FRACTALS, IMAGE EROSION AND DILATION

by

NDAJAH, PETER EMMANUEL
MTECH/2001/2002/758

Submitted To the Department of Mathematics/Computer Science, Federal University of Technology, Minna in Partial Fulfillment for the Award of Master of Technology in Mathematics

01 JULY, 2005

CERTIFICATION

This thesis titled: FRACTALS, IMAGE EROSION AND DILATION by NDAJAH, PETER EMMANUEL meets the regulation governing the award of the degree of Masters of Technology in Mathematics, Federal University of Technology, Minna and is approved for its contribution to knowledge and literary presentation

.....
Prof. K. R. Adeboye
Supervisor

Date 1/7/2005

.....
Dr L.N. Ezeako
Head of Department

Date 01-07-2005

.....
Prof. G.D. Momoh
Dean, School of Science and
Science Education

.....
Date

.....
Prof J.A. Abalaka
Dean, Postgraduate School

Date.....

.....
Prof F.O. Otunta
External Examiner

Date 01-07-05

DEDICATION

To Jesus of Nazareth
My ever abiding friend
Who has always been there for me
Long after other men are gone

ACKNOWLEDGMENT

First of all, I cannot fail to show my profound gratitude to my supervisor, Prof. K.R. Adeboye for the longsuffering and extreme understanding he exercised in my case. Once again I am very grateful. I also extend my thanks to the Head of Department, Mr. L.N. Ezeako who has been more than a guide to me. He is the 'friend that sticks closer than a brother'. I thank my parents, Mr. Emmanuel Ndajah and Mrs Asta Bello for their support. It could never have been greater. They insisted I must go to school. My thanks go to my senior colleagues, Dr. Y.M. Aiyesimi, Dr. N.I Akinwande, Dr. A. Abubakar, Mal. Audu Isah and others in the department of mathematics/Computer Science, Federal University of Technology, Minna. Thank you very much for all the support you have given me in the course of this work.

My brother, Pastor Zephaniah Emmanuel Ndajah and his wife Lucy, his children, Vilita, Mbimba and David cannot be left out. Thank you very much and may God continue to bless you all.

ABSTRACT

This work is an investigation into how images can be generated on the one hand and image analysis on the other. The science of generating images helps us to understand the mathematical principles behind the formation of natural structures. More importantly, the same theory has enabled us to solve problems in a vast number of fields such as economics, geology, physics, meteorology, etc. Image analyses were carried out using the tools of mathematical morphology. Erosion and dilation of images were investigated and it was shown that all other morphological operations are only a combination of erosion and dilation.

TABLE OF CONTENTS

CHAPTER THREE
FRACTALS

CHAPTER FOUR
EROSION AND DILATION

CHAPTER ONE

INTRODUCTION

1.1 The Science of Fractals and the Rebirth of Experimental mathematics

Much scientific research of the past has analyzed human-made machines and the physical laws that govern their operations. The success of science relies on the predictability of the underlying experiments. Euclidean geometry – based on lines, circles, etc. is the tool used to describe spatial relations, while differential equations are essential in the study of motion and growth. However, natural shapes such as mountains, clouds or trees do not fit well into this framework. The understanding of these phenomena has undergone a fundamental change from the last two decades of the 20^{th} century. Fractal geometry as conceived by Mandelbrot [1978–79] provides mathematical model for nature. One of Mandelbrot's key observations has been that these forms possess remarkable statistical invariance under magnification [Peitgen, 1993]. This may be qualified by a fractal dimension, a number that agrees with our intuitive understanding of dimension which may not necessarily be an integer. These ideas may also be applied to time-variant processes.

Another important discovery has been that even in very simple non-linear dynamical systems, such as the double pendulum, long-term predictions are not possible despite exact knowledge of the underlying equations. Such systems exhibit behavioral patterns that we conceive only as erratic or chaotic despite their very simple and deterministic generating mechanisms. Arbitrarily small perturbations of solutions are blown up by such systems until such solutions have lost all correlation with the original ones. This phenomenon has been termed "sensitive

dependence on initial condition" and is the trademark of what is now known as chaos theory [Peitgen, 1996]. There is a strong connection between chaos and fractal geometry, namely, as one follows the evolution of states of a chaotic non-linear system, it typically leaves a trace in its embedding space which has a very complex geometric structure. This trace is a fractal [Mandelbrot, 1982].

But what makes the science of fractals and chaos so interesting is that, this fairly new area of research has created pictures of such power and singularity that a collection of them, for example, has proven to be one of the most successful world-wide series of exhibitions ever sponsored by the Goethe Institute. Since 1985 the exhibition has traveled to more than 100 cities in more than 30 different countries, attracting more than 140,000 visitors in London alone [Peitgen and Saupe, 1996]. More important, however, is the fact that chaos theory and fractal geometry have corrected our outmoded conception of the world.

The magnificent success in the fields of natural sciences and technology had, for many years fed the illusion that the world on the whole functioned like the clockwork mechanism, whose laws were known. It was believed, the evolution or development of things could – at least in principle be ever more accurately predicted. Captivated by the breathtaking advances in the promises of greater command of information, many have put increasing hope in the machine. One conclusion that can be drawn from the new theories, which are admittedly still young, is that stricter determinism and apparently accidental development are not mutually exclusive, but rather their coexistence is more the rule in nature. Chaos theory and fractal

geometry address this issue. In this sense, fractal geometry is first and foremost a new 'language' used to describe complex forms of the 'traditional language' of the Euclidean geometry are basic visible forms such as lines, circles and spheres, those of the new language do not lend themselves to direct observations. They are namely, algorithms, which can be transformed into shapes and structures only with the help of computers. In addition, the supply of these algorithmic elements is inexhaustively large; and they are capable of providing us with a powerful descriptive tool. Once this new language has been mastered, we can describe the form of a cloud as easily and as precisely as an architect can describe a house using the language of traditional geometry.

The correlation of chaos and geometry is anything but coincidental. Rather, it is a witness to their deep kinship. This kinship as we see later is best described in the Mandelbrot set, a mathematical object discovered by Benoit Mandelbrot in 1980. It has been described by some as the most complex, and possibly the most beautiful object ever seen in mathematics. Its most fascinating characteristics however, have only just recently been discovered, namely; that it can be interpreted as an illustrated encyclopedia of an infinite number of algorithms. It is a fantastically and efficiently organized storehouse of images, and as such it is the example per excellence of order in chaos. Fractals and modern chaos theory are linked by the fact that many of the contemporary pace-setting discoveries in their fields were only possible using computers. From the perception of our inherited understanding of mathematics, this is a challenge which is felt by some to be powerful renewal and liberation and by others a denigration. In essence, chaos theory and fractal geometry change

radically our understanding of equilibria – and therefore of harmony and order – in nature as well as in other contexts and offer a new holistic and integrated model which can encompass an edge of the true complexity of nature for the first time. It is highly probable that the new methods and terminologies will allow us, for example, a much more adequate understanding of ecology and climatic development, and thus they could contribute to our more effectively tackling gigantic problems.

Fractal geometry and chaos are often associated with experimental mathematics. Experimental mathematics does not imply an attempted invasion of pure mathematics by applied mathematics. Applied mathematics has always been permeated with science, hence with experiment. This feature greatly contributed to its being thoroughly unpopular with those believing that applied mathematics is bad mathematics. But experimental mathematics means something different: it means injecting experiment back into core parts of mathematics that need not – at least at present have any contact with science. Its most striking impact may be that it underlines the reality of an essential distinction between mathematical fact and mathematical proof. Many times mathematicians insist on defining their fields narrowly, as beginning with proofs and gives short shrift facts accustomed to seeing new mathematical fact almost exclusively suggested by the proofs of old mathematical facts. But history reveals that in the past, the development of mathematics has relied upon many other sources, both of observation and of experimentation. Today's experimental mathematics does not even spurn the kind of observation that has been characteristic of the least sophisticated

among the empirical sciences, but primarily relies upon active experimentation.

New methods of searching for new facts provide mathematics with a powerful front end of unexpected character, one that involves more than just the proverbial pencil and paper. Thus pictures have already demonstrated their astonishing power to help in early stages of both mathematical proofs and physical theory. As this help expands, it may well lead to a new equilibrium and the changes in the prevailing styles completed mathematical proof and of completed physical theory. In other words, we may be witnessing the re-emergence of a new active doublet of experimental and/or theoretical study. As seen in physics, experimental and theoretical physics seldom live in perfect harmony, but they know they must not only coexist but actually listen to each other and otherwise interact. Few in either party want to annihilate the other. In mathematics, the situation is very different. There has been a long history of conflict. The computer is a new tool that has come into being and it has brought two gifts to science. Its first gift is vastly enhanced calculations, which will not concern us in this work. The second gift is graphics, which tells an altogether different story and has brought a profound qualitative change, hence a fair amount of upheaval. Computer graphics allow us the privilege and the delight of taking up theories in physics and mathematics and of proving that if they are suitably transformed, these very same theories are enriched in their own mathematical or physical terms. And they also generate patterns that readily pass for forgeries of life, nature, and even art, in their unfathomable complication.

1.2 Chaos

For many, chaos theory already belongs to the greatest achievement in the natural sciences. Indeed, it can be claimed that very few developments in the natural science have awakened so much public interest. The main maxim of science is its ability to relate cause and effect. On the basis of laws of gravitation, for example, astronomical events such as eclipses and the appearance of comets can be predicted thousands of years in advance. Other natural phenomena, however, appear to be much more difficult to predict. Although the movements of the atmosphere, for example, obey the laws of physics as much as the movements of the planets do, weather prediction is still rather problematic. We speak of the unpredictable aspects of weather just as if we were talking about rolling dice or letting an air balloon loose to observe its erratic path as the relation between cause and effect, such phenomena are said to have random elements. Yet there was little reason to doubt that precise predictability could in principle be achieved. It was assumed that it was only necessary to gather and process greater quantities of more precise information (e.g. through the use of denser networks of weather stations and more powerful computers dedicated solely to weather analysis). Some of the first conclusions of chaos theory however, have recently altered the viewpoint. Simple deterministic systems with only a few elements (e.g. the quadratic iterator) can generate random behaviour, and that randomness is fundamental, gathering more information does not make it disappear. This fundamental randomness has come to be called chaos. An apparent paradox is that chaos is deterministic and generated by fixed rules which do not themselves involve any elements of change. In principle, the future is completely determined by the past, but in practice, small uncertainties much like minute

errors of measurement which enter into calculations, are amplified, with the effect that even though the behaviour is predictable in the short-term, it is unpredictable over the long run. The discovery of such behaviour is one of the most important achievements of chaos theory. Another is the methodologies which have been designed for a precise scientific valuation of the presence of chaotic behaviour in mathematical models as well as in real phenomena. Using these methodologies, it is now possible, in principle to estimate the 'predictability horizon' of a system. This is the mathematical, physical or time parameter limit within predictability is ideally possible and beyond which we will never be able to predict with certainty. It has been established, for example, that the predictability horizon in weather forecasting is not more than about two or three weeks [Peitgen, Jurgen and Saupe, 1996]. this means that no matter how many more weather stations are included in the observation, no matter how much more accurately weather data are collected and analyzed, we will never be able to predict the weather with any degree of numerical accuracy beyond this horizon of time [Brigg, 1992].

For the era of determinism, which was mathematically grounded in calculus, the 'Laplace demon' became the symbol. If we could imagine a consciousness great enough to know the exact locations and velocities of all objects in the universe at the present instant, as well as the forces, then there could be no errors from this consciousness. It could calculate anything about the past or the future from the laws of cause and effect. In its core, the deterministic credo means that the universe is comparable to the ordered running of a tremendously precise clock in which the present state of things is, on the other hand the cause of its future state. Present, past and future are bound

together by the causal relationships; and according to the views of the determinist, the problem of an exact prognosis is only a matter of the difficulty or recording all the relevant data. The deterministic credo was characteristic of the Newtonian era, which for the natural sciences came to an end at the latest through the insights of Weiner Heisenberg in 1927 proclamation of his uncertainty principle, but for other sciences it is still considered valid. Classical determinism in its fearful strictness had to be given up – a turning point of enormous importance.

Indeed, the history of numerical weather prediction illustrates better than anything else the undiminished belief in a deterministic world, for in reality Heisenberg's uncertainty principle did not all mean the end of determinism, it only modified it. The most carefully conducted experiment is after all, newer completely isolated from the influences of the surrounding world, and the state of a system is never precisely known at any point in time. The absolute mathematical precision which Laplace presupposed is not physically realizable; minute imprecision is, as a matter of principle, always present. What scientists actually believed was this: from approximately the same causes follow approximately the same effects in nature as well as in any good experiment. And this is indeed often the case, especially over short time spans. If these were not so, we would not be able to ascertain natural laws, nor could we build any functioning machines. But this apparently very plausible assumptions is not universally true, it does not do justice to the typical course of natural processes over long periods of time. Around 1960, Lorenz discovered this deficiency in the models used for numerical weather prediction and it was he who coined the term "butterfly effect" [Peitgen, 1993].

Thus, Heisenberg's response to deterministic thinking was also incomplete. He concluded that the strong causality principle is wrong because its presumptions are enormous. Lorenz has now shown that the conclusions are also wrong. Natural laws, and for that matter determinism, do not exclude the possibility of chaos. In other words, determinism and predictability are not equivalent. And what is even a more surprising finding of recent chaos theory has been the discovery that these effects are observable in many systems which are much simpler than the 'weather'. In fact, this can be observed in very simple feedback systems, even as simple as the quadratic iterator $x \to ax(1-x)$ (the Logistic equation).

Moreover, chaos and order (i.e. causality principle) can be observed in juxtaposing within the same system. There may be a linear progression of errors characterizing a deterministic system which is governed by the causality principle, while (in the same system) there can also be an experimental progression of errors (i.e. the butterfly effect) indicating that the causality principle breaks down. In other words, one of the lessons coming out of chaos theory is that the validity of the causality is narrowed by the uncertainty principle from one end as well as by the intrinsic instability properties of the underlying natural laws from the other end.

Chaos describes a situation where typical solutions (or orbits) of a differential equation (or typical evolutions of some other models describing deterministic evolution) do not converge to a stationary or periodic function (of time) but continue to exhibit a seemingly

unpredictable behaviour. The dynamical system (or models) describing deterministic evolution considered are differential equations $\dot{x} = F(x)$ with $x \in X, X$ a differentiable manifold and $F : X \to T(x)$ a vector field on X, and differentiable mapping $\phi : X \to X$ which may or may not be invertible. For a given initial state $x_0 \in X$ the corresponding evolution is the solution $x(t)$ of the differential equation with $x(0) = x_0$ or in the case of a mapping, the function $N \to X$ given by $n \mapsto \phi^n(x_0)$. The last case is the discrete-time situation, the first case that of continuous time. Even if the evolution can be defined for negative time, only the part with positive time is considered. Also, only bounded evolutions are considered here i.e. $x(t), x_n \; with \; t \geq 0,$ respectively $n \geq 0$, whose closure as a subset of X is compact. It is assumed that there is a metric defined on X. One says that such a dynamical system is chaotic if there is a subset $\overline{X} \subset X$ which has positive measure (for every measure in the Lebesgue measure class) which is invariant in the sense that every evolution starting in $\overline{X}$ stays in $\overline{X}$ and such that the evolutions in $\overline{X}$ have the following properties

1.2.1 Quasi-Periodic Evolution

No evolution starting in $\overline{X}$ is periodic or quasi-periodic. An evolution $x(t)$ is quasi-periodic if it can be written as $x(t) = F(\omega_1 t_1, \omega_2 t_2, ..., \omega_n t_n)$ with $\omega_1, \omega_2, \omega_3, ..., \omega_n$ independent over the rationals and F periodic with period 1 in all its variables, an evolution x_n is quasi-periodic if it can be written as $x_m = F(\omega_1 n, \omega_2 n, ..., \omega_m n)$ with $1_1, \omega_2, \omega_3, ..., \omega_m$ independent over the rationals and F periodic with period 1 in all its variables. No evolution in $\overline{X}$ tends to a periodic or quasi-periodic evolution as time tends to infinity.

1.2.2 Sensitive Dependence on Initial Conditions

(Sensitive dependence on initial conditions) there is some positive constant A such that for each $x_0 \in X$ and each $\varepsilon > 0$, there is some y_0 in an ε-neighbourhood of x, such that for some positive time the evolution starting in x_0 and y_0 are more than A apart. These conditions are probably not independent. The first two conditions may be a consequence of the third condition. The third condition implies some unpredictability. Even if we know the initial state with arbitrary (but finite) precision, there is some moment in the future at which the state cannot be predicted within a distance A from the information about the initial state. The main examples of chaotic dynamical systems (and dynamical systems which are supposed to be chaotic) are discussed below:

1.2.3 The Logistic Family

This is a one-parameter family [Peitgen and Saupe, 1998] of one-dimension mappings $L_a(x) = 1 - ax^2$, it has been proved that for a large set (of positive Lebesgue measure) of values of the parameter a, this mapping defines a chaotic dynamical system. These mappings were introduced to describe population dynamics under certain conditions.

1.2.4 The Henon Family

This is a 2-parameter family of two-dimensional invertible mappings $H_{a,b}(x, y) = (1 - ax^2 + y, bx), b \neq 0$ in this example, there is only numerical evidence that $H_{a,b}$ defines, for many parameter values and b, a chaotic dynamical system.

1.2.5 The Lorenz Family

This is a three parameter family of differential equations in $\Re^3$

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = rx - y - xz$$
$$z = xy - bz$$

There is a well-developed theory concerning this system. Still there is no complete proof that for any parametric values, this system is chaotic. This chaoticity is strongly suggested by numerical results combined with geometric arguments: what is lacking is a tedious numerical verification. This equation was proposed in connection with convection problems.

1.2.6 General Axiom A Attractors

This is a class of abstract dynamical systems which are chaotic. Among the chaotic dynamical systems they are the most 'regular' and also they are the ones which are most mathematically understood. Finally in a number of physical and chemical experiments, in particular related with weak turbulence and open chemical reactions, far from equilibrium, the experimental data indicate that one should explain these experiments in terms of chaotic dynamical systems. The literature has not yet standardized with respect to definition of chaotic maps.

1.3 Image Processing

Digital image processing involves the manipulation and interpretation of digital images with the aid of a computer. Digital image processing is an extremely broad subject and it often involves procedures which can be mathematically complex. Here we attempt to only introduce the subject matter until later in this work when the subject of erosion and dilation of images will be adequately handled. The central idea behind

digital image processing is quite simple. The digital image is fed into the computer, one pixel at a time. The computer is programmed to insert these data into main equation, or a series of equations, and then store the results of the computation for each pixel. These results from a new digital image that may be displayed or recorded in pictorial format or may itself be further manipulated by additional programs. The possible forms of digital manipulations are literally infinite. However, virtually all these procedures may be categorized into one (or more) of the following.

1.3.1 Image Rectification and Restoration

These operations are designed to correct distorted or degraded image data to create a more faithful representation of the original scene. This typically involves the initial processing of raw image data to correct for geometric distortions, and to eliminate any noise present in the data. Thus, the nature of any particular image restoration process is highly dependent upon the characteristics of the sensor used to acquire the image data. Image rectification and restoration procedures are often termed preprocessing operations because they normally precede further manipulation and analysis of the image data to extract specific information.

1.3.2 Image Enhancement

These procedures are applied to image data in order to more effectively display or record the data for subsequent visual interpretation. Normally, image enhancement involves techniques for increasing the visual distinction between features in a scene. The objective is to create "new" images from the original image data in order to increase the amount of information which can be visually

interpreted from the data. The enhanced image can be displayed interactively on a monitor or they can be recorded in a hand copy format, either in black and white or in colour. There are no simple rules for producing the single "best" image for a particular application. Often several enhancements made from the same "raw" image are necessary.

1.3.3 Image classification

The objective of these operations is to replace visual analysis of the image data with quantitative techniques for automating the identification of features in a scene. This normally involves the analysis of multispectral image data and the application of statistically based decision rules for determining the object's identity of each pixel in an image. When these decisions rules are based solely on the spectral radiances observed in the data, we refer to the classification process as spectral pattern recognition. In contrast, the decision rules may be base don the geometrical shapes, sizes, and [patterns present in the image data. These procedures fall into the domain of spatial pattern recognition. In either case, the intention of the classification process is to categorize all pixels in a digital image into one of several themes.

1.3.4 Data Merging

These procedures are used to combine image data for a given object with other acquired data sets for the same object. These data sets might simply consist of image data generated on other data by the same sensor. Frequently, the intent of the data merging is to combine acquired data with other sources of information.

1.4 Mathematical Concepts in Image Processing and Fractals

1.4.1 Fractals

Fractals were originally defined by B.B. Mandelbrot as point sets with non-integer dimension in the sense of Hausdorff-Besicovitch. Classical examples are the triadic Cantor set and the non-differentiable curve obtained by von Koch. Typically, a fractal is self-similar (i.e. every small looks like the larger whole) in a deterministic or stochastic way. D. Sullivan introduced the concept of quasi-self-similarity. A quasi-isometry is defined by a function f acting on a metric space M with a metric satisfying $\frac{1}{k} d(x,y) < d(f(x), f(y)) < k d(x,y)$

$$for\ all\ x, y \in M$$

A set F is called quasi-self similar if there exists a k and r_0 such that multiplication by $\frac{1}{r}$ of $F \cap D_r(x)$ maps into F by a quasi-isometry for all $r < r_0$ and all $x \in F$ (here $D_r(x)$ is the open ball centered at x of radius r).

Accordingly, a fractal may be defined as a quasi-self-similar set. In some important cases, the similarity transformations with two or more generators. The Julia set of an analytic function $f(z)$ is such a fractal. The inverses of f being the generators of the corresponding semi-group. The variety of ways, but generally accepted definitions are still lacking. In one such a generalization, the fractal dimension is only a local property. Multifractal measures are related to a distribution on a geometric support which would be a fractal set in the ordinary sense. The field of fractals is rapidly expanding, in particular their applications in statistical physics, natural sciences and computer graphics e.g. the use fractals in image processing may give a considerable compression

of relevant data. Many 'objects' in nature such as coastlines, zeolites, patterns of dielectric discharge, Anderson wave functions, dendritic growth and viscous fingers can well be described by deterministic or stochastic (multi-) fractal structures. [Feder, 1996]

1.4.2 Image Processing (Concepts)

In order for any digital computer processing to be carried out on an image, it must first be stored within the computer in a suitable form that can be manipulated by the computer program. The most practical way of doing this is to divide the image up into a collection of discrete (usually small) cells, which are known as pixels (picture elements). Most commonly, the image is divided into a rectangular grid of pixels, so that each pixel is itself a small rectangle. Once this has been done, each pixel is given a pixel value that represents the colour of that pixel. It is assumed that the whole pixel is the same colour, and so any colour variation that did not exist within the area of the pixel before the image was discretized is lost. However, if the area of each pixel is very small, then the discrete nature of the image is often not visible to the human eye. Other pixel shapes and formation can be used, most notably the hexagonal grid in which each pixel is a small hexagon. This has the same advantage in image processing, including the fact that pixel connectivity is less ambiguously defined that with a square grid, but hexagonal grids are not widely used. Part of the reason is that many object capture systems (e.g. digital cameras and scanners), intrinsically discretize the captured image into a rectangular grid in the first instance.

The discretization of images into pixels of a rectangular form make them susceptible to treatments of linear algebra and calculus. The

image pixels form an array and thus a matrix $I_{i,j}$ has a specific binary value that describes its colour and other attributes. Computer programs written to process images take advantage of the array form to process one pixel at a time until the job is done. An image that we look at is given in its standard Euclidean basis. The change in the image occurs when we try to find a new basis where different features (shapes, edges, noise) are represented as basis vectors.

So by looking at an image in its standard basis, we cause changes to the original image. Calculus is useful in image processing because transforms such as the Fourier transforms and some of its sister transforms are used as filters. Filtering usually take place in the frequency domain so an image can be described in terms of frequencies (i.e. colours) as

a. Low frequencies – background, overall shape
b. High frequencies – details, edges, noise, etc.

From (a) and (b) above, we can see that image representation is also a waveform of the form say $A\cos 2\pi ft$. When the waveform $A\cos 2\pi fi$ is fed into a linear time-invariant filter(e.g. electrical filter), the output is also harmonic but has in general a different amplitude and phase; let it be $B\cos(2\pi ft+\phi)$. Then thye filter is completely specified by a certain frequency-dependent complex quantity $T(f)$, whose amplitude is given by B/A and whose phase is ϕ;

Thus $T(f)=\frac{B}{A}e^{i\phi}$

We refer to $T(f)$ as the transfer factor, transfer function, system function, or frequency response of the filter.

1.4.3 Enhancement (Enhance Features)

Enhancement makes images or signal "look" better. The tool for enhancement is the spatial (i.e. more than one dimension) filters. Since an image is usually two-dimensional (although three dimension images are now possible), a two dimension Fourier or wavelet transform are used to enhance the features of an image.

1.4.4 Warping

Warping an image puts the image on an elastic mesh which is stretched. Here the use of partial differential equations becomes imperative. This done by formulating PDEs that describe the mesh (array) deformations. Multi-grid PDE solvers are useful tools for mesh deformation.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

Mandelbrot is often characterized as the father of fractal geometry. Some people, however, remark that many of the fractals and their description goes back to classical mathematics and mathematicians of the past like Gregor Cantor (1872), Giuseppe Peano (1890), David Hilbert (1891), Helve von Koch (1904), Waclaw Sierpinski (1916), Gaston Julia (1918), or Felix Hausdorff(1919), to name just a few. It is true that the creation of these mathematicians played a key role in Mandelbrot's concept of a new geometry. But, at the same time, they did not think of their creations as conceptual steps towards a new perception or a new geometry of nature. Rather, what we know so well as die cantor set, the Koch curve, the Peano curve, the Hilbert curve and the Sierpinski gasket were regarded as exceptional objects, as counter examples, as 'mathematical monsters'. Maybe this is bit overemphasized. Indeed, many of the early fractals arose in the attempt to fully explore the mathematical content and the limits of fundamentals notions (e.g. 'continuous' or 'curve'). The Cantor set, the Sierpinski carpet and the Menger sponge stand out in particular because of their deep roots and essential role in the development of early topology.

But even in mathematical circles their profound meaning had been somewhat forgotten, and they were as shapes, intended to demonstrate the deviation from the familiar rather than typify the normal. Then Mandelbrot demonstrated that these early mathematical fractals in fact have many features in common with shapes found in nature. In other words, we should say that Mandelbrot turned the

manifold mathematical interpretation and value of these fantastic inventions upside down. But in fact, he did more. The best way to describe his contribution is to say that, indeed, some characters, such as the Cantor set, were already there. But he went on to develop a language into which the characters could be embedded. In other words, he noticed that the seemingly exceptional is more like the rule and then develop a systematic language with words and sentences and grammar. According to Mandelbrot himself, he did not follow a grand plan when carrying out the program; but rather summarized in a way, his complex - one is tempted to say nomadic - scientific experience in mathematics, linguistics, economics, physics, medical sciences and communications networks, to mention a few areas where he was active.

Before we begin a discussion of classical fractals, we introduce the concept of self-similarity. It is the underlying theme in all fractals, more pronounced in some of them and variations in others. In a way the word self-similarity needs no explanation and at this point we merely give an example of a natural structure with that property, a cauliflower.

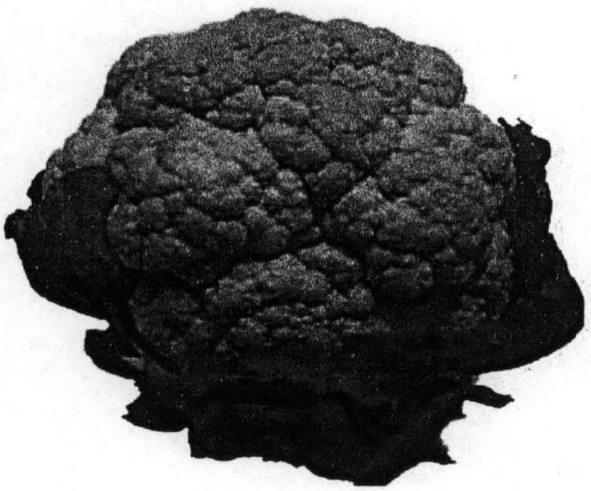


Fig 2.01 The Cauliflower

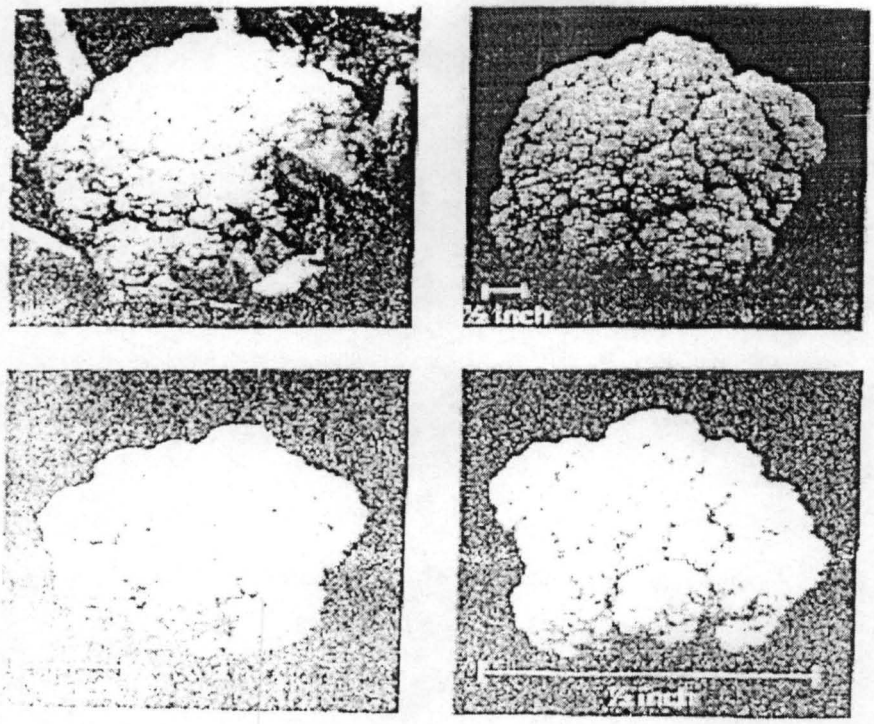


Figure 2.1: The self-similarity of an ordinary cauliflower is demonstrated by dissection and two successive enlargements (bottom). The small pieces look similar to the whole cauliflower head. [Peitegen and Saupe, 1998]

It is not a classical mathematical fractal, but here the meaning of self-similarity is readily revealed without any mathematics. The cauliflower head contains branches or parts, which when removed and compared with the whole are very much the same, only smaller. These clusters again can be decomposed into smaller clusters, which again look very similar to the whole as to the first generation branches. This self-similarity carries through for about three or four stages. After that the structures are too small for further dissection in mathematical idealization, the self-similarity property of a fractal may be continued through infinitely many stages. This leads to new concepts such as fractal dimension which are useful for natural structures that do not have this infinte detail.

2.2 Self-Similarity in the Decimal System

Although the notion of self-similarity is only 20 years old there are many historical constructions which make substantial use of its core idea. Probably the oldest and most important construction in that regard is the decimal system. It is impossible to estimate where mathematics and the natural sciences would be without this ingenious invention. We a re so used to the decimal system that we take it for granted, however, it evolved after a long scientific and cultural struggle and it is very closely related to the material from which fractals are made. It is also the prerequisite of the metric (measuring) system (for length, area, volume, weights, etc). Let us look at a metric stick, which carries markers for decimeters (ten make a metre), and millimeters (ten make a centimeter), a thousand make a metre). In a sense, a decimetre together with its markers looks like a metre with its markers, however, scaled down by a factor of 10. This is not an accident. It is in strict correspondence with the decimal system. When we say 357mm, for example, we mean 3 decimetres, 5 centimetres, and 7 millimeters. On other words, the position of the figure determines their place value, exactly as in the decimal number system. One metre has a thousand millimeters to it and when we have to locate position 357, we go to the 3 decimetre-tick mark, from there to 5 centimetre tick mark, and from there to the 7 millimetre tick mark. Most people take this elegant process for granted but somebody who has to convert miles, yards and inches can really appreciate the beauty of this system.

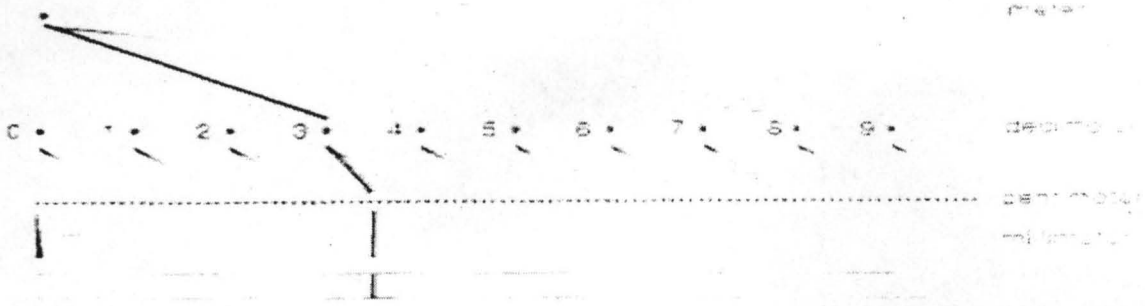


Fig 2.2 The branches of the decimal tree leading to 357 are highlighted

Actually finding a position on the metre stick corresponds to a walk on the branches of tree, the decimal number tree. The structure of the tree expresses the self-similarity of the decimal system very strongly.

2.3 The Cantor Set

Cantor (1845-1918) was a German mathematician at the University of Halle where he carried out his fundamental work in the foundations of mathematics, which we now call set theory. The Cantor set was first published in 1883 and emerged as an example of certain exceptional sets. It is probably fair to say that in the zoo of mathematical monsters — or early fractals - the Cantor set was by far the most important, though it is less visually appealing and more distant to an immediate natural interpretation than in some of the others. It is now understood that the Cantor set plays a role in many branches of mathematics, and in fact, in a very deep sense in chaotic dynamical system, and in somehow hidden as the essential skeleton or model behind many other fractals (for example Julia sets).

The basic cantor set is an infinite set points in the unit interval $[0,1]$. That is, it can be interpreted as a set of certain numbers, as for example, $0,1,\frac{1}{3},\frac{2}{3},\frac{1}{9},\frac{2}{9},\frac{7}{9},\frac{8}{9},\frac{1}{27},\frac{2}{27},\ldots$ plotting this and all other points (assuming we could know where they were) would not make much of a picture at all.

Rather than plotting just points, we plot horizontal lines whose base points are exactly at all the difficult point belonging to the Cantor set. By so doing we are able to see the distribution of these points better.

2.3.1 Construction of the Cantor Set

Start with the interval $[0,1]$. Now take away the open interval $\left[\frac{1}{3},\frac{2}{3}\right]$ i.e. remove the middle third from $[0,1]$but not the numbers $\frac{1}{3}$ and $\frac{2}{3}$. This leaves two intervals $\left[0,\frac{1}{3}\right]$ and $\left[\frac{2}{3},1\right]$ of length $\frac{1}{3}$each and complete a basic construction step. Now we look at the remaining interval$\left[0,\frac{1}{3}\right]$ and $\left[\frac{2}{3},1\right]$and remove their middle thirds which yields intervals of length$\frac{1}{9}$. Continue on in this way. In other words, there is a feedback process in which a sequence of (closed) intervals in generated - one after the first step, two after the second, four after the third step, eight after the fourth step, etc. (i.e2^n intervals of length $\frac{1}{3^n}$ after the n^{th} step.

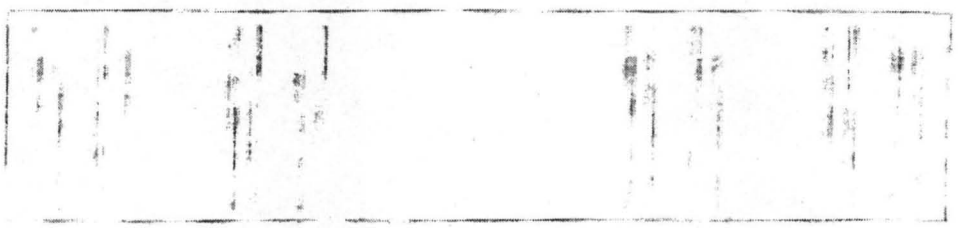


Figure 2.3: The Cantor set represented by vertical lines whose bas points are exactly at all the different points belonging to the set.

Figure 2.4: Some initial steps of the construction

Therefore, the Cantor set is the set of points which remain if we carry out the mentioned steps infinitely often. How is infinitely often explained? A point x, is in the Cantor set if we can guarantee that no matter how often we carry out the removal process, the point x will not be taken out. Obviously $0,1,\frac{1}{3},\frac{2}{3},\frac{1}{9},\frac{2}{9},\frac{7}{9},\frac{8}{9},\frac{1}{27},\frac{2}{27},\ldots$ are examples of such points because they are the end points of the interval which are created in the steps; and therefore must remain All these points have one thing in common, namely, they are related to the power of 3 - or rather, to $\frac{1}{3}$. Triadic numbers are numbers which are represented with respect to base 3.

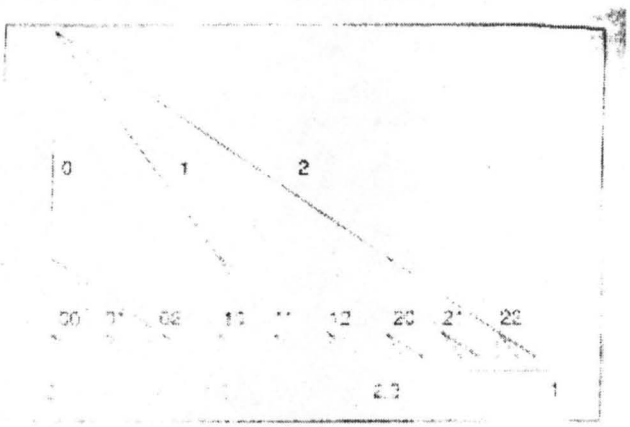


Figure 2.5: A three-branch tree visualizes the triadic expansion of numbers on the unit interval.

A three-branched tree visualizes the triadic expansion of numbers from the limit interval. The first main branch covers all numbers between 0

and $\frac{1}{3}$. Following down the branches all the way to the interval and keeping note of the labels 0, 1 and 2 for choosing the left, middle or right branches will produce a triadic expansion of the number in the interval which is approached in this process.

Now we can completely describe the Cantor set by representing the numbers from [0,1] in their triadic expansion i.e. we switch to the expansion of *x* with respect to the base 3, as in equation 2.1

$$x = a_1 \cdot 3^{-1} + a_2 \cdot 3^{-2} + a_3 \cdot 3^{-3} + a_4 \cdot 3^{-4} + \ldots \qquad (2.1)$$

Thus here, the $x = a_1, a_2, a_3, \ldots$ are numbers from $\{0,1,2,\}$. Let us write some points of Cantor set as triadic numbers $\frac{1}{3} = 0.1$ in the triadic system, and $\frac{2}{9} = 0.02$. In general, we can characterize any point of the Cantor set in the following way.

"The Cantor set C is the set of points in [0,l] for which there is a triadic expansion that does not contain the digit 1".

In the above examples, $\frac{2}{3}$ and $\frac{2}{9}$ are points in the Cantor set according to this statement, since their triadic expansion 0.2 and 0.02 do not contain the digit '1'. However the two examples $\frac{1}{3}$ and $\frac{1}{9}$ seem to contradict this rule, their expansions 0.1 and 0.01 clearly show digit '1'.

2.4 The Sierpinski Gasket and Carpet

Our next classical fractal is about 40 years younger than the cantor set. It was introduced by the great Polish mathematician Waclaw

Sierpinski (1882-1969) in 1916. Sierpinski was a professor at Lvov, a city founded in 1256. The basic geometric construction of the Sierpinski gasket goes as follows. We begin with a triangle in the plane and then apply a repetitive scheme of operations to it. Pick the midpoints of its three sides. Together with the old vertices of the original triangle, these midpoints define for congruent triangles of which we drop the centre one. This completes the basic construction step.

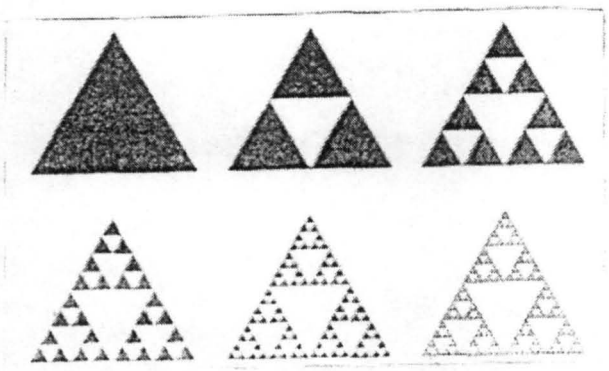


Figure 2.6: The basic construction steps of the Sierpinski gasket

2.4.1 The Sierpinski Carpet

Sierpinski has added another object to the gallery of classical fractals, the Sierpinski carpet. We begin with a square in the plane. Subdivide it into nine little congruent squares of which we drop the centre one and so on. The resulting object which remains if one carries out this process infinitely often can be seen as a generalization of the centre set (in 2 dimensions). Indeed if we look at the intersection of a line which goes through the centre we observe precisely the construction of the Cantor set.

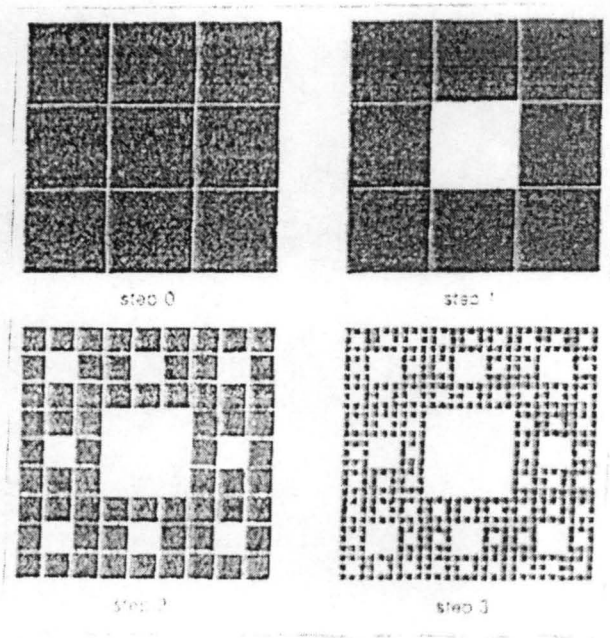


Figure 2.7: The basic construction steps of the Sierpinski carpet

2.5 Koch Curves

Helve von Koch was a Swedish mathematician who in 1904, introduced what is now called the Koch curve. Fitting together three suitably rotated copies of the Koch curve produces a figure which for obvious reasons is called a Koch curve.

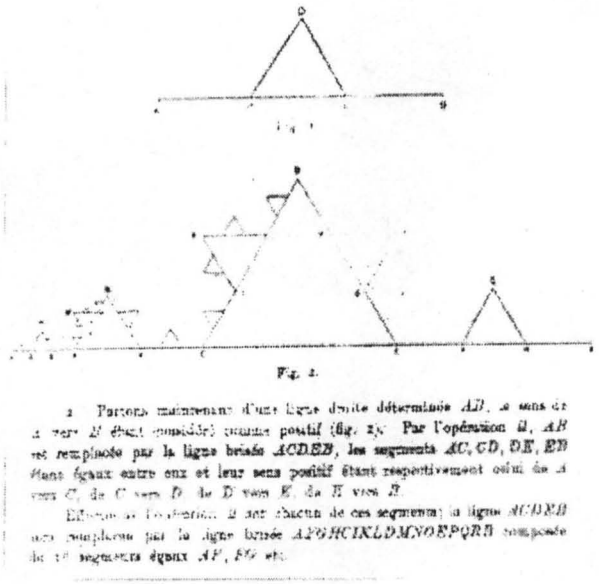


Figure 2.8: Excerpts from von Koch's original 1906 article

Little is known about von Koch, whose mathematical contributions were certainly not on the same category as those of the stars like Cantor, Peano, Hilbert, Sierpinski or Hausdorff. Here the Koch construction must have its way because it leads to many interesting generalizations. The koch curve is a difficult to understand as the cantor set or the Sierpinski gasket. However, the problems with it are of a difficult nature. First of all as the name implies, it is a curve, but this is not immediately clear from the construction. Secondly, the curve contains no straight lines or segments which are smooth in the sense that we could see them as a carefully bent line. Rather, this curve has much of the complexity, folds within folds within folds and so on.

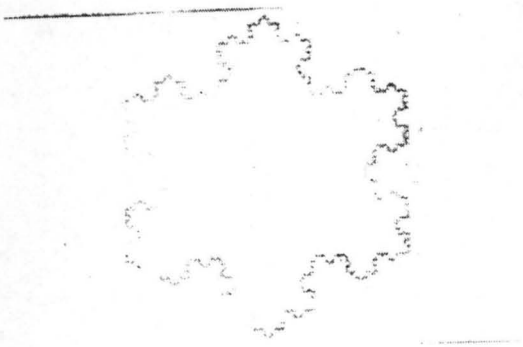


Figure 2.9: The outline of the Koch snowflake (also known as the Koch Island)

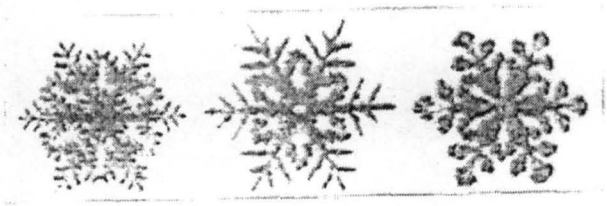


Figure 2.10: Some natural lakes

2.5.1 Construction of the Koch curve

We begin with a straight line. Partition it into three equal parts, then replace the middle third by an equilateral triangle and take away its base. This completes the basic construction step. A reduction of this frame, made in four parts, will be reused in the following stages. It is called the generator. Thus, we now repeat each of the resulting line segments, partitioning them, into three equal parts, and so on.

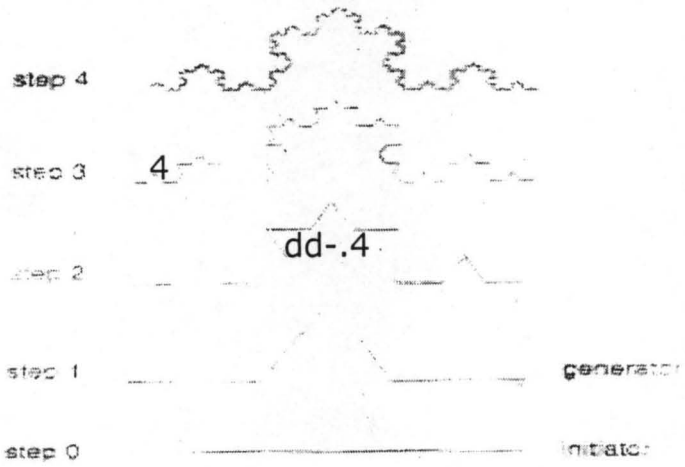


Figure 2.11: Stages in the construction of the Koch curve

Actually, Koch wanted to provide another example for discovery first made by the German mathematician Karl Weistraß, who in 1872 had precipitated a minor crisis in mathematics. He had described a curve that could not be differentiated, i.e. a curve which does not admit a tangent at any of its points. The ability to differentiate (i.e. calculate the shape of a curve from point to point) invented by Newton and Leibniz. If a curve has a corner, then there is a problem. There is no way to fit a unique tangent. The Koch curve is an example of a curve which in a sense is made out of corners everywhere i.e. there is no way to fit a tangent to any of its points.

2.5.2 Generalized Koch Constructions

It is almost obvious how one can generalize the Koch construction to obtain a whole universe of self-similar structures. Such a Koch construction is defined by an initiator, which may be a collection of line segments, and a generator, which is a polygonal line composed of a number of connected line segments. Beginning with the initiator, one replaces each line segment by a properly scaled down copy of the generator curve. Here it is necessary to carefully match end points of the line segment and the generator. This procedure is repeated ad infinitum, of course, one stops, as soon as the length of the largest line segment in the graph is below the graphic device. Whether or not the Koch curve yields a conveying sequence of images or even curves depends on the choice of the initiator and generator.

2.5.3 Length of the Koch Curve

At each stage in the original Koch curve, we obtain a curve. At the first, we are left with a curve which is made up of 4 line segments of the same length, after the second step we have 4x4, and then 4 x 4 x 4 lie segments after the third step, and so on. Ωf the original line had length L, then after the first step a line has length $L \times \frac{1}{3}$ after the second step, we have $L \times \frac{1}{3^2}$, then $L \times \frac{1}{3^3}$, and so on. Since each of the steps produces a curve of line segments, there is no problem in measuring their respective lengths. After the first step it is $4 \times L \times \frac{1}{3}$, then $4^2 \times L \times \frac{1}{3^2}$, and so on. After the k^{th} step, it is $4^k \times L \times \frac{1}{3^k}$. We observe that from step to step the length of the curve grows by a factor of $\frac{4}{3}$. This presents some problems. First of all, the Koch curve is the object which one obtains if one repeats the construction steps infinitely often. But what does this mean? Next even if we could answer this question, why is it a curve which comes out? Ωr why is it that the curves in each step do not intersect themselves.

Figure 2.12: Construction process of the Koch curve

In figure (2.12), we see two curves which we can hardly distinguish. But they are different. The top one shows the result of the construction after 5 steps, while the other curve shows the result after 20 steps. In other words, since the length of the individual line segment is $\frac{1}{3^k}$, where k is the number of steps, it is clear that any of the changes in the construction are soon below visibility unless one works under a microscope. Thus, for practical purposes, one is tempted to be satisfied with a display of something like the 10^{th} step, or whatever is appropriate to fool the eye. But such an object is not the Koch curve. It would have a finite length and would still show its straight line construction segments under sufficient magnification.

2.6 Space-Filling Curves

Talking about dimensions in an intuitive way, we perceive lines to be typical for one-dimensional objects and planes as typical for two-dimensional objects. In 1890 Giuseppe Peano (1858-1932) and immediately after that in 1891, David Hilbert (1862-1943), discussed curves which live in a plane and which dramatically demonstrate that our naïve idea about curves is very limited. They discussed curves which "fill" a plane, i.e. given some patch of the plane, there is a curve which meets every point in that patch.

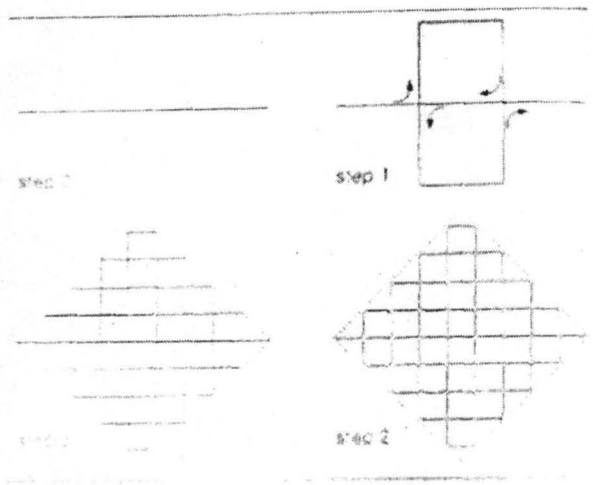
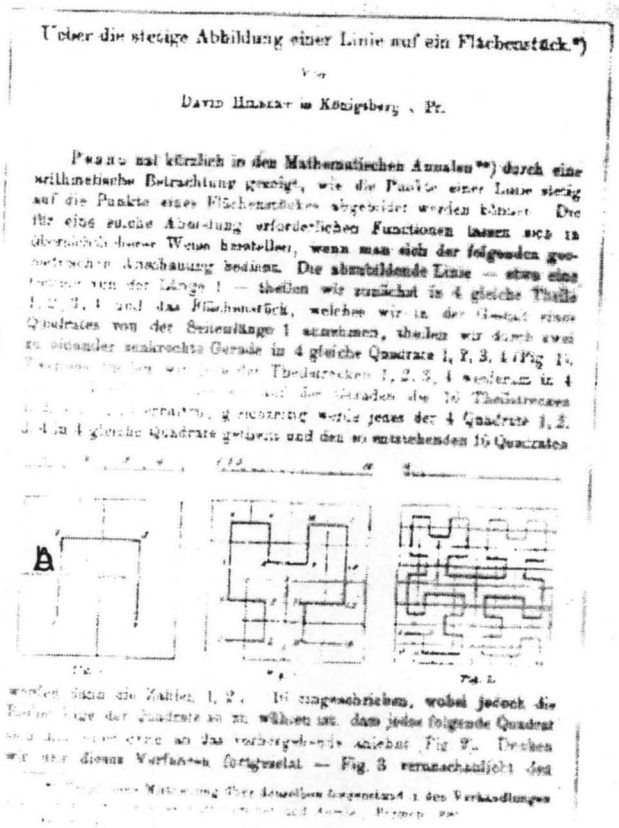


Figure 2.13: Construction of a plane-filling curve with initiator and generator

Figure (2.13) indicates that first steps of the iterative construction of Peano's original curve. In nature the organization of space-filling structures is one of the fundamental building blocks of living beings. An organ must be supplied with necessary supporting substances such as water and oxygen. In many cases these substances will be transported through vessel system that must reach every point in the volume of the organ. For example, the kidney houses three interwoven tree-like vessel systems, the arterial, the veins, and the urinary systems. Fractals solve the problem of how to organize such a complicated structure in an efficient way. Ωf course, this was not what Peano and Hilbert were interested in over 100 years ago. It is only now after Mandelbrot's work, that the omnipresence of fractals has become apparent.

The Peano curve is obtained by anther version of the Koch curve construction. We begin with a single line segment, the initiator, and then substitute the segment as shown in figure 2.13. Apparently, the generator has two points of self-intersection. More precisely, the curve touches itself at two points. Ωbserve that this generator curve fits nicely into a square whose points will be reached by the Peano curve.

Take each straight line piece of the curve in the first stage and replace it by the properly scaled down generator. Ωbviously the scaling factor is 3. This constitutes stage 2. There are a total of 32 self-intersecting points in the curve. Now we repeat, i.e. in each step, line segments are scaled down by a factor of 3. Thus, in the k^{th} step, a line segment has a length $\frac{1}{3^k}$, which is a very rapidly declining number. Since each line segment is replaced nine line segments of one-third



Ueber die stetige Abbildung einer Linie auf ein Flächenstück.*)

Von

David Hilbert in Königsberg i. Pr.

Fig. 1

Fig. 2

Figure 2.14: First page of Hilbert's original; paper 1890

The length of the previous line segments, we can easily calculate the length of the curves in each step. Assume that the length of the original line segment constituting the initiator was 1, then we obtain in stage $1:9\times\frac{1}{3}=3$, and stage $21:9\times9\times\frac{1}{3^2}=9$. Expressed as a general

rule, in each step of the construction, the resulting curve increases in length by a factor of 3. In stage k, the length is 3^k..

2.7 Self-Similarity

The Peano curve construction, though as easy or as difficult, as the construction of the Koch curve, bears within it several difficulties which did not occur or were hidden in the latter construction.

400 David Hilbert. Stetige Abbildung einer Linie auf ein Flächenstück.

nächsten Schritt —, so ist leicht ersichtlich, wie man einem jeden gegebenen Punkte der Geraden einen einzigen bestimmten Punkt der Quadrates zuordnen kann. Man hat nur nöthig, diejenigen Theilstrecken der Geraden zu bestimmen, auf welche der gegebene Punkt fällt. Die mit den ersteren Zahlen bezeichneten Quadrate liegen nothwendig in einander und schliessen in der Grenze einen bestimmten Punkt der Flächenstückes ein. Dieser wird dem gegebenen Punkte zugeordnete Punkt. *Die so gefundene Abbildung* ist *vieldeutig und stetig* und umgekehrt *einem jeden Punkte der Quadrate entsprechen ein, zwei oder mehr Punkte der Linie.* Es erscheint überdies bemerkenswerth, dass durch geeignete Abänderung der Theilungen in dem Quadrate sich leicht *eine eindeutige und stetige Abbildung finden lässt, deren Umkehrung eine nirgends mehr als zweideutige ist.*

Die oben gefundenen stetigen Functionen sind zugleich einfache Beispiele für überall stetige und nirgends differentiirbare Functionen.

Die geometrische Bedeutung der erörterten Abbildung ist folgende: *Es kann sich ein Punkt stets derart bewegen, dass er jeder mit einer endlichen Zeit stetig wachsenden Fläche trifft.* Auch kann man — ebenfalls durch geeignete Abänderung der Theilungen im Quadrate — zugleich beweisen, *dass in unendlich vielen überall dichtvertheilten Punkten des Quadrates eine bestimmte Flächenrichtung* sowohl nach rechts wie auch nach rückwärts existirt.

Wie die analytische Darstellung der stetigen Functionen angestrebt, so folgt aus ihrer Möglichkeit nach einem allgemeinen von K. Weierstrass bewiesenen Satze sofort, dass diese Functionen auch in unendliche nach ganzen rationalen Functionen fortschreitende Reihen entwickelt lassen, welche im ganzen Intervall absolut und gleichmässig convergiren.

Königsberg i. Pr., 6. März 1891.

*) Vergl. Sitzungsberichte der Akademie der Wissenschaften zu Berlin, 9. Juli 1886.

Figure 2.75: *Second page of Hilbert 's paper 1890*

For example, take the intrinsic concept of self-similarity. For the construction of the Koch curve, it seemed that we could say that the final curve has similarity with each of the preceding steps. Ωf you look at the Peano curve in the same intuitive way, each of the steps has similarity with the preceding steps; but if you look at the final curve, essentially we see a filled out square which does not look at all similar to the early steps of the construction. In other words, either the Peano

curve is not self-similar, or we have to be much more careful in describing what self-similar means.

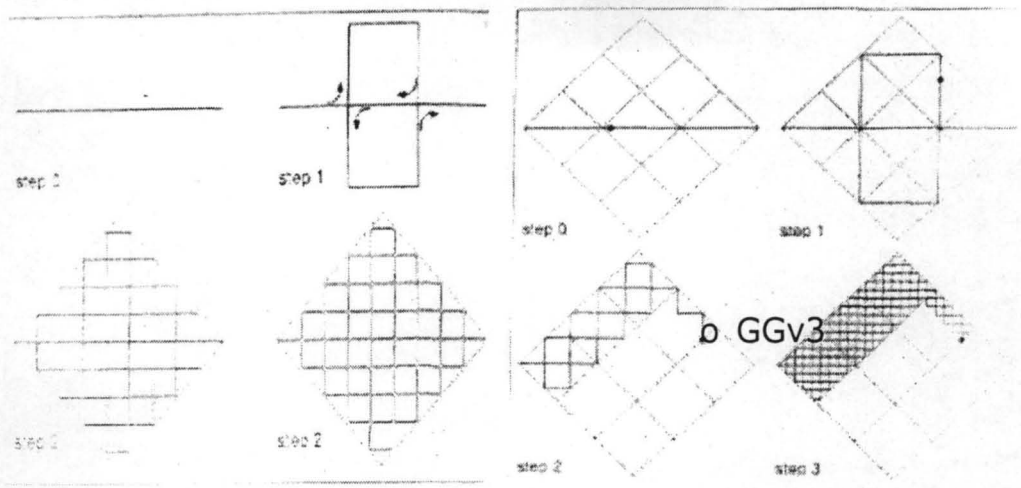


Figure 2.16: Peano curve of four different stages traced out

In conclusion, we have shown that a notion of self-similarity in a strict sense requires a discussion of the object which finally results from the construction of the underlying feedback system. Ωne must carefully distinguish between a finite construction stage and the fractal itself. But if that is so, then how can we discuss the forms and patterns we see in nature, as for example the cauliflower, from that point of view? The cauliflower shows the same forms - clusters are composed of smaller clusters of essentially the same form - over a range of several, say five or six, magnification scales. This suggests that the cauliflower should be discussed in the framework of fractal geometry very much like our plates are suitably discussed within the framework of Euclidean geometry. But a plant is not a perfect sphere and the cauliflower is not perfectly self-similar. First, there are imperfections in self-similarity: a little cluster is not an exact scaled version of a larger cluster. But more importantly, the range of magnification within which we see similar forms is finite. Therefore, fractals can only be used as

models for natural shapes, and one must be always be aware of the limitations.

2.8 Fractals and the Problem of Dimension

The invention of space-filling curves was a major event in the development of the concept of dimension. They questioned the intuitive perception of curves as one- dimensional objects, because they filled the plane (i.e. an object which is intuitively perceived as two-dimensional). Talking about fractals, we usually think of the fractal dimension, Hausdorff dimension or box counting dimension whose original concepts reside in the early development of topology. Topology is a branch of mathematics which has essentially been developed in the 20^{th} century. It deals with the question of form and shape from a qualitative point of view. Two of the basic notions are dimension and homeomorphism. Topology deals with the shapes that can be pulled and distorted in a space that behaves like a rubber.

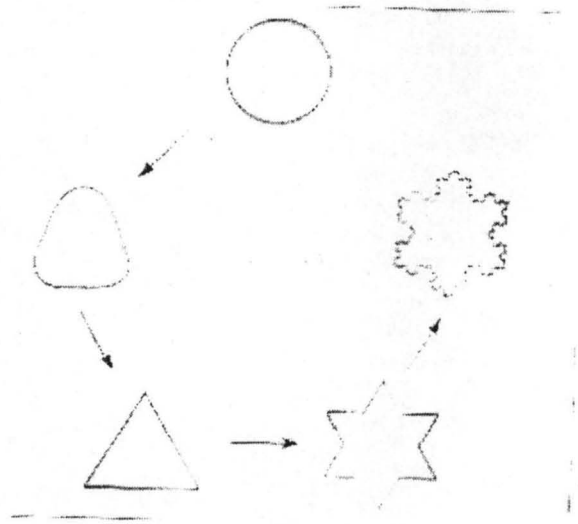


Figure 2.17: A circle can be continuously deformed into a triangle

In topology, straight lines can be bent into curves and circles can be pinched into triangles or pulled out as squares. For example, from the

point of view topology, a straight line and the Koch curve cannot be distinguished. Ωr the coast of a Koch island is the same as a circle. Ωr a plain sheet is equivalent to one which is infinitely crumpled. However, not everything is topologically changeable. Intersection of lines, for example remain intersections. Intersection is invariant; it cannot be destroyed nor can new ones be born, no matter how much the lines are stretched and twisted. The number of holes in an object is also topologically invariant, meaning that a sphere may be transformed into the surface of a horse-shoe, but never into a doughnut. The transformations which are allowed are called homeomorphisms, and when applied, they must not change the invariant properties of the objects. Thus, a sphere and the surface of a cube are homeomorphic, but the sphere and a doughnut are not. A straight line and the Koch curve are topologically the same. Moreover, a straight line is a prototype of an object which has dimension one. Thus, if the concept of dimension is a topological notion, we could expect that the Koch has topological dimension one. This is, however, a delicate matter and it troubled mathematicians around the turn of the twentieth century. The history of the various notions f dimension involves the greatest mathematicians of that time: men like Pointcare, H. Lesbesgue, L.E.J. Brouwer, G. cantor, K. Menger, W. Harowitcz, P. Alexandroff, L. Pontragin, G. Peano, P. Urysohn, E. Cech and D. Hilbert. That history is very closely related to the creation of early fractals. Hausdoff remarked that the problem of creating the notion of dimension is a very complicated one. People had an intuitive idea about dimension: the dimension of an object, say X, is the number of independent parameters (coordinates), which are required for the unique description of its points.

Poincare's idea was inductive in nature and started with a point. A point has dimension 0. Then a line has dimension 1, because it can be split into two parts by a point (which has dimension 0) and a square has dimension 2 because it can be split into 2 parts by a line (which has dimension 1). A cube has dimension 3 because it can be split into 2 parts by a square (which has dimension 2)

In the development of topology, mathematicians looked for qualitative features which would not change when the objects were transformed properly (technically by a homeomorphism). Two objects X and Y (topological spaces) are homeomorphic if there is a homeomorphism $h: X \to Y$ (i.e. a continuous one-to-one and onto mapping that has a continuous inverse h^{-1}). The topological dimension of an object certainly should be preserved. But it turned out that there were severe difficulties in aiming at a proper and detailed notion of dimension which would behave that way. For example, in 1878, Cantor formed a transformation from the unit interval [0,l] to the unit square [0,l]x[0,l] which was one-to-one and onto. Thus it seemed that we need only one parameter for the description of the points in a square. But Cantor's transformation is not a homeomorphism. It is not continuous, i.e. it does not yield a space-filling curve. But then the plane-filling construction of Peano and later Hilbert yielded transformation *g* from the unit interval [0,1] to the unit square [0,1]x[0,1] which were even and continuous. But they were not one-to-one (i.e. there are points, say x_1 and $x_2 (x_1 \neq x_2)$ in the unit interval which are mapped to the same point square $y = g(x_1) = g(x_2)$. This questioned the question - which so far seemed to have an obvious answer - whether or not there is a one-to-one and onto transformation between $I = [0,1]$ and $I^2 = [0,1] \times [0,1]$ which is continuous in both directions. Ωr more generally, is the *m*-

dimensional one, $I^m = [0,1]^m \; n \neq m$. If there is such a transformation, mathematicians felt they were in trouble: a one-dimensional object would be homeomorphic to a two-dimensional one. Thus, the idea of topological invariance would be wrong.

Between 1890 and 1910 several proofs appeared showing that I^n and I^m are not homeomorphic *where* $n \neq m$, but the arguments were not complete. It was the Dutch mathematician Brouwer who ended the crisis in 1917 by an elegant proof which enriched the development of topology enormously. But the struggle for a suitable notion of dimension and a proof that obvious objects - like I^n - had obvious dimensions went on for two more decades. The work of the German mathematician Hausdorff (which led eventually to fractal dimension) also falls in this time span.

During the 20th century mathematicians came up with many different notions of dimension (small inductive dimension, covering dimension, homological dimension) (C. Kovatowski, 1978). Several of them are topological in nature: their value is always a natural number (or 0 for points) and does not change for topologically equivalent objects.

2.9 The Fractal Dimension

Mandelbrot (1982) offers the following tentative definition of a fractal:

"A fractal is by definition a set for which the Hausdorf-Besicovitch dimension strictly exceeds the topological dimension"

This definition requires a definition of the term *set,* Hasudorff-Besicovitch dimension (D) and topological dimension (D_T), which is always an integer. For the present purpose we find that a rather loose

definition of these terms and illustrations - using simple examples - is more useful than the more formal mathematical discussion available. In fact Mandelbrot (1986) has retracted this tentative definition and proposes instead the following:

"A fractal is a shape made of parts similar to the whole in some way"

A neat and complete characterization of fractals is still lacking (Mandelbrot, 1987). The point is that the first definition, although correct and precise, is too restrictive. It excludes many fractals that are useful in physics. A fractal looks the same whatever the scale. An example of this is the cumulus clouds. They consist of big heaps with smaller bulges that have smaller bumps with bumps on them and so on down to the smallest scale you can resolve. In fact, from a picture showing only the clouds we cannot estimate the size of the clouds without extra information. Fractals may be considered to be sets of points in space. For example, the set of points that make up a line in ordinary Euclidean space has the topological dimension $D_T = 1$ and the Hausdorff-Besicovitch dimension $D = 1$. The Euclidean dimension of space is $E = 3$. Since $D = D_T$ for the line it is not a fractal according to Mandelbrot's definition. Similarly, the set of points that form a surface in $E = 3$ space has the topological dimension D_T, and $D = 2$. Again an ordinary surface is not fractal independent of how complicated it is. Also, a ball or sphere has $D = 3$ and $D_T = 3$

The concept of a distance between points in space is central to the definition of the Hausdorff-Besicovitch dimension and therefore of the fractal dimension *D.* How do we measure the 'size' of a set of points in

space? A simple way to measure the length of an object is to divide space into smaller cubes of sides *8* as illustrated in figure

(2.18) space

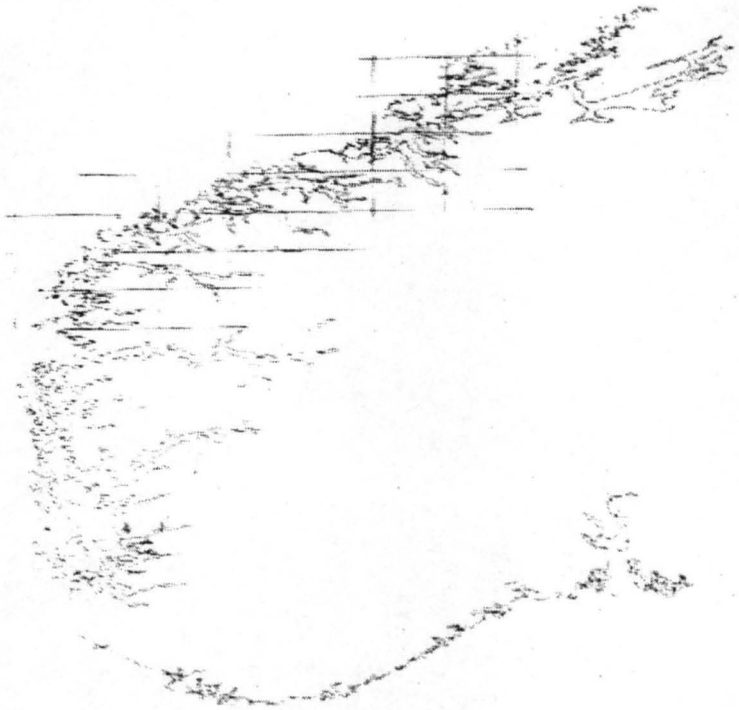


Figure 2.17: The coast of Southern part of Norway

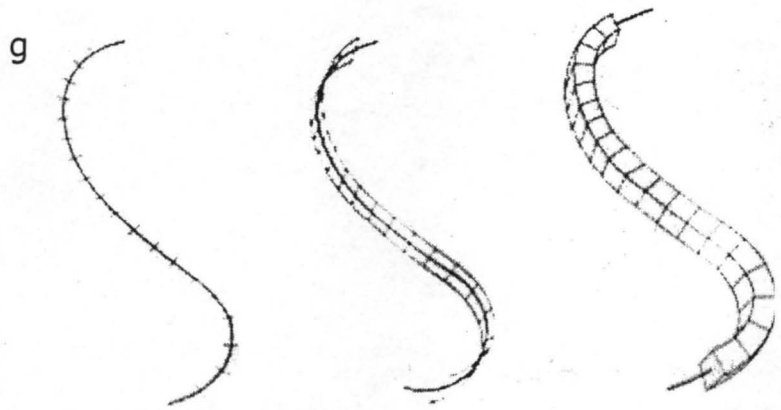


Figure: 2.18: Measuring the size of curves

We might use small spheres of diameter δ instead. If we centre a small sphere on a point in the set then all point that are at a distance

$r < \frac{1}{2}\delta$ from the point at the centre are covered by the sphere. By counting the number of spheres needed to cover the set of points we obtain a measure of the size of the set. A curve can be measured by finding the number $N(\delta)$ of line segments of length δ needed to cover the line. For an ordinary curve we have $N(\delta) = {}^{L_0}\!/_{\delta}$. The length of the curve is given by $L = N(\delta)\cdot\delta \xrightarrow[\delta\to 0]{} L_0\delta^0$ In the limit $\delta \to 0$, the measure L becomes asymptotically equal to the length of the curve and is independent *of* δ. We may choose to associate an area with the set points defining a curve by giving the number of disks or squares needed to cover the curve. This number of squares is again $N(\delta)$, and each square has an area *of* δ^2. The associated area is therefore given by

$$A = N(\delta)\cdot\delta^2 \xrightarrow[\delta\to 0]{} L_0\delta^1$$

Similarly, we may associate a volume, V, with the line as follows

$$V = N(\delta)\cdot\delta^3 \xrightarrow[\delta\to 0]{} L_0\delta^2$$

For ordinary curves both A and V tend to zero as δ vanishes, and the only interesting measure is the length of the curve. We find that for an ordinary surface the number of squares needed to tile it is $N(\delta) = {}^{A}\!/_{\delta^2}$ in the limit of vanishing δ, where A_0 is the area of the surface, we may associate volume with the surface by forming the sum of the volumes of the cube needed to cover the surface.

$$V = N(\delta)\cdot\delta^3 \xrightarrow[\delta\to 0]{} L_0\delta^2$$

What if we associate a length with a surface:

$L = N(\delta)\cdot\delta \xrightarrow[\delta\to 0]{} L_0\delta^{-1}$ which diverges for $\delta \to 0$. This result is measurable since it is impossible to cover a surface with a finite number of line segments. We conclude that the only useful measure of

a set of points defined by a surface in three dimensional space is the area.

We shall see that one may easily define sets of points that are curves which twist so badly that their length is infinite and in fact three curves (Peano curves) that fill the space. Also there are surfaces that fold so wildly that they fill space. In order to discuss such strange sets of points it is useful to generalize the measure of the size first discussed. So far in order to give a measure of the size of a set of points, S, is space we take a test function $h(\delta)=\gamma(\delta)\delta^d$ - a line, square, disk, ball or cube - and cover the set to form the measure $M_d=\sum h(\delta)$ for lines squares and cubes the geometrical factor $\gamma(d)=1$ (Feder, 1988). We have $\gamma=\frac{\pi}{4}$ for disks, $\gamma=\frac{\pi}{6}$ for spheres. In general, we find that, as $\delta\to 0$, the measure $M_d=\sum h(\delta)$ is either zero or infinite depending on the choice of d - the dimension of the measure. The Hausdorff-Besicovitch dimension D of the set S is the critical dimension for which the measure M_d changes from zero to infinity:

$$M_d=\sum\gamma(\delta)\delta^d=\gamma(d)N(\delta)^d\xrightarrow[\delta\to 0]{}\begin{cases}0, d<D\\ \infty, d>D\end{cases} \qquad *$$

M_d is called the d-measure of the set. The value of M_d for $d=D$ is often finite but may be zero or infinite; it is the position of the jump M_d as a function of d that is important. This definition defines the Hausdorff- Besicovitch dimension D as a local property in the sense that it measures properties of sets of points in the limit of vanishing diameter or size δ of the test function and to cover the set. It also follows that the fractal dimension D may depend on position. Actually, there are several fine points that have to be considered. In particular,

the definition of the Hausdorff-Besicovitch dimension allows for a covering of the set by "balls" that are not of the same size, but have diameters less than δ . The δ -measure is then the infimum: roughly the minimal value obtainable in all possible coverings. The case in which the Hausdorff-Besicovitch dimension is non-integer is said to be fractal. The definition (*) of the fractal dimension can be sued in practice. Consider again the coastline shown in Fig (2.18) which have been covered with a set of squares with edge length δ, with the unit of length taken to equal the edge of the frame. Counting the number of squares needed to cover the coastline given the number $N(\delta)$. Now we may proceed as implied by equation (*) and calculate M_d, or we may simply go ahead and find $N(\delta)$for smaller values *of* δ . Since it follows from equation (*), that asymptotically in the limit of small δ,

$$N(\delta) \sim \frac{1}{\delta^D} \qquad (*1)$$

We may determine the fractal dimension of the coastline by finding the slope of $\ln N(\delta)$ plotted as a function *of* $\ln \delta$ The resulting plot for the coastline shown in Fig 2.17 is presented in Fig 2.19

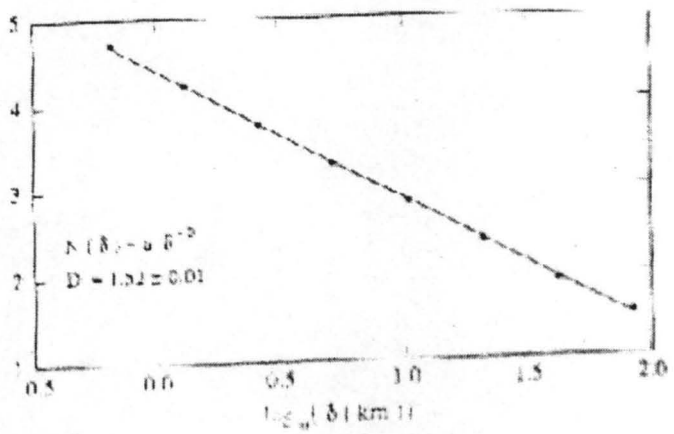


Figure 2.19: The number of 'boxes' needed to cover the coastline in figure in figure 2.17

We find approximately *that* $D \cong 1.5$. The dimension $N(\delta)$, determined for equation (*1) by counting the number of boxes needed to cover the set as a function of the box size, is now called the box-counting dimension or box-dimension. More generally, therefore, a fractal dimension, sometimes used to refer to what is commonly called the capacity dimension, is roughly speaking the exponent *D* in the expression

$n(\varepsilon) = \varepsilon^{-D}$ where $n(\varepsilon)$ is the minimum number of open sets of diameter *s* needed to cover the set.

2.10 Mathematical Foundations in Image Processing

Image processing is divided into three parts, corresponding to as many goals. The first one derives from the discrete nature of images and the search for their minimal representation in terms of digital memory. This discipline is called image compression. The second goal is the restoration of a better version of an image, given a generation model with noise an blur and other perturbations. The third goal is analysis, which means in Greek "breaking into parts". In analysis, all spurious details disappear leaving only the main structures. The aim is not denoising or compression but to construct an invariant code putting in evidence the main parts and permitting a fast recognition in a large database of shapes.

2.11 The Heat Equation

The heat equation arises naturally in the image generation process. Indeed, according to Shannon's theory, an image can be correctly represented as a discrete set of values, the "samples", only if it has been previously smoothed. We start with U_0 the original image. Then a

blur kernel K is applied, i.e. we convolve U_0 with K to obtain a new image $K * U_0$. A subsequent subsampling is thereafter possible, where the distance between samples is related to the bandwidth of the blur kernel by the Nyquist rule. Stability of the image representation is maintained. This simple remark that smoothing is necessary part of image formation, leads to our first PDEs. The difference between the original and the blurred image is roughly proportional to its Laplacian (Gabor, 1960). In order to formalize this remark, we have to notice that K is spatially concentrated and that we may introduce a scale parameter for K, namely

$$K_n(x) = \frac{1}{n} K\left(\frac{1}{h^{1/2}}\right)$$

Then,

$\frac{U_0 * K_n(x) - U_0(x)}{h} \to \Delta U_0(x)$ so that when h gets smaller, the blur process looks more and more like the heat equation $\frac{\partial U}{\partial t} = \Delta U, U(0) = U_0$ Conversely, Gabor deduced that we can, in some extent, deblur an image by reversing time in the heat equation

$$\frac{\partial U}{\partial t} = -\Delta U, U(0) = U_{restored}$$

Numerically, this amounts to subtracting its Laplacian from the observed image:

$$U_{restored} = U_{observed} - h\Delta U_{observed}$$

This equation can be repeated several times with some small values of h until it blows up. The reversed heat equation is extremely ill-posed. All the same, Gabor' s method is efficient and can be applied with some success to most digital images obtained from an optical device. We therefore set two directions. Ωne is to improve, to stabilize the reverse heat equation. This can be done through nonlinear models.

The second direction is to go on with the heat equation: we can numerically simulate a further blurring of the image. This leads to the wavelet theory and its applications to optimal multiscale sampling and compression. Second, iterated linear and nonlinear smoothing (i.e. nonlinear PDEs) will relevant to image analysis. We can improve the time-reverse ht equation. A pseudoinverse is used where the propagation term Du - is tuned by the sign of the Laplacian

$$\frac{\partial U}{\partial t} = -sign\Delta U|Du|$$

This equation is called a "shock filter". This equation propagates, with constant speed, the level lines of the image in the same direction as the reverse heat equation would do. It therefore enhances the image. The equation more or less equivalent to a nonlinear filter due to Kramer filter can be interpreted as a partial differential equation, by the same kind of heuristic arguments which Gabor developed to derive the heat equation. The equation is

$$\frac{\partial U}{\partial t} = -signD^2u(Du, Du)|Du|$$

thus, the Laplacian is replaced by a directional second derivative of the image $D^2u(Du, Du)$. Kramer's version yields a slightly better version of the shock filter. The third deblurring method is to the best knowledge, the best version. It poses the deblurring problem as an inverse problem. Given the observed image U_0 , we try to find a restored version U such that $K * U$ is as close as possible to U_0 and the oscillations of U is nonetheless bounded:

$$U_{restored} = Arg\min\left(\int|Du| + \lambda(K * U - U_0)^2\right)$$

The parameter λ tunes the oscillations we allow for the restored version. If λ is large, the restored version will satisfy accurately the

equation $K * U = U_0$, but may be very oscillatory. If instead λ is small, we get a smooth but inaccurate solution. This parameter can be computed in principle as a Lagrange multiplier. The obtained restoration can be remarkable.

In image analysis, the heat equation has had s very different use: Marr, Hildreth, Canny, Witkin, Koendrink proposed in the eighties to analyze an image by applying the heat equation. The heat equation is easily proved to be the only good candidate if the image analysis has to be linear. The question arises of what derivations should be computed in an image. The early research in computer vision proposed "edge detection" as a main tool: it is assumed that the apparent contours of the objects and also the boundaries and facets of objects, result in step discontinuities in the image, while inside those boundaries, the image oscillates only mildly. The apparent contour points or "edge points" will be computed as points where the gradient is in some sense largest. Two ways to so: Hildreth and Marr proposed the points where A (7 crosses zero. A significant improvement was done by Canny, who proposed to compute the points where Du is maximal on the gradient lines. Such points satisfy $D^2u(Du, Du) = 0$

The heat equation under sound invariant requirements is the only good linear filter. But there are nonlinear ways to smooth an image. The first one was proposed by Perona and Malik. The idea is roughly to smooth out what has to be smoothed, the irrelevant, homogenous regions and enhance instead the boundaries. Thus, the diffusion should look like the heat equation when $|Du|$ is small and an inverse heat equation should instead be applied when $|Du|$ is large. Here is the

equation in divergence form $\frac{\partial U}{\partial t} = div\left(g\left(|Du|^2\right)\right)$"where $g(s) = \frac{1}{1+\lambda s^2}$ decreases when s increases. It is easily checked that we have diffusion equation when $|Du| \le \lambda$ and an inverse diffusion equation when $|Du| > \lambda$. In order to do so, we rewrite the equation in the following way. We consider the second derivative of U in the derivative of Du

$$U_{\eta\eta} = D^2u\left(\frac{Du}{|Du|}, \frac{Du}{|Du|}\right)$$

and the second derivative in the orthogonal direction,

$$U_{\xi\xi} = D^2u\left(\frac{Du\perp}{|Du|}, \frac{Du^{\perp}}{|Du|}\right)$$

where

$Du = (U_x, U_y)$ and $Du^{\perp} = (-U_y, U_x)$. The Laplacian can be rewritten in the intrinsic coordinates (ξ, η) as $\Delta U = U_{\xi\xi} + U_{\eta\eta}$. The Perona-Malik equation rewrites as

$$\frac{\partial U}{\partial t} = \frac{U_{\xi\xi}}{1+\lambda^2|Du|^2} + \frac{\left(1-\lambda^2|Du|^2\right)U_{\eta\eta}}{1+\lambda^2|Du|^2}$$

So the first term always appears as a one-dimensional heat equation in the directional heat equation, or reverse heat equation in the direction of the gradient. So this model mixes the heat equation and the reverse heat equation.

The Perona-Malik model attempts to put in a single operator two very different goals which we already mentioned, namely restoration and analysis. But it comes at a cost: the model contains a "contrast threshold" which can only be fixed manually. Mathematical existence

and uniqueness are not guaranteed, despite some attempts by Kichenassamy and Weickert. We summarize the involved parameters: we need to fix both λ and the smoothing scale$(s), t$ and the threshold on the gradient in Canny's edge detector as well. This means that we have a two parameter game: How this will be dealt with in automatic image analysis has no general answer for the time being.

2.12 Contributions to Fractals and Image Processing

Contributions in image processing have come through four main approaches. Two of which rely on geometric space techniques and the other two abstract space techniques.

Under geometric space techniques we have:

1. Linear Models comprising of
   (a) Convolution
   (b) Fourier analysis
   (c) Tomography
   (d) Kriging, splines

2. Nonlinear Models
   (a) Morphological filtering
   (b) Granulometry
   (c) Random sets
   (d) Watersheds

Under abstract spaces, we have

1. Statistical models using techniques such as
   (a) Multivariate analysis
   (b) Neuronalnets
   (c) Sterelogy

2. Syntactical Models such as
   (a) Semantic approaches
   (b) Grammars
   (c) Neuronal nets

2.12.1 Linear Models

Linear models have the vector space as a working structure i.e. a set of vectors *V* such that

(i) *V* is a commutative group

(ii) *K* is a field

and there exists can external law of multiplication between scalars and vectors.

The main vehicle for linear models in image processing is the linear heat equation, first suggested by Gabor, 1960 and he came up with the following smoothing equation

$$\lim_{h\to 0}\frac{U_0 * K_h(x) - U_0(x)}{h} = U_0(x)$$

and the deblurring equation

$$\frac{\partial U}{\partial t} = -\Delta U, U(0) = U_{restored}$$

Rudin and Ωsher in 1987 and 1992 proposed a pseudoinverse equation where the propagation term *(Du)* is tuned by the sign of the Laplacian

$$\frac{\partial U}{\partial t} = -sign\Delta U|Du|$$

the equation is called a shock filter.

Nonlinear Models

Perona and Malik in 1987 developed the following which is nonlinear for which no mathematical existence and uniqueness have been guaranteed despite efforts by Kichenassamy and Weikert. It is given as

$$\frac{\partial U}{\partial t} = \frac{U_{\xi\xi}}{1+\lambda^2|Du|^2} + \frac{\left(1-\lambda^2|Du|^2\right)U_{\eta\eta}}{1+\lambda^2|Du|^2}$$

a lot of nonlinear image analysis models have been tried in the last 15 years. Actually almost all possible nonlinear parabolic equations have been proposed, some given below.

The Rudin-Ωsher-Fatemi's Model

This consists, for the smoothing term, of minimizing the total variation of U. The gradient descent for $\int|Du|$ writes

$$\frac{\partial U}{\partial t} = \left(\frac{Du}{|Du|}\right) = \frac{1}{|Du|}U_{\xi\xi}$$

written in this way, the method appears as diffusion in the direction orthogonal to the gradient, tuned by the magnitude of the gradient.

Casselles and Coll proved that this equation is indeed well-posed in the space of bounded variation. A variant was proposed by Alvarez et al,

$$\frac{\partial U}{\partial t} = \frac{|Du|}{|K*Du|}div\left(\frac{Du}{|Du|}\right) = \frac{1}{|K*Du|}U_{\xi\xi}$$

where the tuning of the gradient in nonlocal.

Kimia, Tonnenbaum and Zacker proposed, endowed in a more general analysis framework, the equation

$$\frac{\partial U}{\partial t} = |Du| div\left(\frac{Du}{|Du|}\right) = D^2u\left(\frac{Du\perp}{|Du|}, \frac{Du^{\perp}}{|Du|}\right) = U_{\xi\xi}$$

this equation had been proposed before in another context by Sethian as a tool for front propagation algorithms. This equation, which we call in continuation "curvature equation", is a "pure" diffusion in the direction orthogonal to the gradient. The Weickert equation is a variant of the curvature equation with nonlocal estimate of the direction orthogonal to the gradient: the diffusion direction

$d = sEigen(K * (Du \otimes Du))$

is computed as the eigenvector of the least eigenvalue of

$K * (Du \otimes Du)$ if the convolution kernel, the eigenvector is simply $Du^{\perp}$.

Ωther diffusions have been considered as well: for interpolation goals, Caselles et al., proposed a diffusion which may be interpreted as the strongest possible image smoothing,

$$\frac{\partial U}{\partial t} = D^2u(Du, Du)$$

Zhong and Camona proposed a diffusion in the direction of $d = sEigen(Du^2)$ of the eigenvector with least eigenvalue *of* Du^2. Sochen, Kimmel and Malladi proposed instead a nondegenrate diffusion, associated with a minimal surface variational formulation: their idea was to make a gradient descent for the area of the graph U, $\int\sqrt{1+|Du^2|}$, which leads to the diffusion equation

$$\frac{\partial U}{\partial t} = div\left(\frac{Du}{\sqrt{1+|Du|^2}}\right)$$

2.12.2 Contribution to Fractals

Fractals came from chaos theory. Therefore, no discussion of fractals without reference to chaos theory is complete. The science of fractals must be credited first to Mandelbrot. Here we present renditions of the Mandelbrot set.

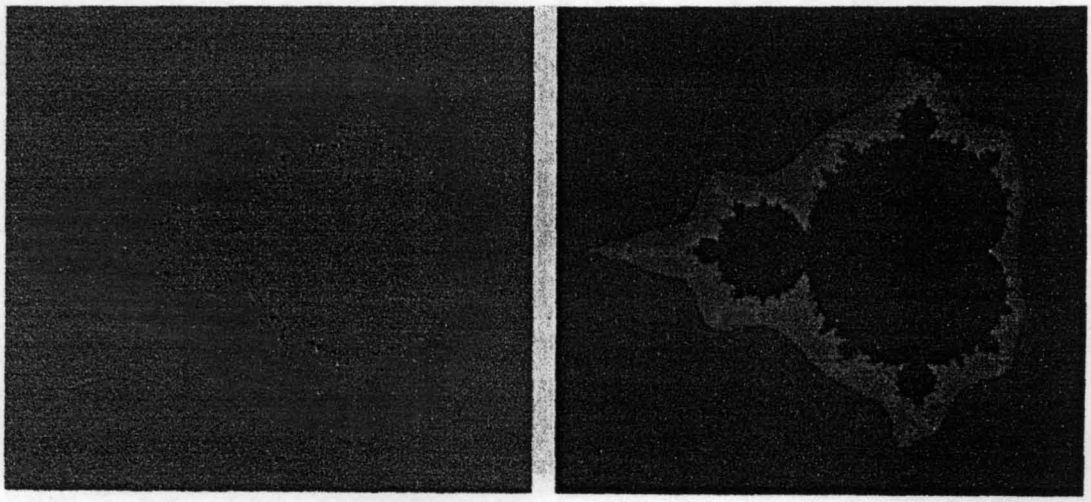


Figure 2.20: Renditions of the Mandelbrot set

In atmospheric physics the Lorenz equation has the following fractal plot

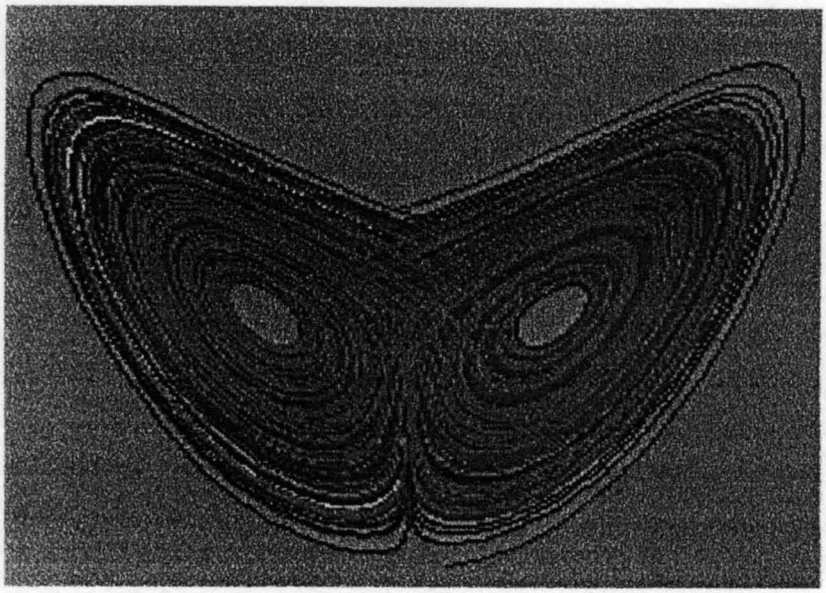


Figure 2.21: The Lorenz equation

We present Julia sets due to Gaston Julia

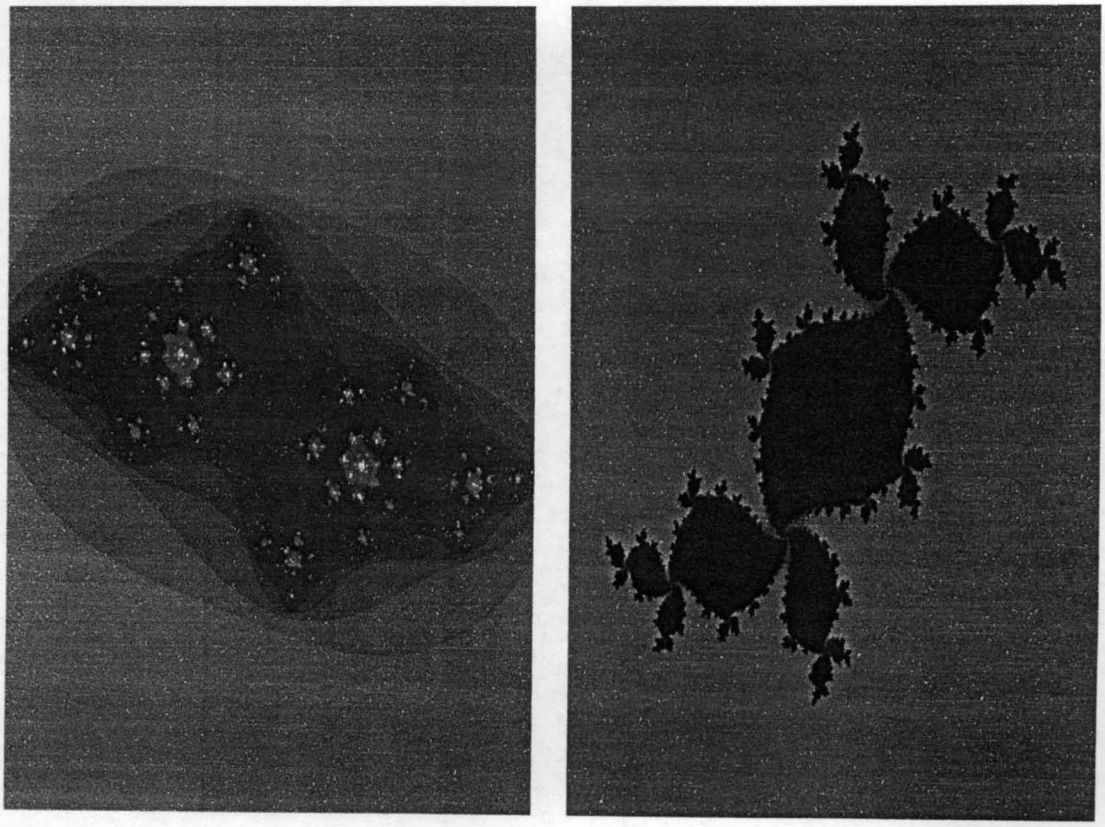


Figure 2.22: *Julia sets*

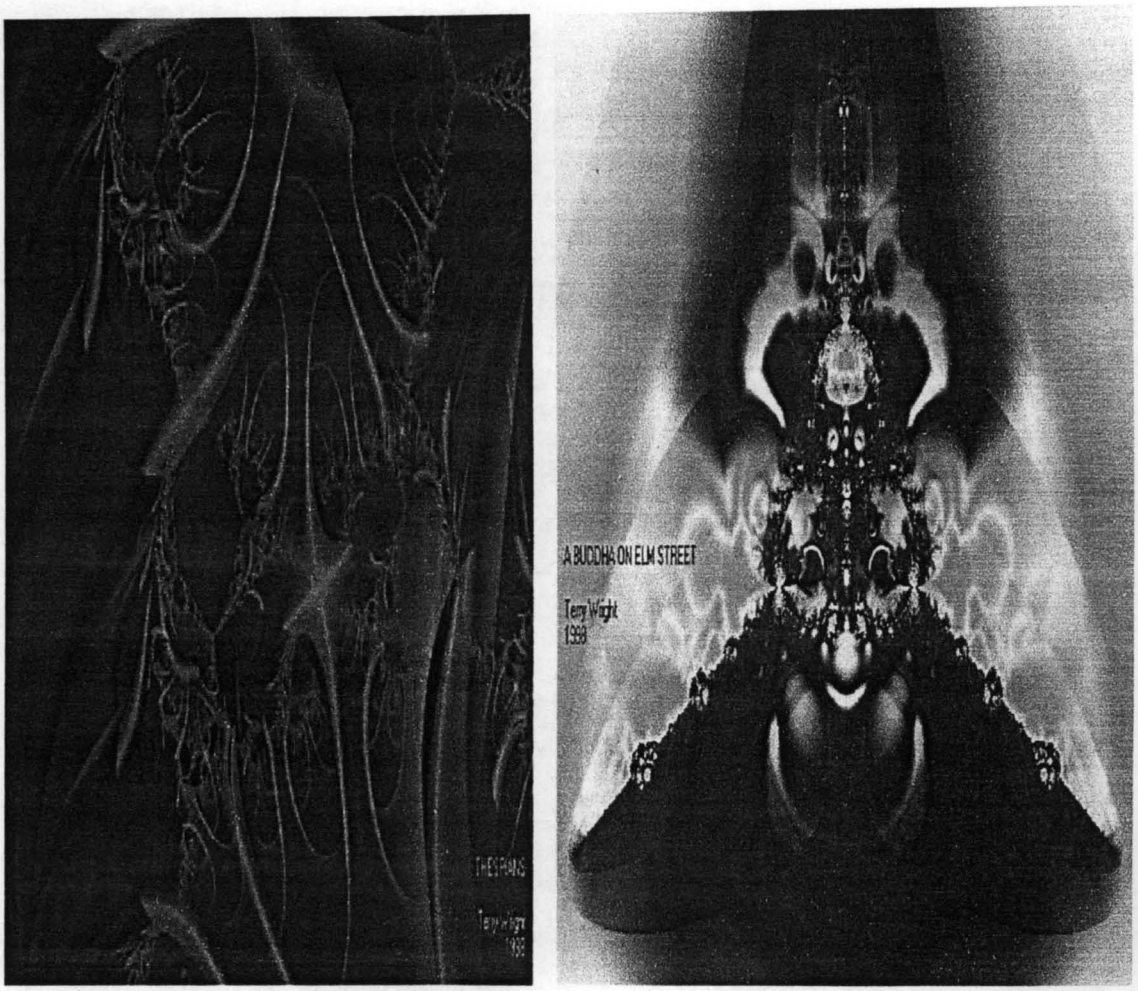


Fig 2.23: Fractals due to Terry Wright

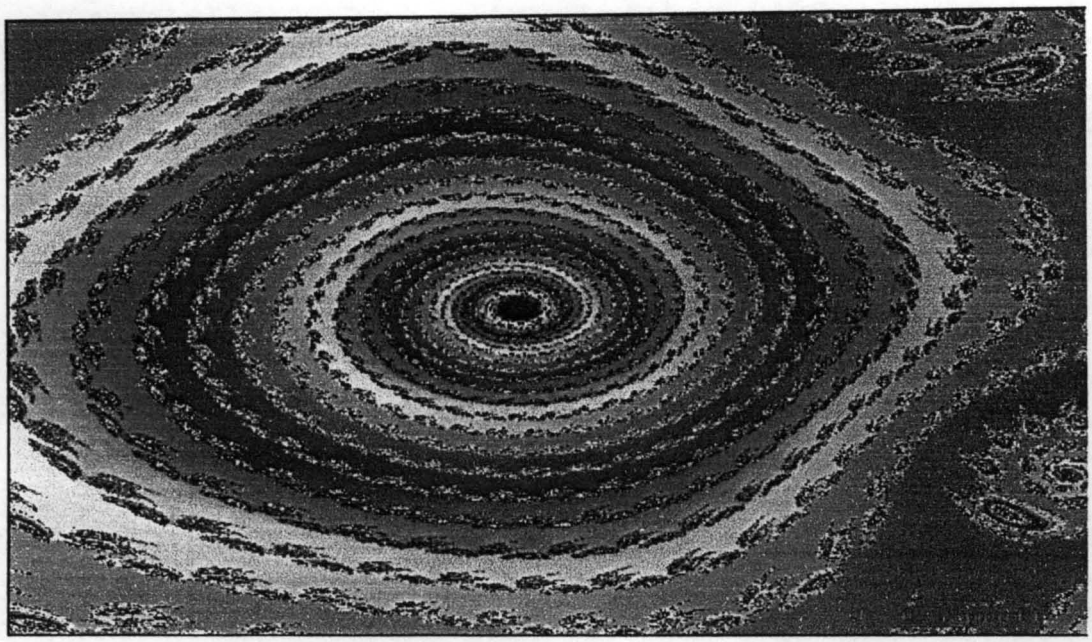


Fig 2.24: Quadratic Attractors

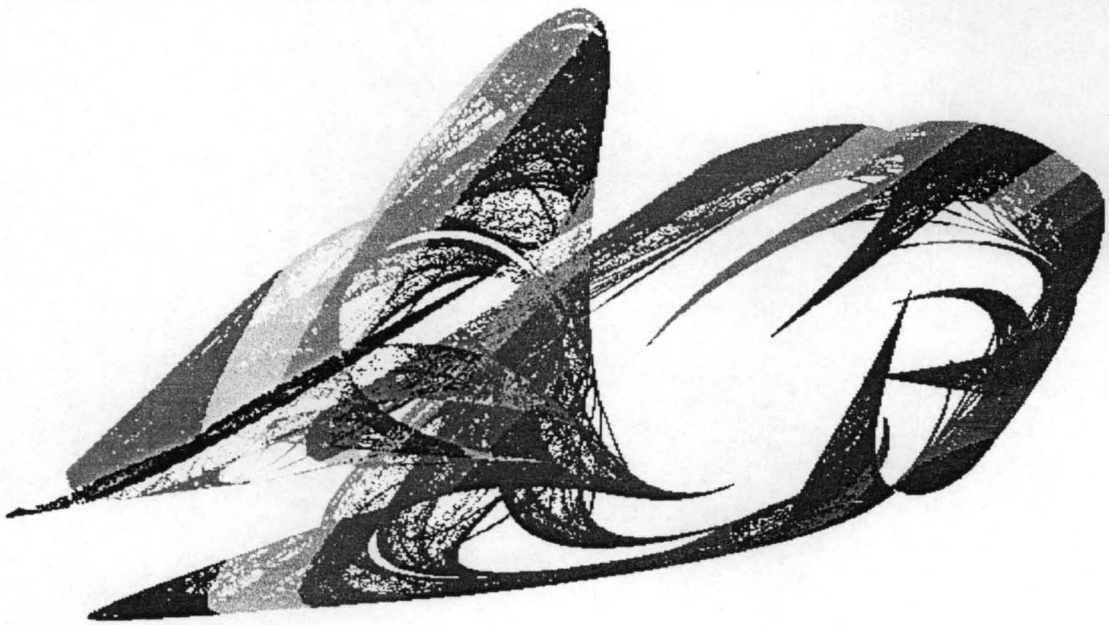


Fig 2.25: Strange Attractors

CHAPTER THREE

FRACTALS

3.1 Feedback and the Iterator

When we think of fractal images, figures or structures, we usually perceive them as static objects. This is a legitimate initial standpoint in many cases as *for* example when we deal with natural structures like the ones in figures 3.1 to 3.3 below

Fig. 3.1 Red Lake Peak, the Sierra Nevada

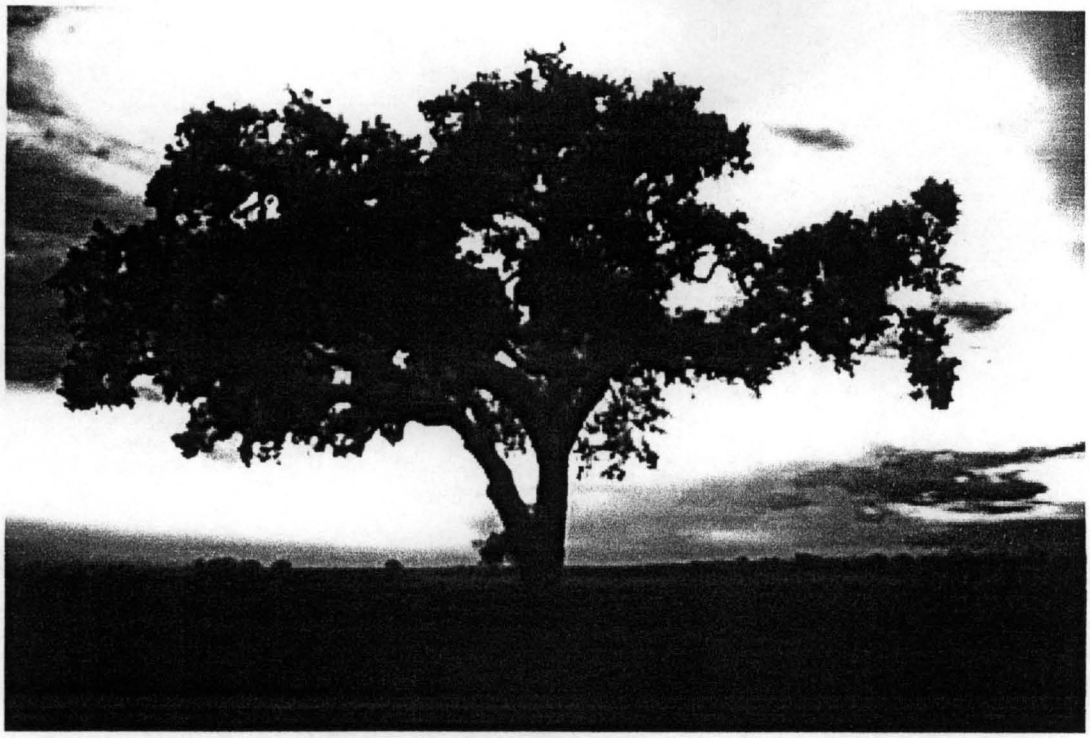


Fig. 3.2 California Oak Tree, (Quercus lobata), Valley Oak

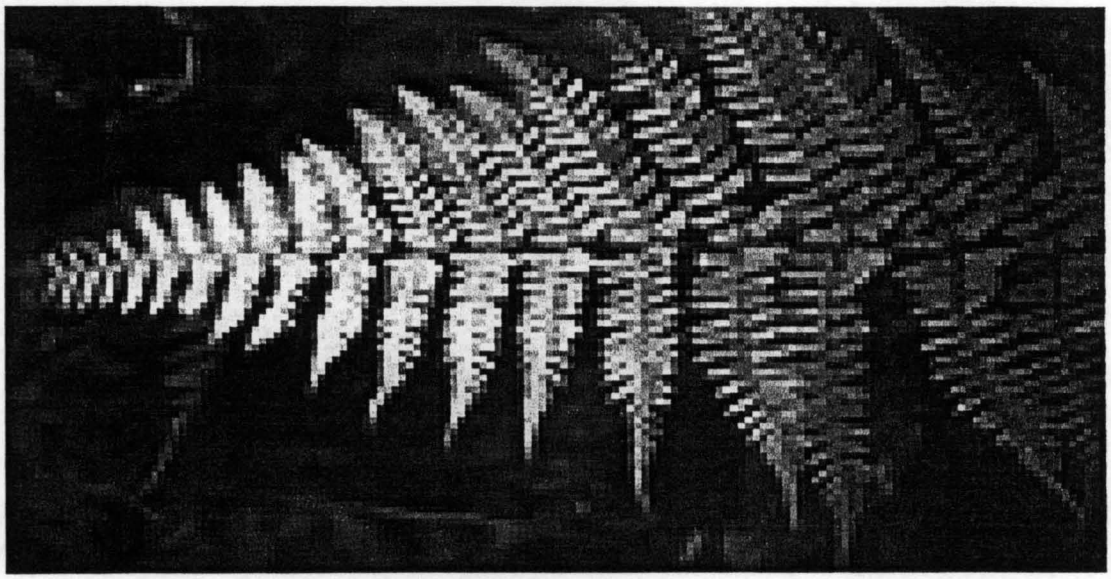


Fig *3.3: The fern. Verlag Gustav Fischer, Stuttgart*

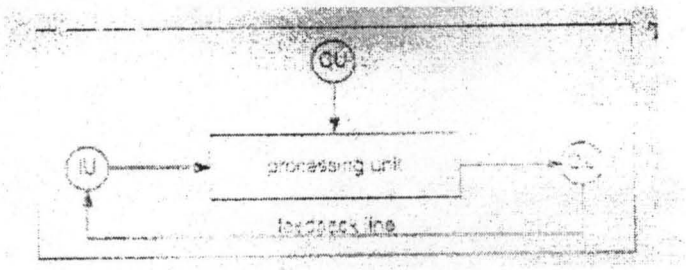
But this point of view tells us little about the evolution or generation of a given structure, often as, for example in botany; we like to discuss more than just the complexity of ripe plant. In fact any geometrical

model of a plant which does not also incorporate its dynamic growth plan will not lead very far. The same is true for mountains, whose geometry is a result of past tectonic activity as well as erosion processes which still and will forever shape what we see as mountains. We can also say the same for the deposits of zinc in an electrolyte experiment.

3.1.1 The Principle of Feedback

The most important example of a simple process with very complicated behaviour is the process determined by expressions such as $x^2 + c$ where c is considered to be a fixed constant or $p + rp(1-p)$ where r a constant.

Feedback processes are fundamental in all exact sciences. In fact they were first introduced by Isaac Newton and Gottfried Leibniz some 300 years ago.



Fig, 3.4 *The Feedback machine with IU = input Unit, OU = Output Unit, CU = Control Unit*

The feedback machine has three storage units (IU = input unit, OU = Output Unit, CU = control Unit), and one processor all connected by four transmission links [Peitgen, Jurgen ans Saupe, 1992]

3.1.2 Types of Feedback Machines

We consider as simple examples of feedback machines which process numbers. One-step machines are characterized by an iteration formula $x_{n+1} = f(x_n)$, where $f(x)$ can be any function of x. It requires some number as input and returns a new output e.g $f(x_n) = x_n^2 + 1$). The formula can be controlled by a fixed parameter (e.g. $x^2 + c$, i.e. with control parameter c), but in any case the output depends only on the input. The numbers are indexed in order to keep track of the time (cycle) in which they were obtained.

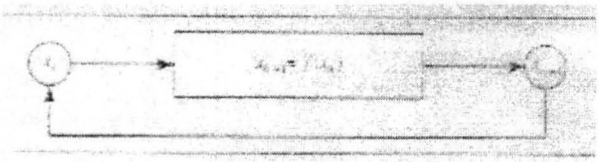


Fig. 3.5 Principle of the one-step feedback machine

One-step machines are very useful mathematical tools and have been developed in particular for numerical solution of complex problems. One-step processes represent only a particular class of a family of feedback methods. Multistep feedback processes are computed by a formula such as

$$x_{n+a} = g(x_{n+a-1}, x_{n+a-2}, x_{n+a-3}, \ldots, x_0) \quad \ldots\ldots 1$$

an example is the 2-step law which generates the Fibonacci numbers

$$g(x_n, x_{n-1}) = x_n + x_{n-1} \quad \ldots\ldots 2$$

machines with memory are typical of our computer age. While a machine without memory reacts to their inputs always in the same way, a machine with memory may react differently upon taking its own state or content of the memory into account.

3.2 Self-Similarity in Fractals

Self-similarity is a concept which can be understood without any trouble. The new bread romanesco (see figure below) a crossing between cauliflower and broccoli, illustrates the concept.

Fig. 3.6 The brocolli romanesco, exhibits striking self-similarity

Macroscopically, we see a form which is best described as a cluster. That cluster is composed of smaller clusters which look almost identical to the entire cluster, however scaled down by some factor. Each of these clusters again is composed of smaller ones, and these again of even smaller ones, without difficulty, we can identify at least three generations of clusters on clusters. The second, third and all the following generations are essentially scaled down versions of the previous ones. In a rough sense, this is what self-similarity means. Below, we find the classical Mandelbrot fractal and a magnification of a section of it. When the fractal is fully developed that is when the image-generating iteration (eqn 1) goes to infinity the image reaches a

limit and becomes invariant under more iterations. At this point, the self-similarity in the fractals is fully developed.

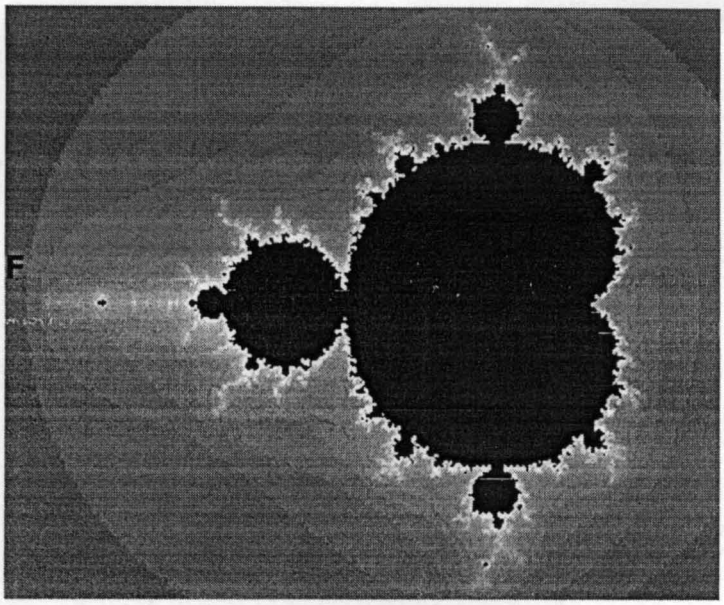


Fig 3.7: Mandelbrot classical fractal

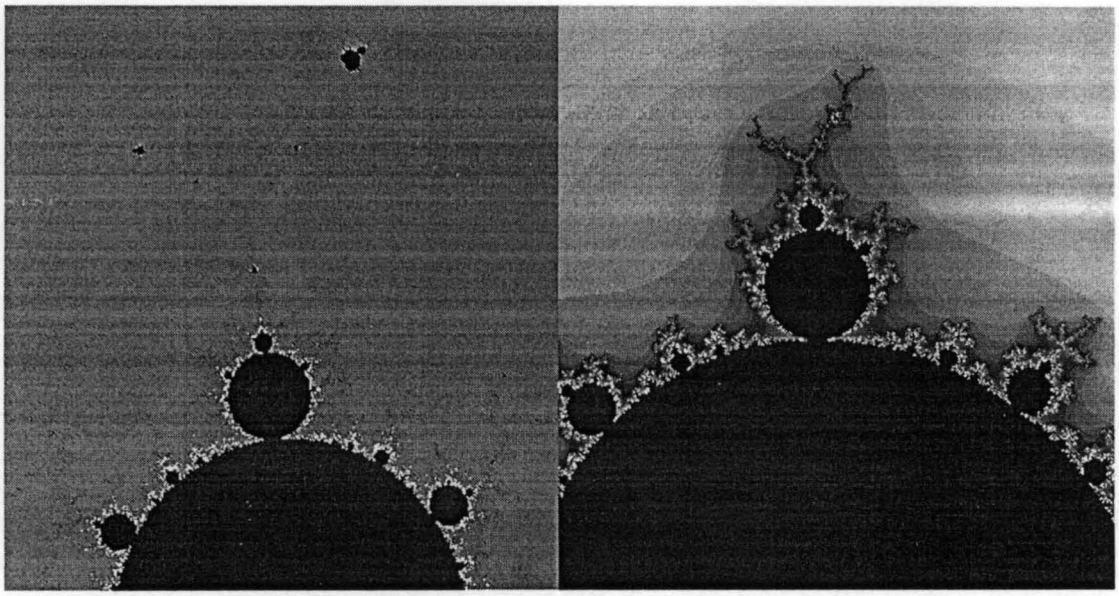


Fig 3.8: Zooms of Mandelbrot classical fractal

3.3 Limits and Self-similarity

A rigorous discussion of the concept of self-similarity is intimately related to concepts of limits. The visual observation in nature, however, is simple and immediate. Fractals add new dimension to the problem of dealing with limits; but also a new perspective from which to understand the concept of limits. On one hand fractals may visualize the limit object in a feedback process; on the other hand some fractals demonstrate self-similarity in its pure form. In fact .many fractals can be completely characterized and defined by their self-similarity properties.

Self-similarity extends one of the most fruitful notions of elementary geometry: similarity. Two objects are similar if they have the same shape, regardless of their size. Corresponding angle, however, must be equal and corresponding line segments must all have the same factor of proportionality. For example, when a photo is enlarged it is enlarged by the same factor in both horizontal and vertical directions. We call this enlargement factor the scaling factor. The similarity between the images is called the similarity transformation.

3.3.1 Similarity Transformations

Similarity transformations are compositions involving a scaling, a rotation and a transition. In the plane, we denote points P by their coordinate pairs $P(x, y)$. We now apply scaling, rotation and transition to one point, $P = (x, y)$, of a figure. First a scaling operation denoted by S, takes place, yielding a new point $P' = (x', y')$, with

$x' = sx$

$y' = sy$

where $s > 0$ *is the scaling factor*

a scale reduction occurs of $P(x,y)$ *if* $s<1$ and an enlargement will occur when $s<1$. Next a rotation R, is applied to $P'(x',y')$ yielding

$$P''=(x'',y'')$$
$$x''=\cos\theta x'-\sin\theta y'$$
$$y''=\sin\theta x'+\cos\theta y'$$

this describes a counterclockwise (mathematically positive) rotation of P' about the origin of the coordinate system by an angle of θ.

Finally, a translation T of P'' a displacement (T_x,T_y) is given by

$$x'''=x''+T_x$$
$$y'''=y''+T_y$$

which yields the point

$$P'''=(x''',y''')$$

Summarizing,we may write

$$P'''=T(P'')=T(R(P))=T(R(S(P)))$$

or using the notation

$$W(P)=T(R(S(P)))$$
$$we\ have\ P'''=W(P)$$

W is the similarity transformation. In the formulas

$$x'''=s\cos\theta x-s\sin\theta y+T_x$$
$$y'''=s\sin\theta x+s\cos y+T_y$$

applying W *to* all points of an object in the plane produces a figure which is similar to the original.[Crownover, 1995]

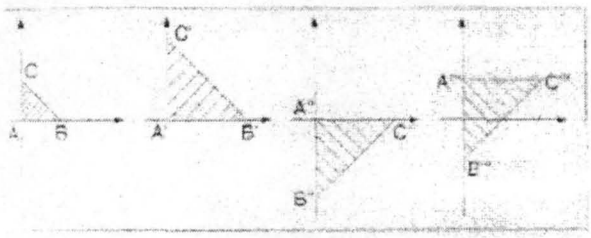


Fig. 3.7 A similarity transformation is applied to the triangle ABC. The scaling factor ***is*** $s=2$, *the rotation is by 270°, and the translation is given by* $T_x=0$ *and* $T_y=1$

3.3.2 Similarity and Growth

We look at the logarithmic spiral. A spiral drawn on a disk seems to grow continuously as it is turned around its center; in fact the logarithmic spiral is special in that magnifying it is the same as rotating the spiral.

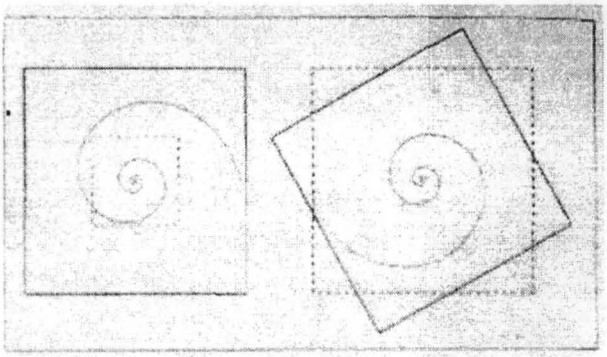


Fig. 3.8: The magnifying of a logarithmic spiral by a factor b shows the same spiral, however rotated by an angle θ (about 210°)

Fig 3.8 illustrates this remarkable phenomenon, which as such is another example of a self-similar structure.

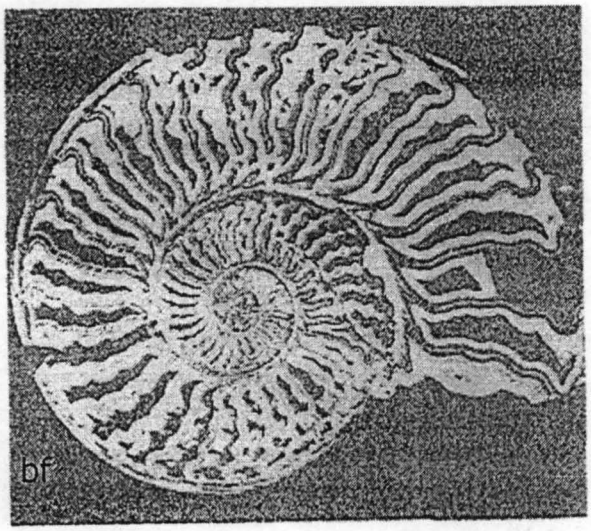


Fig. 3.9: The growth of an ammonite follows a logarithmic spiral

Figure 3.9 shows an ammonite which is a good example of a logarithmic spiral in nature. In other words an ammonite grows according to a law of similarity.

It grows in such a way that its shape is preserved. Most living things however, grow by a different law. An adult is not simply a baby scaled up, In the growth from baby to adult, different parts of the body are scaled up, each with a different scale factor. Two examples are:

a. Relative to the size of the body, a baby's head is much larger than an adult's
b. If we measure the arm length or head size for humans of different ages and compare it with body height, we observe that humans do not grow in a way that maintains geometric similarity. The arm, which at birth is one-third as long as the body is by adulthood closer to two-thirds as long.[Peitgen, Jurgens and Saupe, 1992]

We can discuss two different phases: one that fits early development, up to the age of about three years, and another that fits development after that sometime called isometric growth. After the age three years, however, the ratio drops significantly, indicating that body height is growing relatively faster than head size. This is called allometric growth.

3.3.3 What is Self-Similarity?

Intuitively, it seems the word self-similarity seems clear and hardly needs a definition. However, talking in precise mathematical terms about self- similarity is a much more difficult undertaking. For example, in the romenesco plant or for that matter in any physically existing object, the self-similarity may hold only for a

few powers of magnitude. Below a certain scale, matter decomposes into a collection of molecules, atoms, and going a bit further, elementary particles. Having reached that stage, of course, it becomes ridiculous to consider miniature slowed-down copies of the complete object. Also, in a structure like the cauliflower, the part can never exactly be equal. Some variation must be accounted for. Thus, it is already clear at this point that there are several variants of mathematical definitions of self-similarity. In any case, we like to think of mathematical fractals as objects which possess recognizable details at all microscopic scales – unlike real physical objects. When considering cases of fractals where the small copies, while looking like the whole, have variants, we have the so-called statistical self-similarity. Moreover, the miniature copies may be disturbed in other ways, for example, somewhat skewed. For this case there is the notion of self-affinity.

3.3.4 Self-similarity and the Koch Curve

To exemplify the concept, we choose the Koch curve which is already familiar. The Koch curve looks like it is made up of four identical parts. Let us look at one of these, say the one on the extreme left. We take a variable zoom lens and observe that at exactly x3 magnifying power the little piece seems to be identical to the entire curve. Each one of the little pieces breaks into four identical pieces again, and each of them seems so identical to the entire Koch curve when we apply a magnification lens of x9, and so on ad infinitum. This is the self-similarity property in its purest form.

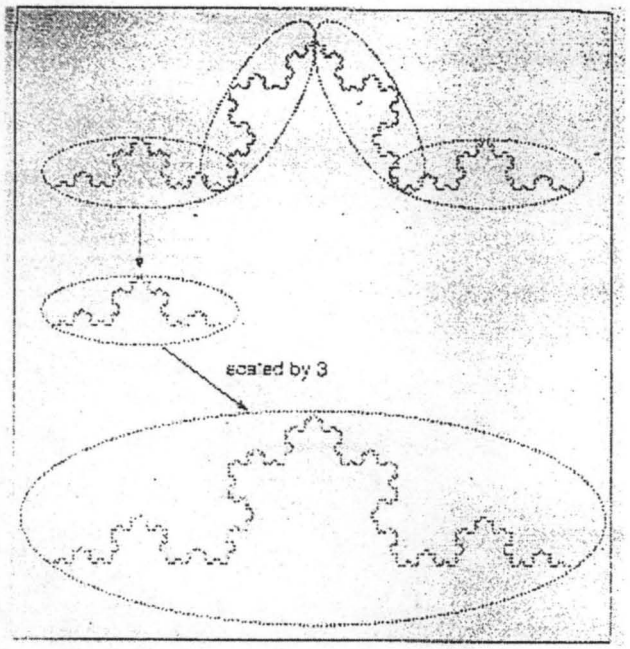


Fig. 3.10: *One-quarter of a Koch curve (top) is magnified by a factor of 3. due to the self-similarity of the Koch curve the result is a copy of the whole curve*

But even in this case, where copies of the whole appear at all stages and are exact and are not disturbed in any way, there are still various degrees of self-similarity possible. Consider for example, a cover of a book that contains on it a picture of a hand holding that very book. As we look deeper and deeper into the design, we see more and more of the rectangle covers. Contrast that with an idealized structure of a two-branch tree as shown in the figure below. Also pictures in the self-similar Sierpinski gasket. All three examples are self-similar structures. They contain small replicas of the whole. However, there is a significant difference. Let us try to find points which have the property that we can identify small replicas of the whole in their neighborhoods at any degree of magnification. In the case of the book design the copies are arranged in a nested sequence, and clearly the self-similarity property can be found only at one particular point. This is the limit point at which the sizes of the copies tend to zero. The book cover is self-similar at this point.

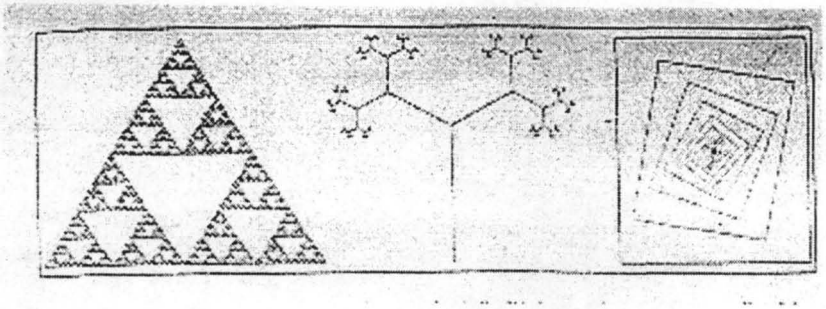


Fig. 3.11: the Sierpinski Gasket (left) is self-similar at all of its points, while the two-branch tree (middle) is self-similar only at the leaves. The structure on the right is self-similar only at the centre point

The situation is much different in the two-branch tree. The complete tree is made up of the stem and two reduced copies of the whole. Thus, smaller and smaller copies accumulate near the leaves of the tree. In other words, the property of self-similarity condenses in the set of the eaves. The whole tree is not strictly self-similar but self-affine. The stem is not similar to the whole tree but we can interpret it as an affine copy which is compressed to a line. In the Sierpinski gasket, similar to the Koch curve above, we can find copies of the whole near every point of it. The gasket is composed of small but exact copies of itself. Considering these differences, we call all those objects self-similar, while only the Sierpinski gasket and the Koch curve are in addition called strictly self-similar Also the set of leaves without the stem and all the branches is strictly self-similar. The cauliflower is a physical approximation of self-similarity, but not strictly self-similar as in the two-branch tree.

3.3.5 Geometric Series and the Koch Curve

Fractals such as the Koch curve, the Sierpinski gasket and many others are obtained by a construction process. Ideally, however, this process should never terminate. Any finite stage of it produces an object, which may have a lot fine structures depending on how far the

process has been allowed to proceed; but essentially it is still far from a true fractal. Thus the fractal only exists as an idealization. It is what we would get if we let the process run indefinitely. In other words fractals really are limit objects, and their existence is not as natural as it may seem. Limits often lead to new quantities, objects or qualities; this is true particularly for fractals. However, given an object, there are cases where it is not immediately obvious whether a limit exists at all. As for example the first sum in

$\sum \frac{1}{k} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \ldots$ is divergent i.e. the sum is infinite while

$\sum \frac{1}{k^2} = \frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \ldots$ converges to $\frac{\pi^2}{6}$ (as shown by Euler)

The geometric series has an analogy in the construction of basic fractals. 'There is an initial object and a scaling factor. The important property of the scaling factor is that it be less than 1 in magnitude. Then there is a construction process

Step 1: Start with initial object (P)

Step 2: Scale down P by the scaling factor q and add

Step 3: Scale down P by the scaling factors q.q and add

Step 4: ...

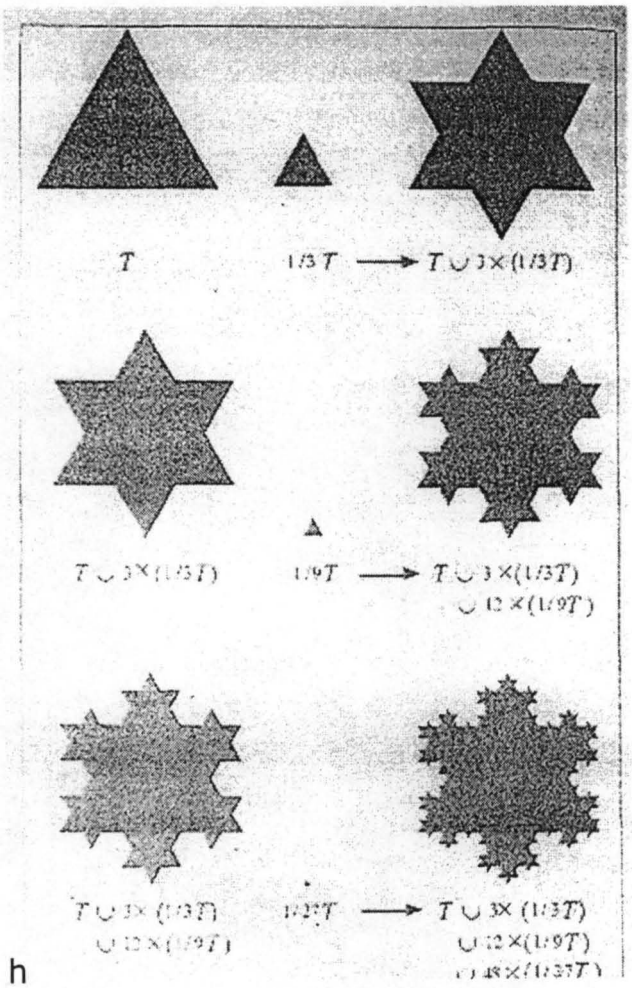
The point is that this infinite construction leads to a new object, representing the process - the limit of geometric series. The Koch island, which we see in its basic construction is obtained in an analogous manner except that rather than adding up numbers, we 'add up' geometric objects. 'Addition' of course, is here interpreted as a union of sets, and the important point is that in each step we add a certain number of scaled down version of the initial set.

Step 1: We choose an equilateral triangle T with sides of length a

Step 2: We scale down T by a factor of $1/3$ and paste on three copies of the resulting little triangles as shown. The island is now bounded by 3.4 line segments, each of length $a/3$.

Step 3: We scale down T by a factor $1/3 \cdot 1/3$ and paste 3.4 copies of the resulting little triangles as shown. The resulting island is bounded by 3.4.4 straight segments, each of length $1/3 \cdot 1/3 \cdot a$

Step 4: ...



h

Fig. 3.12: The Koch island is the limit of the construction and has area $A = \frac{2}{5} 3a^2$

The point here is that the infinite construction leads to a new geometric object, the Koch. In fact, the analogy between the geometric process and the geometric series goes much further.

3.3.6 The Area of the Koch Curve

At the beginning we have the area A, for initial triangle T, and calculate $A_1 = \frac{\sqrt{3}}{4}a^2$. In each step k, we have to add the area of n_k little equilateral triangles with sides s_k. $n_1 = 3, n_2 = 3 \cdot 4, n_3 = 3 \cdot 4 \cdot 4 \ldots$ In other words $n_k = 3 \cdot 4^{k-1}$. The sides s_k of the little triangles are obtained by successively scaling down the side of the original triangle by a factor of $1/3$. In other words, $s_k = \left(1/3\right)^k a$. Combining these results, we get

$$
\begin{aligned}
A_{k+1} &= A_k + n_k \cdot \frac{\sqrt{3}}{4} s_k^2 \\
&= A_k + 3 \cdot 4^{k+1} \frac{\sqrt{3}}{4} \frac{1}{3^{2k}} a^2 \\
&= A_k + \frac{\sqrt{3}}{12}\left(\frac{4^{k-1}}{9^{k-1}}\right) a^2
\end{aligned}
$$

In other words, if we develop the terms step by step, we have the series

$A_{k+1} = A_1 + \frac{\sqrt{3}}{4}\left(1 + \frac{4}{9} + \frac{4^2}{9^2} + \frac{4^3}{9^3} + \ldots + \frac{4^{k-1}}{9^{k-1}}\right)a^2$. The expression in the bracket is a partial sum of the geometric series $1 + \frac{4}{9} + \frac{4^2}{9^2} + \frac{4^3}{9^3} + \ldots$ which has a a limit $1 - \frac{1}{1 - 4/9} = \frac{9}{5}$. That means that the Koch island, the geometric objects of the limit, has area $A = A_1 + \frac{\sqrt{3}}{12} \cdot \frac{9}{5} a^2$ and since $A_1 = \frac{\sqrt{3}}{4} \cdot a^2$, we finally obtain $A = \frac{2}{5}\sqrt{3} \cdot a^2$. This is quite a convincing argument that there

is indeed a new geometric object resulting from the infinite process. There are properties of limit that are not shared by any of its finite stage approximations. The most important property is that of self-similarity. For example the self-similarity of the Koch curve is reflected by the fact that the curve is made up of four identical parts.
Another property of the Koch curve which is not shared by any of its finite stage approximations is that its length is infinite. As the Koch curve is one- third of the boundary of the Koch is; and, we have that the boundary of the island is also infinitely long. In contrast to this, the area of the Koch island is finite, well-defined number as seen above.

3.4 Encoding Images by Simple Transformations

So far, we have discussed two extreme ends of fractal geometry. We have explored fractal monsters, such as the Cantor set, the Koch curve, and the Sierpinski gasket; and we have argued that there are many fractals in natural structures and patterns, such as coastlines, blood vessel systems, and cauliflower. We have discussed features, such as self-similarity, scaling properties, and fractal dimensions shared by those natural structures and the monsters; but we have not yet seen that they are close relatives in the sense that may be a cauliflower is just a 'mutant' of a Sierpinski gasket, and a fern is just a Koch curve let loose.

We may regard fractal geometry as a new language in mathematics [Falconer, 1989]. As the English language can be broken down into letters and the Chinese language into characters, fractal geometry promises to provide a means to break down the patterns and forms of nature into primitive elements, which then can be composed into "words' and 'structures' describing these efficiently[Lauerier, 1991].

Here, we will discuss one of the major dialects of fractal geometry as if it were a language. Its elements are primitive transformation, and its words are primitive algorithms, we introduce the metaphor of the Multiple Reduction Copy Machine (MRCM), which will be our centre of interest.

3.4.1 The Multiple Reduction Copy Machine

The Multiple Reduction Copy Machine (MRCM) provides a good metaphor for what is known as a deterministic Iterated Function System (IFS). Both terminologies can be used interchangeably. The crucial idea is that the machine runs in a feedback loop; its own output is fed back in its new input again and again.

Consider an MRCM with three lens system, each of which is set to reduce by a factor $1/2$. The resulting copies are assembled in the configuration of an equilateral triangle.

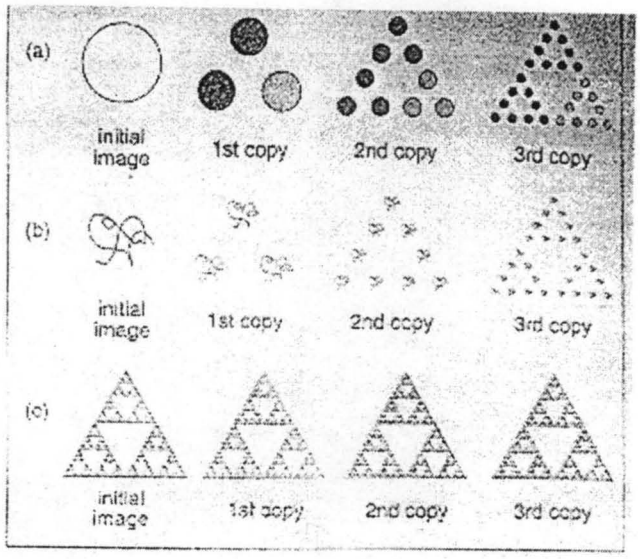


Fig. 3.13: Three iterations of an MRCM with three different initial images

Fig 3.13 shows the effect of the machine run three times beginning with different initial images. In (a), we take a disk and we use different shadings to keep track of the effect of the individual lens systems. In (b) we try a truly "arbitrary' image. In just a few iterations, the machine, or abstractly speaking the process, throws out images which look more and more like a Sierpinski gasket. In (c) we start with a Sierpinski gasket and observe that the machine has no effect on the image. The assembled reduced copies are the same as the initial image. That is because of the self-similarity property of the Sierpinski gasket. Let us summarize this first experiment. No matter which initial image we take and run the MRCM with. We obtain a sequence of images which always tends towards one and the same final image. This is called the attractor of the machine process. Moreover, when we start the machine with the attractor, then nothing happens. One says the attractor is left invariant or fixed. The MRCM machine always has been a unique final image as an attractor, and this final image is invariant under the iteration of the MRCM.

3.4.2 Composing Simple Transformations

The Multiple Reduction Copy Machine is based on a collection of contractions. The term contraction means, roughly speaking that points are moved closer together when one contraction is applied, we may also say use transformations which reduce by one factor, say $\frac{1}{3}$. Horizontally and by a different factor, say $\frac{1}{2}$, vertically is also allowed. A similarity transformation, however, leaves angles unchanged, while more general transformation may not.

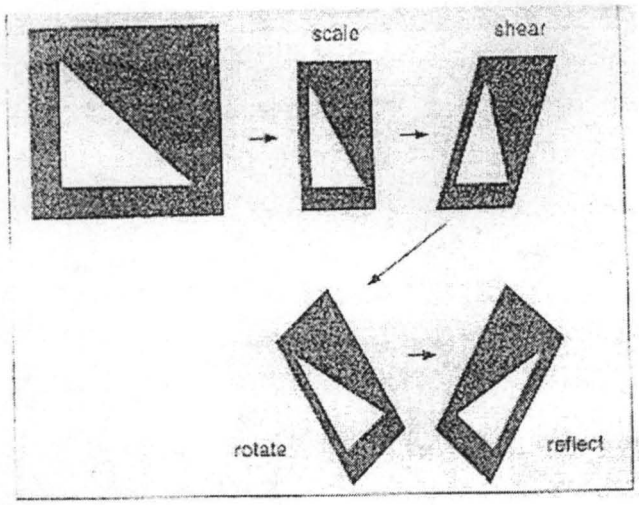


Fig. 3.14: Transformations with scaling, shearing, reflection, rotation and translation are admissible in an MRCM

The lens of the MRCM can be described by affine linear transformations of the plane. Talking about a plane means that we fix a coordinate system, an x-axis and a y-axis. Relative to that coordinate system every point P in the plane can be written as a pair (x, y). Sometimes we write $P = (x, y)$. In this way, points can be added together and can be multiplied by real numbers: if

$P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ then

$P_1 + P_2 = (x_1 + x_2, y_1 + y_2)$

and $sP = (sx, sy)$

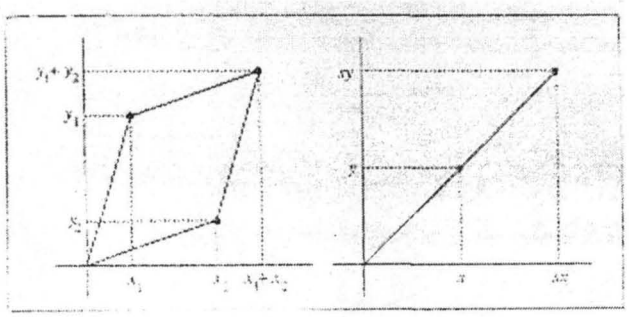


Fig. 3.15: (Left) Two points (x_1, y_1) and (x_2, y_2) are added $(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$. (Right) A point is multiplied by a number: $s(x, y) = (sx, sy)$

A linear mapping F is a transformation which associates with every point P in the plane a point $F(P)$ such that $F(P_1 + P_2) = F(P_1) + F(P_2)$ for all points P_1 and P_2 and $F(sP) = sF(P)$

For any real numbers and all points P. A linear transformation F can be represented with respect to the given coordinate frame by a matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$where\ if\ P = (x, y)\ and F(P) = (u, v),\ then$

$u = ax + by$

$v = cx + dy$

in other words, a linear transformation is determined by four coefficients,

a, b, c, d. These are special representation which helps us to discuss contractions more conveniently. To this end we write the four coefficients in our matrix as

$$\begin{bmatrix} r\cos\phi & -s\sin\varphi \\ r\sin\phi & s\cos\varphi \end{bmatrix}$$

such a representation is always possible. We set

$r = \sqrt{a^2 + c^2}$

and

$$\phi = \arccos \frac{a}{\sqrt{a^2 + c^2}}$$

to obtain r and ϕ. Similar formulas hold for s *and* φ. In this way, it is easier to discuss reductions, rotations and reflections.

3.4.2.1 When $s = r, 0 \le r < 1\ and\ \varphi = \phi$

This fixes a mapping which reduces by a factor of r and simultaneously rotates by the angle ϕ counterclockwise (the mapping is just a reduction if $\phi = 0$

3.4.2.2 When $s = r, 0 \le r < 1 \; and \; \varphi = \pi \; and \; \varphi = 0$

This fixes a mapping which reduces by a factor of r and simultaneously reflects with respect to the y-axis.

3.4.2.3 When $r = a, and \; s = b, 0 \le a < 1, \; 0 \le b < 1 \; and \; \varphi = \phi = 0$

$r = a, and \; s = b, 0 \le a < 1, \; 0 \le b < 1 \; and \; \varphi = \phi = 0$ fixes a mapping which reduces by a factor b in the y-direction

3.4.2.4 When $s = r > 0, and \; \varphi = \varphi$

$s = r > 0, and \; \varphi = \varphi$ defines a similarity transformation given by a rotation by an angle of φ and a scaling by a factor of r.

Affine linear mappings are simply the compositions of a linear mapping together with a translation. In other words, if F is linear and Q is a point, then the real mapping $w(P) = F(P) + Q$ where P is any point in the plane, is said to be affine linear. Affine linear mapping allows us to describe contractions which involve positioning in the plane (i.e. the translation by Q is given by a pair of coordinates, say (e, f), an affine linear mapping w is given by six numbers

$$\left[\begin{array}{cc|c} a & b & e \\ c & d & f \end{array}\right]$$

and if $P = (x, y)$ and $W(P) = (u, v)$ then

$$u = ax + by + e$$
$$v = cx + dy + f$$

Another notation for the same equations can also be used

$$w(x, y) = (ax + by + e, cx + dy + f)$$

in the discussion of Iterated Function Systems, it is crucial to study the objects which are left invariant under iteration of an IFS.

Given an affine linear mapping w we can ask which are left invariant Under w? This is an exercise with a system of linear equations. Instead, $w(P) = P$ means

$$x = ax + by + e$$
$$y = cx + dy + f$$

Solving this system of equations yields exactly one solution, as long as the determinant $(a-1)(d-1) - bc \neq 0$. This point $P = (x, y)$ is called the fixed point of w. Its coordinates are

$$x = \frac{-e(d-1) + bf}{(a-1))(d-1) - bc}$$
$$y = \frac{-f(a-1) + ce}{(a-1)(d-1) - bc}$$

3.4.3 IFS and the Hutchinson Operator

Let $w_1, w_2, w_3, \ldots$ be N contractions of the plane. We define a mapping – the Hutchinson Operator as follows: Let *Abe* any compact subset of the plane. Here we think of *A* as an image. Then by the collage obtained by applying the /^contractions to *A* and assembling the results can be expressed by the collage mapping

$$W(A) = w_1(A) \cup w_2(A) \cup w_3(A) \cup \ldots \cup w_N(A)$$

the Hutchinson Operator turns the repeated application of the metaphoric MRCM into a dynamical system: an IFS. Let A_0 *be* an initial set (image) then we obtain

$$A_{k+1} = w(A_k), k = 0,1,2,\ldots$$

a sequence of sets (images), by repeatedly applying w. An IFS generates a sequence which tends towards a final image, A_∞, which we call the attractor of the IFS (or MRCM), and which is left invariant by the IFS. In terms of w, this means that $w(A\infty) = A$

The image coding problem has led to a central question: how images can be compared or what the distance between two images is. Felix Hausdorff proposed a method of determining this distance. The Hausdorff distance has two marvelous consequences. First, we can talk about the sequence of images A_k having limit A_∞ in a very precise sense: A_∞ is the limit of the sequence $A_0, A_1, A_2, \ldots$ provided that the Hausdorff distance $h(A_0, A_k)$ *goes* to 0 ask k goes to ∞.[Peitgen and RIcter, 1997]

But even more importantly, Hutchinson showed that the operator *W* which describes the collage $W(A) = w_1(A) \cup w_2(A) \cup w_3(A) \cup \ldots w_N(A)$ is a contraction with respect to the Hausdorff distance. That is, there is a constant c, with $0 \le c < 1$ such that $h(w(A), w(B))$ for all compact sets A and B in the plane.

In establishing this fundamental property, Hutchinson was able to inject into consideration one of the most powerful and beautiful principles in mathematics - the contraction mapping principle. The Hausdorff distance determines the distance of images. It is based on the concept of distance of points. Expressed generally, the distance between points of a space X can be

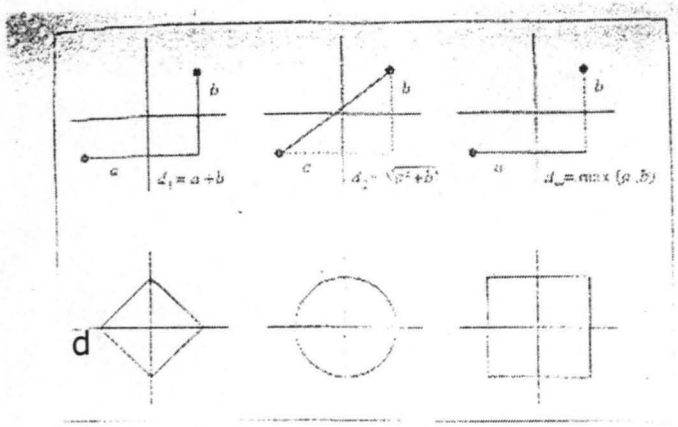


Fig. 3.16: Three methods of measuring distance in the plane (the lattice distance, the Euclidean distance, the maximum norm distance) and the corresponding unit sets (the set of points which have the distance 1 to the origin of the coordinate system).

Measured by a function $d: X \times X \to \Re$ Here $\Re$ denotes the real numbers and the function d must have the properties.

i. $d(x,y) \geq 0$

ii. $d(x,y) = 0 \; iff \; x = y$

iii. $d(x,y) = d(y,x)$

iv. $d(x,y) \leq d(x,z) + d(y,z) \, triangle \; inequality \,)$

holds for all $x, y, z \in X$ where d is the metric. [Kryszig, 1978].

Once we have a metric space X, we can talk about limits of sequences. Let $x_0, x_1, x_2, \ldots$ be sequence of points from X and an element *of* X . Then a is the limit of the sequence provided

$\lim_{k \to \infty} d(x_k, a) = 0$

In other words, for any $\varepsilon > 0$, we can find a point x_n in the sequence so that any point later in the sequence has distance to a less than ε

$d(x_n, a) < \varepsilon, k > n$

In this case we say that the sequence converges to a. The central problem of dynamical systems theory is to forecast the long-term

behaviour. Often that behaviour will not depend on the initial choice a_0. That is exactly the environment for the contraction mapping principle. It provides everything which we can use to make the forecast. But having in mind the variety of both wild and tame behaviour which feedback processes can produce/ it is clear that the principle will select some subclass of feedback systems for which it can be applied. Here, we state two features which characterize this class.

3.4.3.1 The Space

The objects, numbers, images, transformations, etc., which we call a_n must belong to a set in which we can measure the distance between any two of its elements, for example, the distance between x and y is $d(x,y)$. Furethermore, the set must be subtracted in some sense. That means, if an arbitrary sequence satisfies a special test which examines the possible existence of a limit, then a limit exists and belongs to the set.

3.4.3.2 The Mapping

The sequence of objects is obtained by a mapping, say, f. That means that for any initial object a_0, a sequence $a_0, a_1, a_2, \ldots$ is generated by $a_{n+1} = f(a_n), n = 0,1,2,3,\ldots$ Furthermore, f is a contraction. That means that for any two elements of the space, say x and y, the distance between $f(x)$ and $f(y)$ is always strictly less than the distance between x and y. For this class of feedback systems the contraction mapping principle gives the following remarkable result.

3.4.3.3 The Attractor

For any initial object a_0, the feedback system $a_{n+1} = f(a_n)$ Will always have a predictable long-term behaviour.There is an object a_∞ (the limit of the feedback system) to which the system will go. We call a_∞ the unique attractor of the feedback system.

3.4.3.4 Invariance

The feedback system leaves a_∞ invariant. In other words, if we start with a_∞, then a_∞ is a fixed point of f i.e. $f(a_\infty) = a_\infty$

3.4.3.5 The Estimate

We can predict how fast the feedback system will arrive close to a_∞ when it is started at a_0. We only have to test the feedback loop once on the initial object. That means, if we measure the distance between a_0 and $a_1 = f(a_0)$ we can already safely predict how often we have to run the system to arrive near a_∞ within a prescribed accuracy. Moreover, we can estimate the distance between a_0 and a_∞ [Barnsely, 1993].

3.5 The Attractor of a Contraction Mapping

A mapping f is a contraction of the metric space provided that there is a constant $c, 0 \le c < 1$ such that for all x, y in X one has that $d(f(x), f(y)) \le cd(x, y)$

The constant c is called the contraction factor for f. Let $\{a_\infty\}$ be a sequence of elements from a complete metric space X defined $a_{\infty+1} = f(a_n)$ The following holds true

a. There is a unique attractor $a_\infty = \lim_{n \to \infty} a_n$

b. a_∞ is invariant, $f(a_\infty)=a_\infty$

c. There is an a priori estimate for the distance from a_n *to* the attractor, $d(a_n,a_\infty)\le\frac{cd(a_0,a_\infty)}{(1-c)}$ [Barnsely, 1993]

From the contraction property of f we derive $d(f(a_0),f(a_\infty))=cd(a_0,a_\infty)$

Applying the triangle inequality, we further obtain

$$d(a_0,a_\infty)\le d(a_0,f(a_0))+d(a_0,f(a_\infty))\le d(a_0,f(a_0))+cd(a_0,a_\infty)$$

thus

$$d(a_0,a_\infty)\le\frac{cd(a_0,f(a_0))}{(1-c)}$$

and likewise

$$d(a_n,a_\infty)\le\frac{cd(a_n,a_{n+1})}{(1-c)}$$

for all $n=0,1,2,\ldots.$

Finally, with

$$d(a_n,a_{n+1})\le cd(a_{n-1},a_n)\le c^2d(a_{n-2},a_{n-1})\le\ldots\le c^nd(a_0,a_1)$$

we arrive at the result

$$d(a_n,a_\infty)\le\frac{c^nd(a_0,a_1)}{(1-c)}$$

this allows us to predict n so that a_n is within a prescribed distance to the limit.

3.6 The Hausdorff Distance

Given an image A, the ε-collar of A, written A_ε, which is the set *A* with together with all points in the plane which have a distance from A of not more than ε . Hausdorff measured the distance between two compact sets A and B in the plane using ^-collar.

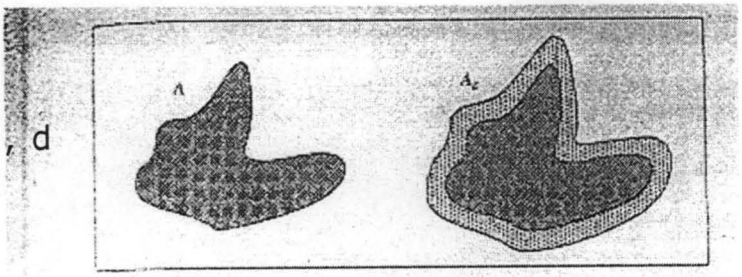


Fig. 3.17: The ε -collar of a set A in the plane. The ε-collar of A includes A and is not just the set of points close to A.

In precise mathematical terms the definition of the Hausdorff distance is as follows:

Let X *be* a complete metric space with a metric d. For any compact subset A of X and $\varepsilon > 0$, define the ε-collar of A by

$$A_\varepsilon = \{x \in X \mid d(x, y) \leq \varepsilon \text{ for some } y \in A\}$$

for any compact subset A and B *of* X , the Hausdorff distance is

$$h(a, b) = \inf\{\varepsilon \mid A \subset B_\varepsilon \text{ and } B \subset A_\varepsilon\}$$

according to Hausdorff, the space of ail compact subsets of X , equipped with the Hausdorff distance, is another complete metric space. This implies that the space of all compact subsets of X *is* a suitable environment for the contraction mapping principle.

3.7 Physical Applications

Several methods have been proposed for the automatic solution of the inverse problem .e. the encoding of images, but none has yet really proven itself to be the right choice. Therefore, we discuss a few ideas, some of which go back to Barnsley in the early 1980s. These ideas, however do not lead to automatic algorithms, they are more suitable for interactive computer programs requiring an intelligent human operator.

Assume that we already have approximated a given original image by an MRCM. The blueprint of an MRCM is already determined by its first copy. The copy is a collage of transformed images. Applying the MRCM to the original image, called the target image, one also determines the quality of the approximation. When the copy is identical to the original, then the corresponding IFS codes the target image perfectly. When the distance of the copy to the target is small, then we know from the contraction mapping principle, that the attractor of the IFS is not far from the initial image.

These properties enables us to find the code for a given target image, in particular for target images which contain apparent self-similarities such as the fern.

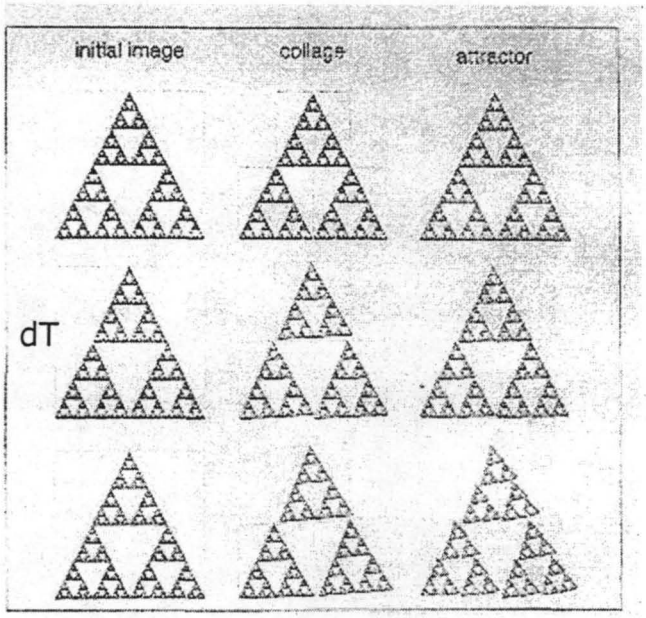


Fig 3.18: Application of three MRCMs to the Sierpinski gasket. Top: the correct MRCM leaves the image invariant: middle: a reasonable approximation; bottom: bad approximation

We exploit the contraction mapping principle to analyze the results of Figure 3.18. The a priori estimate for a sequence $a_0, a_1, a_2, \ldots$ which is generated by a contraction f in a metric space with attractor a_∞, yields

$$d(a_n, a_\infty) \leq \frac{c^n d(a_0, a_1)}{(1-c)}$$

Here c is the contraction factor of f and $a_{k+1} = f(a_k), fork = 0,1,2,...$ in particular, this means that

$$d(a_0, a_\infty) \leq \frac{d(a_0, f(a_0))}{(1-c)}$$

Thus, a single iteration starting from the initial a_0 gives us an estimate for how far a_0 is from the attractor a_∞ with respect to the metric d . Now let us interprete this result for the Hutchinson operator W with respect to the Hausdorff distance h. Let c be the contraction factor of W *and let* P be an arbitrary image (formally a compact subset of the plane). We would like to test how a good Hutchinson operator will encode the given image P. This can be obtained from the equation 3.2. Indeed, in this setting, (3.2)
now reads

$$h(P, A_\infty) \leq \frac{1}{(1-c)} h(P, w(P)) \quad \ldots\ldots 3.3$$

where A_∞ is the attractor of the IFS given by W. In other words, the quality of the encoding measured by the Hausdorff distance between P and A_∞ is controlled by applying the Hutchinson operator just once to P and quantifies by $h(P, w(P))$. [Barnsley, 1993]. Barnsely calls (3.3) the "collage theorem for Iterated Function systems
Fractals are greatly applied in artworks and textile designs and other such areas. Here, we present a few such examples.

CHAPTER FOUR

IMAGE EROSION AND DILATION

4.1 Mathematical Morphology

Mathematical Morphology is a tool for extracting image components that are useful for representation and description. The technique was originally developed by Matheron and Serra [cmm.ensmp.fr/] at the Ecole des Mines in Paris. It is a set-theoretic method of image analysis providing a quantitative description of geometrical structures. (At the Ecole des Mines they were interested in analysing geological data and the structure of materials). Morphology can provide boundaries of objects, their skeletons, and their convex hulls. It is also useful for many pre- and post-processing techniques, especially in edge thinning and pruning.

Generally speaking most morphological operations are based on simple expanding and shrinking operations. The primary application of morphology occurs in binary images, though it is also used on grey level images. It can also be useful on range images. (A range image is one where grey levels represent the distance from the sensor to the objects in the scene rather than the intensity of light reflected from them).

4.2 Set Operations

The two basic morphological set transformations are *erosion* and *dilation*. These transformations involve the interaction between an image A (the object of interest) and a structuring set B, called the *structuring element*.

Typically the structuring element B is a circular disc in the plane, but it can be any shape. The image and structuring element sets need not be

restricted to sets in the 2D plane, but could be defined in 1, 2, 3 (or higher) dimensions.

Let *A* and *B* be subsets of Z^2. The *translation* of *A* by *x* is denoted A_x and is defined as

$$A_x = \{c : c = a + x, \text{for } a \in A\}.$$

The *reflection* of *B*, denoted $\hat{B}$, is defined as

$$\hat{B} = \{x : x = -b, \text{for } b \in B\}.$$

The complement of *A* is denoted A^c, and the difference of two sets *A* and *B* is denoted *A* - *B*.

4.3 Erosion

Erosion of the object *A* by a structuring element *B* is given by

$$A \ominus B = \{x : B_x \subseteq A\}.$$

The *translation* of *B* by *x* is denoted B_x and is defined as

$$B_x = \{c : c = b + x, \textit{ for } b \in B\}$$

Binary erosion uses the following for its mask:

1 1 1

1 1 1

1 1 1

This means that every pixel in the neighborhood must be 1 for the output pixel to be 1. Otherwise, the pixel will become 0. No matter what value the neighboring pixels have, if the central pixel is 0 the output pixel is 0. Just a single 0 pixel anywhere within the neighborhood will cause the output pixel to become 0. Erosion can be used to eliminate unwanted white noise pixels from an otherwise black area. The only condition in which a white pixel will remain white in the

output image is if all of its neighbors are white. The effect on a binary image is to diminish, or erode, the edges of a white area of pixels.

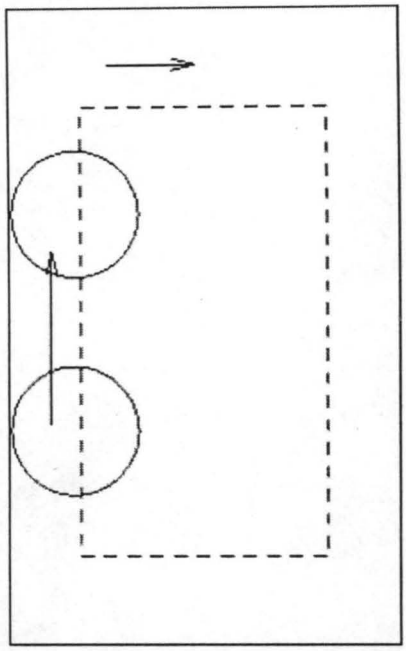


Figure 4: *A* is eroded by the structuring element *B* to give the internal dashed shape.

Dilation and erosion are duals of each other with respect to set complementation and reflection. That is,

To see this, consider first the left hand side:

$(A \ominus B)^c = \{x : B_x \subseteq A\}^c.$

Now, if B_x is contained in *A*, then $B_x \cap A^c = \emptyset$, and so

$(A \ominus B)^c = \{x : B_x \cap A^c = \emptyset\}^c.$

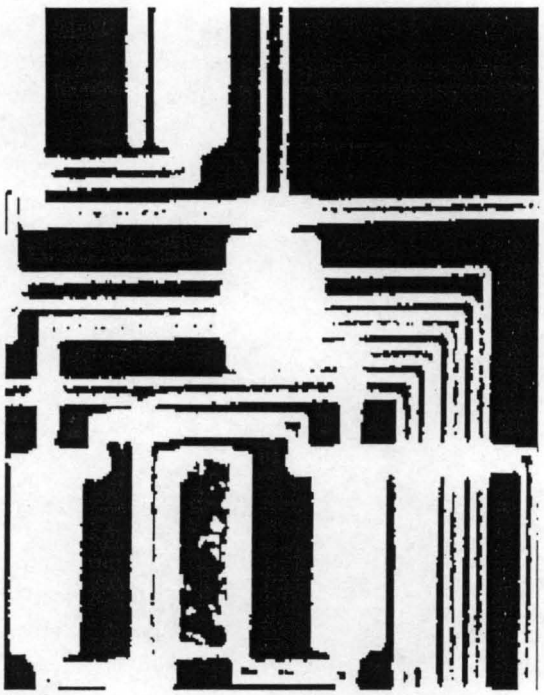
But the complement of the set $B_x \cap A^c = \emptyset$ of all *x*s that satisfy is just the set of all *x*s such that

$B_x \cap A^c \neq \emptyset$

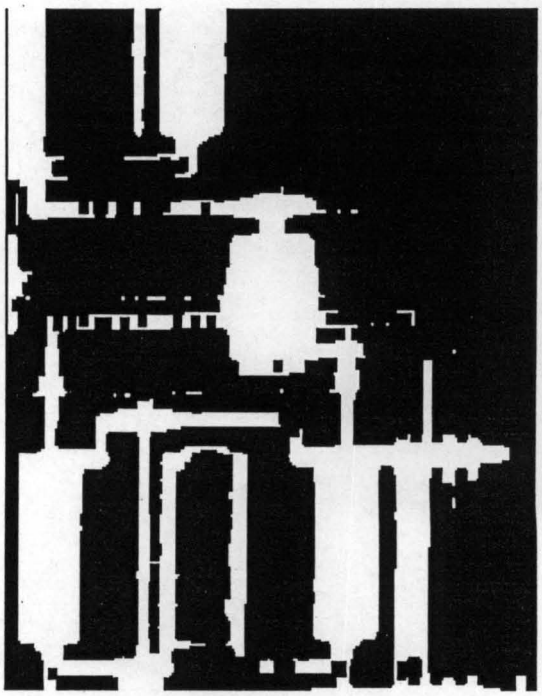
Application

MATLAB CODE

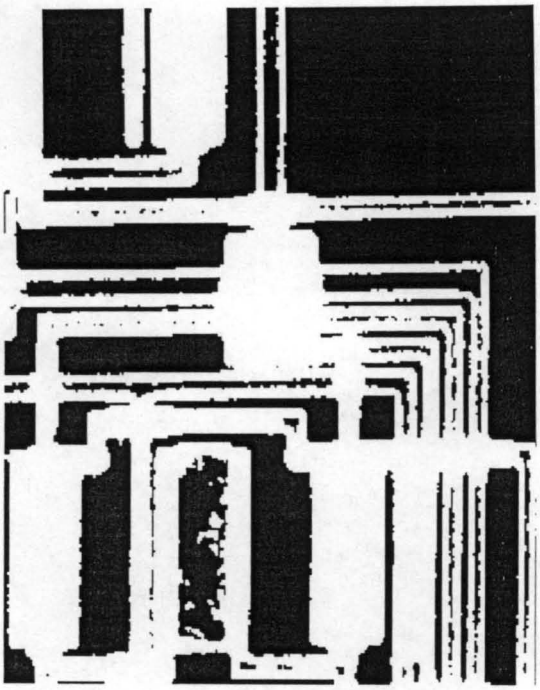
```
BW1 = imread('circbw.tif');
SE=eye(5);
BW2=erode ((BW1,SE);
IMSHOW (BW1)
FIGURE, imshow (BW2)
```



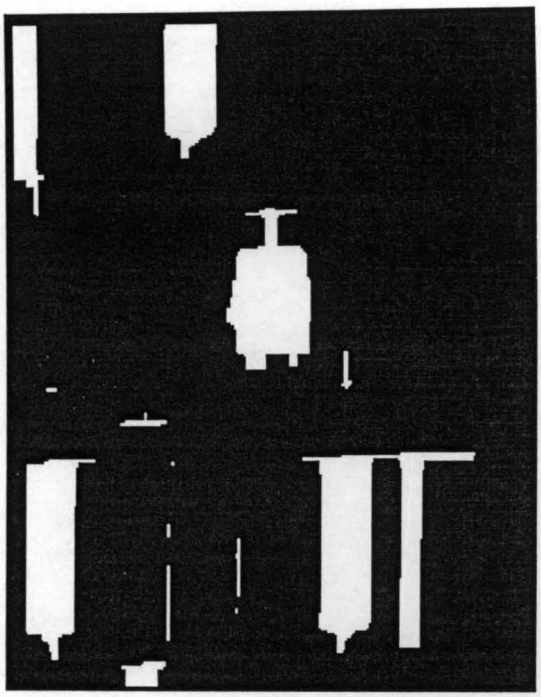
Before



After (Using a structuring element of 5x5 matrix)



Before



After (Using a structuring element of 12x12 matrix)

4.4 Dilation

Dilation of the object *A* by the structuring element *B* is given by

$$A \oplus B = \{x : \hat{B}_x \cap A \neq \emptyset\}.$$

Dilation is the opposite of erosion. Its mask is:

0 0 0

0 0 0

0 0 0

This mask will make white areas grow, or dilate. The same rules that applied to erosion conditions apply to dilation, but the logic is inverted - use the NAND rather than the AND logical operation. Being the opposite of erosion, dilation will allow a black pixel to remain black only if all of its neighbors are black. This operator is useful for removing isolated black pixels from an image. The result is a new set made up of all points generated by obtaining the reflection of *B* about its origin and then shifting this reflection by *x*.

Consider the example where A is a rectangle and B is a disc centred on the origin. (Note that if B is not centred on the origin we will get a translation of the object as well.) Since B is symmetric, $\hat{B} = B$. This definition becomes very intuitive when the structuring element B is viewed as a convolution mask.

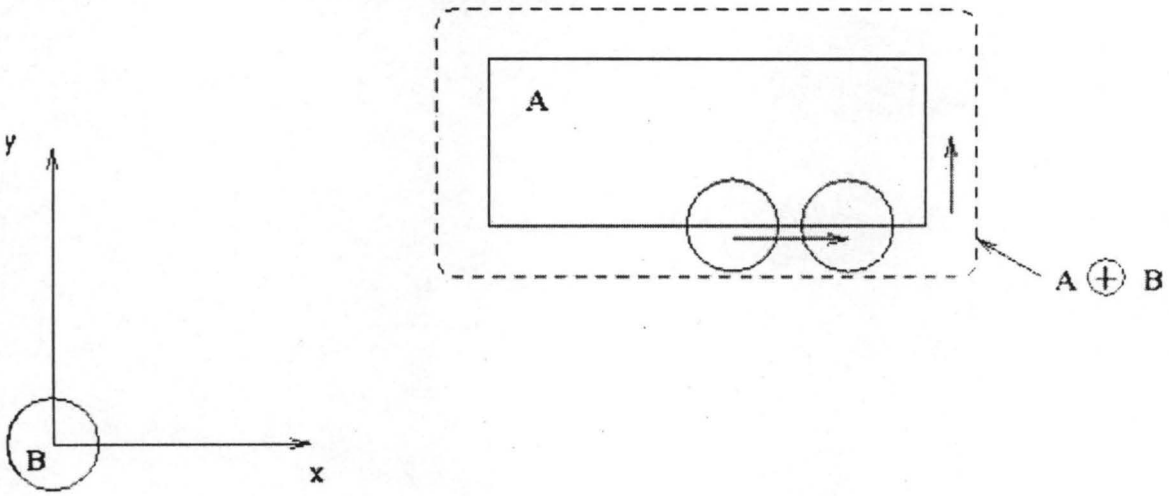


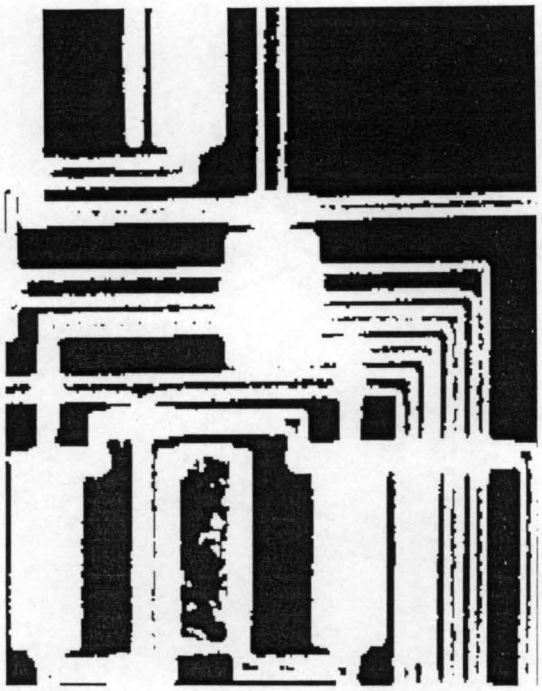
Figure 3: A is dilated by the structuring element B.

Application

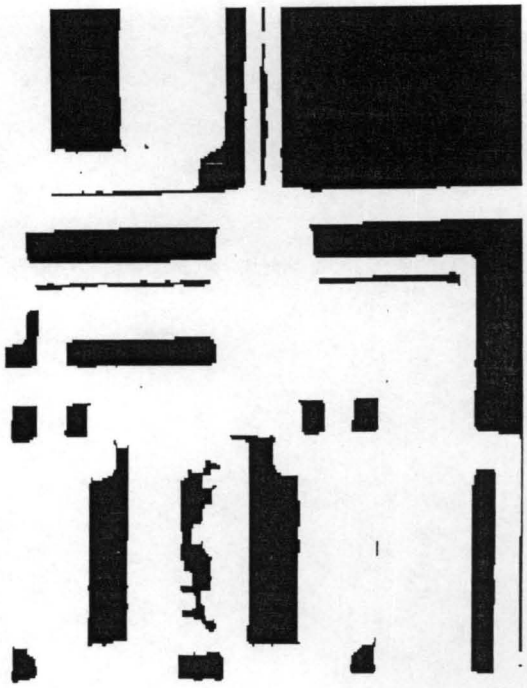
MATLAB CODE

```
BW1 = imread('circbw.tif');
SE = zeros(5,5);
BW2 = dilate(BW1,SE);
imshow (BW1)
figure, imshow (BW2)
```

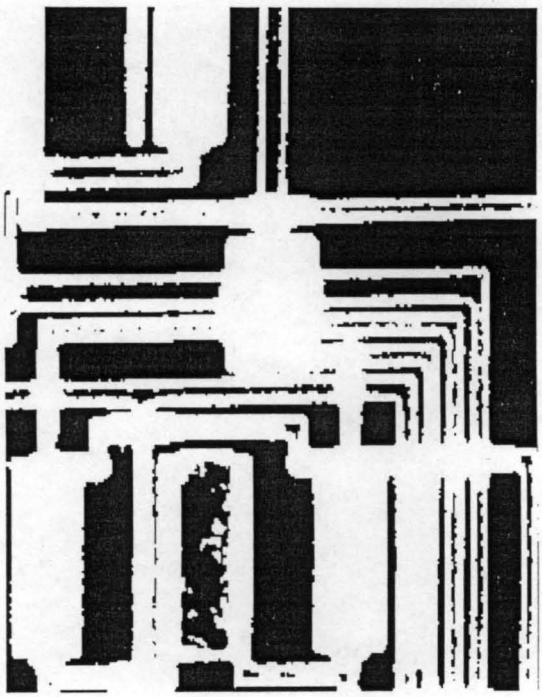
DILATION



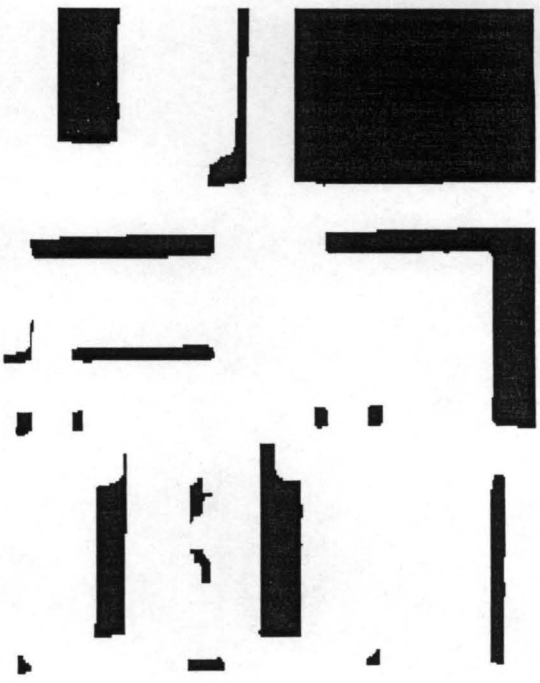
Before



After (Using a structuring element of 5x5 matrix)



Before



After (Using a structuring element of 12x12 matrix)

4.5 Other Morphological Operations

Erosion and dilation can be used in a variety of ways, in parallel and series, to give other transformations including thickening, thinning, skeletonisation and many others.

Two very important transformations are *opening* and *closing*. Now intuitively, dilation expands an image object and erosion shrinks it. Opening generally smoothens a contour in an image, breaking narrow isthmuses and eliminating thin protrusions. Closing tends to narrow smooth sections of contours, fusing narrow breaks and long thin gulfs, eliminating small holes, and filling gaps in contours.

4.5.1 Opening

The opening of A by B, denoted by $A \circ B$, is given by the erosion by B, followed by the dilation by B, that is

$$A \circ B = (A \ominus B) \oplus B.$$

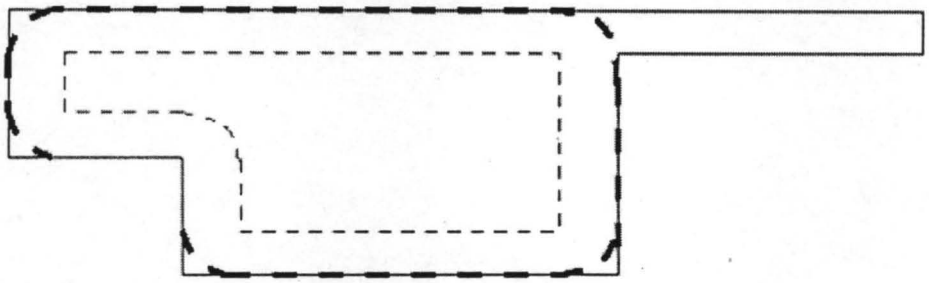


Figure 5: The opening (given by the dark dashed lines) of A (given by the solid lines. The structuring element B is a disc. The internal dashed structure is A eroded by B.

Opening is like `rounding from the inside': the opening of A by B is obtained by taking the union of all translates of B that fit inside A. Parts of A that are smaller than B are removed. Thus

$$A \circ B = \bigcup \{B_x : B_x \subseteq A\}.$$

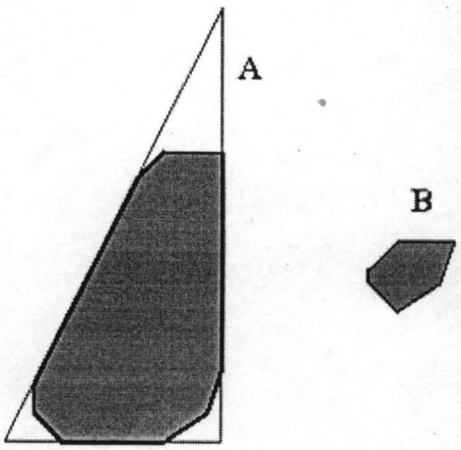


Figure 6: The opening of *A* by the structuring element *B*.

4.5.2 Closing

Closing is the dual operation of opening and is denoted by $A \bullet B$. It is produced by the dilation of *A* by *B*, followed by the erosion by *B*:

$$A \bullet B = (A \oplus B) \ominus B.$$

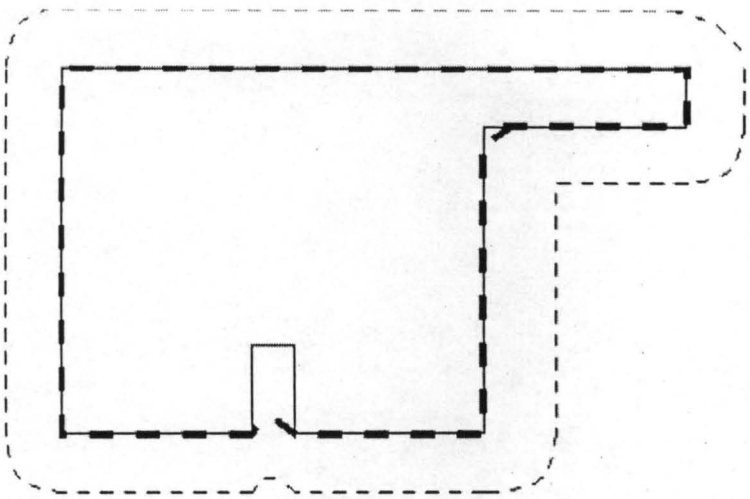


Figure 7: The closing of *A* by the structuring element *B*. This is like `smoothing from the outside'. Holes are filled in and narrow valleys are `closed'.

Just as with dilation and erosion, opening and closing are dual operations. That is

$(A \bullet B)^c = (A^c \circ B^c)$.

The opening operation satisfies the following properties:

a. $A \circ B$ is a subset of A.

b. If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$.

c. $(A \circ B) \circ B = A \circ B$

.

Similarly

1. A is a subset of $A \bullet B$.
2. If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$.
3. $(A \bullet B) \bullet B = A \bullet B$

.

Property 3, in both cases, is known as *idempotency*. It means that any application of the operation more than once will have no further effect on the result.

The morphological filter $(A \circ B) \bullet B$ can be used to eliminate `salt and pepper' noise. Salt and pepper noise is random, uniformly distributed small noisy elements often found corrupting real images. It will appear as black dots or small blobs on a white background, and white dots or small blobs on the black object. The background noise is eliminated at the erosion stage, under the assumption that all noise components are physically smaller than the structuring element B. Erosion on its own will increase the size of the noise components on the object. However, these are eliminated at the closing operation.

The important thing to note is that morphological operations preserve the main geometric structures of the object. Only features `smaller than' the structuring element are affected by transformations. All other

features at `larger scales' are not degraded. (This is not the case with linear transformations, such as convolution).

The *boundary* of a set *A*, denoted ∂A, can be obtained by first eroding *A* with *B*, where *B* is a suitable structuring element, and then performing the set difference between *A* and its erosion. That is

$$\partial A = A - (A \ominus B).$$

Typically, *B* would be a 3 x 3 matrix of 1s.

Region filling can be accomplished iteratively using dilations, complementation, and intersections. Suppose we have an image *A* containing a subset whose elements are 8-connected boundary points of a region. Beginning with a point *p* inside the boundary, the objective is to fill the entire region with 1s.

Since, by assumption, all non-boundary points are labeled 0, we begin by assigning 1 to *p*, and then construct

$$X_k = (X_{k-1} \oplus B) \cap A^c, \text{for } k = 1, 2, \ldots$$

where $X_0 = p$, and *B* is the `cross' structuring element shown in figure 8. The algorithm terminates when $X_k = X_{k-1}$. The set union of X_k and *A* contains the filled set and its boundary.

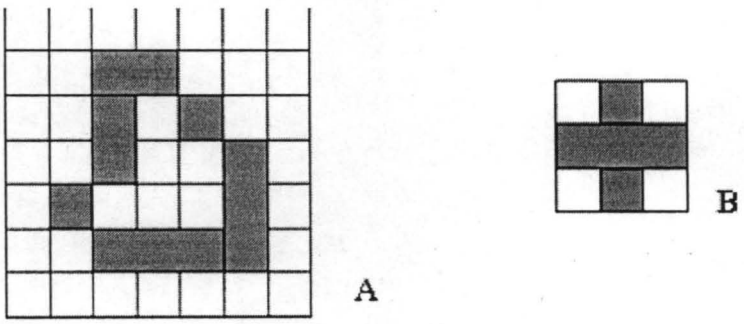


Figure 8: The region in *A* is filled using the structuring element *B*.

Likewise, *connected components* can also be extracted using morphological operations. If *Y* represents a connected component in

an image A and a point p in Y is known, then the following iterative expression yields all the elements of Y:

$$X_k = (X_{k-1} \oplus B) \cap A, \text{for } k = 1, 2, \ldots$$

where $X_0 = p$ and B is a 3×3 matrix of 1s. If $X_k = X_{k-1}$ the algorithm has converged and we let $Y = X_k$.

An important step in representing the structural shape of a planar region is to reduce it to a graph. This is very commonly used in robot path planning. This reduction is most commonly achieved by reducing the region to its *skeleton*.

The skeleton of a region is defined by the medial axis transformation (MAT). The MAT of a region R with border B is defined as follows: for each point p in R, we find its closest neighbour in B. If p has more than one such closest neighbour, then p belongs to the medial axis (or skeleton) of R. Of course, closest depends on the metric used. Figure 9 shows some examples with the usual Euclidean metric.

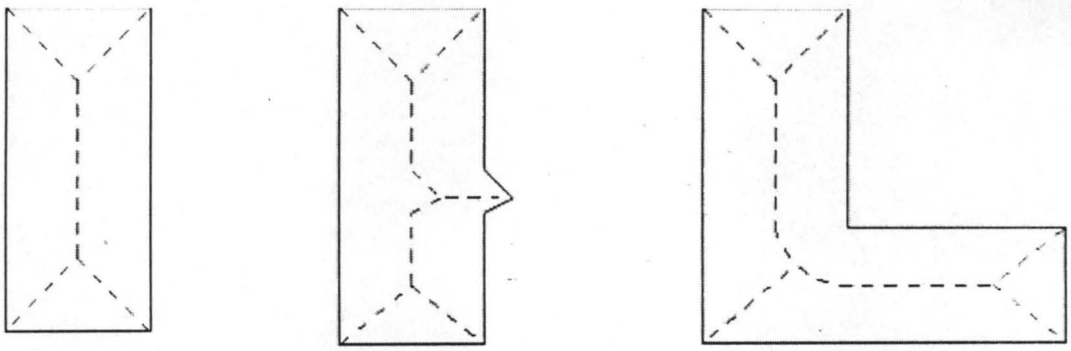


Figure 9: The skeletons of three simple regions

Direct implementation of the MAT is computationally prohibitive. However, the skeleton of a set can be expressed in terms of erosions and openings. Thus, it can be shown that

$$S(A) = \bigcup_{k=0}^{K} S_k(A),$$

where

$$S_k(A) = \bigcup_{k=0}^{K} \{(A \ominus kB) - [(A \ominus kB) \circ B],$$

$(A \ominus kB)$ indicates k successive erosions of A, and K is the last iterative step before A erodes to an empty set.

Thus A can be reconstructed from its skeleton subsets $S_k(A)$ using the equation

$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB),$$

where

$S_k(A) \oplus kB$

represents k successive dilations of $S_k(A)$.

4.6 Outlining

Other functions can be performed using erosion and dilation as their basic operation. One of these is *outlining*. It is possible to perform a single erosion operation and then subtract the resultant image from the original. The result will be an image that shows a one-pixel outline of all objects. If two erode operators are performed before the subtraction, a two-pixel outline would be created. If desired, a dilation operation can be performed before the erosion as a way to clear up any unwanted 'holes" in the white areas and may produce a cleaner outline image. This is optional because, while making the image cleaner, it might also affect the border of the original image.

4.7 Binary Hit-or-Miss Operators

Two operator masks have been discussed so far, one filled with l's to perform erosion and another filled with 0's to perform dilation. There are other masks that could be useful for other types of conditional processing. For example, the following masks can be used to check to see if a pixel is four-connected to its neighbors:

```
0 0 0    0 1 0    0 0 0    0 0 0
0 1 1    0 1 0    1 1 0    0 1 0
0 0 0    0 0 0    0 0 0    0 1 0
```

A similar set of masks can be used to check for eight-connectivity. Bridges, which are defined to be single-pixel connections between groups of similar pixels, can be identified by the following masks:

```
1 0 1    1 1 1
1 1 1    0 1 0
1 0 1    1 1 1
```

There also are masks that check for corners or interior pixels or other conditions.

Performing multiple passes on the same image to check for every possible condition of interest can become time consuming. To solve this problem, a concept can be borrowed from the image point operators - look-up tables. Because each pixel in a binary image is either one or zero, it can become a bit that is grouped with other pixels in the neighborhood to form a numerical value. The neighborhood of 9 binary pixels becomes a 9-bit number that can be used as an index into a look-up table to determine if the output pixel should be a hit or a miss. This table is known as a 9-to-1 LUT since the 9-bit input value results in a 1-bit output value. The table has 512

entries, the number of possible conditions of the 3 x 3 binary pixel neighborhood.

Obviously, the challenge of using this technique is generating the proper look-up table, because all possible conditions of pixel neighborhoods must be considered. Once this task is completed, however, the resultant processing is much faster.

4.8 Pipelined Processing

A number of morphological operators have been performed by applying a single 3 x 3 pixel mask. There are others, such as *shrinking*, *thinning*, and *skeletonization*, for which 3 x 3 will not suffice. A 5 x 5 mask is needed to perform these functions. But that mask size creates over 33 million conditional patterns that must be checked for each pixel! A very efficient method is to use a two-stage pipeline processing technique, with both stages using 3 x 3 masks [http://www.homepages.inf.ed.ac.uk/]. The first stage of the procedure is to process an image, checking for pixels that might be operated upon. This first stage of the pipeline generates a new binary image that marks the likely candidates. The second stage of the pipeline then uses the original binary image and the marked image to determine whether each pixel is a hit or a miss for the desired function. The look-up table method of processing is used, so these checks become very fast. The result is performing the equivalent of 33 million checks per pixel in two passes of a look-up table.

4.9 Shrinking

Shrinking will reduce objects in a binary image to a single point located at the geometric center of the object. This can be thought of as finding the center of mass of an object. For objects that do not have

holes in them, a single point is generated. If there is a hole, the process will produce a ring of pixels that surrounds the hole and is equidistant from the nearest boundary.

4.10 Thinning

The *thinning* function is similar to shrinking, except that thinning generates a minimally connected line that is equidistant from the boundaries. Some of the structure of the object is maintained. Thinning also is useful when the binary sense of the image is reversed, creating black objects on a white background. If the thinning function is used on this revered image, the results, are minimally connected lines that form equidistant boundaries between the objects.

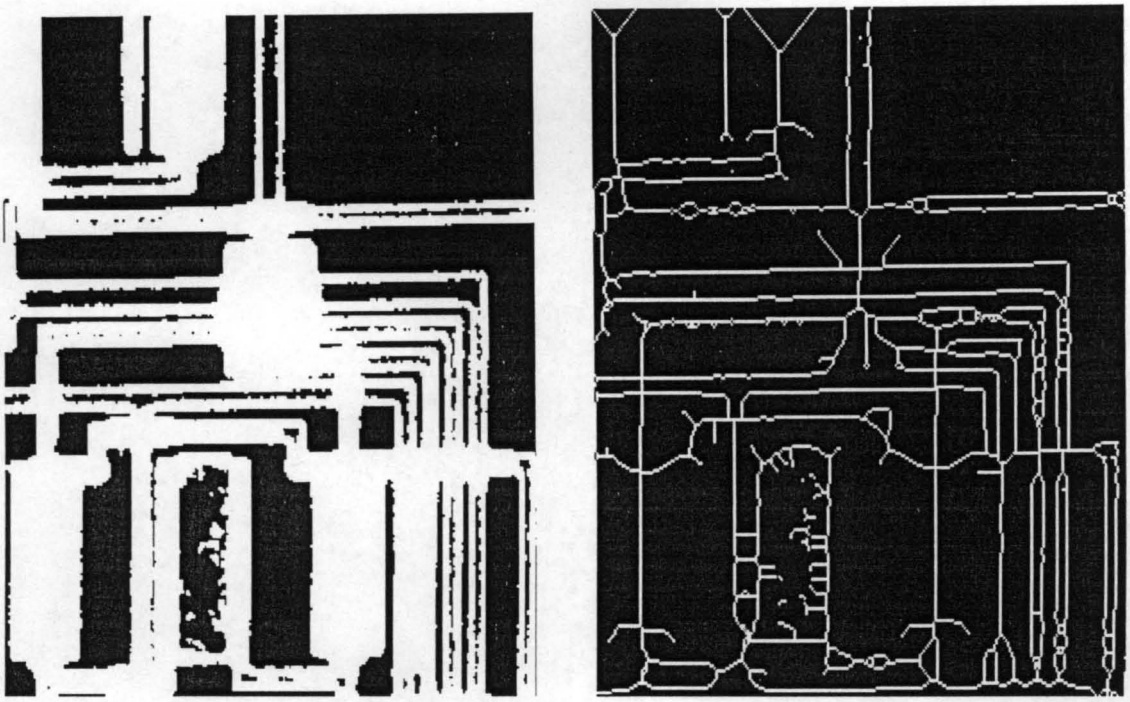
4.11 Skeletonization

Skeletonization also is similar to thinning, except that it maintains more information about the internal structure of objects. The classic way to think about skeletonization is to set fire (mentally, of course) to pixels around the outer edge of an object simultaneously. As the fire burns inward toward the center of the object, eventually it will meet burning pixels from the opposite direction. When two opposing fires meet they extinguish one another, leaving behind a single (or double) pixel boundary, or skeleton, of the object.

Application

MATLAB CODE

```
BW1=imread ('circbw.tif');
BW2 = bwmorph (BW1, 'skel', inf);
Imwrite (BW2, 'skeleton.tif')
Imshow (BW1)
Figure, imshow (BW2)
```



4.12 Grayscale Morphological Operations

While morphological operations usually are performed on binary images, some processing techniques also apply to grayscale images. These operations are for the most part limited to erosion and dilation. Grayscale erosions and dilations produce results identical to the nonlinear minimum and maximum filters.

The minimum operator will interrogate a 3 x 3 (or any other size) neighborhood and select the smallest pixel value to become the output value. This has the effect of causing the bright areas of an image to shrink, or erode. Similarly, grayscale dilation is performed by using the maximum operator to select the greatest value in a neighborhood.

Morphological functions that are based on hit-or-miss processing, such as thinning and skeletonization, do not translate well to grayscale images.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

This work is an investigation into how images can be formed on the one hand and image analysis on the other. The work was undertaken because of the tremendous usefulness of the science of fractals and image analysis. These fields are young and have promise of greater usefulness and applications in the future. We have seen that for most natural objects are indeed fractals of some sort. After Mandlebrot plotted the first fractal, fractals have continued to be discovered not just in the form of arts or synthesized landscapes but also used in solving problems such as percolation problems [Feder, 1996], stock market behaviour [Mandelbrot, 2004] and other events for which results seemed unpredictable and chaotic such as weather behaviour.

What can be said about fractals is that even though the shapes generated look so intricate, yet they obey sound mathematical principles. In general, for a fractal to form the following conditions must be met.

a. Its dimension must be a fraction, not an integer
b. there must be an iteration scheme
c. the iteration scheme must be convergent
d. in particular, the scheme must obey the contraction mapping principle
e. As a result of (3), there must be an attractor i.e. a final image that forms (the limiting image). This is analogous to the limiting point of a sequence.
f. The sequence of images must be compact. This means that any sequence of images must also converge within the initial image set.

While the technology does not yet exist to derive the attractor (which can only be obtained after an infinite number of iterations), we can come within a reasonable neighbourhood of the attractor in our construction.

Erosion and dilation are the basic operations of mathematical morphology. All other operations of mathematical morphology are based on different combinations of erosion and dilation. Erosion is sometimes called Minkowski subtraction and dilation is called Minkowski addition [Serra, 1996]. Mathematical morphology is based on binary images i.e. the image is in black and white form (although

REFERENCES

1. Barnsely, N.F. (1993). Fractals Everywhere 2nd Edition. Academy Press, New York

2. Bloch, I., Maitre, H. (1995). Fuzzy Mathematical Morphology: A Comparative Study. In Pattern, Vol 28, No. 9 pp 1341-1387

3. Briggs, John (1992). Fractals: The pattern of Chaos. Touchstone Publishers, New York.

4. Crownover, Richard M. (1995). Introduction to Fractals and Chaos. Jones And Bartlett, Ontario

5. De Baets, B. Kerre, E. Gupta, M. (1995). The Fundamentals of Fuzzy Mathematical Morphology. In: International Journal of General Systems, Vol 23, Pp 155-171

6. Di Gesu, V. (1988). Mathematical Morphology and Image Analysis. In: Proceedings of Workshops on Knowledge Based Systems and Modules for Logical Reasoning, Agyten

7. Falconer, K.J. (1989). Fractal Geometry: Mathematical Foundations and Applications. Wiley And Sons, New York

8. Feder, Jens (1988). Fractals. Plenum Press, London

9. Guichard, F. (2000). Problems in Image Processing: Image Iterative Smoothing and PDEs. International Centre For Theoretical Physics, Trieste

10. Kryszig, Erwin (1978). Introductory Functional Analysis. John Wiley and Sons, Canada

11. Lauerier, H. (1991). Fractals: Geometrical Figures. Penguin Books, Sydney

12. Parker, James R. (1996). Algorithms for Image Processing and Computer Vision. John Wiley And Sons, Canada

13. Peitgen,H.O. and P.H. Richter (1986). The Beauty of Fractals. Springer Verlag, New York

14. Peitgen, H.O. and Saupe, D (1998). The Science of Fractals. Springer Verlag, New York

15. Peitgen, H.O., Jurgens, H. and Saupe, D. (1992). Fractals for the Classroom Vol 1. Springer Verlag, New York.

16. Peitgen, H.O., Jurgens, H. and Saupe, D. (1992). Fractals for the Classroom Vol 2. Springer Verlag, New York

17. Richard, A.P. (1997). Mathematical Morphology for Angel-Valued Images. In: Proceedings of The SPIE, Non-Linear Image Processing VIII

18. Roerdink, H.B.T.M (1996). Computer Vision and Mathematical Morphology, Computing Supplement, pp 131-148

19. Roerdink, H.B.T.M (2000). Pattern Recognition 33(6) pp 877-895

20. Serra, Jean (1996). Course on Mathematical Morphology. Sydney International Statistical Congress, Sydney, Australia.

21. Serra,Jean and Soille, Pierre (1994).Mathematical Morphology and Its Applications To Image Processing (Computational Imaging and Vision Vol 2). Kluwer Academic Press, London

22. Van Droogenbooeck, M. (2002). Algorithms for Opening of Binary and Label Images with Rectangular Structuring Element. In: H. Talbert and R. Beare, Editors, Mathematical Morphology pp 197-207. CSRIO Publishing, Sydney, Australia

23. Van Droogenbooeck, M. (1994).On The Implementation of Morphological Operations. In J. Serra and P. Soille, Editors, Mathematical Morphology and Its Applications to Image Processing Pp 241-248. Kluwer Academic Press, Dordretch

Web Sites Visited

24. http://www.umr.edu/~bunyak/vision/morph.htm
25. http://www.mmorph.com/resources.htm
26. http://www.dcc.unicomp.bv/~pascual/mm.html
27. http://www.ime.usp.br/~cesar/revision/morpho.htm
28. http://www.cmis.csiro.au/ismm2002/proceedings/PDF/introduction.htm

29. http://www.reutrino.co.jp/abincivi/PDF
30. http://www.efg2.com/lab/hiloray/image processing algorithms.htm
31. http://www.industrialvision.co.uk
32. http://www.math.rice.edu/~lanius/frac/
33. http://spanky triumph.ca
34. http://www.mbfractals.com/
35. http://www.fractals.com/
36. http://www.math.umass.edu.nmconnors/fractal.html
37. http://www.mathjmandl.org/chaos/
38. http://www.made.bn.edu/DYSYS/arcadia/
39. http://www.fractalarts.com/ASF/
40. http://cmm.ensmp.fr/Recherche/pages/nav0b.htm
41. http://www.homepages.inf.ed.ac.uk/

APPENDIX

PROGRAM LISTING

THE MANDELBROT SET (C++ PROGRAM)

```
// MainFrm.cpp : implementation of the CMainFrame class

#include "stdafx.h"
#include "Mandel.h"

#include "MainFrm.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////
/////////////////
// CMainFrame

IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
     //{{AFX_MSG_MAP(CMainFrame)
          // NOTE - the ClassWizard will add and remove
mapping macros here.
          //    DO NOT EDIT what you see in these blocks of
generated code !
     //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////
/////////////////
// CMainFrame construction/destruction

CMainFrame::CMainFrame()
{
     // TODO: add member initialization code here

}

CMainFrame::~CMainFrame()
{
}

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
```

```
    if( !CFrameWnd::PreCreateWindow(cs) )
        return FALSE;
    // TODO: Modify the Window class or styles here by
modifying
    //  the CREATESTRUCT cs

    return TRUE;
}

/////////////////////////////////////////////////////////////
/////////////////
// CMainFrame diagnostics

#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const
{
    CFrameWnd::Dump(dc);
}

#endif //_DEBUG

/////////////////////////////////////////////////////////////
/////////////////
// CMainFrame message handlers

// MandelView.cpp : implementation of the CMandelView class
//

#include "stdafx.h"
#include "Mandel.h"

#include "MandelDoc.h"
#include "MandelView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

// define Mandelbrot set constants:
#define CIMAX 1.2
```

```
#define CIMIN -1.2
#define CRMAX 1.0
#define CRMIN -2.0
#define NMAX 128

// colors used to create Mandelbrot pattern:
DWORD ColorTable [6] =
     {    0x0000ff, // red
          0x00ff00, // green
          0xff0000, // blue
          0x00ffff, // yellow
          0xffff00, // cyan
          0xff00ff};     // magenta

/////////////////////////////////////////////////////////////////
/////////////////
// CMandelView

IMPLEMENT_DYNCREATE(CMandelView, CView)

BEGIN_MESSAGE_MAP(CMandelView, CView)
     //{{AFX_MSG_MAP(CMandelView)
     ON_WM_SIZE()
     //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////
/////////////////
// CMandelView construction/destruction

CMandelView::CMandelView()
{
     // TODO: add construction code here
     m_Col = 0;
}

CMandelView::~CMandelView()
{
}

BOOL CMandelView::PreCreateWindow(CREATESTRUCT& cs)
{
     // TODO: Modify the Window class or styles here by
modifying
     //  the CREATESTRUCT cs

     return CView::PreCreateWindow(cs);
```

```
}
/////////////////////////////////////////////////////////////////////////
//////////////////
// CMandelView drawing
void CMandelView::OnDraw(CDC* pDC)
{
     CMandelDoc* pDoc = GetDocument();
     ASSERT_VALID(pDoc);
     // TODO: add draw code for native data here
     m_Col = 0;
     m_CR = (float)CRMIN;
}
/////////////////////////////////////////////////////////////////////////
//////////////////
// CMandelView diagnostics

#ifdef _DEBUG
void CMandelView::AssertValid() const
{
     CView::AssertValid();
}
void CMandelView::Dump(CDumpContext& dc) const
{
     CView::Dump(dc);
}

CMandelDoc* CMandelView::GetDocument() // non-debug version
is inline
{
     ASSERT(m_pDocument-
>IsKindOf(RUNTIME_CLASS(CMandelDoc)));
     return (CMandelDoc*)m_pDocument;
}
#endif //_DEBUG

/////////////////////////////////////////////////////////////////////////
//////////////////
// CMandelView message handlers

void CMandelView::OnSize(UINT nType, int cx, int cy)
{
     CView::OnSize(nType, cx, cy);

     // TODO: Add your message handler code here

     if(cx <= 1 || cy <= 1) // avoid divide-by-zero
          return;
```

```
    m_ColMax = cx;
    m_RowMax = cy;

    m_DCR = (float)((CRMAX - CRMIN) / (m_ColMax - 1));
    m_DCI = (float)((CIMAX - CIMIN) / (m_RowMax - 1));
}
void CMandelView::DrawCol()
{
    CClientDC ClientDC (this);
    float CI;
    int ColorVal;
    float I;
    float ISqr;
    float R;
    int Row;
    float RSqr;

    if(m_Col >= m_ColMax || GetParentFrame()->IsIconic())
        return;

    CI = (float)CIMAX;
    for(Row = 0; Row < m_RowMax; ++Row)
    {
        R = (float)0.0;
        I = (float)0.0;
        RSqr = (float)0.0;
        ISqr = (float)0.0;
        ColorVal = 0;
        while(ColorVal < NMAX && RSqr * ISqr < 4)
        {
            ++ColorVal;
            RSqr = R * R;
            ISqr = I * I;
            I *= R;
            I += I + CI;
            R = RSqr - ISqr + m_CR;
        }
        ClientDC.SetPixelV(m_Col, Row,
ColorTable[ColorVal % 6]);
        CI -= m_DCI;
    }
    m_Col++;
    m_CR += m_DCR;
}
```

```
// Mandel.cpp : Defines the class behaviors for the
application.
//

#include "stdafx.h"
#include "Mandel.h"

#include "MainFrm.h"
#include "MandelDoc.h"
#include "MandelView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////
/////////////////
// CMandelApp

BEGIN_MESSAGE_MAP(CMandelApp, CWinApp)
     //{{AFX_MSG_MAP(CMandelApp)
     ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
          // NOTE - the ClassWizard will add and remove
mapping macros here.
          //    DO NOT EDIT what you see in these blocks of
generated code!
     //}}AFX_MSG_MAP
     // Standard file based document commands
     ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
     ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////
/////////////////
// CMandelApp construction

CMandelApp::CMandelApp()
{
     // TODO: add construction code here,
     // Place all significant initialization in
InitInstance
}

/////////////////////////////////////////////////////////////
/////////////////
```

```
// The one and only CMandelApp object

CMandelApp theApp;

/////////////////////////////////////////////////////////////////////////
/////////////////
// CMandelApp initialization

BOOL CMandelApp::InitInstance()
{
     // Standard initialization
     // If you are not using these features and wish to
reduce the size
     //  of your final executable, you should remove from
the following
     //  the specific initialization routines you do not
need.

#ifdef _AFXDLL
     Enable3dControls();           // Call this when using
MFC in a shared DLL
#else
     Enable3dControlsStatic();     // Call this when
linking to MFC statically
#endif

     // Change the registry key under which our settings
are stored.
     // TODO: You should modify this string to be something
appropriate
     // such as the name of your company or organization.
     SetRegistryKey(_T("Local AppWizard-Generated
Applications"));

     LoadStdProfileSettings();  // Load standard INI file
options (including MRU)

     // Register the application's document templates.
Document templates
     //  serve as the connection between documents, frame
windows and views.

     CSingleDocTemplate* pDocTemplate;
     pDocTemplate = new CSingleDocTemplate(
          IDR_MAINFRAME,
          RUNTIME_CLASS(CMandelDoc),
```

```
          RUNTIME_CLASS(CMainFrame),       // main SDI
frame window
          RUNTIME_CLASS(CMandelView));
     AddDocTemplate(pDocTemplate);

     // Parse command line for standard shell commands,
DDE, file open
     CCommandLineInfo cmdInfo;
     ParseCommandLine(cmdInfo);

     // Dispatch commands specified on the command line
     if (!ProcessShellCommand(cmdInfo))
          return FALSE;

     // The one and only window has been initialized, so
show and update it.
     m_pMainWnd->ShowWindow(SW_SHOW);
     m_pMainWnd->UpdateWindow();

     m_pMainWnd->SetWindowText("Mandelbrot Demo");

     return TRUE;
}


/////////////////////////////////////////////////////////////////////
/////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
     CAboutDlg();

// Dialog Data
     //{{AFX_DATA(CAboutDlg)
     enum { IDD = IDD_ABOUTBOX };
     //}}AFX_DATA

     // ClassWizard generated virtual function overrides
     //{{AFX_VIRTUAL(CAboutDlg)
     protected:
     virtual void DoDataExchange(CDataExchange* pDX);    //
DDX/DDV support
     //}}AFX_VIRTUAL

// Implementation
```

```
protected:
     //{{AFX_MSG(CAboutDlg)
          // No message handlers
     //}}AFX_MSG
     DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
     //{{AFX_DATA_INIT(CAboutDlg)
     //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
     CDialog::DoDataExchange(pDX);
     //{{AFX_DATA_MAP(CAboutDlg)
     //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
     //{{AFX_MSG_MAP(CAboutDlg)
          // No message handlers
     //}}AFX_MSG_MAP
END_MESSAGE_MAP()

// App command to run the dialog
void CMandelApp::OnAppAbout()
{
     CAboutDlg aboutDlg;
     aboutDlg.DoModal();
}

//////////////////////////////////////////////////////////////
/////////////////
// CMandelApp message handlers


BOOL CMandelApp::OnIdle(LONG lCount)
{
     // TODO: Add your specialized code here and/or call
the base class

     CWinApp::OnIdle(lCount);

     CMandelView *PView = (CMandelView *)((CFrameWnd
*)m_pMainWnd)->GetActiveView();
```

```
    PView->DrawCol();
    return TRUE;
}
```

THE FERN LEAF (BASIC CODE)

```
SCREEN 12
INPUT "Number of Iterations (5000):", imax
left = 30
w = 300
w1 = w + left
e1 = .5 * w: e2 = .57 * w: e3 = .408 * w: e4 = .1075 * w
f1 = 0 * w: f2 = -.036 * w: f3 = .0893 * w: f4 = .27 * w
REM FIXED POINT MAP 1
x = e1
y = 0
FOR i = 1 TO imax
r = RND
REM map 1 (stem)
50 IF r > .02 GOTO 100
xn = 0 * x + 0 * y + e3
yn = 0 * x + .27 * y + f1
GOTO 400
REM map 2 (right leaf)
100 IF r > .17 GOTO 200
xn = .139 * x + .263 * y + e2
yn = .246 * x + .224 * y + f2
   GOTO 400
   REM map 3 (left)
200 IF r > .3 GOTO 300
   xn = .17 * x - .215 * y + e3
   yn = .222 * x + .176 * y + f3
GOTO 400
REM map 4 (top of fern)
300 xn = .781 * x + .034 * y + e4
    yn = -.032 * x + .739 * y + f4
REM draw game point
400 PSET (xn + left, w1 - yn)
x = xn
   y = yn
   NEXT i
   END
```

THE SIERPINSKI GASKET (BASIC PROGRAM)

```
DEFINT x,y
FOR y = 0 TO 255
     FOR x = 0 TO 255
```

```
        IF (x AND y) = 0 THEN PSET (x + 30, y + 30)
    NEXT x
NEXT y
END
```

THE KOCH CURVE (BASIC PROGRAM)

```
DIM xleft(10), xright(10), yleft(10), yright(10)
INPUT "Peak offset (0.29):", r
Level = 5
xeft(level) = 30
xight(level) = 30 + 300
yeft(level) = 190
GOSUB 100
END
REM DRAW LINE AT LOWEST LEVEL OF RECURSION
100  IF level > 1 GOTO 200
LINE (xleft(1), yleft(1) - (xright(1), yright(1))
GOTO 300
REM BRANCH INTO LOWER LEVELS
200 level = level + 1
REM LEFT BRANCH
xleft(level) = xleft(level + 1)
yleft(level) = yleft(level + 1)
xright(level) = .333*xright(level +1) + .667*xleft(level +
1)
yright(level) = .333*yright(level +1) + .667*yleft(level +
1)
GOSUB 100
REM MIDDLE LEFT BRANCH
xleft(level) = xright(level)
yeft(level) = yright(level)
xright(level) = .5*xright(level + 1) + .5xleft(level + 1) -
r*(yleft(level + 1) - yright(level + 1))

yright(level) = .5*yright(level + 1) + .5yleft(level + 1) -
r*(xleft(level + 1) - xright(level + 1))
GOSUB 100
REM MIDDLE RIGHT BRANCH
xleft(level) = xright(level)
yleft(level) = yright(level)
xright(level) = .667*xright(level + 1) + .333*xleft(level +
1)
yright(level) = .667*yright(level + 1) + .333*yleft(level +
1)
GOSUB 100
REM RIGHT BRANCH
```

```
Xleft(level) = xright(level)
yleft(level) = yright(level)
xright(level) = xright(level + 1)
yright(level) = yright(level + 1)
GOSUB 100
Level = level + 1
300 RETURN
```

MATLAB CODE FOR FIG 1 & 2

```
BW1 = imread('circbw.tif');

SE=eye(5);

BW2=erode ((BW1,SE);

IMSHOW (BW1)

FIGURE, imshow (BW2)
```

MATLAB CODE FOR FIG 3 & 4

```
BW1 = imread('circbw.tif');
SE = zeros(5,5);
BW2 = dilate(BW1,SE);
imshow (BW1)
figure, imshow (BW2)
```

MATLAB CODE FOR SKELETONIZATION

```
BW1=imread ('circbw.tif');
BW2 = bwmorph (BW1, 'skel', inf);
Imwrite (BW2, 'skeleton.tif')
Imshow (BW1)
Figure, imshow (BW2)
```