition in automatic mode type

```
Mediproof.exe␣"LoadGraf"␣%1
```

where %1 is an image name (JPEG, BMP or TIFF formats are supported).

#### System Requirements

- Windows 98/ME/2000/XP
- CD-ROM Drive
- 256 MB RAM
- 30 MB hard disk space

#### Picture Requirements

- Input format: JPEG, BMP or TIF
- Image size: 640x480 to 1024x728
  (higher image resolution requires larger RAM of 512 Mb and more)
- Good focus, no motion blur
- Uniform lighting
- Centered mole

### 6.3.7 Estimation of the Minimal Number of Samples

**Initial Data** There are approximately 65,000 new cases [101] of skin cancer each year in the UK, which is about 5% [101] of total number of patient

examined annually. Let $p$ be the probability of unrecognized cases. Then $q = 1 - p$ is the probability of recognized cases and the number of mistakes is $S^k n = P\{k \text{ misidentification out of } n\} = C_n^k p^k q^{n-k}$ where $n$ is number of experiments and $C_n^k = \frac{n!}{k!(n-k)!}$

**Statement of Problem** Estimate the number of images (tests) $n$ required to estimate the error probability within the range $\pm 1\%$ (0.01) and degree of confidence $\alpha = 0.99$.

**Theoretical Background**

Assuming the error is normally distributed, the probability of estimating the sampling fraction $w$ within the degree of confidence $\Delta$ is defined by

$$P(|w - p| \leq \Delta) = \Phi(t), \tag{6.1}$$

where

$$\Phi(t) = \frac{2}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx \tag{6.2}$$

is the Laplace integral, $p$ is the probability of the error and $t = \frac{\Delta}{\sigma_w}$.

**Numerical Results** Let us find $p$ with 0.01 and let $p'$ be the real probability. Set the confidence interval at $\alpha = 0.99$. Then our minimal error will be

$$P(|p' - p| < 0.01) \tag{6.3}$$

multiplicater with $n$

$$P(|S_k^n - pn|0.01n) < \alpha \tag{6.4}$$

by the law of large numbers

$$0.01 \frac{\sqrt{n}}{\sqrt{pq}} \geq Z_\alpha \tag{6.5}$$

for $\alpha = 0.99$ and $Z = 2.58$

$$\sqrt{n} \geq 100 \sqrt{pq} Z_\alpha \tag{6.6}$$

$$n \geq 10^4 pq Z_\alpha^2 \tag{6.7}$$

For all $p : pq \leq \frac{1}{4} \Rightarrow n \geq 10^4 \frac{1}{4} 6.6564 = 16641$.

If $p < 0.1$ then $q > 0.9$ and $pq < 0.1$. We can then evaluate the minimal

requirement $n$:

$$n \geq 10^4 \frac{1}{10} 6.6564 = 6656$$

Practically, this number is not enough to assess the accuracy of the recognition system due to the following reasons:

- The assumption that $p$ and $w$ are constant for all types of moles is very doubtful. It is necessary to carry tests with varieties of moles, types of skin etc.
- The recognition quality will significantly vary in time during the test process since the knowledge database is constantly updated.

The empirical assessment is $n = 6,656$.

### 6.3.8 Comparison with other approaches

Today's world wide market offers a number of aids and tools for skin estimation. Some of them are used as a database to estimate the velocity and level of pathology and requires extra time factors to make a decision. Other products calculate several properties and represent them on a screen. Medical staff are then used to make a final decision.

More interesting techniques involve the capture mole images using different sensors or a multiplicity of different dimension. However, these systems are as yet, not approved for clinical diagnosis and are not a referenced form of dermatoscopy.

The following list provides some of the more common products in the field:

(i) MoleMAX - http://www.molechecks.com.au

(ii) DermLite - http://www.dermlite.com/mmfoto.html

(ii) DermoGenius Lite - http://www.dermogenius.de

(iiii) MelaFind - www.melafind.com

None of these products have as yet been approved by the FDA. Comparing these products with the methods developed for this thesis, it is clear that there are

no other automatic recognition systems with self-adjusting procedures and self-controlled functions. The tests undertaken to date, have established the capacity for Mediproof to be used in routine clinical conditions provided extensive training of the system has been undertaken.

| Button | Action |
|---|---|
| **Load Image** | loads image in JPEG, BMP or TIF formats |
| **Teach All** | performs teaching in automatic mode |
| **Teach Selection** | performs teaching in manual mode |
| **Recognise All** | performs recognition in automatic mode |
| **Recognise Selection** | performs recognition in manual mode |
| **Exit** | closes Mediproof |
| **Zoom** | switches on/off the zoom mode (use the left mouse button to zoom in and the right button to zoom out) |
| **Show Structure** | displays the mole boundary and features |

| Menu Item | Action |
|---|---|
| **New knowledge DB** | resets knowledge database |
| **Open knowledge DB...** | loads knowledge database from a kdb file) |
| **Save knowledge DB...** | saves knowledge database to kdb file |
| **Open Image...** | loads image in JPEG, BMP or TIF formats |
| **Exit** | closes Mediproof |


Figure 6.12: File menu

# Chapter 7

# Conclusion and Future Research Work

This thesis has been concerned with the task of developing a methodology and implementing applications that are concerned with two key tasks: (i) the analysis of an image in terms of its fractal structure and the fractal properties that characterize that structure; (ii) the use of a fuzzy logic engine to classify an object based on its fractal properties. The combination of these two aspects has been used to define a processing engine that is unique in its modus operandi but entirely generic in terms of the applications to which it can be applied.

In order to develop this thesis, it has been necessary to research a relatively broad range of material. In Chapter 2, the fundamentals of physical optics were presented leading to the basic model for the generation of an incoherent image starting with Kirchhoff diffraction theory through which we can express image formation in terms of a linear systems theory. This provides us with the fundamental image model which expresses the relationship between the object plane and the image plane via the point spread function of the imaging system being used and its noise characteristics. Chapter 3 considered a number of basic image processing methods that are useful as bespoke operations and of importance to some of the pre-processing operations used in the systems developed (e.g. the Wiener filter) to restore the recorded image and optimize its characteristics with regard to noise. This chapter also introduced the principles of Fractal Geometry and in particular, the properties of fractal images. Fractal geometry is a central kernel to the problems addressed in this thesis. The essential connection

is that fractal images are textural and hence, by analyzing appropriate images in terms of their fractal properties (e.g. the fractal dimension), it is possible to derive a data set that quantifies the textural properties of an image. This is the fundamental basis for the pattern recognition methods discussed in this work. In order to use fractal image analysis for pattern recognition, it is necessary to first isolate the objects of interest in the image plane to which this form of analysis can be applied. Method of object segmentation are discussed in Chapter 4 which introduces a number of new and original filters for the isolation of objects and in particular, their boundaries. Chapter 5 discusses the principles of object recognition and classification developing the basis for the fuzzy logical classification methods used in this thesis to develop a supervised learning system using a multiple parametric space or feature vector composed of fractal, Euclidean and statistical parameters computed from the object itself and its boundary characteristics. The system developed has been tailored for application to three major problems: (i) cervical smear analysis; (ii) skin cancer screening and (iii) process control engineering.

The research has investigated numerous processes for pattern recognition using fractal geometry as a central processing kernel. This has led to the design of the new library of fundamental pattern recognition algorithms. The image types considered contain about 80 percent useful environmental information for the human. With rapid advances in video technology the content of a video stream is increasing at a rate that is far beyond the human brain capacity for decision making. This necessitates a need for developing an automatic image processing and decision making system using artificial intelligence. Such systems are required in search engines, information databases, navigation in unknown terrain, interpretation of two dimensional data, etc.

The first system to be implemented was developed for surface inspection of metallurgical production. The system was developed further for medical applications and in particular, for the diagnosis of cancer. Preliminary tests have been made for the detection of skin cancer and abnormal states in cervical smears. A significant amount of programming and software engineering has been required to implement the methodology and the systems to which it has been applied. This has involved the design of conventional and novel algorithms for object recognition, classification and systems implementation. In all cases, the approach has been designed, developed and successfully tested using a MATLAB programming

environment.

The creation of logic and general purpose hardware for artificial intelligence is a basic theme for any future research and development based on the results of this thesis for the applications developed and beyond. The results of the current research project can be utilized in a number of different areas although medical imaging would appear to be one of the most natural fields of interest because of the nature of the images available, their complex structures and the difficulty of obtaining accurate diagnostic results which are efficient and time effective. A further extension of this research is to consider the effect of replacing the fuzzy logic engine used to date with an appropriate Artificial Neural Network. It is not clear as to whether the application of an ANN could provide a more effective system and whether it could provide greater flexibility with regard to the type of images used and the classifications that may be required.

# Appendix A

# Evaluation of the $n$-dimensional Fourier transform of $r^\beta$ where $\beta$ is a floating point number

**Theorem:** If $\beta \neq 2m$ or $-n-2m$ where $m = 0, 1, 2, ...,$ then

$$\int_{-\infty}^{\infty} r^\beta \exp(-i\mathbf{k}\cdot\mathbf{r}) d^n\mathbf{r} = \frac{(\frac{1}{2}\beta + \frac{1}{2}n - 1)!}{(-\frac{1}{2}\beta - 1)!} 2^{\beta+n}\pi^{n/2}k^{-\beta-n}$$

where $\mathbf{k}$ and $\mathbf{r}$ are the $n$-dimensional vectors $(k_1, k_2, ..., k_n)$ and $(r_1, r_2, ..., r_n)$ respectively, $r \equiv |\mathbf{r}| = \sqrt{r_1^2 + r_2^2 + ... + r_n^2}$ and $k \equiv |\mathbf{k}| = \sqrt{k_1^2 + k_2^2 + ... + k_n^2}$.

Note: $\int_{-\infty}^{\infty} f(\mathbf{r}) \exp(-i\mathbf{k}\cdot\mathbf{r}) d^n\mathbf{r}$ is taken to mean

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}...\int_{-\infty}^{\infty} f(r_1, r_2, ..., r_n) \exp[-i(k_1r_1 + k_2r_2 + ... + k_nr_n)] dr_1 dr_2 ..., dr_n$$

**Proof:** The proof of this result follows the approach of Jones [93] and is based two results:

(i) If $f$ is a function of $r$ only, then

$$F(\mathbf{k}) = \left(1 - \frac{\partial^2}{\partial k_1^2} - \frac{\partial^2}{\partial k_2^2} - ... - \frac{\partial^2}{\partial k_n^2}\right)^N (2\pi)^{n/2} \int_0^{\infty} \frac{f(r) r^{n-1}}{(1+r^2)^N} \frac{J_{\frac{n-2}{2}}(kr)}{(kr)^{(n/2)-1}} dr$$

where $N$ is a positive integer and $J_{(n-2)/2}$ is the Bessel function (of order $(n-2)/2$).

(ii) A known result for Bessel Functions [92]

$$\frac{(2\pi)^{n/2}}{k^{(n/2)-1}}\int_0^\infty \frac{r^{\beta+(n/2)}}{(1+r^2)^N} J_{\frac{n-2}{2}}(kr)dr$$

$$= \frac{\pi^{n/2}(\frac{1}{2}\beta+\frac{1}{2}n-1)!(N-\frac{1}{2}\beta-\frac{1}{2}n-1)!}{(N-1)!(\frac{1}{2}n-1)!} {}_1F_2(\tfrac{1}{2}\beta+\tfrac{1}{2}n;\tfrac{1}{2}\beta+\tfrac{1}{2}n-N+1,\tfrac{1}{2}n;\tfrac{1}{4}k^2)$$

$$+\frac{\pi^{n/2}k^{2N-\beta-n}(\frac{1}{2}\beta+\frac{1}{2}n-N-1)!}{(N-\frac{1}{2}\beta-1)!2^{2N-\beta-n}} {}_1F_2(N;N-\tfrac{1}{2}\beta,N+1-\tfrac{1}{2}\beta-\tfrac{1}{2}n;\tfrac{1}{4}k^2) \quad (A1)$$

where

$${}_1F_2(a;b,c;x) = 1+\frac{a}{1!bc}x+\frac{a(a+1)}{2!b(b+1)c(c+1)}x^2+\ldots$$

The first of these results can be obtained by choosing a polar axis to lie along the direction of $\mathbf{k}$ so that $\mathbf{k}\cdot\mathbf{r} = kr\cos\theta_1$ and

$$F(\mathbf{k}) = \int_{-\infty}^{\infty} f(r)\exp(-i\mathbf{k}\cdot\mathbf{r})d\mathbf{r} = \int_0^\infty f(r)r^{n-1}\int_0^\pi \exp(-ikr\cos\theta_1)\sin^{n-2}\theta_1 d\theta_1$$

$$\times\int_0^\pi\ldots\int_0^{2\pi}\sin^{n-3}\theta_2\ldots\sin\theta_{n-2}d\theta_2\ldots d\theta_{n-1}dr$$

$$= \int_0^\infty f(r)r^{n-1}\frac{2\pi^{(n-1)/2}}{(\frac{1}{2}n-\frac{3}{2})!}\int_0^\pi \exp(-ikr\cos\theta_1)\sin^{n-2}\theta_1 d\theta_1 dr$$

using

$$\int_0^\pi \sin^\nu d\theta = \frac{(\frac{1}{2}\nu-\frac{1}{2})!\pi^{1/2}}{(\frac{1}{2}\nu)!}$$

Now

$$-\left(\frac{\partial^2}{\partial k_1^2}+\frac{\partial^2}{\partial k_2^2}+\ldots+\frac{\partial^2}{\partial k_n^2}\right) = \int_{-\infty}^{\infty} f(r)(r_1^2+r_2^2+\ldots+r_n^2)\exp(-i\mathbf{k}\cdot\mathbf{r})d^n\mathbf{r}$$

and therefore

$$\left(1-\frac{\partial^2}{\partial k_1^2}-\frac{\partial^2}{\partial k_2^2}-\ldots-\frac{\partial^2}{\partial k_n^2}\right)^N = \int_{-\infty}^{\infty} f(r)(1+r^2)^N\exp(-i\mathbf{k}\cdot\mathbf{r})d^n\mathbf{r}$$

Hence, we can write

$$F(\mathbf{k}) = \left(1-\frac{\partial^2}{\partial k_1^2}+\frac{\partial^2}{\partial k_2^2}-\ldots-\frac{\partial^2}{\partial k_n^2}\right)^N (2\pi)^{n/2}\int_0^\infty \frac{f(r)r^{n-1}}{(1+r^2)^N}\frac{J_{\frac{n-2}{2}}(kr)}{(kr)^{(n/2)-1}}dr \quad (A2)$$

The ratio of two successive terms $u_{n+1}/u_n$ in the infinite series for ${}_1F_2$ is $(a+n)x/[(n+1)(b+n)(c+n)]$ which tends to zero as $n \to \infty$ for any finite $x$. Thus, the series for ${}_1F_2$ converges absolutely and uniformly with respect to $x$ and the same is true of its derivatives (provided that neither $b$ or $c$ is a negative integer or zero when the series diverges). Therefore,

$$\left(\frac{\partial^2}{\partial k_1^2}+\frac{\partial^2}{\partial k_2^2}+\ldots+\frac{\partial^2}{\partial k_n^2}\right){}_1F_2(a;b,\tfrac{1}{2}n;\tfrac{1}{4}k^2)$$

$$=\frac{(b-1)!(\frac{1}{2}-1)!}{(a-1)!}\left(\frac{\partial^2}{\partial k_1^2}+\frac{\partial^2}{\partial k_2^2}+\ldots+\frac{\partial^2}{\partial k_n^2}\right)\sum_{s=0}^{\infty}\frac{(a+s-1)!(\frac{1}{2}k)^{2s}}{(b+s-1)!(\frac{1}{2}n+s-1)!s!}$$

$$=\frac{(b-1)!(\frac{1}{2}n-1)!}{(a-1)!}\sum_{s=0}^{\infty}\frac{(a+s-1)!(\frac{1}{2}k)^{2s-2}}{(b+s-1)!(\frac{1}{2}n+s-2)!(s-1)!}$$

The term $s=0$ disappears so that, by replacing $s$ by $s+1$ we obtain

$$\left(\frac{\partial^2}{\partial k_1^2}+\frac{\partial^2}{\partial k_2^2}+\ldots+\frac{\partial^2}{\partial k_n^2}-1\right){}_1F_2(a;b,\tfrac{1}{2}n;\tfrac{1}{4}k^2)$$

$$=\frac{(b-1)!(\frac{1}{2}n-1)!}{(a-1)!}\sum_{s=0}^{\infty}\frac{(a+s-1)!(\frac{1}{2}k)^{2s}}{(b+s)!(\frac{1}{2}n+s-1)!s!}(a+s-b-s)$$

$$=\frac{a-b}{b}{}_1F_2(a;b+1,\tfrac{1}{2}n;\tfrac{1}{4}k^2)$$

Hence,

$$\left(\frac{\partial^2}{\partial k_1^2}+\frac{\partial^2}{\partial k_2^2}+\ldots+\frac{\partial^2}{\partial k_n^2}-1\right)^N{}_1F_2(a;b,\tfrac{1}{2}n;\tfrac{1}{4}k^2)$$

$$=\frac{(a-b)(a-b-1)\ldots(a-b-N+1)}{b(b+1)\ldots(b+N-1)}{}_1F_2(a;b+N,\tfrac{1}{2}n;\tfrac{1}{4}k^2 \qquad (A3)$$

In the first term of equation (A1) $a=\frac{1}{2}(\beta+n), b=\frac{1}{2}(\beta+n)-N+1$ so that $a-b=N+1$ with the result that the right hand side of the equation vanishes.

For the second term of equation (A1), consider, with $b>0$

$$\left(\frac{\partial^2}{\partial k_1^2}+\frac{\partial^2}{\partial k_2^2}+\ldots+\frac{\partial^2}{\partial k_n^2}\right)k^{2b}{}_1F_2(a;b+\tfrac{1}{2}n,b+1;\tfrac{1}{4}k^2)$$

$$=\frac{(b+\frac{1}{2}n-1)!b!}{(a-1)!}\sum_{s=0}^{\infty}\frac{(a+s-1)!k^{2b+2s-2}}{4^{s-1}(b+\frac{1}{2}n-2+s)!(b+s-1)!s!}$$

as above. Hence,

$$\left(\frac{\partial^2}{\partial k_1^2}+\frac{\partial^2}{\partial k_2^2}+\ldots+\frac{\partial^2}{\partial k_n^2}-1\right)k^{2b}{}_1F_2(a;b+\tfrac{1}{2}n,b+1;\tfrac{1}{4}k^2)=\frac{(b+\frac{1}{2}n-1)!b!}{(a-1)!}$$

$$\times \left[ \frac{(a-1)!4k^{2b-2}}{(b+\frac{1}{2}n-2)!(b-1)!} + \sum_{s=0}^{\infty} \frac{(a+s-2)!(a-1)k^{2b+2s-2}}{4^{s-1}(b+\frac{1}{2}n-2+s)!(b+s-1)!s!} \right]$$

from which is evident that

$$\left( \frac{\partial^2}{\partial k_1^2} + \frac{\partial^2}{\partial k_2^2} + \ldots + \frac{\partial^2}{\partial k_n^2} - 1 \right) k^{2b}{}_1F_2(1; b+\tfrac{1}{2}n, b+1; \tfrac{1}{4}k^2) = (b+\tfrac{1}{2}n-1)4bk^{2b-2} \tag{A4}$$

and

$$\left( \frac{\partial^2}{\partial k_1^2} + \frac{\partial^2}{\partial k_2^2} + \ldots + \frac{\partial^2}{\partial k_n^2} - 1 \right) k^{2b}{}_1F_2(a; b+\tfrac{1}{2}n, b+1; \tfrac{1}{4}k^2)$$

$$= 4b(b+\tfrac{1}{2}n-1)k^{2b-2}{}_1F_2(a-1; b+\tfrac{1}{2}n-1, b; \tfrac{1}{4}k^2), \quad a \neq 1$$

Consequently, if $a \neq 1$ or 2, then since

$$\left( \frac{\partial^2}{\partial k_1^2} + \frac{\partial^2}{\partial k_2^2} + \ldots + \frac{\partial^2}{\partial k_n^2} \right) k^{\beta} = \beta(\beta+n-2)k^{\beta-2}$$

for all $\beta$ except those for which $\beta + n = 2, 0, -2, -4, \ldots,$

$$\left( \frac{\partial^2}{\partial k_1^2} + \frac{\partial^2}{\partial k_2^2} + \ldots + \frac{\partial^2}{\partial k_n^2} - 1 \right) k^{2b}{}_1F_2(a; b+\tfrac{1}{2}n, b+1; \tfrac{1}{4}k^2)$$

$$= 4^2 b(b-1)(b+\tfrac{1}{2}n-1)(b+\tfrac{1}{2}n-2)k^{2b-4}{}_1F_2(a-2; b+\tfrac{1}{2}n-2, b-1; \tfrac{1}{4}k^2)$$

where, in deriving this result, since it cannot be assumed that $b-1 > 0$, with $b = N - \frac{1}{2}\beta - \frac{1}{2}n$ we impose the condition $\beta = 2m (m = 0, 1, 2, \ldots)$. Thus, using equation (A4) we can write

$$\left( \frac{\partial^2}{\partial k_1^2} + \frac{\partial^2}{\partial k_2^2} + \ldots + \frac{\partial^2}{\partial k_n^2} - 1 \right)^N k^{2N-\beta-n}{}_1F_2(N; N-\tfrac{1}{2}\beta, N+1-\tfrac{1}{2}\beta-\tfrac{1}{2}n; \tfrac{1}{4}k^2)$$

$$= \left( \frac{\partial^2}{\partial k_1^2} + \frac{\partial^2}{\partial k_2^2} + \ldots + \frac{\partial^2}{\partial k_n^2} - 1 \right) 4^{N-1} \frac{(N-\frac{1}{2}\beta-\frac{1}{2}n)!(N-\frac{1}{2}\beta-1)!k^{-\beta-n+2}}{(-\frac{1}{2}\beta-\frac{1}{2}n+1)!(-\frac{1}{2}\beta)!}$$

$${}_1F_2(1; -\tfrac{1}{2}\beta+1, -\tfrac{1}{2}\beta-\tfrac{1}{2}n+2; \tfrac{1}{4}k^2) = \frac{(N-\frac{1}{2}\beta-\frac{1}{2}n)!(N-\frac{1}{2}\beta-1)!4^N k^{-\beta-n}}{(-\frac{1}{2}\beta-\frac{1}{2}n)!(-\frac{1}{2}\beta-1)!} \tag{A5}$$

Using equations (A5) and (A3) in equations (A1) and (A2) we find that

$$F(\mathbf{k}) = \frac{(N-\frac{1}{2}\beta-\frac{1}{2}n)!(\frac{1}{2}\beta+\frac{1}{2}n-N-1)!}{(-\frac{1}{2}\beta-\frac{1}{2}n)!(-\frac{1}{2}\beta-1)!} 2^{\beta+n}(-1)^N \pi^{n/2} k^{-\beta-n}$$

Finally, using the formula

$$z!(-z)! = \frac{\pi z}{\sin \pi z}$$

we have

$$(N-\tfrac{1}{2}\beta-\tfrac{1}{2}n)!(\tfrac{1}{2}\beta+\tfrac{1}{2}n-N-1)! = \frac{\pi}{\sin \pi(\frac{1}{2}\beta+\frac{1}{2}n-N)}$$

$$= \frac{(-1)^N \pi}{\sin \frac{1}{2}\pi(\beta+n)} = (\tfrac{1}{2}\beta + \tfrac{1}{2}n - 1)!(-\tfrac{1}{2}\beta - \tfrac{1}{2}n)!(-1)^N$$

so that

$$F(\mathbf{k}) = \frac{(\frac{1}{2}\beta + \frac{1}{2}n - 1)!}{(-\frac{1}{2}\beta - 1)!} 2^{\beta+n} \pi^{n/2} k^{-\beta-n}$$

# Appendix B

# Research software

During algorithm development process for this research thesis, it was necessary to analyse statistical and dynamic data on a range of images. The system for undertaking this task was designed whose graphical user interface is shown in figure B.1.

The basic window is activated using the mouse button to view an area of interest in the image. The program maps the digital information for the locality of the required. The information is mapped line by line and bottom-up on a free field from the image and contains the following data:

a) Values of the color channels HSV, variance between channels, distribution.

b) Values of the colour channels RGB, variance between channels, distribution.

c) Name of the object and the file preference for saving data.

d) Mean and STD for a window consisting of 13x13 pixels and an estimation of the gradient around of these points.

e) Consideration of a ring of 72 points around of a point of interested, mean and STD, distribution, estimation of the maximum and minimum.

f) For a central ring of points in a window of 9x9 pixels, mean and STD, distribution and value after application of a Gaussian filter.

On a right upper supplementary window, it is possible to see the distribution of a cluster of points in the RGB channels. The lower right window removes the

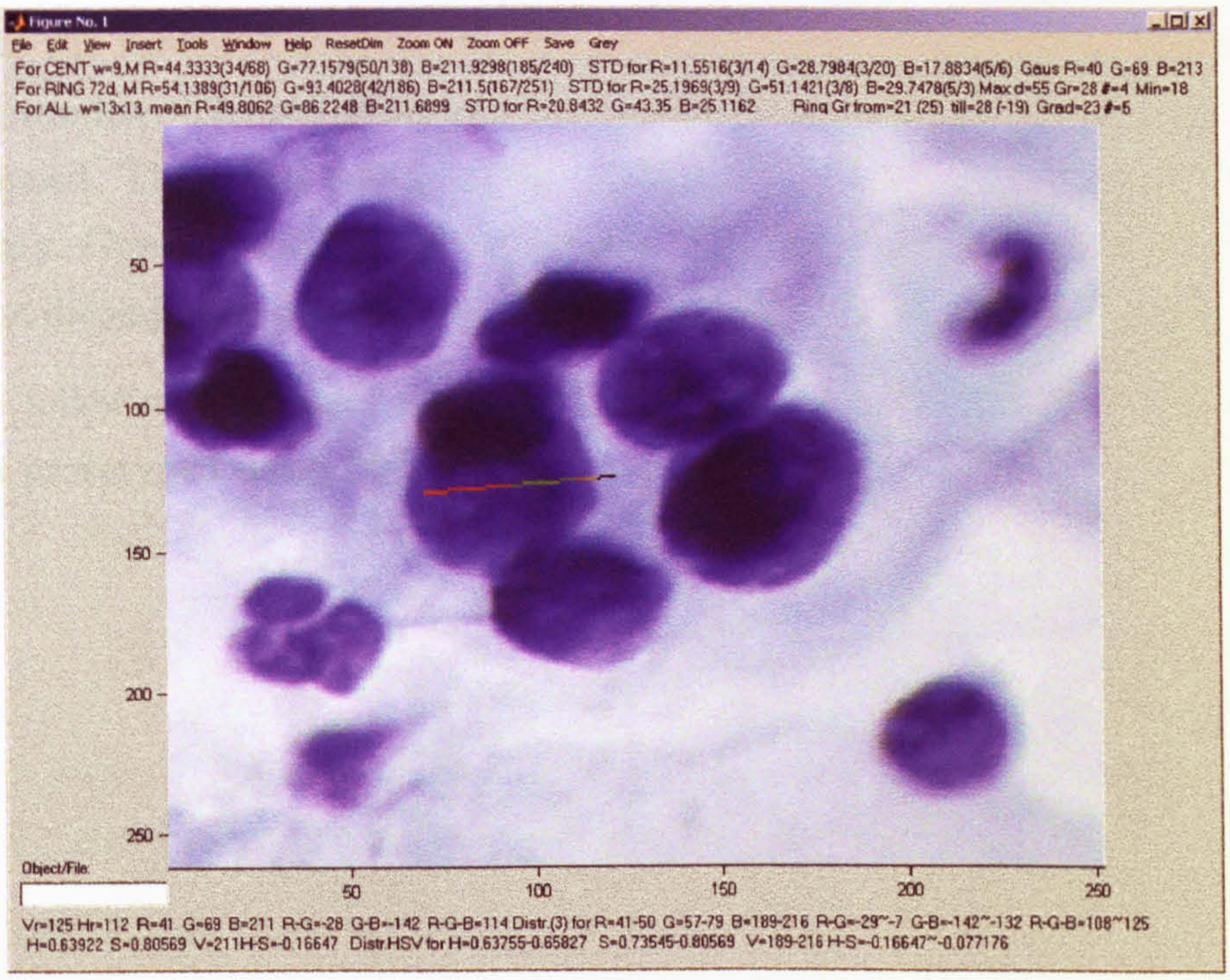



Figure B.1: Main window of research program

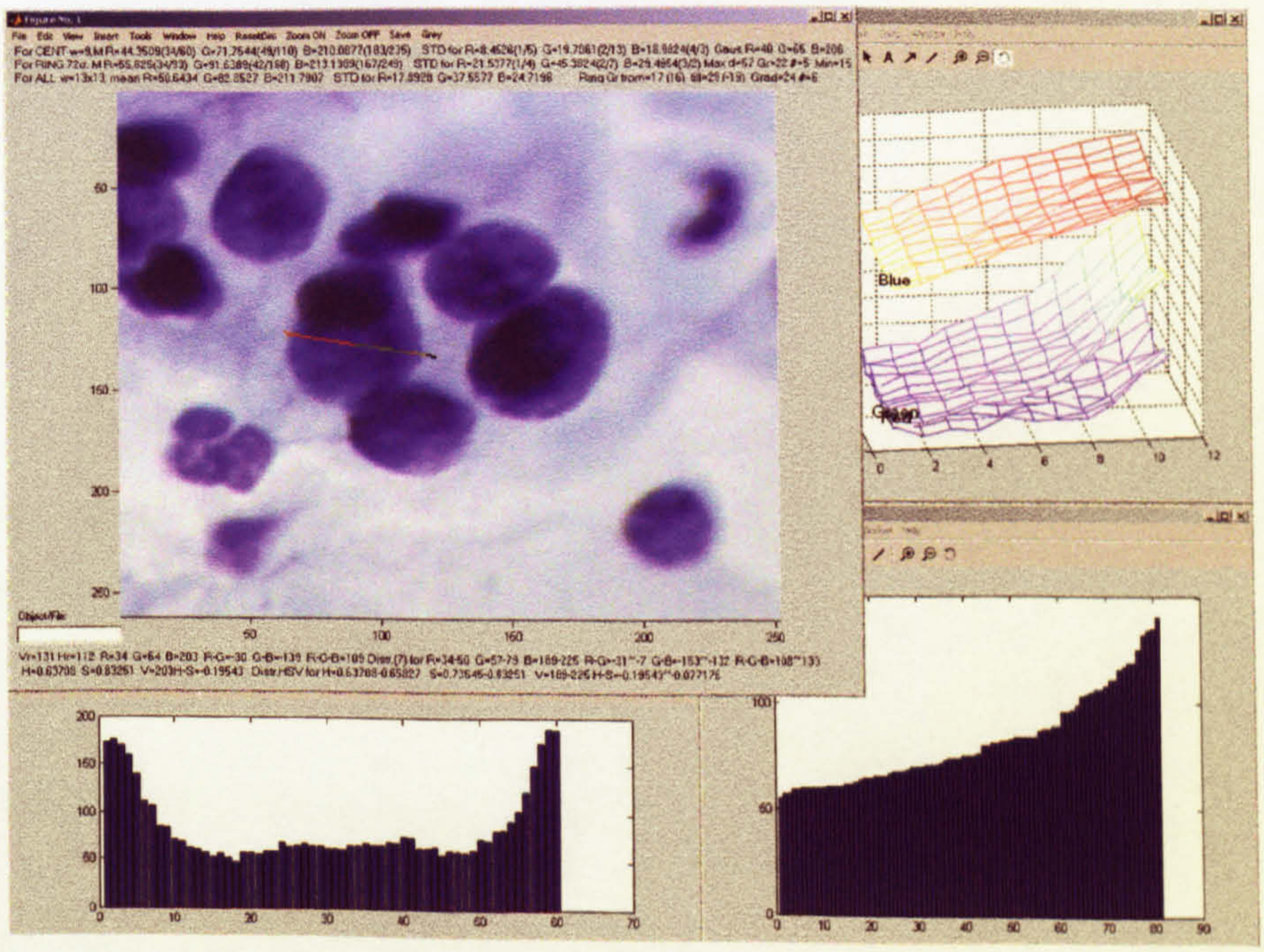



Figure B.2: Supplementary window of research program

sequence, ordered in ascendence of the image point values. The lower left window contains the values of those points of the image profile along a line imaged on a central window. The red point displays the position where the left mouse button was pressed. The yellow colour selects a line up to the outer position of the object boundary. Green marks the distance up to the object centre. The brown colour of a line displays the position on the opposite edge of the object disposed.

The following functions are used in different applications of the developed system prototyped using MATLAB.

# Appendix C

# List of C++ WINAPI procedures

This appendix presents a complete list of the C++ functions which are used in different applications of the developed system.

*Detour by object contour*

```
void WINAPI ObjectContour(long *ppf, ...
            long *ListDotsX,long *ListDotsY, long *Lpr)
```

This procedure has ppf -input image ListDotsX, ListDotsY - list coordinates Lpr - list parameters StartX, StartY, Width, Height

*Obtaining the tops coordinates of polygon*

```
void WINAPI ObjectSpider(long *ppf, ...
            long *ListDotsX,long *ListDotsY, long *Lpr)
```

This procedure has ppf -input image ListDotsX, ListDotsY - list coordinates Lpr - list parameters StartX, StartY, Width, Height

*Self adjustable filter for enhance objects sharpness*

```
void WINAPI ObjectAdFiltr(long *ppf, long *Lpr)
```

This procedure has ppf -input image Lpr - list parameters StartX, StartY, Width, Height

*Object oriented filter*

```
void WINAPI ObjectOtFiltr(long *ppf, long *Lpr)
```

This procedure has ppf -input image Lpr - list parameters StartX, StartY, Width, Height

# Bibliography


[1] I.E. Abdou and W.K.Pratt. Quantitative design and evaluation of enhancement thresholding edge detectors. *Proc. IEEE*, (67):753–763, 1979.

[2] Robert B. Macy Abhijit S.Pandya. *Pattern recognition with neural networks in C++.* IEEE Press, Florida Atlantic University, 1995.

[3] Anna Beck Albert Einstein. *The Collected Papers of Albert Einstein.* Princeton University Press, USA, 1989.

[4] Jeffrey D Ullman Alfred V Aho, John E Hopkroft. *Data structure and Algorithms.* Addison-Wesley Publishing, 2000.

[5] A.Papoulis. A new algorithm in spectral analysis and band-limited extrapolation. *IEEE trans.*, 22, Sept. 1975.

[6] B.Lipkin and A.Rosenfeld. *Picture Processing and Psychopictorics.* Academic Press, New York, 1970.

[7] Benoit B.Mandelbrot. *Fractals : form, chance, and dimension.* W.H.Freeman, San Francisco, 1977.

[8] Benoit B.Mandelbrot. *The Fractal Geometry of Nature.* W.H.Freeman, New York, 1983.

[9] C. Brice and C. Fennema. Scene analysis using regions. *Artificial Intelligence*, (1):205–226, 1970.

[10] C.A.Harlow and S.A.Eisenbeis. The analysis of radio-graphic images. *IEEE Trans. Computer*, (C-22):678–6881, 1973.

[11] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (PAMI-8):679–698, 1986.

[12] Wei-Chung Lin Chin-Tu Chen, Jin-Shin Chou and C.A. Pelizzari. Edge and surface searching in medical images. *In Medical Imaging II*, (SPIE 41):594–599, 1988.

[13] A I Dale. *Most honourable rememberance : The life and work of Thomas Bayes.* New York- Berlin- Heidelberg, 2003.

[14] E.R. Davies. Circularity - a new principle underlying the design of accurate edge orientation operators. *Image and Vision Computing*, (2):134–142, 1984.

[15] E.R. Davies. A new framework for analysing the properties of the generalized hough transform. *Pattern Recognition*, (6):1–7, 1987.

[16] E.H.Mamdani. Advances in linguistic synthesis of fuzzy controllers. *J.Man Mach.*, 8:669–678, 1976.

[17] E.R.Davies. *Machine Vision: Theory, Algorithms, Practicalities.* Academic press, London, 1997.

[18] E.Sanchez. Resolution of composite fuzzy relation equations. *Inf.Control*, 30:38–48, 1976.

[19] J.A. Feldman and Y. Yakimovsky. Decision theory and artificial intelligence: I. a semantic-based region analyzer. *Artificial Intelligence*, (5(4)):349–371, 1974.

[20] Janet Finlay and Alan Dix. *An introduction to Artificial Intelligence.* UCL Press, London, 1996.

[21] S. A.; Flannery, B. P.; Teukolsky and W. T. Vetterling. *Digital Filtering in the Time Domain.* Cambridge University Press, Cambridge, England, 1992.

[22] J. Foglein. On edge gradient approximations. *Pattern Recognition*, (1):429–434.

[23] M.R.B. Forshaw. Speeding up the marr-hildreth edge operator. *Computer Vision, Graphics, and Image Processing*, (41):172–185, 1988.

[24] Herbert Freeman. *Machine vision. Algorithms, Architectures, and Systems.* Academic press, London, 1988.

[25] W.E.L. Grimson. *Object recognition by computer, Thr Role of Geometric Constraints.* MIT Press, 1990.

[26] Ernest L. Hall. *Computer Image Processing and Recognition.* Academic press, London, 1979.

[27] Hentchel and I.Procaccia. The infinite number of generalized dimensions of fractals and strange attractors. *Physica*, 3D:435–444, 1983.

[28] Shunji Mori Hideyuki Tamura and Takashi Yamawaki. Textual features corresponding to visual perception. *IEEE Man. and Cybernetics*, 6, 1978.

[29] H.O.Peitgen and D Saupe. *The Science of Fractal Images.* Springer-Verlag, New-York, 1988.

[30] Marcia L. McLure James J. Moynihan. *EDI: A Guide to Electronic Data Interchange and Electronic Commerce Applications in the Healthcare Industry.* McGraw-Hill Companies, 1996.

[31] J.Goguen. The logic of inexact colcepts. *Synthese*, 19:325–373, 1969.

[32] J.Heikkonen. Recovering 3d-motion parameters from optical flow field using randomized hough transform. *Pattern Recognition*, (9):971–978, 1995.

[33] J.Keller and S.Chen. Texture description and segmentation through fractal geometry. *Computer Vision, Graphics and Image Processing*, (45):150–166, 1989.

[34] J.M.S.Perwitt. *Object enhancement and extraction.* Academic Press, New York, 1970.

[35] A.Zisserman J.Mundy. *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, 1992.

[36] H. Buxton J.S. Wiejak and B.F. Buxton. Convolution with separable masks for early image processing. *Computer Vision, Graphics, and Image Processing*, (32):279–290, 1985.

[37] K.A.Neusypin K.A.Pupkov. *Quistions of theory an application control and navigation systems.* Moscow, 1997.

[38] K.C.Hayes. Texture coarseness: Further experiments. *IEEE Trans.*, (4):467–472, 1974.

[39] K.Clarke and D.Schweizer. Measuring the fractal dimension of natural surfaces using a robust fractal estimator. *Cartography and Geographic Information Systems*, (18):27–47, 1991.

[40] K.Clarke. Scale based simulation of topographic relief. *The American Cartographer Vol.5*, (2):173–181, 1988.

[41] K.Falconer. *Fractal Geometry.* Wiley, 1990.

[42] L.DeCola. Fractal analysis of a classified landsat scene. *Photogrammetric Engineering and Remote Sensing*, 55(5):601–610, 1989.

[43] L.G.Roberts. *Machine perception of three-dimensional solids.* MIT Press, Cambridge, 1965.

[44] Jr Louis J. Galbiati. *Machine vision and digital image processing fundamentals.* State University of New York, New-York, 1990.

[45] Zadeh L. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE*, 3(4):22–44, 1973.

[46] D. Marr and E.Hildreth. *Theory of edge detection.* Number B 207. Proc. R. Soc., London, 1977.

[47] Patrick R.Andrews Martin J.Turner, Jonathan M.Blackledge. *Fractal Geometry in Digital Imaging.* Academic Press, London, 1998.

[48] M.Bennamoun and G.J.Mamic. *Object Recognition: Fundamentals and Case Studies.* Springer-Verlag, London, 2002.

[49] M.Goodchild. Fractals and the accuracy of geographical measures. *Mathematical Geology*, 12(2):85–98, 1980.

[50] Fairhurst Michael. *Computer vision for robotic system: an introduction.* Prentice-Hall, New York, London, 1988.

[51] Roger Boyle Milan Sonka, Vaclav Hlavac. *Image Processing, Analysis and Machine Vision.* PWS, USA, 1999.

[52] N.Lam M.Shelberg and H.Moellering. Measuring the fractal dimensions of surfaces. *Proc. 6th International Symposium on Automated Cartography*, pages 319–328, 1993.

[53] M.Shelberg. *The Development of a Curve and Surface Algorithm to Measure Fractal Dimensions.* Ohio State University, 1982.

[54] Veps'al'ainen A M and Ma J. Estimating the fractal and correlation dimension from 2d and 3d images. *SPIE, Visual Communication and Image Processing IV*, 1199:416–430, 1989.

[55] Vishvjit S Nalwa. *A guided tour of computer vision.* Addison-Wesley Publishing company, 1993.

[56] V.S Nalwa and T.O.Binford. On detecting edge. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (PAMI-8):699–714, 1986.

[57] R. Nevatia and K.R. Babu. Linear feature extraction and description. *Computer Graphics and Image Processing*, (13):257–269, 1980.

[58] Heinonen P. Neuvo Y. Nieminen, A. A new class of detail-preserving filters for image processing. *PAMI(9)*, 1, Jan 1987.

[59] N.Lam and L.De Cola. *Fractal in Geography.* Prentice-Hall, 1993.

[60] N.Lam. Description and measurement of landsat tm images using fractals. *Photogrammatic Engineering and Remote Sensing*, 56(2):187–195, 1990.

[61] N.Rescher. *Many-valued logic.* McGraw-Hill, New York, 1969.

[62] B.Chaudhuri N.Sarkar and P.Kundu. Improved fractal geometry based texture segmentation technique. *IEEE Proceedings Vol.140*, (5):233–241, 1993.

[63] N.Sarkar and B.Chaudhuri. An efficient approach to estimate fractal dimension of textured images. *Pattern Recognition Vol. 25*, (9):1035–1041, 1992.

[64] N.Vadiee. *Fuzzy rule based expert system-I.* Prentice Hall, Englewood, 1993.

[65] F.Yamammoto N.Yokoya and N.Funakubo. Fractal based analysis and interpolation of 3d natural surface shapes. *Comuter Vision Graphics and Image Processing*, (46):284–302, 1989.

[66] R. Ohlander. *Analysis of Natural Scenes.* PhD thesis, Carnegie-Mellon University, 1975.

[67] Theo Pavlidis. *Algorithms for graphics and image processing.* Computer Science Press, USA, 1982.

[68] P.Blicher. *Edge Detection and Geometric Methods in Computer Vision.* PhD. University of California, California, 1984.

[69] P.Kube and A.Pentland. On the imaging of fractal surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (10/5):704–707, 1988.

[70] P.Maragos and F-K Sun. Meassuring fractal dimension: Morfological estimates and iterative estimation. *SPIE Visual Communications and Image Processing IV*, (1199):431–438, 1989.

[71] P.P.Ohanian and R.C.Dubes. Performance evaluation for four classes of textural features. *Pattern recognition*, (25(8)):819–833, 1992.

[72] R. Ohlander K. Price and D.R. Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics Image Processing*, (8(3)):313–333, 1978.

[73] K.K. Pringle. *Visual perception by a computer Automatic Interpretation and Classification of Images.* Academic Press, New York, 1969.

[74] T.V. Robrtson P.W.Swain and K.S. Fu. *Multispectral image partitioning.* Technical report, School of Electrical Engineering, Purdue Univ., 1973.

[75] Wesley E.Snyder Hairong Qi. *Machine Vision.* Cambridge University Press, England, 2004.

[76] R.A.Kirsh. *Computer determination of the constituent structure of biological images.* MIT Press, Comput. Biomed. Res., 1971.

[77] R.Bajcsr. Computer description of textured surfaces. *Procecdings of the International Conference on Artificial Intelligence*, (August):572–579, 1973.

[78] O.Faugeras R.Derichle. Tracking line segments. *Image Vision and Computing*, 8(4):261–270, 1990.

[79] R.F.Brammer. Unified image computing based on fractals and chaos model techniques. *Optical Engineering*, 28(7):726–734, 1989.

[80] B.D. Ripley. *Pattern recognition and neural networks.* Academic Press, Oxford, 1996.

[81] R.Maydole. Many-valued logic as a basis for set theory. *Logic*, 4:269–291, 1975.

[82] R.M.Haralick. Edge and region analysis for digital image data. *Computer graphics and Image Processing*, (12):60–73, 1980.

[83] R.M.Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, (PAMI-6):58–68, 1984.

[84] R.N.Bracewell. *The Fourier Transform and Its Applications.* McGraw-Hill Book Co., New York, 1978.

[85] G.S. Robinson. Edge detection by compass gradient mask. *Computer Graphics and Image Processing*, 6:429–501, 1977.

[86] A. Rodin. Sinergetica i determinizm. `http://www.synergetic.ru/philosophy/index.php?article=rodin`.

[87] A. Rosenfeld. *Digtal Picture Processing*, volume 1,2. Academic Press, New York, 1982.

[88] P.E. Hart R.O. Duba. *Pattern Classification and Scene Annalysis.* Academic Press, New York,Wiley, 1973.

[89] R.Voss. Random fractals: Characterisation and measurement,in scaling phenomena in disordered systems. *Eds. R Pynn and A Skjeltorps,Plenum*, 1986.

[90] R.W.Gerchberg. Super resolution through error energy reduction. *Opt.Acta*, 21, 1974.

[91] D.Quattrochi S.Jaggi and N.Lam. Implementation and operation of three fractal measurement algorithms for analysis of remote sensing data. *Computers and Geosciences*, 19(6):745–767, 1993.

[92] Gradshteyn I S and Ryzik I M. *Table of Integrals, Series and Products.* Academic Press, 1980.

[93] Jone D S. *Generalised Functions.* McGraw-Hill, 1966.

[94] Cornelius T.Leondes. *Image Processing and Pattern Recognition.* Academic press, London, 1998.

[95] V.Tom T.Peli and V.Lee. Multi-scale fractal and correlation signatures for image screening and natural clutter suppression. *SPIE Visual Communications and Image Processing IV*, 1199:402–415, 1989.

[96] N.N.Beloglazov U.A.Andreev. *Correlative - extreme video sensitive systems for robots.* Tomsk, 1986.

[97] H. Waelbroeck and F. Zertuche. Discrete chaos, 1998. `http://papaya.nuclecu.unam.mx/~nncp/chaos98.ps`.

[98] Y.Y. Yakimovsky and J. Feldman. A semantics-based decision theoretic region analyzer. *3-th int. Conference Artificial Intelligence*, (1):580–588, 1973.

[99] S.Marchand-Maillet Y.M.Sharaiha. *Binary Digital Image Processing. A Discrete Approach.* Academic press, London, 2000.

[100] Lotfi A. Zadeh. *Fuzzy sets and their applications to cognitive and decision processes.* Academic Press, New York, 1975.

[101] Cancer research uk. `http://www.cancerresearchuk.org/aboutcancer/reducingyourrisk/9314`.