TR diss 2203 S

STELLINGEN

behorende bij het proefschrift

SCALING AND TEXTURE

PYRAMIDS AND FRACTALS

door
Ernst Jan Eijlers

1. Bij het gebruik van maten afkomstig uit de fractaltheorie voor het kwantificeren van textuur blijkt dat het verondersteldelineaire verband in de zogenaamde log-log ruimte inderdaad voor een beperkte klasse van texturen gevonden wordt, maar voor een meer algemene maat dient deze eis aangepast te worden.

2. Er bestaan twee soorten van initialisatie van de piramidale datastructuur. Het eerste type is gebaseerd op de reductie van de variantie, het tweede type beschrijft de beeldfenomenen op verschillende niveaus van resolutie. Het praktisch nut van het eerste type is beperkt, juist het tweede type is over het algemeen het argument om te kiezen voor een piramidale datastructuur.

3. Ondanks dat in de literatuur het gebruik van de definitie van de box-dimensie voor de bepaling van de dimensies conform het algemene dimensie-model wordt gepropageerd voor signaalanalyse, blijkt het voor twee-dimensionale beeldverwerkingstoepassingen als lokale textuuranalyse en textuursegmentatie niet geschikt.

4. Met de massa-piramide is een snelle, op de de fractaltheorie geïnspireerde textuuranalyse-methode verkregen, die textuureigenschappen op verscheidene niveaus van resolutie kwantificeert.

5. De informatie-inhoud van een signaal zoals bepaald met de traditionele informatiematen uit de informatietheorie, kan een overwaardering tonen in vergelijking met de informatie-inhoud zoals die gevonden wordt met technieken die binnen de chaostheorie gangbaar zijn.

6. In het bijzonder voor industriële beeldverwerkingsapplicaties geldt dat een goed gebruik van werktuigbouwkundige technieken tot een reductie van de complexiteit van de beeldverwerkingsalgoritmen kan leiden.

7. Met de introductie van digitale fotografie, zoals die op dit moment plaatsvindt, nemen niet alleen de mogelijkheden van data-acquisitie en visualisatie voor de digitale beeldwerking toe, maar ook de herkenbaarheid van mogelijke toepassingen van digitale beeldverwerkingstechnieken door een groter publiek.

8. Als gevolg van de wens tot standaardisatie en het opkomen voor nationale belangen verschuift de complexiteit van ontwikkelingen op het gebied van consumentenelektronika van het technisch vlak naar het internationaal politieke vlak.

9. Het toenemend aanbod aan wetenschappelijke tijdschriften suggereert een stimulerende werking te hebben op de wetenschapsbeoefening. Door de verstikkende werking zal daarentegen een reductie van het aantal een grotere stimulans betekenen.

10. Door studies aan te bieden waarvoor geldt dat na afloop het vinden van passend werk nagenoeg uitgesloten is, gaat de onderwijsgever de verantwoordelijkheid aan om het curriculum zodanig in te richten dat de kans op werk alsnog verhoogd wordt.

11. De politieke en ethische discussie over aspecten van niet-traditioneel Nederlandse religies wordt in Nederland makkelijker gevoerd, dan die over aspecten van de traditioneel Nederlandse religies.

575229
3'708405
TR diss 2203

TR diss
2203

SCALING AND TEXTURE

pyramids and fractals

Ernst Jan Eijlers

2200

SCALING AND TEXTURE
pyramids and fractals

Proefschrift

ter verkrijging van de graad van doctor aan de
Technische Universiteit Delft, op gezag van
de Rector Magnificus, prof. drs. P.A. Schenck,
in het openbaar te verdedigen ten overstaan van
een commissie aangewezen door het College van
Dekanen op dinsdag 6 april 1993 te 16.00 uur

door

Ernst Jan Eijlers,

geboren te Woerden,
elektrotechnisch ingenieur.

BIBLIOTHEEK TECHNISCHE UNIVERSITEIT
Prometheusplein 1
2628 ZC
DELFT

Dit proefschrift is goedgekeurd
door de promotor prof.dr.ir. E. Backer.

Promotiecommissie:

Rector Magnificus
Prof.dr.ir. E. Backer (promotor)
Dr.ir. J.J. Gerbrands (toegevoegd promotor)
Prof.dr.ir. J. Biemond
Prof.dr.ir. F. W. Jansen
Prof.dr. M. Kleefstra
Prof.dr. H. Koppelaar
Dr. P.W. Verbeek

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Eijlers, Ernst Jan

Scaling and texture : pyramids and fractals / Ernst Jan
Eijlers. - [S.l. : s.n.]. -Ill.
Thesis Technische Universiteit Delft. - With ref. - With
summary in Dutch.
ISBN 90-9005909-1
NUGI 841
Subject headings: texture analysis / pyramidal
datastructures / fractals.

Contents

Summary **vii**

1 Introduction **1**
- 1.1 Digital Image Analysis 1
- 1.2 The Scope of this Thesis 5

2 Texture **7**
- 2.1 Introduction 7
- 2.2 Analysis versus Synthesis 13
- 2.3 Analysis 15
  - 2.3.1 SGLDM: The Spatial Grey Level Dependence Method 16
  - 2.3.2 GLDM: Grey Level Difference Method 19
  - 2.3.3 GLRLM: Grey Level Run Length 20
  - 2.3.4 Max-Min Measure 21
  - 2.3.5 Textural Edgeness 23
  - 2.3.6 The Long-Correlation Model 23
  - 2.3.7 Multi-Channel Filtering 24
- 2.4 Synthesis 25
- 2.5 Concluding Remarks 26

3 Pyramidal Data Structures **27**
- 3.1 Introduction 27
- 3.2 Examples of Pyramids 29
  - 3.2.1 The Grey Value Pyramid 29
  - 3.2.2 The Laplacian Pyramid 36
  - 3.2.3 The Binary Pyramid 40

3.2.4 The Linked Pyramid 42
3.3 Pyramids and Scale Space 47
3.4 Pyramids for the Study of Texture 48

4 Fractal Theory 55
4.1 Introduction 55
4.2 Fractal Brownian Motion 60
4.2.1 Brownian Motion and Fractal Brownian Motion . 60
4.2.2 Properties of the Fractal Brownian Motion Model 63
4.3 Generation Techniques 65
4.4 The Box-Counting Dimension 70
4.5 Texture Measures based on the Fractal Dimension 73
4.6 The Generalized Dimension Model 82
4.7 Texture Measures based on the Generalized Dimension Model 84
4.8 Discussion 86

5 Scaling and Texture: Pyramids and Fractals 89
5.1 Introduction 89
5.2 Thoughts on the Integration 92
5.3 The Blanket Technique expanded 93
5.4 The Box-Counting Technique 102
5.5 The Mass Pyramid 105
5.6 Discussion 118

6 Conclusions and recommendations 123

References 127

Samenvatting 143

Acknowledgements 147

Curriculum Vitae 149

Summary

With the development of computer systems that are able to process and even understand digital images, one is often confronted with scenes of which texture is part of. As texture is still undefined, attempts can be undertaken to describe it, but often it is explained by giving examples. Typical examples of textures are those of carpets, clouds, textile, leather, and so on. Although one might be tempted to define texture as a structure, it certainly does not have to be structured. It is the fact that a texture does not have to be structured to be interpreted by a human observer as being homogeneous that hampers the definition and even the description of what is meant by texture. Consequently, the design of an operator for the analysis of textures by a digital image processing system is hampered by this lack of a definition. At the initial stage of segmentation such an operator is already required. During this stage, the image is segmented into regions that are considered as homogeneous. This means that the texture operator is expected to output a constant value if the underlying texture is considered to be the same.

Some of the descriptions found in the literature explicitly mention the aspect of scaling. It is true that the appearance of a texture may change dramatically if the scale on which it is studied is changed. A wide variety of texture operators are suggested in the literature. Some of these operators have the ability to tune the level of scaling at which the texture is to be quantified. However, this requires a priori knowledge of the scale or the range of scaling in which the texture has to be studied. In practice, this knowledge is not always available, and, therefore, an analysis method is required that studies the texture on a wide range of scaling. Such a method could be based on what is called

a *pyramidal data structure* or *pyramid.* This structure consists of a number of layers that describe the image on several levels of resolution. A typical example of a pyramid is the Gaussian pyramid. If the sizes of the original image are, for instance, 256 × 256, a pyramid is made that consists of 9 layers with the dimensions: 256 × 256, 128 × 128, 64 × 64, 32 × 32, 16 × 16, 8 × 8, 4 × 4, 2 × 2, and 1 × 1. If we imagine that these layers are placed above each other, the pyramidal structure can be recognized. Now, the Gaussian pyramid is based on a resolution reduction operator that consists of a Gaussian filter and a subsampling stage. If the original input image is placed at the bottom level, the next level is obtained by applying this resolution reduction operator. In the end, we have obtained 9 low-pass filtered representations of the original image.

As the concept of pyramidal data structures is in wide use for image processing applications, a wide variety of pyramids can be found. This variety is a result of the type of information which the pyramid contains and the way that the resolution is reduced. Other examples are the binary pyramid and the Laplacian pyramid. This latter one consists of band-pass filtered representations of the original input image.

In the literature, two types of use of the pyramidal data structure can be found. The first type of use aims at the reduction of the variance in the data at the lowest level in the pyramid. This means that the variance decreases with the height in the pyramid. In practice, this is the easiest way to initialize the pyramid. The second type of use is based on the idea that levels higher in the pyramid should correspond to a coarser description of the image. In this thesis, we consider this type of use to be more according to the concept of pyramidal data structures. Generally, it appears that the design of a resolution reduction operator for this type of use is not trivial. This is certainly the case for the texture pyramid, which describes the image from a textural point of view at several levels of scaling. It is the design of such a resolution reduction operator that is the central theme of this thesis.

As we have mentioned before, some of the existing texture operators do have the ability to select the level of scaling on which the texture is studied. However, even with this ability, it is not garantueed that the initialization of the texture pyramid is a trivial problem. In practice, the border artefacts, as a result of the limited sizes of the layers with respect to the window sizes of the texture operator, will increase with the height in the pyramid. The optimal initialization of the texture pyramid should be such that the quantification of the texture properties bubbles up in the pyramid and such that the level in the pyramid stands for the level of scaling with which the texture is quantified. This new operator should be based on a model of which scaling is an intrinsic aspect. Such a model could be based on the fractal theory, central to which is the description of phenomena on different levels of scaling.

The idea of fractal theory is often illustrated with the question: "How long is the coastline of Britain?" It appears that the more accurately one measures the length, the more accurately the coastline is followed, and therefore the longer the coast length appears to be. Therefore, it is not possible to answer this apparently simple question consistently. To be able to answer this question, the *fractal dimension* has been introduced. This fractal dimension might be considered as a measure that quantifies the whimsicality of a phenomenon, a property that seems to meet our need for quantifying texture.

The fractal dimension belongs to an infinite number of dimensions that are defined by the generalized dimension model. This model has a parameter that is used as a weighting mechanism, such that different aspects of the phenomenon can be highlighted. Each value for the parameter corresponds to a dimension definition. Theoretically, it is recommended to supplement the fractal dimension with at least one other dimension. It can be proved that signals might appear dissimilar to the human observer, whereas the fractal dimensions are equal. By supplementing the fractal dimension with another dimension, an increase in the discriminability might be obtained. Therefore, the resolution reduction operator to be designed should be based on the generalized dimension model.

Now, for the design of the resolution reduction operator that is based on the idea that higher levels should correspond to a coarser description, it is required to weaken a constraint of the fractal theory regarding the expected behavior in scale space. The theory strictly prescribes the expected behavior over a wide interval. As the scale space is divided up by the pyramidal data structure into small intervals, we measure the dimensions only for these small intervals.

In this thesis, several attempts for the design of an operator are discussed. The final and most succesful one is based on the idea that grey value may be considered as mass. By dividing the mass by the total mass in the window of the operator, the mass density is obtained. From the mass density distribution in the window, features similar to those prescribed by the fractal theory are obtained.

Experiments have shown that the preformance in discriminatory sense is good. For some texture pairs, the use of two dimensions is recommended (vector texture discrimination), however, also texture pairs which need only one dimension (scalar discrimination) have been found. Further, texture pairs where texture discrimination was only possible at the lower levels of the pyramids have also been found, whereas other texture pairs could only be discriminated at higher levels. With these results, the benefit of this type of use of a pyramidal data structure for texture analysis has been proved. By weakening the constraints of the fractal theory, the operator has won in generality. Finally, it should be stressed that the method requires limited processing time and with some adaptions, the implementation into hardware could be straightforward.

Chapter 1

Introduction

1.1 Digital Image Analysis

Nowadays, digital image analysis -of which the development started in the sixties- has a considerable number of running applications in a wide variety of disciplines. The architecture of the original digital analysis systems could be considered to be rather rigid. The dataflow was one-way directed. Today, the architecture of digital image analysis sytems is based on the concepts of *knowledge guided* or *knowledge based systems*, and therefore, the dataflow is more flexible. To explain the functioning of a digital image analysis system, we can still make use of the architecture of what might be called *the first generation*, which is shown in Fig. 1.1. It must be stressed that for a number of applications, this type of architecture might still be appropriate. Therefore, speaking of generations of digital image analysis systems does not imply that the architecture of the current generation of digital image analysis systems supersedes the first generation of systems.

The analysis starts at the imaging module. This might be a video camera coupled with a frame grabber [1]. For this type of data acquisi-

---

[1] A frame grabber is a piece of hardware, which can be connected to the computer bus. It consists of a A/D converter, which is connect to the video camera -or in general the video source- and a large memory block in which the digitized image is stored. This memory can be read by the CPU of the computer. Sometimes the frame grabber has some computational power, so that image operations can be done in this memory block.

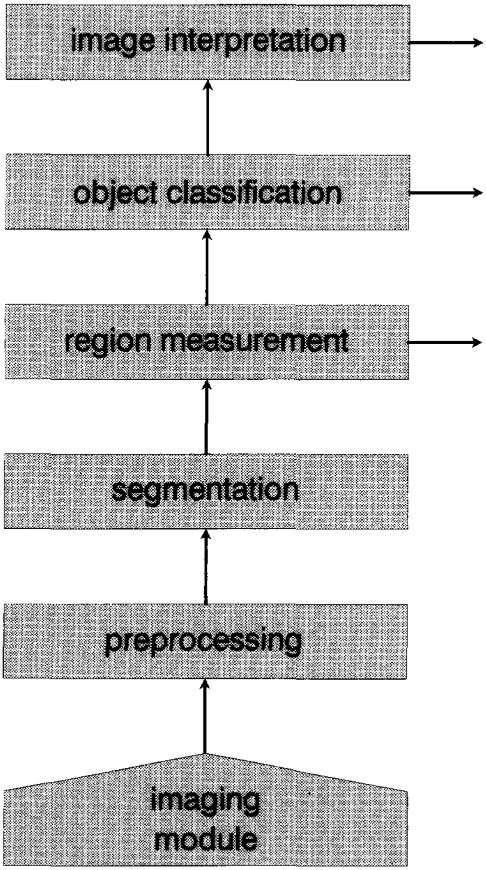

Figure 1.1: The architecture of the "first" generation digital image analysis systems.

tion, the images to be analysed have typically dimensions of 256 × 256, 512 × 512 or 1024 × 1024. The number of grey levels to be distinguished are typically 256, where 0 represents black and 255 white. Now, the term picture element (mostly abbreviated to *pixel* [2]) is used to denote a resolution cell in the image.

For some applications only single images are considered. Other applications require the analysis of a sequence of images. Such a sequence might be a sequence in time, or a sequence in place. For the latter situation, one speaks of 3-dimensional [3] image analysis. In this thesis we will mainly study 2-D images with sizes of 256 × 256 and with 256 grey levels.

Besides the video camera, other types of image sources are radars, seismic sensors, Magnetic Resonance Imaging sensors, etc. Another type of imaging unit which is sometimes forgotten are algorithms which generate a 2-dimensional output. An example of such a case is the time-varying spectrum of a 1-D signal.

After having acquired the image, a preprocessing step is mostly required. This step might consist of a number of operations. One of the first operations to think of is noise reduction. The more that is known about the noise to be expected, the more the operator can be adapted to it. Besides the reduction of noise, further enhancement of the image quality might be required. The image could be distorted by defocussing, relative motion between camera and object, lens artefacts, etc. It is the *image restoration* that focusses on these types of problems.

The preprocessing module could also consist of a transformation of the image into another domain. This might be the case if the image contains textural regions. The best way to describe such regions in this stage is as regions that are not homogeneous in grey value but in another property, such that it appears to human observers as homogeneous. See, for instance, Figs. 2.1 - 2.4 in Chapter 2 which show examples of such textures. The transformation aims to give the same texture value for all pixels in a homogeneous textural region, whereas dissimilar textural regions should result in different texture values. The resulting image is called a *texture map*. For discrimination purposes, the use of several of

---

[2] The 3-dimensional equivalent is *voxel*, which stands for *volume element.*

[3] In the sequel the abbreviation n-D is used for n-dimensional.

these texture maps might be required, they are the result of different texture operators or of one operator which has been used with different tunings.

The preprocessing module is followed by a module which segments the image in homogeneous regions. It is important that the homogeneity criterium is precisely defined. The input image might be the original image which has been freed from noise and/or distortions. It might also be one or more texture maps, which possibly might be complemented with the grey value image.

First, after having segmented the image, the real analysis can start. At this moment the representation of the image becomes more abstract. It starts with determining the properties of the regions. One could think of the average grey value, the average texture value, size, number of holes, shape, etc. For some applications the analysis is now considered to be finished.

However, the analysis might continue. Now, the relations between the regions and the properties of the regions themselves are used for recognizing objects in the image. Note that the data representation at this stage is even more abstract. An example of an analysis system for which the analysis stops at this module is a system which is used for robot vision. The image data is used for the identification of the parts to be assembled, their position on the conveyor belt, etc. The information obtained by this system can be used for guiding the robot during the assembly process.

In the final stage of the analysis the interrelationships between the objects are studied. The output of this analysis stage can be a sentence or a description. For instance, for medical application the sentence might consists of a diagnosis. This diagnosis could possibly be supplied with a certainty factor.

As mentioned before, the analysis system discussed is based on the classical concept. The output of one module is used as an input for the next module. The scheme is, therefore, considered to be rigid. There is no flexibility and no optimizing to the image contents. For instance, the preprocessing is based on general ideas of how the average image should be preprocessed. After some analysis, more precise ideas could come into being and be used to preprocess specific parts of the image with a selected operator which is fine tuned to these specific parts. The same need for flexibility counts for the segmentation module. If such feedback would exist, more precise analysis results might be obtained. The drawback of such an approach is its complexity. This complexity results in an increase in the complexity of the algorithms and an increase in the computer power demands. These flexible systems are often called *knowledge guided* or *knowledge based*, and their architecture can be based on that of a blackboard system. To return to Fig. 1.1, we may conclude that the architecture of the newest generation of image analysis systems could be sketched as shown in Fig. 1.1, but where feedback loops have been added between the several modules.

1.2 The Scope of this Thesis

In Section 1.1, we already mentioned the study of textures. The discussion was rather shallow. In the study of textures, it is important to realize that "texture" is still undefined. Therefore, there is no immediately obvious way to design an algorithm for analysis purposes. However, despite the fact that texture is still undefined, we can make many observations on it. In Chapter 2, we discuss more elaborately texture as an image processing phenomenon. It appears that the appearance of a texture differs with scale. At greater distance, coarser aspects of the texture dominate the perception; whereas on smaller scales, the finer details dominate. It is this scaling aspect that justifies the use of the *pyramidal data structures.*

A pyramidal data structure consists of a number of representations of an image. Each representation corresponds to a scaling level. For instance, if we start with an image of 256×256, the pyramidal data structure consists of representations with sizes 256×256, 128×128, 64×64, 32×32, 16×16, 8×8, 4×4, 2×2, and 1×1. By placing the representations above each other, the pyramidal structure can be recognized. The name of pyramidal data structure is often abbreviated to *pyramid.* The most elementary operator to obtain each next level in the pyramid is based on Gaussian filtering and subsampling. In Chapter 3, we discuss a number of applications of the pyramidal data structures. We will show that the initialization of a pyramid for the analysis of texture and the segmentation of images containing textural regions is not trivial. Preferably, one would start with a model in which the scaling behavior is integrated. Such a model could be based on the *fractal theory.*

The fractal theory is the subject of Chapter 4. Over the last few years, the interest of its applicability for image analysis has been rapidly growing. The strength of the fractal theory is its ability to describe phenomena which appear very complex in a very simple fashion. The description is based on a defined behavior of the phenomenon on different scales. Thus, scaling *is* intrinsic to the fractal theory and its derived models. Therefore, we study in this thesis the interrelations between texture, pyramids and fractal theory. In Chapter 5, these interrelations are discussed and the derived results examined. Finally, in Chapter 6 the reader will find the conclusions of the study underlying this thesis.

Chapter 2

Texture

2.1 Introduction

Texture plays an essential role in the visual world. Therefore, it has been a research subject from almost the beginning of digital image processing. Many images contain textural regions. Examples of such images are: satellite images, medical images, microscopic images, and images taken from an autonomous vehicle. What follows after the data acquisition is dependent on the application. Sometimes one or more preprocessing steps are required to reduce noise and distortions. If the scene consists of one homogeneous textural region, the analysis will be based on the complete image. For instance, in a microscopic image which shows a product from the food industry in its final, or in an intermediate stage, one could extract process parameters from textural features. This can be done by applying a *texture operator* on the image. This operator transforms the input image into an image containing texture values, the *texture map.* A perfect (so a non existent) operator would give a homogeneous output image for a homogeneous textural input image. The choice of the operator is dependent on the application. Considerations can be based on the hardware aspects, the known physical laws underlying the applications, or the applicability proven by an intensive empirical study.

In most cases one is confronted with images containing several regions. This means that a segmentation step is required. After transforming the input image into one (or more) texture maps by applying one (or more) texture operators, the image can be segmented. Recent developments in image processing consider such segmentation results as preliminary, because a better segmentation result might be obtained by a segmentation module which is guided by knowledge. However, before such a fine tuning of the segmentation can be carried out, a preliminary segmentation result is required.

Sometimes one is not interested in the textural regions as such, but one needs to find them for the analysis of the whole image. An example of such a case is that of images taken from an airplane. Such images might be used to map new roads. For such an application it is not necessary to classify the type of texture. However, testing on the presence of textured regions might help in the detection or localization of roads. A similar type of use of texture information can be found in medical applications, where the localization of (for instance) blood vessels could partly be based on texture information.

Besides the use of texture information for segmentation purposes, it can also be applied in classification. An example of such a study is the classification of vegetation in radar images. In the literature, examples are known where all kind of vegetation and even diseases of vegetation are classified. Another example is the classification of tissue types in medical images.

Texture analysis can also be applied for recognizing shapes. Several types of approaches are in use in such studies. A detailed discussion on this kind of use of texture information is given by Kanatani and Chou in [61] and by Tomita and Tsuji in [112].

We have already mentioned the use of texture operators to transform a textured image into a texture map. Such an operator would function optimally if a homogeneous output image is obtained, given a homogeneous texture image. Examples of homogeneous textures are given in the Figures 2.1 - 2.4. These images are taken from Brodatz's book [14], which consists of a collection of 112 textures. This book was originally meant for artists and designers, but has grown to be a standard library of textures for the image processing community as well. The book consists of textures in the range of almost purely deterministic to almost purely stochastic.

To develop the *perfect* operator one needs to know what exactly is

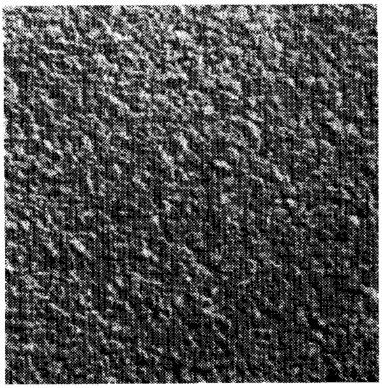

Figure 2.1: Pressed cork. Picture d4 from [14].

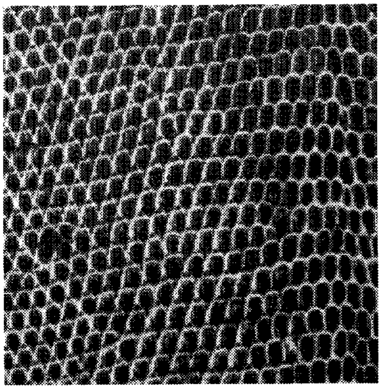

Figure 2.2: A reptile skin. Picture d22 from [14].

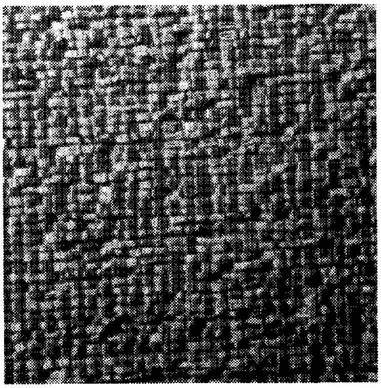

Figure 2.3: Raffia looped to a high pile. Picture d84 from [14].

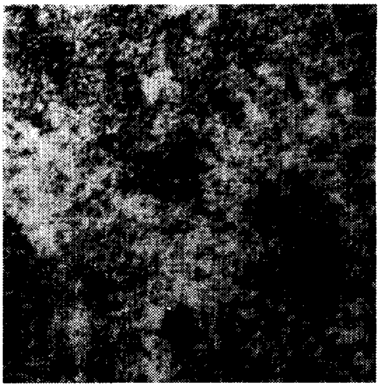

Figure 2.4: Ice crystals on an automobile. Picture d100 from [14].

meant by texture. This seems to be the essential problem of texture research. We are not able to define exactly what is meant by texture. Despite the fact that texture has been studied for a couple of decades in the image processing community and in other scientific disciplines as well as in arts, it is still undefined. A poor try at defining texture is based on the idea that a homogeneous textural region is homogeneous in a property other than grey value. However, this definition is too general to be useful.

The central problem is that we still do not understand the functioning of the human visual system. The optimal texture operator would be able to simulate the functioning of the human visual system in relation to the classification and segmentation of textured images. The name Julesz should be mentioned in conjunction with this research.

Originally, Julesz concluded that the human visual system was not able to discriminate textures for which the first and the second order statistics match. The appearance of counter examples, however, forced Julesz to some minor adjustments to his theory, which he later abandoned. This means that he abandons the idea of discriminability based on global statistics. In 1981, he introduced the *texton theory.* Textons are visual events, which are assumed to play a role in texture discrimination. Terminations, which are the end points of line segments or corners, are examples of such events. Locations where differences in textons occur, or where differences in the densities of textons occur, are used for segmentation purposes. More information on this subject can be found in [97], [98], [72], and [113].

In another attempt to understand human visual perception, it is assumed that it is based on spatial frequency analysis. Therefore, this study is related to the wavelet theory. The algorithms for texture analysis based on this idea are called multi-filtering algorithms, the discussion of which is postponed till Section 2.3.7.

Despite the lack of a definition, a lot of research on texture analysis and synthesis has been carried out. It must be stressed that it is not unusual in science to work with poorly defined phenomena. In the literature, many texture operators have been developed. A more elaborate discussion of these operators can be found in Section 2.3. A detailed discussion of existing synthesis techniques is outside the scope of this thesis. However, in Section 2.4 an enumeration of some important techniques is given.

Studying the literature [49], [48], [40], [44], it can be concluded that there is a certain consensus that there are two types of textures to be distinguished. These types of textures can be combined into one more general oriented "model".

The first type of texture is based on primitive texture elements, also called textels. The orientation of these elements with respect to each other may be based on a functional, stochastical, or a deterministical rule. An example of such a texture is a brick wall (see Fig. 2.5) or, more extreme, a chessboard type of texture. A second example of such a texture is the skin of a reptile as shown in Fig. 2.2. It is typical of this concept that the texture is based on a repeating pattern, or on repeating patterns. Analysis techniques which are based on the detection of such a repeating pattern should be first considered. In this case one could think of methods based on the frequency spectrum or the correlation function. Within the context of these types of textures, one also speaks of microtexture and macrotexture. By microtexture one means the primitives of the textures, while macrotexture refers to the orientation of the primitives with respect to each other.

The second type of texture is not built up of textural elements. This is a purely stochastic type of texture. Examples of this type of texture are: tissues, clouds, etc. In Fig. 2.4, ice crystals on an automobile are shown as an example of this type of texture. The analysis of this

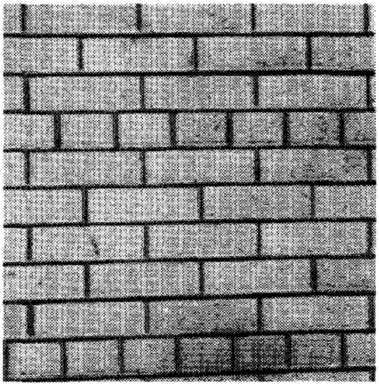

Figure 2.5: A ceramic-coated brick wall. Picture d26 from [14].

type of texture is based on stochastic operators which determine some probabilistic measures.

In [48], Haralick gives an elaborated discussion of the first type of texture. He calls textural primitives *tonal primitives.* The tonal aspects and the regional aspects of these primitives can be studied. Studying the tonal aspects of the primitives implies that the primitivities are described in terms of the average, minimum or maximum grey value. The regional aspects refer to a description based on features like shape or area.

Haralick's discussion continues with the question as to whether or not a distinction could be made between the tonal and the textural features. It seems that this is impossible. Therefore, Haralick prefers to speak of a *tone-texture* concept. This concept can be compared with the particle-wave concept. Light can be supposed to be a wave phenomenon for certain studies, but from another point of view it might be considered as a particle phenomenon. The discussion about texture and tonals may be regarded as comparable. If one studies only a very small part of the images, the number of tonal primitives will be small and therefore the dominant property of the part will be tone. When studying a larger part, more tonal primitives are considered, and the region will be dominated by the textural properties.

Gagalowicz [40] has based his work on a combination of the two types of textures. In his opinion, a texture consists of at least two levels, where each level is based on either the structured type of texture, or the purely stochastic type of texture. In his discussion, he introduces two conceptions e.g. *invariance under translation* and *textural resolution* . The concept *invariance under translation* is used to denote the property whereby a texture leaves the same visual impression whatever part of the texture is observed. The minimal window size of the operator required to garantuee invariance under translation is called the textural resolution. These definitions are crucial for texture analysis in general. He continues his discussion and develops a synthesis routine based on these aspects, which are outside the scope of this discussion.

In the following we discuss the duality of synthesis and the analysis of textures. Thereafter, we discuss the most well-known techniques for analysis and for synthesis. Finally, we conclude with some remarks regarding trends in analysis. The discussion of techniques based on fractal theory is postponed to Chapter 4.

2.2 Analysis versus Synthesis

The search for texture analysis operators is always based on a model. However, some models can be so general that it is not possible to verify whether or not the texture under analysis behaves according to the model, which means that such an analysis is, in general, not supplemented with an error measure. Examples of such models are the SGLDM[1]-approach and the grey level run-length approach, which are discussed in Section 2.3.

Having a model means that we can develop an analysis operator as well as a synthesis algorithm (see Fig. 2.6) which results in a duality between analysis and synthesis. We can generate images according to the model, with which we can verify the functioning of the analysis tool. This garantuees conditioned experiments, where the noise and the distortions to be expected in the real application can be simulated. Following this approach, we are able to learn the functioning of the analysis operator. It also means that if we have obtained the required parameters during the analysis, we can generate textures which should look like the texture under study. Or, more practically, knowing the parameters, we can imagine how the texture would look. Thus, the

---

[1] SGLDM = Spatial Grey Level Dependence Method.

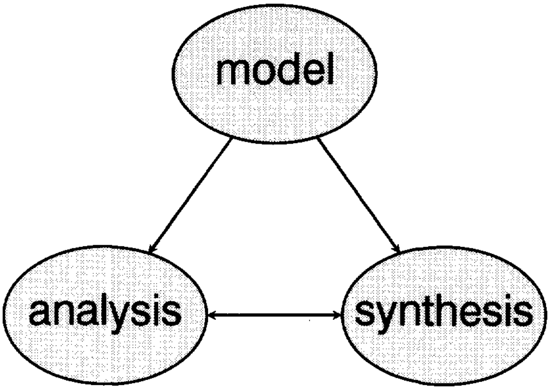

Figure 2.6: A model may be used for analysis as well as synthesis techniques. The relationship between analysis and synthesis can be of use for testing analysis operators.

parameters are interpretable. Image coding is a case where this duality is vital, where regions might be modeled and where only the model parameters are stored or transmitted. This duality may also result in interaction with other scientific disciplines. The most related disciplines are psychology, in particularly Gestalt psychology, and computer graphics.

These psychological studies aim to increase the understanding of the human visual system. Part of that study is the understanding of the discrimination and classification of textures. An example of such work is that of Julesz [59], as we discused in Section 2.1. His work is carried out on the edge between the world of image processing and the world of psychology. Despite the partly overlapping goal between the image processing community and the psychological studies of the human visual system, the interaction seems to be limited.

Compared to the influence of the work of psychologists on texture research for image processing, the influence of computer graphics is possibly more limited. With the growing number of computer graphics applications, the need for algorithms which generate natural looking textures is steadily increasing. Medical training systems, simulators for training defense personnel and computer games are examples of such applications. The fact that the influence of computer graphics research on texture research for image processing is limited is caused by the fact that one is only interested in an algorithm that will generate a natural looking texture. Such algorithms do not necessarily need to be applicable for texture analysis.

From this discussion it follows, that because of the complexity of texture as an image processing phenomenon, the analysis approaches of the image processing community are almost stand alone with respect to other scientific disciplines that study texture as well.

2.3 Analysis

Most texture operators calculate the texture value for a window of given dimensions. The minimal window sizes are dictated by the textural resolution as discussed on page 13. In practice, the window is never smaller than 8×8. A larger window size, however, is preferable. Although a much larger window could result in ill-defined edges for scenes with more than one region. Now, suppose that the grey values are in the range of $[0, 255]$ and we have a window of 8×8. Then there exist 256^{64} =13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006084096 realizations. Of course, there is no equality of probability in the occurrence of all realizations in natural fields. Further, this number should not be interpreted as being the number of classes. It is the task of the texture operator to cluster similar textures to almost equal values, while dissimilar textures should result in dissimilar texture values. However, this number illustrates that it is hard to expect a good performance for a whole range of applications from a single operator. In most practical situations, a combination of different operators is used, or one operator is used with different parameter values. An example of the latter situation is the use of a directionally sensitive operator. Such an operator is especially useful for textures which show a certain directionality. In such a case, we could apply the operator twice, where the direction in which the measurements take place is fitted to the direction of the texture. The values could be combined to an average value, but we could also use them separately during the analysis. In the following sections, we discuss the most common type of texture operators and those operators which are typical for a class of operators. However, we should stress that for a certain type of texture the frequency domain should first be considered. Some textures have characteristic properties in this domain which might be used for the analysis. The study of the frequency domain will probably be based on the power spectrum. And because this spectrum is related to the autocorrelation function according to the Wiener-Khintchine theorem, we can consider this study to be based on the autocorrelation function.

2.3.1 SGLDM: The Spatial Grey Level Dependence Method

The *Spatial Grey Level Dependence Method* stands for a whole range of texture measures which are based on the co-occurrence matrix. Haralick has played an essential role in the development of the two-dimensional co-occurrence matrices for texture discrimination purposes [49], [48], [50]. He developed a technique which is based on the estimation of the second order joint probability density function. The use of this probability density function was based on research carried out by Julesz, who argued that the second order statistics played an essential role in texture discrimination by man. As discussed in Section 2.1, Julesz later abandoned the theory of discriminability which is based on global statistics. This, however, does not exclude the usefulness of the texture operators developed by Haralick for certain applications.

The co-occurrence matrix, which is also called the Gray-Tone Spatial-Dependence Matrix, is essential for Haralick's operators . The number of rows and the number of columns of this matrix are equal to the number of grey values one wishes to distinguish in the image. This number does not need to be equal to the number of grey values which can be distinguished in the image. Originally, the sizes of the matrix were based on an optimalization between processing time, memory requirements and the quality of the measurements obtained. However, today's computer power justifies the use of all available grey values. The matrix is initialized to 0. Before filling the matrix, a vector has to be defined with a certain length and a certain orientation. This vector is moved over the image, within the window of the texture operator and the grey value at the tail of the vector and the head of the vector are considered during the measurement. The elements in the co-occurrence matrix correspond to pairs of grey values, the values of the elements correspond to the frequency of occurrence of the particular combination in the window. For example, if the grey value at the tail of the vector is a and the grey value at the head of the vector is b, then the elements (a, b), respectively (b, a) will be increased by one. Note that if the grey values are equal, the value of the corresponding element is increased by two! At the end, all matrix elements are normalized, which results in an estimate of the joint second-order probability density function. This matrix is used for the calculations of texture measures. Note that the matrix is symmetrical and we could therefore reduce the memory requirements by almost 50%.

The features which are based on the cooccurence matrix are a measure for the distribution of the counts over the matrix. For the normalized co-occurrence matrices Haralick proposes a number of features. An example of such a measure is the *contrast*, which is defined as follows:

$$\text{Contrast} = \sum_{n=0}^{N-1} n^2 \left\{ \begin{array}{l} \sum_{i=1}^{N} \sum_{j=1}^{N} \; p(i,j) \\ |i-j| = n \end{array} \right\}, \tag{2.1}$$

where N stands for the number of rows, and the number of columns. The matrix element is designated by $p(i, j)$, where i corresponds to the grey value at the tail of the vector and j to the grey value at the head of the vector. The measure is called contrast, because it stresses those elements that corresponds to pixel pairs with a large difference. Bear in mind that elements in the neighborhood of the main diagonal correspond to pixel pairs with a small difference, whereas elements more remote from the main diagonal correspond to pixel pairs with a large difference. The measures proposed can be divided into three classes, viz.

- Measures that stress those elements that correspond to pixel pairs with small differences.
- Measures that stress those elements that correspond to pixel pairs with large differences.

- Measures that treat all elements equally.

An example of the latter case is the *entropy*, which has been defined as follows:

$$\text{Entropy} = -\sum_i \sum_j p(i,j) \log\left(p\left(i,j\right)\right). \tag{2.2}$$

This measure will come back in Chapter 4.

As this method is based on measuring the grey values at the tail and the head of a vector, we have to know how to choose the orientation and the length of the vector. If the texture under study does not show a directionality, the choice of the orientation of the vector is arbitrary. If, however, the texture shows a certain directionality it is recommended to apply several vectors with different orientations. The choice of the length [2] of the vector is dependent on the type of texture. If the texture under study appears to be a more stochastically oriented one, as for instance the texture shown in Fig. 2.4, the influence of the length seems -according to Besuijen [9]- to be limited. From a computational point of view, smaller values for the length are to be prefered, because more measurements are obtained within the window. For this type of texture, one often uses a length of 1.

The choice of the length is more critical for textures which seem to be built up of texture elements. An example of this case is shown in Fig. 2.7. In this figure, we have shown that the vector can be chosen too small in comparison to the size of the texture elements. We have also shown the case where the vector has been chosen too large. The right choice for the length is where the length is more or less equal to the size of the texture elements. [3]

For a texture which appears to be built up of texture elements, we see that there is an optimal "scale" on which the texture should be studied. In this case scale is defined as the length of the vector. Choosing a vector length which is not in correspondence to the texture elements results in a decrease in the performance of the texture operator.

---

[2] Haralick uses the city-block distance metric for his definition of the length of the vector.

[3] Of course, the length could be chosen as an integer multiple of the size of the texture elements. However, this will reduce the number of measurements that can be carried out in the window and is therefore not preferable.

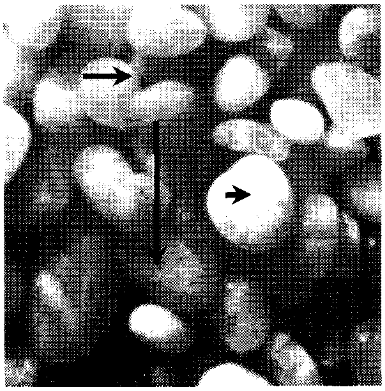

Figure 2.7: A typical example of a texture (from [14]) which is built up of texture elements. The three vectors shown correspond to the cases where the vector for the SGLDM is chosen too small, too large and in correspondence with the size of the texture elements.

The way the SGLDM operators deal with scale is limited. Scale, as dicussed before, is for these operators defined as the length between pixel pairs. Normally, by scale we mean the level of resolution. This means that if we study a texture on a coarser scale, a larger area patch is considered, rather than pixel pairs at increasing distance. The Max-Min operator, which is discussed in Section 2.3.4, is an example of an operator that studies textures on several levels of resolution.

The SGLDM has been extended by Davis et al. [29], such that the co-occurence matrix is not based on grey levels, but on the spatial distribution of textural features, like edge pixels. It originally came from a synthesis approach where the texture was thought to be built up of texture elements as discussed in Section 2.1. Another variant to the SGLDM distinguishes the grey value at the tail and the head of the vector. The grey value at the tail corresponds with the row-index of the matrix and the grey value at the head of the vector corresponds with the column-index.

2.3.2 GLDM: Grey Level Difference Method

The Grey Level Difference Method can be seen as a less time and memory consuming variant of the SGLDM approach. Instead of a matrix we use a vector. This vector represents the number of counts of possible absolute grey level differences, measured over a given vector. Thus, if we distinguish N grey levels, our grey level difference (in the sequel abbreviated to *GLD*) vector will have N elements. The first element corresponds with a grey level difference of 0, the second element corresponds with a grey level difference of 1, and so on. As for the co-occurence matrices, we can normalize the resulting vector. Note that the grey level difference vector can be obtained from the co-occurrence method, by summing up the entries along axes parallel and symmetrical with respect to the main diagonal of the matrix. An early variant on this approach has been discussed by Bacus and Gose [7].

As for the SGLDM method, we can also define here a number of features, which are based on this vector. The features in use for the GLDM are similar to the SGLDM features. For instance, the GLDM variant for contrast feature is defined as:

$$\text{Contrast} = \sum_{i=1}^{N} i^2 p(i). \tag{2.3}$$

In [121], Weska et al. describe a comparative study of texture measures. In this study, it is concluded that the SGLDM and GLDM perform almost equally well. Therefore, from a computational point of view, the GLDM might be prefered. Conners and Harlow [24] also made a comparative study of texture measures. In their theoretical study it is shown that the discriminable pairs of textures for the GLDM are to be a subset of the discriminable pairs of the SGLDM. This means that for applications, for which the performance of the GLDM appears to be insufficient from a discriminating point of view, one should test the SGLDM on its suitability. Of course, the way scale is treated by the GLDM is not different from the way it is treated by the SGLDM. Therefore, the choice of the vector length might be critical.

2.3.3 GLRLM: Grey Level Run Length

In 1975, Galloway [41] introduced the Grey Level Run Length Method. In this technique, a matrix is filled with statistics of the run lengths. Runs are defined as consecutive pixels which have the same grey value.

Now, the length of a run is equal to the number of pixels of which it consists. Of course, for each possible direction such a matrix can be defined.

The matrices to be initialized are called the run length matrices. The rows of these matrices correspond to the grey level, while the columns correspond to the run lengths. Before the analysis, Galloway groups the grey levels. She gives an example in which the original image consists of 64 grey levels. This image is transformed to an image where the grey values lie in the range of $[0, 7]$. The transformation used is linear. This rescaling prevents the situation that only small run lengths will be found. For practical reasons, the possible run lengths are also grouped. For the 64×64 image Galloway gave an example in which she grouped the run lengths in the ranges: 1, $2-3$, $4-7$, $8-15$, $16-31$, and $32-64$. In these groupings, the resulting matrices were of the size 8×6.

As for the texture measures discussed before, we can determine a number of features. The features of this technique are not significantly different from the discussed features of the SGLDM and the GLDM.

Although this method is well known in the image processing community, its use is rather limited, which is because of the strong noise sensitivity. After a comparative study, Weszka et al. [121], and Conners and Harlow [24], came to the conclusion that the Grey Level Run Length Method should not in general be recommended. Further, we must conclude that it is not possible to select the level of scaling on which the texture has to be studied. In the next section, the Max-Min measure is discussed as an example of a measure which is based on the quantification of scaling behavior.

2.3.4 Max-Min Measure

In [85], Mitchell et al. propose a line-scan based texture measure which counts extrema on different levels of scaling. Initially, the data is smoothed in order to make the technique less sensitive to noise. Then, the image is filtered by a non linear 1-D filter. Suppose that the original line is denoted as $I(x)$, where x is a position on the line and $I(x)$ the grey value on that position and let the resulting line be denoted by $I'(x)$. The filter is supplied with a threshold parameter T, which determines

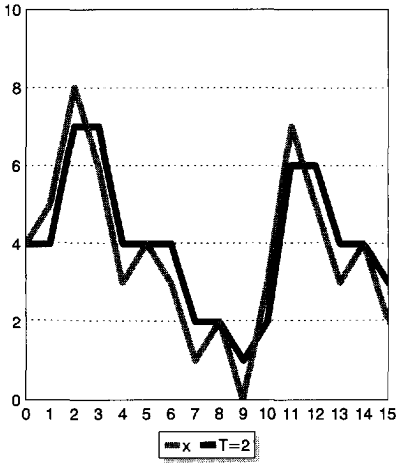

Figure 2.8: An illustration of the filter operation which is used by Mitchell [85]. The resulting curve is obtained where $T = 2$.

the level of scaling. A larger value for T corresponds to a measurement on a larger scale. Now, the filter scheme is defined as follows:

If	Then
$I'(k) < I(k+1) - \frac{T}{2}$	$I'(k+1) = I(k+1) - \frac{T}{2}$
$I(k+1) - \frac{T}{2} \leq I'(k) \leq I(k+1) + \frac{T}{2}$	$I'(k+1) = I'(k)$
$I(k+1) + \frac{T}{2} < I'(k)$	$I'(k+1) = I(k+1) + \frac{T}{2}$

An example of this filter process where $T = 2$ is shown in Fig. 2.8. After filtering the data, the extrema (that means the minima and the maxima) can be determined. It is this number that is determined on several levels of scaling.

In order to make the algorithm insensitive to multiplicative illumination changes, the authors use the logarithm of the grey values. To make the operator insensitive to the absolute number of extrema encountered, the ratio between the number of extrema at one threshold to the number at the next threshold is determined. This ratio can be obtained from a curve where the logarithm of the number of extrema is drawn against the threshold value. Now, the slope of the resulting curve corresponds to the ratio. For each type of texture, a charateristic curve is obtained. According to the authors, the classification results obtained with this technique are comparable to the results obtained with the SGLDM and GLDM. However, the method is less complex and considerably faster.

2.3.5 Textural Edgeness

In [100], Rosenfeld and Thurston propose a texture measure which is based on the textural edgeness. Before calculating the average edgeness of a textural scene, one has to compute the edges within the region of interest. The authors apply Robert's cross operator. This discrete difference operator is based on the following operation:

$$I' = |I(i,j) - I(i+1,j+1)| + |I(i,j+1) - I(i+1,j)|, \tag{2.4}$$

where I' denotes the resulting image and I the original image. From the resulting image the average edgeness per unit area is computed.

2.3.6 The Long-Correlation Model

The long-correlation model is a texture model *en pure sang* which has been developed by Kashyap [64],[63]. Kashyap mentions the applicability for analysis and synthesis. The name long-correlation model refers to the main characteristic of the model. In the traditional stationary autoregressive models, the correlation decays exponentially with the lag (or a linear combination of the exponentials.) In contrast with these traditional models, the correlation for the long-correlation model decreases with the lag k in a manner like k^α. This property is related to the fractional Brownian function which is discussed in Chapter 4. Such a slowly decreasing correlation is typical for a number of natural textures.

Suppose that $\{y(i,j)\}$ is a sequence of intensities which follows a two-dimensional *long-correlation* model and $\{\zeta(i,j)\}$ is a two-dimensional white noise sequence with variance ρ. Then the 2-D long-correlation model is represented by the following equation:

$$y(i,j) = (1 - z_1^{-1})^{-c}(1 - z_2^{-1})^{-d}\zeta(i,j), \tag{2.5}$$

where z_1^{-1} and z_2^{-1} are unit delay operators in the i and j directions, respectively. The parameters c and d characterize the properties of this model along with the variance of the noise. These parameters can be estimated in the Fourier domain by the least squares method.

2.3.7 Multi-Channel Filtering

A tendency which is influenced by the development of the wavelet theory is based on multi-channel filtering. This technique is justified by the assumption that the human visual system makes use of spatial frequency analysis. The idea is to filter the image with a range of narrow-band filters. Studying the output of such a filter bank results in a selection of filters on which one or more features might be based. The selection of filters is based on the resolution required in the spatial domain and the frequency domain. The resolution in the spatial domain is of importance for the accuracy with which texture edges are found. The resolution in the frequency domain is of importance for the discrimination performance. An approach based on real-valued even-symmetric Gabor filters is discussed by Farrokhnia and Jain in [37], and by Jain and Farrokhnia in [57]. The impulse response of these filters is given by:

$$h(x,y) = \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\}\cos\left(2\pi u_0 x\right), \tag{2.6}$$

where u_0 is the frequency of a sinusoid plane wave along the x-axis (i.e. the 0° orientation), and σ_x and σ_y determine the width of the Gaussian envelope along the x and y axes. The impulse response in the Fourier domain is given by:

$$\begin{aligned} H(u,v) \quad = \quad & A\exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} + \\ & A\exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\}, \end{aligned} \tag{2.7}$$

where $\sigma_u = \frac{1}{2\pi\sigma_x}$, $\sigma_v = \frac{1}{2\pi\sigma_y}$ and $A = 2\pi\sigma_x\sigma_y$. The Gabor filter optimizes the resolution in the spatial domain as well as in the frequency domain. A more detailed and general discussion by Bovik on the multichannel approach can be found in [13].

2.4 Synthesis

In Section 2.2, we mentioned that if there is a model available, this model can be used for analysis purposes, as well as for synthesis purposes. However, there are models that appear to be particularly suited for analysis purposes, whereas other models can be more suited for synthesis purposes. Because we already have discussed a number of analysis techniques, we will now mention a few of the synthesis techniques that are suggested for the study of textures. Techniques based on fractal theory are discussed in Section 4.3.

The most well-known techniques are based on ARMA models. A detailed discussion on the use of 2-D non-causal autoregressive models is described by Chellapa and Kashyap in [22]. An elementary discussion of this technique is also given by McCormick and Jayaramamurthy in [83]. The use of the hidden Markov model has been described by Gong and Huang in [43].

In 1976, Zucker [127] introduced a synthesis technique which is of interest for textures which can be characterized by some textural primitives. These primitives might be distorted in shape; and their orientation to each other is based on a deterministical, functional or a probabilistical rule. In [125], Yokoyama and Haralick introduce a synthesis technique that takes into account the probabilistic aspects of textures as well as the deterministic aspects. The technique consists of two phases. The first phase starts with the creation of a symbolic image. This symbolic image is initialized such that all values are set to zero. The next step of the first phase consists of the distribution of *seed cells*. This means that in a random or in a deterministic fashion the values of the resolution cells are changed in a value that corresponds to a symbol type. Cells that are non-zero are called *seeds* and after having completed the distribution they will grow to structures. The growing process on its turn can take place in a random fashion or in a deterministic fashion. The second phase consists of the transformation of the symbolic image to the final grey value image. This transformation can again be random oriented as well as deterministc oriented. The elegance of this synthesis technique is the consequent distinction made between the deterministic aspects of textures and the probabilistic aspects.

2.5 Concluding Remarks

In the preceding discussion we have listed a number of texture analysis operators. Some of them are to be considered as "real" model oriented. Others are so general that the model aspects are hardly to be recognized. It is, therefore, that operators of the latter type are considered to be probabilistic measures. As a result of the limited applicability of the available texture models, the general oriented texture operators are still in wide use.

New texture models are introduced by, for instance, the fractal theory. The fractal theory describes how the appearance of phenomena changes by varying the scale. Scale is -as we have discussed in this chapter- also of importance for the study of texture. The appearance of a texture might dramatically change by varying the scale on which it is observed. Studying an image processing phenomenon on different levels of scaling is often carried out in a *pyramidal data structure* or *pyramid.* In the next chapter we will discuss this datastructure and its use for image processing tasks. In Chapter 4, the fractal theory will be discussed and its use for the study of textures. Finally, in Chapter 5 an analysis technique is developed that studies the underlying texture on a number of scaling levels by using the pyramidal datastructure, and which is based on ideas from the fractal theory.

Chapter 3

Pyramidal Data Structures

3.1 Introduction

The name *pyramidal data structure* (or *pyramid*) refers to the way this structure can be visualized. It is a type of structure which is in wide use for image processing purposes and which was introduced by Kelly in 1971 [69]. The program where Kelly was working on aimed at the extraction of the outline of human heads in digital pictures. This outline could later be used for identification purposes. Kelly demanded that the technique required only a minimum of processing time. Further constraints were the reduced sensitivity to noise and distortions. In order to fulfill the requirements he followed a method which was called *planning*, and which originally came from artificial intelligence research and was introduced by Minsky [84]. Planning consists of three steps, viz.:

1. Simplify the problem.
2. Solve the simplified problem.
3. By using the solution to the simplified problem, try to solve the original problem with the same solution method.

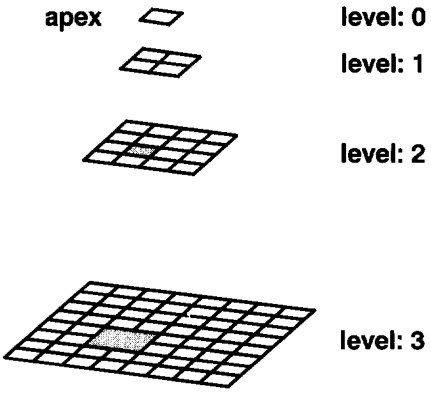

Figure 3.1: The pyramid as proposed by Kelly [69].

This concept is translated by Kelly in the following way:

1. Extract a new and smaller image from the original image.
2. Locate the desired edges in the smaller image.
3. Use the edges found in the smaller image for locating the edges in the original image.

The smaller image was obtained by a combination of averaging and subsampling. The sizes of the reduced image were reduced by a factor of 8 by taking the average value of non overlapping windows of 8×8. Although Kelly only used two resolution levels, in his discussion he suggested applying more resolution levels. He suggested that for some applications, it might be preferable to apply a sequence of reduction factors, like, for instance: 2, 4, 8, etc. This suggestion was the introduction of what would later be called the pyramidal data structure. The structure is often visualized by putting the reduced images above each other in such a way that the sizes decreases (see Fig. 3.1).

In the literature, two styles of numbering the levels within the pyramid are in use. The first one starts at the top level of 1×1 (also called the apex), which is given the index 0 and the index is incremented with each level underneath. In this way the index is directly related to the size of the level. For instance, a level with index i corresponds with an image with sizes $2^i \times 2^i$. The other type of numbering starts at the bottom level and goes to the apex. It is the first type of numbering we here apply.

In 1972, Uhr [114] applied the pyramidal datastructure in its full extent, but called it a *recognition cone*, which refers to the human visual system [115]. The term "cone", however, suggest the use of a circular type of image. Later the term *pyramid* or *pyramidal datastructure* has been introduced which is more in accordance to the square type of images to be studied.

The original pyramid is also called a *grey value pyramid*, because the nodes (or pixels) in the pyramid are filled with grey value information. A pyramid can also be filled with other information, which requires another type of resolution reduction. Today, a whole range of pyramids are known. In this chapter, we aim to give an impression of the use of pyramidal data structures, what types of pyramids are in use, and what should be taken into account when one is planning to design a new type of pyramid, or a new type of initialization of the pyramid. In the following section we give an overview of the pyramids which can be found in literature, after which the use of the pyramidal data structures for texture analysis and segmentation is discussed. The discussion of implementations of the pyramidal datastructures into hardware is outside the scope of this thesis.

3.2 Examples of Pyramids

3.2.1 The Grey Value Pyramid

The initialization of the grey value pyramid is a recursive process, which starts at the bottom level. This level is a direct copy of the original input image. Each level above is obtained by a combination of a low-pass filtering step and a subsampling step. The operator that carries out this process is called the *resolution reduction operator*. Each new level is obtained by applying this operator on the level underneath. In practice, the filtering and subsampling step will be combined, such that a reduction in processing time is obtained. Besides the linear filtering technique as discussed here, examples are to be found where non-linear filters were used. For instance in [86], Morales and Acharya introduce

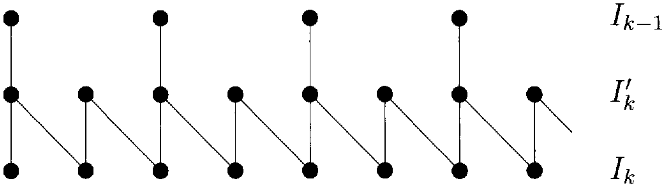

Figure 3.2: Resolution reduction as a process built up in two phases.

what they call a *morphological pyramid* and in [105] the median filter is used by Shneier to prevent that linear features in the pyramid are blurred.

The Resolution Reduction Operator from a Frequency Point of View

Because of the subsampling step in the resolution reduction, the choice of the averaging filter is not trivial. Originally a 2 × 2 averaging filter was proposed. That this filter is not optimal from a frequency point of view can be seen if we describe the filter mathematically. Suppose that I_k corresponds with the pyramid level k. Then, $I_k(i,j)$ denotes the grey value of the pixel with coordinates (i,j). This level is supposed to be the input for the level which is denoted by I_{k-1}. For our analysis, we suppose that there is an intermediate level I'_k, which has the same sizes as $I_k(i,j)$, but which contains the result of the filtering step (see Fig. 3.2). We can now formulate how I'_k can be obtained from I_k:

$$\begin{aligned} I'_k(i,j) &= \frac{1}{4}[I_k(i,j) + I_k(i+1,j) + \\ &\quad I_k(i,j+1) + I_k(i+1,j+1)] \end{aligned} \tag{3.1}$$

Transforming this expression into the z-domain results in:

$$\mathcal{I}'_k(z_1,z_2) = \frac{1}{4}(1 + z_1 + z_2 + z_1 z_2)\mathcal{I}_k(z_1,z_2), \tag{3.2}$$

where $\mathcal{I}'_k$ stands for the z-transform of I'_k and $\mathcal{I}_k$ for the z-transform of I_k. The z-domain can easily be transformed into the Fourier domain

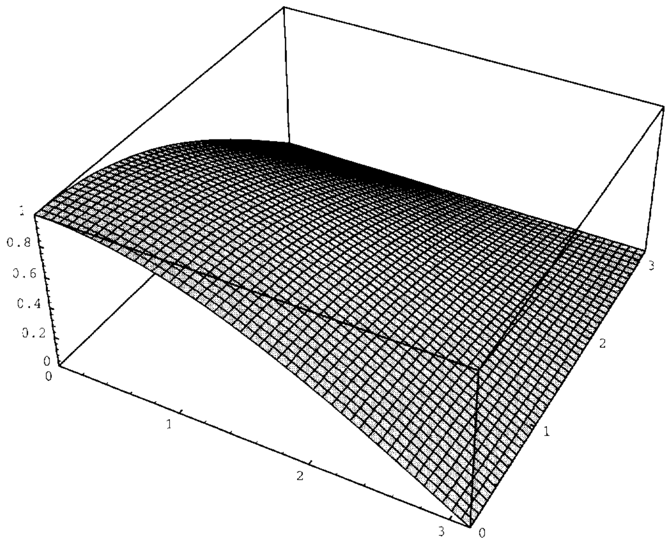

Figure 3.3: The magnitude of the transfer function of the 2 × 2 uniform averaging filter.

by the substitution $z_1 = e^{j\omega_1}$ and $z_2 = e^{j\omega_2}$. This results in:

$$\mathcal{I}'_k(\omega_1, \omega_2) = \frac{1}{4}(1 + e^{j\omega_1} + e^{j\omega_2} + e^{j\omega_1}e^{j\omega_2})\mathcal{I}_k(\omega_1, \omega_2). \qquad (3.3)$$

By rewriting this, we find for the transfer function $H(\omega_1, \omega_2)$

$$H(\omega_1, \omega_2) = \cos(\frac{\omega_1}{2})\cos(\frac{\omega_2}{2})e^{\frac{j\omega_1}{2}}e^{\frac{j\omega_2}{2}} \qquad (3.4)$$

The magnitude of this transfer function is shown in Fig. 3.3

It is characteristic of this averaging filter that it still passes a significant amount of high frequency energy. These high frequency components result after subsampling in aliasing. To prevent aliasing, the filter should reject frequencies for which $\frac{\pi}{2} < \omega_1 \leq \pi$ or $\frac{\pi}{2} < \omega_2 \leq \pi$. Tanimoto and Pavlidis give in [109] a more detailed discussion on this type of filtering. In [110], Tanimoto deals with several types of distortions which might occur higher up in a pyramid.

Another type of filter which has also been applied in literature is the uniform filter of sizes 4 × 4. This filter suppresses the unwanted high frequency components more strongly than the previously studied

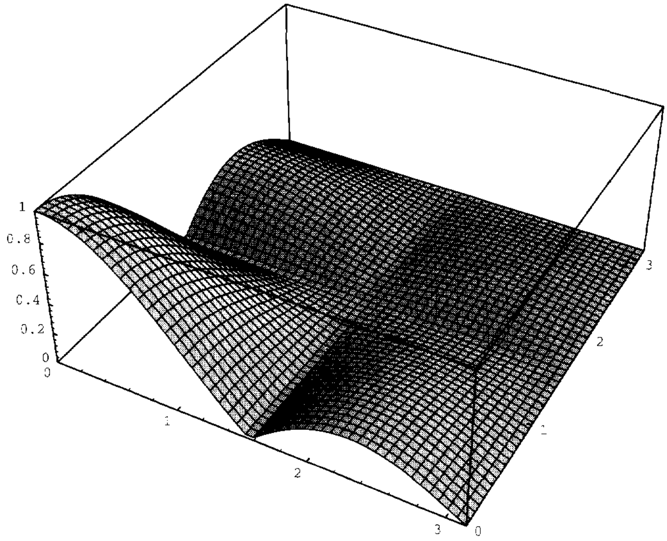

Figure 3.4: The magnitude of the transfer function of the 4 × 4 uniform averaging filter.

filter. The analysis of this filter can be done in the same way, which results in the following transfer function:

$$H(\omega_1, \omega_2) = \cos(\frac{\omega_1}{2})\cos(\frac{\omega_2}{2})\cos(\omega_1)\cos(\omega_2)e^{\frac{j\omega_1}{2}}e^{\frac{j\omega_2}{2}} \tag{3.5}$$

The magnitude of this function is shown in Fig. 3.4.

A better filter for the initialization of the grey value pyramid is the Gaussian averaging filter. The use of this filter was proposed by Koenderink [70] among others who came to this conclusion after an analytical study of scale space, where one of the constraints was that the filter should not introduce artefacts higher up in the pyramid. Scale space means in our case a 3-dimensional space, where two of the three dimensions correspond with the spatial domain of the image, and where the third dimension corresponds to the resolution. The relation of pyramidal data structures with scale space is postponed to Section 3.3 In Fig. 3.5 an illustration of a grey value pyramid is given, which is realized with a Gaussian filter with sizes 5 × 5. The magnitude of the transfer function of this filter is shown in Fig. 3.6.

Burt [15] [16] has given some rules for designing Gaussian-like filters.

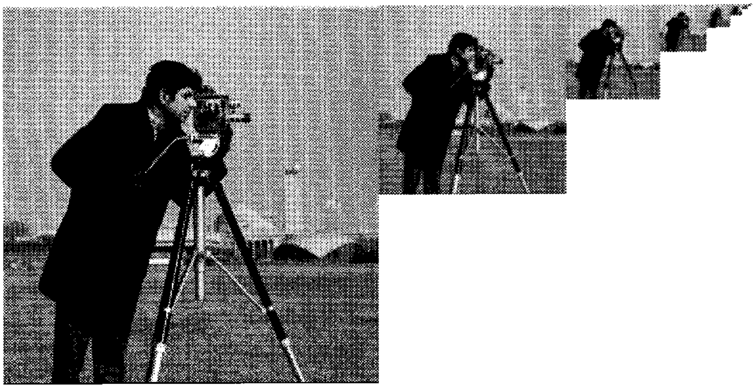

Figure 3.5: An example of a grey value pyramid.

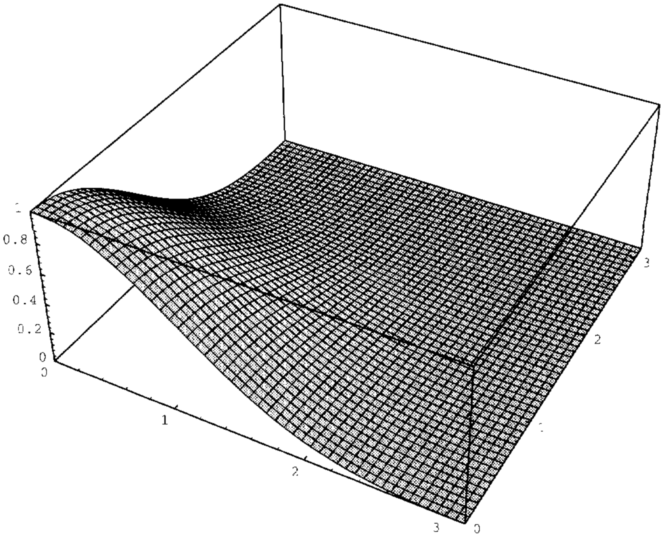

Figure 3.6: The magnitude of the transfer function of the 5×5 Gaussian weighted filter.

Gaussian-like, because the filters are not exactly Gaussian. However, the difference is negligible. In his discussion he aims at the development of a design algorithm for filters with equivalent shapes but different widths. In order to ensure that the equivalent kernels are unimodal, symmetric, and centered at $(0,0)$, Burt adopts the following four constraints:

1. Normalization
$$\sum_{i=-m}^{m} \sum_{j=-m}^{m} w(i,j) = 1, \tag{3.6}$$
2. Symmetry
$$w(i,j) = w(-i,j) = w(i,-j) = w(-i,-j), \tag{3.7}$$
3. Unimodal
$$w(i,j) \geq w(k,l) \text{ for} |i| \leq |k| \text{ and} |j| \leq |l|, \tag{3.8}$$
4. Equal Contribution
$$\sum_{i=-m}^{m} \sum_{j=-m}^{m} w(\hat{\imath}+ir, \hat{\jmath}+jr) = \frac{1}{r^2}, \tag{3.9}$$
for $0 \leq \hat{\imath}, \hat{\jmath} < r$,

where the filter weights are denoted by $w(i,j)$, with $-m \leq i,j \leq m$. The last constraint ensures that each pixel in a pyramidal environment contributes with equal weight to every level above. For the Gaussian case the filter is separable, which means that:

$$w(i,j) = w_x(i) w_y(j), \tag{3.10}$$

where w_x and w_y stand for the one-dimensional kernels. Now, suppose we want to apply the Gaussian filter as a separable filter, so that we filter the image first in the horizontal direction and afterwards in the vertical direction, our problem is simplified to a one-dimensional problem. The constraints can be reformulated to the one-dimensional case. Suppose we are interested in a one-dimensional Gaussian filter with five weighting coefficients. Let $w_x(0) = a, w_x(1) = b$ and $w_x(2) = c$. Then we find according to the constraints:

$$\left.\begin{array}{rcccl} w(-1) & = & w(1) & = & b \\ w(-2) & = & w(2) & = & c \end{array}\right\} \quad \text{Symmetry}$$

$$a + 2b + 2c = 1 \qquad \text{Normalization}$$

$$a \geq b \geq c \geq 0 \qquad \text{Unimodal}$$

$$a + 2c = 2b \qquad \text{Equal contribution}$$

Burt shows that for $a = 0.4$ a Gaussian-like filter will be obtained. The other coefficients are obtained by solving the equations. This results in $a = 0.4, b = 0.25$ and $c = 0.05$.

Applications of the Grey Value Pyramid

Correlation matching is a typical application of this type of pyramid. This type of application is described by Glazer in [42], by Hall et al. in [46], and by Rosenfeld and Vandenbrug in [101]. By applying a pyramidal data structure, a significant reduction of processing time can be achieved, and dependent on the complexity of the scene, a more precise result might be reached. The matching process starts at a high level in the pyramid, and after having found the result for this level, the process continues one level lower within the region of interest. This region of interest can be compared with the band of edge pixels Kelly processed after having found an initial outline of the face.

Another type of application is that of blob-detection as described by Blanford and Tanimoto in [12], and by Shneier in [104]. Some applications require the localization of one or more bright blobs in the image. In the first instance, one might consider the use of a *maximum* operator. However, the presence of noise makes this operator less appropriate, a Gaussian resolution reduction operator or a uniform weighted operator might be preferable for the initialization of the pyramid. With this pyramid the search for the maximum value is started.

In [105], Shneier describes the use of a grey value pyramid for the extraction of linear features. Such features can occur in aerial photographs and an algorithm for the detection of these features is of use for the automatic mapping of roads. Shneier describes an algorithm, where a line detector operates at each significant level of the pyramid.

The output for each level is used to detect regions in the original image that correspond to linear features. The threshold required to extract the linear features in the original image is based on properties within these regions. In contrast to the original grey value pyramid as described earlier, the author used a median filter with a size of 2×2. For this application, such a filter is preferable, because the linear features would otherwise be blurred, and this would hamper the detection of these features.

A completely different type of grey value pyramid is Peleg's adaptive pyramid [91]. Until now we have discussed pyramids with a uniform resampling scheme. Peleg proposes a pyramid with an resampling scheme which is adaptive to the contents of the underlying region. Therefore, he defines a "busyness" measure. From an information theory point of view this approach is of particular interest. Regions with more information will be more densely resampled than smooth regions. The resulting image looks like images reflected in a carnival mirror.

3.2.2 The Laplacian Pyramid

The Laplacian pyramid is an extension to the Gaussian pyramid. It was introduced by Burt in 1983 [17] [16]. A generalization of this type of pyramid is also known as the *DOLP* (= Difference Of Low-Pass) pyramid, and as the *DOG* (= Difference Of Gaussian) pyramid. The initialization starts as the initialization of a Gaussian pyramid. Thereafter, each level above the bottom level is expanded by interpolation such that the sizes equal the sizes of the level underneath. The resulting image is subtracted from the level underneath. The difference is stored in the *Laplacian pyramid* (see Fig. 3.7). What remains in this resulting image are the high frequency components which were suppressed during the resolution reduction step. Thus, the resulting pyramid consists of band-filtered versions of the original image. In Fig. 3.8, we show an example of such a pyramid. In this figure, the grey scale has been normalized and shifted, such that the values fall within the visual range. This explains the average grey-tone. The initialization as described above can be formulated as follows. Suppose that I_N is the bottom level of the pyramid, which equals the original input image. Then, for

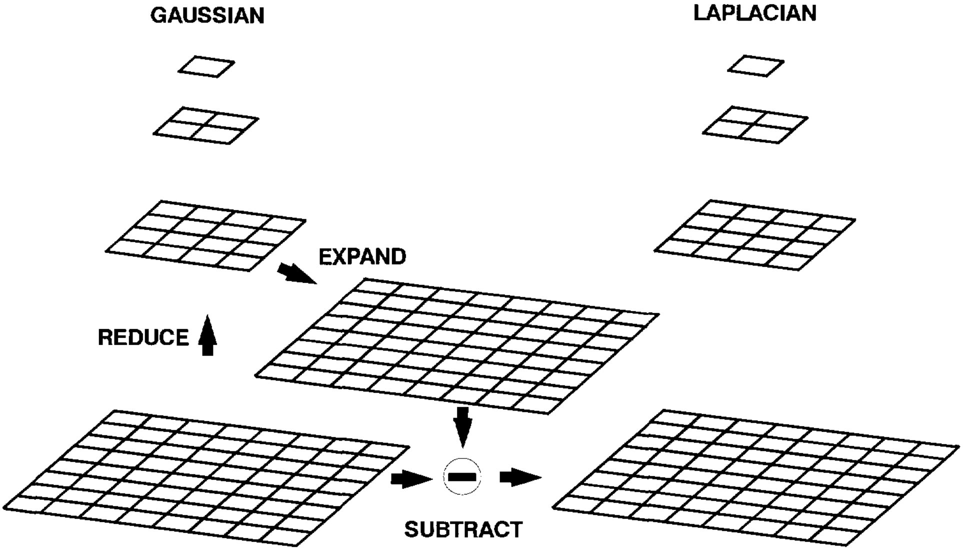

Figure 3.7: The initialization of the Laplacian pyramid.

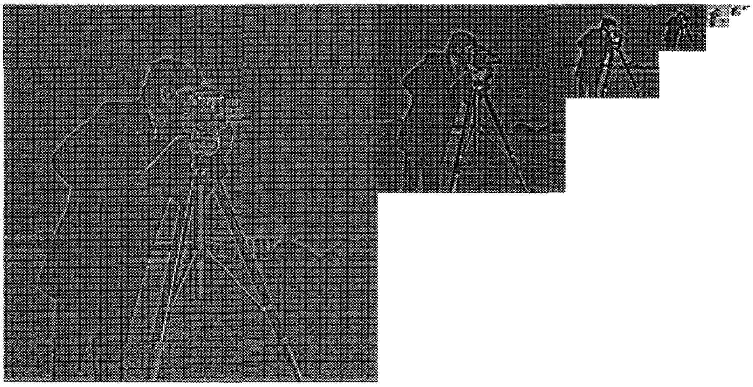

Figure 3.8: An example of a Laplacian pyramid, where the values have been scaled to the interval [0,255].

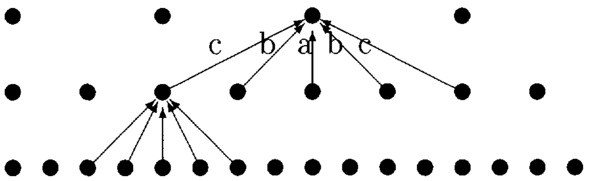

Figure 3.9: The reduction process for the 1-dimensional case visualized.

$0 \leq k \leq N$ we say:

$$I_{k-1} = REDUCE[I_k] \tag{3.11}$$

According to the Gaussian reduction as discussed before, we write:

$$I_{k-1}(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) I_k(2i+,2j+n) \ , \ 0 \leq i,j \leq 2^{k-1} - 1 \tag{3.12}$$

where w stands for the Gaussian kernel. An illustration of this process is shown in Fig. 3.9. Besides the *REDUCE* operator, we need for the initialization of the Laplacian pyramid an *EXPAND* operator. Now, suppose that I'_k is the expanded version of I_{k-1}. The sizes of I'_k equal the sizes of I_k. Expansion is based on reversing the arrows in Fig. 3.9. This results in:

$$I'_k(i,j) = 4 \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) I_{k-1}\left(\frac{i+m}{2}, \frac{j+m}{2}\right) \tag{3.13}$$

It must be stressed that only those terms for which the index is an integer contribute to the sum. Now, the Laplacian pyramid is obtained by subtracting the images I'_k from the images I_k. Thus, the resolution reduction operator of the Laplacian pyramid is characterized by the difference of two Gaussian filter operators. This corresponds with the Laplace operator, which is used for enhancing edges. Therefore, this pyramid is given the name Laplacian pyramid. In the following section we discuss some of the applications of this type of pyramid.

Applications

For some applications one is interested in the zero-crossings of bandpass filtered images. The Laplacian pyramid can be useful in this type of approach. The zero-crossing analysis is related to the scale-space study which is discussed in Section 3.3. An introduction by Marr to this type of analysis and its relation to the human visual system can be found in [82]. Burt [17] [16] suggests the use of the Laplacian pyramid for image coding purposes. Burt also gives in [16] an example where the Laplacian pyramid is used for texture analysis and segmentation. He generates a Laplacian pyramid, based on a given input image, containing some textural regions. The values within the Laplacian pyramid are squared. Thereafter the values are smoothed. The resulting values might be considered as the local (power) spectral estimates which can be used for texture segmentation purposes. An example of the use of this approach can be found in [34]. In [19], Burt discusses a similar type of use of the Laplacian pyramid for automated surveillance. This algorithm should register if a movement is detected, where the movement is detected by taking the differences of succeeding frames.

In [16] and [18], Burt also suggests the use of the Laplacian pyramid for image fusion purposes and for the study on human binocular perception. Image fusion is desirable within certain applications, where one considers images from several sources. By fusing the information from these sources, one could recognize information which is not visible in the original images. The example of Burt is based on the fusion of a left image and a right image. Burt starts his fusion scheme by the initialization of two Laplacian pyramids (corresponding to two image sources), which are denoted by LL and LR. The resulting binocular pyramid is denoted by LB. The resulting fusing scheme is:

$$LB_l(i,j) = \begin{cases} LL_l(i,j) \text{ if } |LL_l(i,j)| > |LR_l(i,j)| \\ LR_l(i,j) \text{ otherwise} \end{cases}$$

The index l denotes the level index of the pyramids.

In [111], Toet gives an alternative fusion scheme. Toet defines the ROLP pyramid, which stands for Ratio Of Low-Pass pyramid. This type of fusion scheme garantuees a better preserving behavior for visually important details. The initialization of the ROLP pyramid is given by the following scheme:

$$
\begin{aligned}
R_i &= \frac{G_i}{EXPAND[G_{i-1}]} \text{ for } 1 \leq i \leq N && (3.14)\\
R_0 &= G_0, && (3.15)
\end{aligned}
$$

where G_i stands for the level i in the Gaussian pyramid and R_i for the i-th level in the ROLP pyramid. The merging scheme is similar to the Burt's merging scheme:

$$
RB_l(i,j) = \begin{cases} RL_l(i,j) \text{ if } |RL_l(i,j) - 1| > |RR_l(i,j) - 1| \\ RR_l(i,j) \text{ otherwise} \end{cases}
$$

The final-result image is obtained by expansion of the pyramid levels and a multiply reconstruction procedure.

In the literature [25] [26] [28] [27], Crowley describes the use of the Laplacian pyramid for the description of shapes. The algorithm searches for peaks and ridges within the Laplacian pyramid. By linking the adjacent peaks in the layers of the pyramid, one obtains a multiresolution tree which describes shapes. This description can be used for matching purposes.

In the following section, we discuss the binary pyramid, of which the initialization could be based on the Laplacian pyramid, however it could also be based on a logic operator.

3.2.3 The Binary Pyramid

When one starts from a binary image, the most useful operator is the logical *OR* operator. This reduction process can be denoted by:

$$
I_{k-1}(i,j) = I_k(2i,2j) \vee I_k(2i+1,2j) \vee I_k(2i,2j+1) \vee I_k(2i+1,2j+1), \qquad (3.16)
$$

where $\vee$ stands for the logical *OR* operator. It is this operator that garantuees that objects in the input image will still be connected higher up in the pyramid. In Fig. 3.10 an illustration of this type of pyramid is shown.

The logical OR-operator is also used for the initialization of the binary PXY-pyramid as suggested by Backer et al. [6]. The PXY-pyramid is a coded version of the binary pyramid using the PXY-table

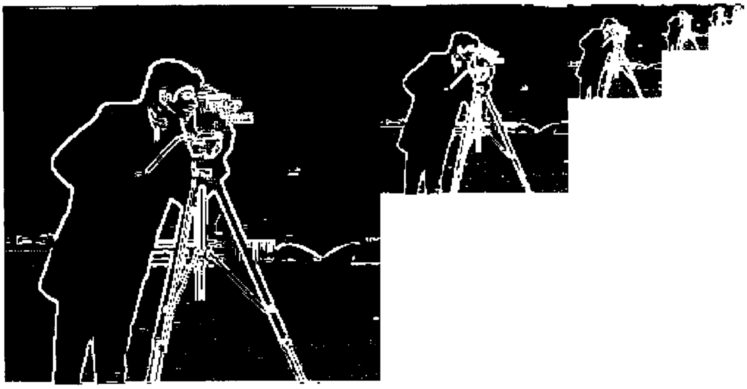

Figure 3.10: An example of a binary pyramid.

coding technique as suggested by Young et al. in [126]. This coding technique starts with scanning row by row the image from the top left corner to the bottom right corner. The table is filled in such a way that the odd index elements of the PXY-table correspond to the starting position of object runs, whereas the even elements correspond to the starting points of the background runs. The element with index 0 gives the number of elements containing PXY table information. This coding technique enables a fast extraction of object features, such that in time critical applications a fast identification of the objects can be achieved. The application mentioned by the authors is that of robot vision.

The initialization of the binary pyramid could also be based on information of another type of pyramid that already is available. Such an approach is discussed by Burt in [16]. His goal is the analysis of texture with a pyramidal data structure. He starts with the initialization of a Gaussian pyramid. After which an edge detector is applied to each level of the pyramid. This results in a binary pyramid. This binary pyramid is used for the initialization of a pyramid which contains edge density values. The edge density measure is obtained by applying a local averaging operator to the binary pyramid. The output value of the operator is used for the initialization of the edge-density pyramid.

3.2.4 The Linked Pyramid

A completely different type of pyramid is the linked pyramid. This pyramid was introduced by Burt et al. in 1981 [20]. It was developed for segmentation purposes. According to the authors during the segmentation stage of image processing one is confronted with two major problems:

1. The goal of segmentation is to partition an image into regions which have homogeneous properties. Horowitz [55] defined segmentation more mathematically as follows:

   > The result of image segmentation is a partitioning of the image X into disjoint subsets X_1, X_2,...,X_n such that:
   >
   > (a)
   >
   > $$\bigcup_{i=1}^{N} X_i = X \qquad (3.17)$$
   >
   > (b)
   >
   > $$X_i \ , i = 1, 2, ..., N \qquad (3.18)$$
   >
   > is connected.
   >
   > (c)
   >
   > $$P(X_i) = TRUE \text{ for } i = 1, 2, ..., N, \qquad (3.19)$$
   >
   > where P stands for a logical predicate.
   >
   > (d)
   >
   > $$P(X_i \bigcup X_j) = FALSE \text{ for } i \neq j \qquad (3.20)$$
   >
   > where X_i and X_j are adjacent.

   However, we can only determine whether or not a region is homogeneous if we already have a region. Therefore, the segmentation could be characterized as a chicken-egg problem.

2. The second problem is that the sizes of the pattern elements and segments are not known a priori, and different sizes are appropriate for different areas of an image.

Therefore, the authors develop an approach based on the following strategies:

1. The segmentation is performed at several levels of resolution.
2. The segmentation and image properties are recomputed in a co-operative, iterative fashion.

For this approach, the authors developed the linked pyramid. The pyramids, as discussed so far, have fixed relations between the father nodes [1] and the son nodes. For instance, the node with coordinates (i, j) at level $k-1$ (this level k should be above the bottom level) has the following son nodes: $(2i, 2j)$, $(2i+1, 2j)$, $(2i, 2j+1)$, and $(2i+1, 2j+1)$. The linked pyramid, however, is not characterized by fixed relationships and these may change during each iteration. Instead of speaking of son nodes, within the linked pyramid we speak during the iteration process of candidate son nodes. Similarly, we do not speak of father nodes, but of candidate father nodes. For a node with index (i, j) we distinguish now 16 candidate son nodes with indices:

$$\begin{array}{cccc}
(2i-1, 2j-1) & (2i-1, 2j) & (2i-1, 2j+1) & (2i-1, 2j+2) \\
(2i, 2j-1) & (2i, 2j) & (2i, 2j+1) & (2i, 2j+2) \\
(2i+1, 2j-1) & (2i+1, 2j) & (2i+1, 2j+1) & (2i+1, 2j+2) \\
(2i+2, 2j-1) & (2i+2, 2j) & (2i+2, 2j+1) & (2i+2, 2j+2).
\end{array}$$

And a node with index (i, j) has four candidate father nodes, viz.:

$$\begin{array}{c}
(\frac{i-1}{2}, \frac{j-1}{2}), (\frac{i-1}{2}, \frac{j+1}{2}) \\
(\frac{i+1}{2}, \frac{j-1}{2}), (\frac{i+1}{2}, \frac{j+1}{2}),
\end{array}$$

where only integer values of the indices are considered.

The initialization of the linked pyramid starts in the same way as the initialization of a grey value pyramid. After the initialization, each son is linked to its most similar father. Similarity is measured by calculating the difference in value. After setting the new links, the values of the nodes are recomputed. This computation is done bottom up, where the father node values are based on a weighting average of the momentary

---

[1] The term node may be read as pixel in this case.

son nodes. The weighting function is based on the number of pixels at the bottom level that correspond to each of the nodes. Thus, a son node which corresponds with a larger region in the original image will be given a higher weight in the averaging. After having recomputed the node values, the relations should be evaluated again. This process continues in an iterative fashion, and the convergence has been proven by Kasif and Rosenfeld [65]. This proof is based on a comparison with the ISODATA algorithm.

As soon as the linking process stabilizes, the segmentation result can be extracted from the pyramid. Each node in the pyramid can be seen as a starting point for a tree structure. (Although there may be nodes without son nodes!) The value of a node can be given to the corresponding pixels at the bottom level. Therefore, one should select a level in the pyramid which contains a sufficient number of nodes in relation to the number of segments to be expected. After the selection of the level, a clustering of nodes might be required, otherwise homogeneous regions might be divided up into several regions. This is caused by the limited reach of nodes. This reach is fully determined by the height of the node in the pyramid. Only the nodes of level 1 have a reach such that the tree can contain all pixels of the image. Nodes lower in the pyramid have a smaller reach, which result in a division of the homogeneous segments. It should be stressed that this segmentation algorithm does not necessarily result in connected regions.

In [3], Antonisse discusses two problems which can occur with a segmentation based on the linked pyramid. The first problem is called *the Island Problem.* This means that a small region is completely swallowed up by a larger surrounding region. In his article he gave a one-dimensional example (see Fig. 3.11). This figure shows the input signal and the initialized pyramid, which is based on a weighted filter with coefficients $(0.1, 0.4, 0.4, 0.1)$. The linking process finally results in that shown in Fig. 3.12. From this illustration we see that the island consisting of two zeroes is not recognized by the segmentation algorithm as a separate segment and is swallowed up by the surrounding segment.

The second problem dealt by Antonisse -the *checkerboard problem*- is a degeneration of the island problem. An illustration of the final linked pyramid is shown in Fig. 3.13. Now we have no dominating region and the segmentation result is one complete segment, instead of

0.375 0.862

0.13 0.62 0.75 0.862

0 0.1 0.9 0.5 0.5 1 1 1

0 0 0 0 1 1 1 0 0 1 1 1 1 1 1

Figure 3.11: The initialized linked pyramid for illustrating the island problem as discussed by Antonisse [3].

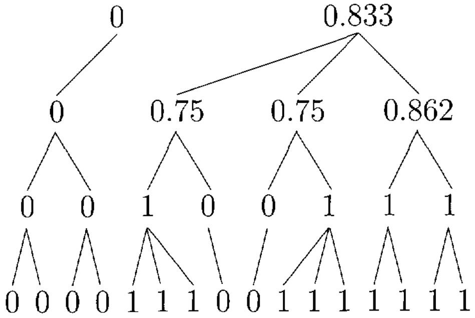

Figure 3.12: The final linked pyramid for illustrating the island problem as discussed by Antonisse [3].

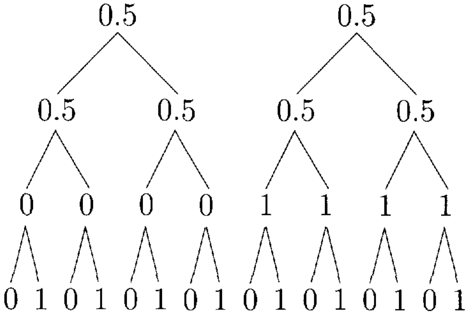

Figure 3.13: The final linked pyramid for illustrating the checkerboard problem as discussed by Antonisse [3].

16 separate segments.

In the literature, a number of modifications to the original linked pyramid approach have been proposed. Burt [20] suggests a few alternative refinements to the algorithm. One of these refinements suggests that a node should not be forced to be linked with a father node, if there is not enough similarity. This modification might prevent the occurrence of the two problems as previously discussed. Of course, the linked pyramid approach is not limited to a 1-dimensional feature space. One might also consider more features such as, for instance, color, texture, etc. More variants are discussed by Hong et al. in [53].

Applications

In [53], Hong et al. describe an algorithm for the extraction of borders of objects in a noisy image. For this purpose, he initializes a grey value pyramid. The levels in the grey value pyramid are used as the input for an edge detector, which results in an edge pyramid. Then the edges in adjacent levels are linked in the same way as discussed above. This is done bottom-up. (Note that if the linking process were done top-down, a son might be linked to more than one father.) In contrast to the initialization to the original linked pyramid approach, the links will not be re-evaluated. A son is linked to a father if the difference in the angles of the edges is small enough. Otherwise, the son becomes the root of a tree.

In [54], Hong et al. propose some alternatives to the weighting scheme, with which the actual node values are computed. These schemes are studied for the extraction of compact regions. In [52], a technique is described by Hong et al. for the extraction of compact objects which is based on the linked pyramid. The search is carried out on several levels of resolution, so that compact regions of different sizes can be detected. The algorithm starts with the initialization of a linked pyramid, containing grey value information. From this grey level pyramid a linked pyramid containing edge information is obtained. This pyramid contains the edge direction as well as the magnitude. The information in the edge pyramid is used for the calculation of what are called the surroundedness scores. This measure represents how far a node might be considered an interior point. With this measure and these pyramids, the compact regions are extracted.

In [106], Spann et al. use a linked pyramid for the detection of thin structures. The algorithm uses a local dynamic thresholding operator, which successively reduces the object width until it has disappeared. Since the object width can vary along its length, a multi-resolution approach is preferable. Pietikäinen describes in [96] the use of a linked pyramid for the segmentation of textured images. He also proposes a variant to the original linked pyramid approach in which linking of the nodes is done top-down as well as bottom-up. This refinement leads, according to Pietikäinen, to a significant improvement. In this approach, global information obtained from the upper pyramid levels is used to locate the large homogeneous areas, while more accurate boundary information about these areas is obtained by linking nodes on lower levels to the nodes representing these major areas. Comments on his initialization technique can be found in Section 3.4 of this thesis.

3.3 Pyramids and Scale Space

In a search for a technique to describe signals qualitatively, Witkin [123] introduced a concept called *scale space filtering*. The description aimed at is based on the extrema of a signal and the extrema of its first few derivatives. The range of scale on which these extrema are still visible is essential to the description. A set of images showing the phenomena within the image on different scales can be obtained by filtering the image with a filter of which the sizes are varied. The disadvantage of this approach is that we only possess a limited number of representations of the original image on different scales. The relationships between extrema on adjacent levels are not unambiguous. Therefore, Witkin introduced a continous scale. This filtering process is called *scale-space filtering* and an image representation on a certain scaling level is called a *scale-space image.* For a number of reasons, such as well-behavedness, Witkin chooses the Gaussian filter for the scale-space filtering. For the 1-dimensional case, where the signal $f(x)$ is convolved with the Gaussian kernel $g(x)$ we find:

$$F(x,\sigma) = f(x) * g(x,\sigma) = \int_{-\infty}^{\infty} f(u) \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-u)^2}{2\sigma^2}} du, \qquad (3.21)$$

where σ is the *parameter of scale* and determines the width of the filter. For the description, Witkin studies the zero-crossings of the second derivative for which the third derivative is unequal to zero. A more detailed discussion by Marr on the study of zero-crossings can be found in [82].

In [70], Koenderink proves that if any feature higher in the scale space is required to possess a "cause" at a lower level, the family of Gaussian kernels is unique. Koenderink also proves that a logarithmic spacing of the planes in a discrete scale space is required. Further, he shows that the pyramidal data structures are too coarse from a mathematical point of view. However, because of the powerful algorithms applied, the consequences of the coarseness are limited. By following another theory, Babaud et al. have later also proven the uniqueness of the Gaussian kernel [5].

3.4 Pyramids for the Study of Texture

To now, we have given a summary of existing pyramid types and some corresponding applications. The discussion aimed to give an overview of the work done with respect to the pyramidal data structures and the aspects to be considered when such structures are to be applied. The goal of this thesis is to study the usefulness of pyramidal data structures

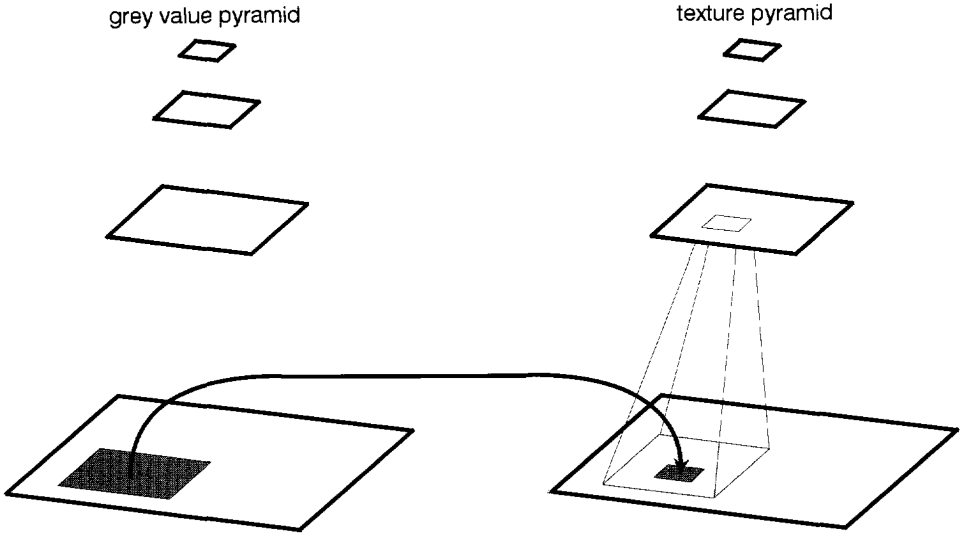

Figure 3.14: Initialization of a texture pyramid.

for texture analysis and segmentation. We already referred to the work of Pietikäinen [96]. He divides the input image, which contains textural regions, into blocks of 8 × 8, and determines for the blocks a texture value. This results in a texture map with reduced sizes. As a result of the small block size on which the texture values are determined, the texture values show a significant variance. In order to smooth out the variance, Pietikäinen suggests the use of a median filter to smooth the resulting texture map. This map is used as the input for a linked pyramid. The calculation of the node values is based on unweighted averaging of the corresponding son nodes. The initialization of the texture pyramid is illustrated in Fig. 3.14. Because of the type of information in the pyramid, the pyramid is called a *texture pyramid.* For this type of initialization we find that nodes higher in the pyramid do not correspond with a coarser description of the phenomena. The pyramidal data structure is only used as a mechanism for decreasing the variance of the texture values. Therefore, the type of approach might be compared with the *Split-and-Merge* segmentation algorithm.

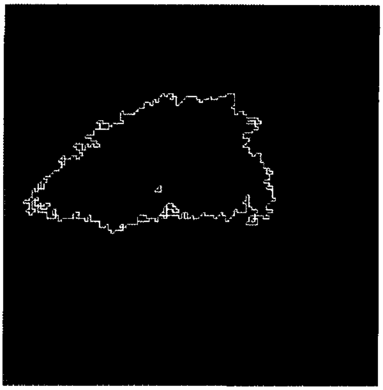

Figure 3.15: Segmentation result based on the first type of initialization.

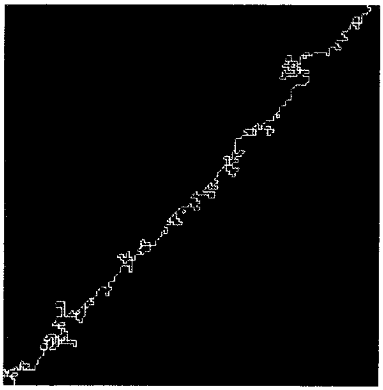

Figure 3.16: Segmentation result based on the first type of initialization.

In [34], Eijlers shows segmentation results which are based on a texture pyramid which has been initialized as discussed and segmented with an alternative region growing procedure. Two typical segmentation results are shown in Figures 3.15 and 3.16. Besides this type of initialization, we can distinguish two other types of initialization.

As stated above, we are interested in a technique such that nodes at higher levels in the texture pyramid correspond to a quantification of the texture aspects on a coarser scale. An example of such an initialization technique is the second type of initialization which is shown in Fig. 3.17. The idea behind this initialization technique is that the texture operator is applied on the original image, but the sizes of the

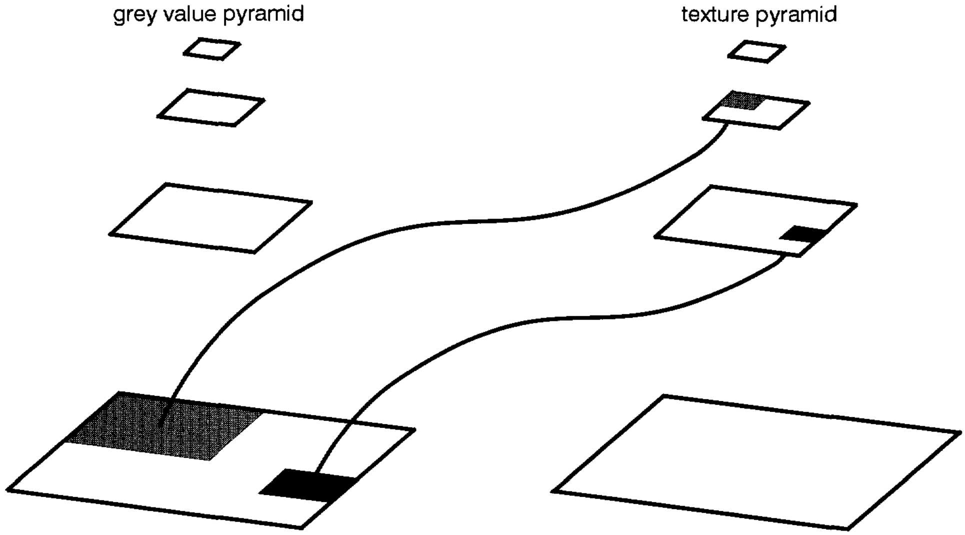

Figure 3.17: The second type of initialization of a texture pyramid.

texture operator depend on the level of the texture pyramid to be initialized. For instance, if we apply on level l a window with sizes $2^a \times 2^a$, then on level $l-1$ we should apply a window with sizes $2^{a+1} \times 2^{a+1}$. However, resizing of the window is not always enough. For operators based on the Spatial Grey Level Dependence Method (Section 2.3.1) or the Grey Level Difference Method (Section 2.3.2) we should enlarge the vector with a factor of two as well. In the figure we have shown a grey value pyramid, of which only the bottom level -thus the original image- is used. The complete grey value pyramid is shown for comparison with the other types of initialization. The main disadvantage of this method is that the higher the node in the pyramid the more inaccurately it can be determined. After all, the ratio between the image sizes and the window sizes changes with the level in the pyramid, which means that the artefacts at the borders are more significant higher in the pyramid. Because of this disadvantage, this type of initialization is not recommended.

The third type of initialization is shown in Fig. 3.18. It is only

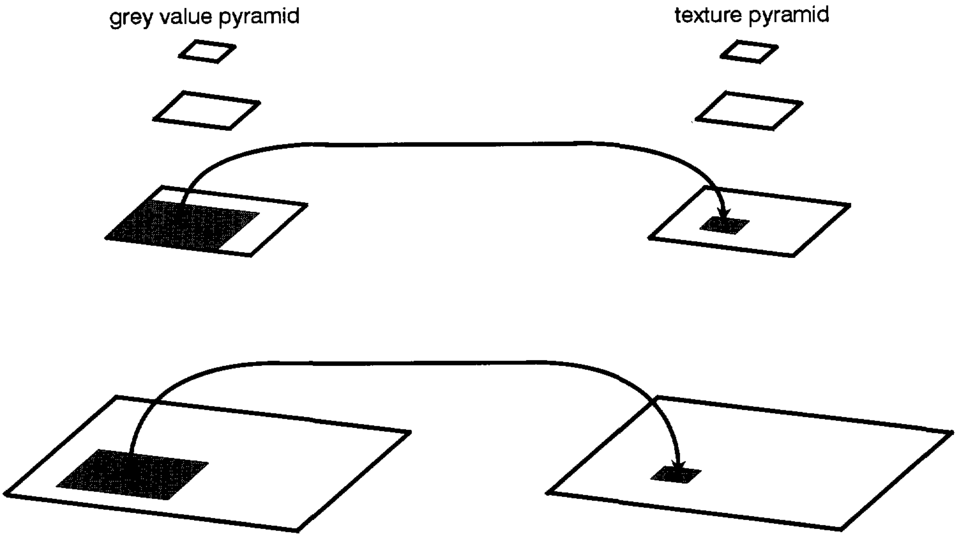

Figure 3.18: The third type of initialization of a texture pyramid.

this type of initialization which requires the existence of a grey value pyramid. Texture values are now determined by applying the same texture operator with the same window sizes to the corresponding level of the grey level pyramid. Because of the border artefact, this type of initialization is not recommendable either.

Based on the preceding considerations, one might conclude that the usefulness of pyramidal structures for texture analysis and segmentation is limited. However, there is one type of initialization which has not been discussed yet. This type of initialization is based on a texture model. The values in the texture pyramid now correspond to the parameters of the texture model. An equivalent for this approach for the segmentation based on motion information has been described by Bierhuizen in [10].

To proceed, a texture model has to be chosen. Of course, from such a model we expect a good performance in discriminatory sense. Further, a model is to be preferred, in which scaling behavior is incorporated in the model. This behaviour is not present explicitly in most of the models discussed in Chapter 2. However, models which are based on the fractal theory explicitly quantify scaling behavior. This theory and its use for texture analysis and segmentation is discussed in the next chapter.

Chapter 4

Fractal Theory

4.1 Introduction

In the preceding chapters we concluded that we are interested in a theory that ables us to describe phenomena on a number of discrete scaling values. Such a theory is the *fractal theory.* The study of the applicability of this theory for image processing purposes started in the early eighties. In other disciplines, it had already shown to be of interest for the description of a range of natural phenomena. In this thesis, we focus on the applicability of this theory for the analysis and segmentation of textures in pyramidal data structures. Therefore, the discussion of the fractal theory is limited to those aspects of fractal theory that are of use within the focus of attention.

The name *fractal theory* originates from Mandelbrot. However, the study of phenomena which showed fractal behavior started much earlier, however, those studies lacked the ability to display the mathematical functions. When Mandelbrot started his study, display facilities connected to digital computers had begun to appear. He made an intensive study of functions showing fractal behavior and illustrated its applicability in a broad range of disciplines. The interest in fractal theory was given impetus by the beautiful illustrations found in [75], which has later been replaced by [77].

The essence of fractal theory is often illustrated with the question: *How long is the coastline of Britain?* (see Fig. 4.1). To answer this

Figure 4.1: The coast of Britain.

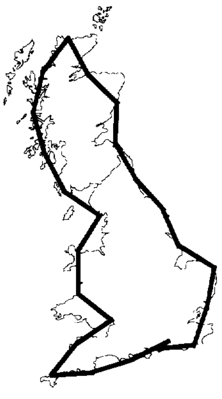

Figure 4.2: Measuring method 1.

problem, Mandelbrot suggests four types of measuring techniques.

The first type of method suggested is the most obvious one. The measurement is based on a yardstick of length ϵ. During the measurement the begin point of the yardstick is laid against the end point of the preceding step. Now, the length can be found by multiplying the length of the yardstick by the number of steps. As shown in Fig. 4.2 this types of measuring results in a simplification of the coastline as a sequence of polygons. Note, that the coastline cannot be coverered by an integer number of yardsticks.

With the second method the coastline is followed at a maximal distance of ϵ. The path to be followed should be the shortest. Again, the length is measured with a yardstick. (see Fig. 4.3).

In contrast to the second method, the third method is based on a symmetrically defined band around the coastline. This band has a width of 2ϵ. Now, the length is estimated by dividing the area by the width (see Fig. 4.4).

The fourth type of method is based on the pointillist painting style. The coastline is completely covered by the minimum number of circles with radius ϵ. The centres of the circles may lie in the sea, as well as on the shore. The length can be estimated by dividing the total area of the circles by 2ϵ.

One might expect that all these methods would result in consistent answers as to the length of the coastline. However, if we measure the

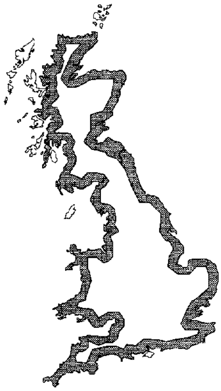

Figure 4.3: Measuring method 2.

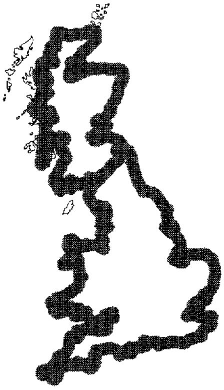

Figure 4.4: Measuring method 3.

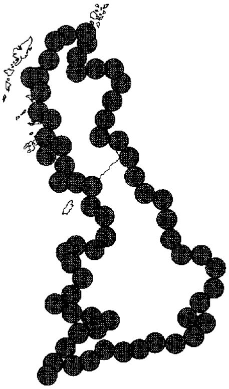

Figure 4.5: Measuring method 4.

coast length with a smaller ϵ, we will follow the coastline more accurately, which results in an increase in the length. Although, the dependency found between the length and the resolution of measurement is not valid for each range of resolution, in general, we could define the coast length L as a function of the resolution ϵ, so $L(\epsilon)$. This finding results from the irregularity of the coastline. The relation found has been discussed earlier in the literature, and has been modeled as:

$$L(\epsilon) \sim F\epsilon^{1-D}, \tag{4.1}$$

where F stands for a constant and D for the *fractal dimension.* It was Mandelbrot who suggested calling the parameter D the fractal dimension. The fractal dimension is supposed to be real-valued. The fact that the dimension is supposed to be real-valued underlies the nomenclature. The term fractal originates from the Latin adjective *fractus.* The corresponding Latin verb *frangere* means *to break.* This fractal dimension exceeds the topological dimension of 1, and therefore the coastline is defined as a *fractal pattern.*

We have seen that the coastline is very irregular, this implies complex behavior. Complexity, however, does not exclude the possibility of a certain degree of structure. If we take a part of the coastline and we enlarge it to the size of a larger part, we will see that those parts appear similar. This scaling behavior is called self-similarity. [1]

To proceed, we now introduce the *Koch-curve* as a model of the coastline which is an example of an exactly self-similar fractal. This model is based on an *initiator* and a *generator.* The initiator is in this case is a straight interval of length 1. The generator consists of four line segments of length $\frac{1}{3}$ ordered as shown in Fig. 4.6. Now, the generator is scaled such that it fits within the interval corresponding with the initiator, then each straight line segment of the generator is considered as a new initiator. All new initiators will be replaced by a scaled version of the generator. This process proceeds recursively and will never end. Note that the resulting curve is nowhere differentiable. The correspondence with the coastline is limited because of its strong underlying

---

[1] Note that classical information theory does not recognize such an underlying structure. The information measures will, therefore, overestimate the information contents.

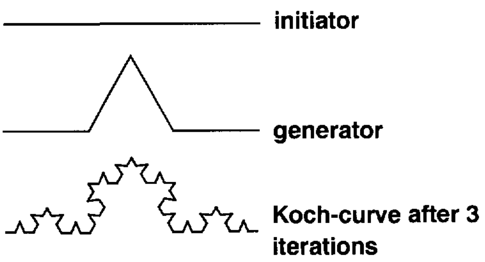

Figure 4.6: The Koch-curve.

structure. However, the model is illustrative for the discussion of the length of the coastline.

The length of the Koch-curve after the first iteration is equal to $\frac{4}{3}$. After the second iteration the length is increased by a factor $\frac{4}{3}$. Thus, for the Koch-curve we can write $L(\frac{\epsilon}{3}) = \frac{4}{3}L(\epsilon)$. Equation 4.1 can now be solved, which results in a fractal dimension of $D = \frac{\log 4}{\log 3}$.

To here, we have only mentioned the fractal dimension. However, a whole range of dimension definitions exists. The choice of the dimension definition to be used is most often based on practical considerations. For a number of definitions, equal values might be obtained.

Before defining a fractal pattern, the *topological dimension* D_T has to be explained. Suppose, we are studying objects in an Euclidean space $\mathbb{R}^E$. Then these objects have at least a topological dimension of 0 and at most E. The topological dimension is always an integer. Examples of objects are: a point ($D_T = 0$), a line ($D_T = 1$), a plane ($D_T = 2$), etc. Now an object (or a set) is defined to be a fractal if:

$$D \geq D_T \tag{4.2}$$

It must be stressed that objects which differ significantly in topology cannot always be discriminated by their topological dimension, whereas the fractal dimension might discriminate.

The Koch curve we have discussed is a well-known example of a family of deterministic fractals. For the Koch-curve the initiator consists of a straight interval of length 1. One could imagine that other initiators, which are based on a polygonic curve might also result in an interesting fractal. Of course, the shape of the generator can be changed, such that a completely differently shaped fractal will be obtained. To make a more stochastic type of fractal, one could define more generators. Each time that a part of the fractal has to be replaced by a smaller generator, one of the generators is chosen at random.

In this section, we have given an example of a fractal. We have seen that the fractal dimension is a sort of measure for the irregularity of objects. The scaling behavior which is included in the definition of self-similarity is essential to fractal theory. In the next section we will discuss fractal Brownian motion, which is an extension to the original Brownian motion, and which is a typical example of a stochastic fractal.

4.2 Fractal Brownian Motion

In the preceding section, we presented the Koch-curve as a fractal model of the coast line. This type of fractal has two disadvantages. Firstly, the fractal has too strong a topological similarity, which results in an unnatural appearance. Secondly, the curves show self-similarity only on discrete scaling steps. For the modeling of natural phenomena a more stochastic type of model is desirable. The fractal Brownian motion is an example of a stochastic fractal model[2], which is an extension to the ordinary Brownian motion model $B(t)$. The ordinary Brownian motion is defined as a real random function with independent Gaussian distributed increments such that $B(t_2) - B(t_1)$ has mean zero and variance $|t_2 - t_1|$, and such that $B(t_2) - B(t_1)$ is independent of $B(t_4) - B(t_3)$ if the intervals (t_1, t_2) and (t_3, t_4) do not overlap. The fractal Brownian function is discussed in the following section .

4.2.1 Brownian Motion and Fractal Brownian Motion

Suppose that the ordinary Brownian motion is denoted by $B(t)$, where t stands for the time.[3] Now, the fractional Brownian function (fBm)

---

[2] The motion model is also called fractional Brownian motion.

[3] Mandelbrot uses in [81] the notation $B(t, \omega)$, where B stands for the Brownian motion, t stands for the time and ω designates the set of all values of a random function. Therefore, his notation might be considered as more formal. However, in this discussion ω plays no role and is therefore omitted.

is defined as an extension to the ordinary Brownian motion. For this model, the spectral exponent H is introduced. Mandelbrot has proven in [81] that H should satisfy $0 < H < 1$. The fractal Brownian motion of exponent H is a moving average of $dB(t)$ in which the past increments are weighted by the kernel $(t-s)^{H-\frac{1}{2}}$. For the case that $t > 0$, it is called the *reduced fractal Brownian motion.* This function is denoted by $B_H(t)$ and formulated as follows:

$$\begin{aligned} B_H(0) &= b_0 \\ B_H(t) - B_H(0) &= \frac{1}{\Gamma(H+\frac{1}{2})}\left\{\int_{-\infty}^{0}\left[(t-s)^{H-\frac{1}{2}} - (-s)^{H-\frac{1}{2}}\right] dB(s) + \right. \\ &\qquad \left. \int_0^t (t-s)^{H-\frac{1}{2}}\, dB(s)\right\} \end{aligned} \tag{4.3}$$

The introduction of $\frac{1}{\Gamma(H+\frac{1}{2})}$ is motivated by Mandelbrot as a coefficient that garantuees that if $H - \frac{1}{2}$ is an integer, the integral results in an ordinary repeated integral. In the discussion of the fractional integrals, which follows in the intermezzo on page 62, the introduction of this factor will be made clearer. Note that for $b_0 = 0$ and $H = \frac{1}{2}$ we find that the fractal Brownian motion is equal to the ordinary Brownian motion. It is this value for H, which divides the behavior of the fractal Brownian motion into two intervals. For the first interval of $0 < H < \frac{1}{2}$ the increments are negatively correlated, which result in a more chaotic behavior . For the second interval, the increments are positively correlated, which result in more or less smooth behavior.[4] In the literature one often states $b_0 = 0$. Equation 4.3 can be written more symmetrically as follows:

$$\begin{aligned} B_H(t_2) - B_H(t_1) &= \frac{1}{\Gamma(H+\frac{1}{2})}\left\{\int_{-\infty}^{t_2}(t_2-s)^{H-\frac{1}{2}}\, dB(s) - \right. \\ &\qquad \left. \int_{-\infty}^{t_1}(t_1-s)^{H-\frac{1}{2}}\, dB(s)\right\}. \end{aligned} \tag{4.4}$$

---

[4] Mandelbrot also introduced the term *fractal Brownian noises*, which is used for successive differences between points of a fractal Brownian motion signal.

Now, it can be proven[5] that the spectral exponent H of the fractal Brownian motion is related to the fractal dimension as follows :

$$D = 2 - H. \tag{4.5}$$

The model, as suggested, is one-dimensional. However, the extension to more dimensions is trivial. As is discussed by Falconer in [36], the extension to a more dimensional space is such that each component is a fractal Brownian motion in itself.

Intermezzo

Before we continue we here make an excursion into the fractional integrals *as discussed by Weyl in [122]. These integrals show a significant similarity with the fractal Brownian motion. The theory of the fractional integrals is based on iterative integration of functions. Suppose:*

$$\begin{aligned} f_1(x) &= \int_0^x f(x)\,dx && (4.6)\\ f_2(x) &= \int_0^x f_1(x)\,dx && (4.7)\\ f_3(x) &= \int_0^x f_2(x)\,dx && (4.8)\\ f_n(x) &= \int_0^x f_{n-1}(x)\,dx && (4.9)\end{aligned}$$

This can be written more compactly as:

$$f_n(x) = J^n f(x), \tag{4.10}$$

where J symbolizes the integration process. This results in:

$$J^{n+1} f(x) = \frac{1}{n!}\int_0^x (x-\xi)^n f(\xi)\,d\xi \ (n=0,1,2,...) \tag{4.11}$$

This result is only valid for positive integer values of n. An equivalent for positive real-values is:

$$J^{\alpha} f(x) = \frac{1}{\Gamma(\alpha)}\int_0^x (x-\xi)^{\alpha-1} f(\xi)\,d\xi. \tag{4.12}$$

---

[5] See for a clearly written proof [36].

> *Instead of the parameter n, we now use the parameter α. Note that Γ-function generalizes the $n!$-normalization coefficient.*

The fractals we have discussed up to now were characterized by a space of which the components were equal in meaning. The graph of the 1-dimensional fractal Brownian motion, however, is characterized by two different types of scale. The first one is the signal level, and the other one is the time scale. This makes it theoretically[6] impossible to speak in terms of distance, circular, or square. For the components, the rescaling ratios of t and B_H are different. Therefore, self-similarity is not the right term to describe the scaling behavior. The term to describe this behavior is called *self-affinity*. A more detailed discussion by Mandelbrot on this subject can be found in [79].

4.2.2 Properties of the Fractal Brownian Motion Model

Before we can study the use of the fractal Brownian motion as a model for texture, we need to investigate the properties. Some of the properties we deal with are a direct consequence of its definition. Other properties might be intuitively correct, but for the complete proof the reader is referred to the corresponding article. The first three properties to be mentioned are a direct consequence of the definition and are of less importance for quantifying texture properties.

1. One of the properties which are mentioned in the literature is that increments of the fractal Brownian motion have zero mean. This is because of the zero mean of the increments of the ordinary Brownian motion. The weighting introduced by Mandelbrot for the introduction of the fractal Brownian motion is of no influence to this mean value.

2. From the definition it follows that the increments are stationary.

---

[6] In practice, we can define boxes in this space and determine the fractal dimension. This approach is discussed in Section 4.4.

3. The fBm is almost nowhere differentiable. This property follows from its definition.

The following properties are especially of interest for image processing applications.

4. The increments of the fractal Brownian motion are self-similar. This means:

   $$B_H(t+T) - B(t) \triangleq h^{-H}\left[B_H(t+Th) - B_H(t)\right], \tag{4.13}$$

   where $\triangleq$ is used to denote identical in distribution function. The proof can be based on a proof given by Weyl [122], where Weyl proves a similar property.

5. For the standard deviation of the increments of the fBm a T^H-law is valid. This means:

   $$E\left[\left(B_H(t+T) - B_H(t)\right)^2\right] = T^{2H} V_H, \tag{4.14}$$

   where

   $$V_H = \frac{1}{\Gamma\left(H+\frac{1}{2}\right)^2}\left\{\int_{-\infty}^{0}\left[(1-s)^{H-\frac{1}{2}} - (-s)^{H-\frac{1}{2}}\right]^2 ds + \frac{1}{2H}\right\}. \tag{4.15}$$

   This result can be found by writing out the left side of Eq. 4.14. In [81], Mandelbrot discusses this proof as well. Keller and Seo [68] have used Equation 4.14 for the determination of the fractal dimension. The discussion of their results in comparison to other techniques is postponed to Section 4.5.

6. In [81], Mandelbrot proves that the spectral density of a fBM signal is proportional to f^{-2H-1}. A more detailed discussion of this property by Voss -with an extension to more dimensions and the accent on synthesizing- can be found in [119] and in [120]. A general discussion of techniques for the generation of textures which are based on the fractal theory is given in Section 4.3 of this thesis. In Section 4.3 also examples of textures are given which are based on this property. Pentland [93], [94] used this property for the estimation of the fractal dimension of textural scenes. The discussion of his technique is postponed to Section 4.5

7. A property that directly follows from its definition, and the properties discussed so far, can be denoted as:

$$Pr\left\{\frac{B_H(t_2) - B_H(t_1)}{|t_2 - t_1|^H} < y\right\} = F(y), \tag{4.16}$$

   where $F(y)$ is a cumulative distribution function of the random variable y. This property is used by Pentland [94] for deriving the following properties with respect to image transformations:

   (a) Pentland proves in [94] that a linear transformation of a fractional Brownian function results in a fractional Brownian function with the same fractal dimension.

   (b) In [94], Pentland proves that when a 3-D surface with a spatially isotropic fractal Brownian shape is projected to an image, the resulting intensity surface is fractal Brownian as well, and the fractal dimension is identical to that of the surface normal. The proposition is, however, restricted to a Lambertian surface reflectance function and constant illumination and albedo.

   (c) Pentland's latter proposition is also valid vice versa as has been proven by him in [94]. This means, that if an image intensity surface is a two-dimensional fractional Brownian then the image 3-D surface must be spatially-isotropic fractional Brownian as well. For this proof it is also assumed that the surface is Lambertian and the illumination and albedo are constant.

4.3 Generation Techniques

To here, we have discussed the concept of fractal theory and the fractal Brownian motion as an example of a fractal model. With this knowledge we can develop generation techniques which enable us to generate fractal signals based on the general fractal concept, or which are based on the fractal Brownian motion model. In Section 2.2, we discussed the advantage of possessing such tools. We concluded that we can test models on their appropriateness for certain classes of textures, and we can test analysis tools on generated data. By the use of this approach, fully conditioned experiments can be carried out. Noise behavior and sensitivity to other distortions which are in agreement with the practical situation can be investigated independently. In this section we discuss the most common types of generation techniques. Not all of the techniques to be discussed result in a purely fractal-type signal. However, these techniques might be sufficient for the applications for which they were developed.

In [119], Voss discusses that generation can be based on the summing up of Gaussian bumps. This technique is adapted to a pyramidal data structure by Pentland in [95]. The generation starts with filling the levels in the pyramid with Gaussian distributed noise. The variance at the apex is equal to σ^2. The variance of the noise at the level directly beneath the apex is equal to $\sigma^2(1 - H)^2$. Level 2 is filled with noise with variance $\sigma(1 - H)^4$, and so on. After having filled the complete pyramid with noise, the levels are summed up. This technique is based on the interpolation technique as discussed in Chapter 3. With this generation technique each node manifests in the final result image as a Gaussian bump. The image obtained is not a true fractal surface, but might be considered to be a close approximation.

A technique suggested by Fournier et al. [39] is called the *Midpoint Displacement Method.* This method has been critized by Mandelbrot in [76], because the product of the method would not be sufficiently fractal-like. A reply by the authors to Mandelbrot's reaction can be found at the end of [76]. The method starts with initializing the corner values. Then a recursive process starts. Each midpoint is replaced by the average value of its corner points, and this average value is perturbed by an amount which is related to the size of the square. The main advantage of the method is its reduced processing time. Further information on this type of generation technique is given by Voss in [119]. Some alternative refinements are given by Jeffery in [58].

Because we know the correlation behavior of the increments of a fBm process, it is also possible to develop a method which generates a signal of which the correlation of the increments is in agreement with the fBm model. This technique has been discussed by Lundahl et al. [74] as well as by Ohley and Lundahl in [89]. The process starts with the initialization of the image, by filling it with a Gaussian random generator. Then the image samples are correlated by a correlation step, such that the correlation obtained is in agreement with the original fBm process. The final image is obtained after an integration step. What is interesting about their technique is the fact that it is based on a discrete version of the fractal Brownian motion model.

In [119], Voss discusses a method to generate a Brownian relief. This technique is based on summing up randomly placed randomly oriented faults in the plane. The profile of the fault is that of a step function and its amplitude is random. Because the method can only generate signals for which $H = \frac{1}{2}$ and because the method is computationally expensive, it is of limited use.

A method which has been called the *Cylindrical Integration Method* is mentioned by Dodd in [32] and [31]. Initialization starts with generating an image which is filled with a Gaussian random field. This image is fitted on a cylinder, such that the two vertical borders are connected. The result is obtained by integration over a path defined on the cylinder. The disadvantage of this method that it can only generate textures for which $H = \frac{1}{2}$.

In Section 4.2.2, we mentioned that the magnitude of the spectrum of a fractal Brownian motion signal is of the form f^{-2H-1}. In [119], Voss describes a technique which starts with an image which is filled with white noise. This image is transformed to the Fourier domain. In this domain the magnitude is forced to the correct value. The random behavior remains in the phase image. By transforming back, the resulting image is obtained. This is the type of technique we have used. Examples of images with different fractal dimensions are shown in Figs. 4.7 - 4.15. One can use such images for testing the performance of man in estimating the fractal dimension. An extensive study carried out by Dodd [31] shows that the minimum resolvable fractal dimension was found to be 0.06. Pentland describes in [94] an experiment where naive subjects were asked to estimate the roughness of fractal surfaces on a scale of one to ten. There was a correlation of 0.98 between the estimated roughness and the fractal dimension of the surface.

In this section we have discussed the generation of monochrome textures signals. The generation is, however, not limited to monochrome textures. In [31], the technique of generating colored fractal textures

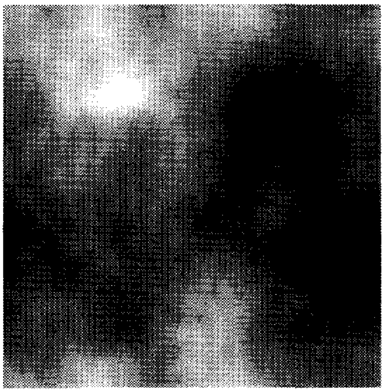

Figure 4.7: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a fractal dimension of 2.1.

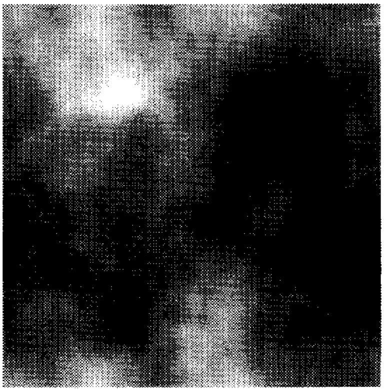

Figure 4.8: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a fractal dimension of 2.2.

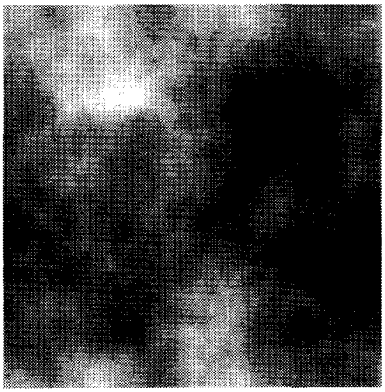

Figure 4.9: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a fractal dimension of 2.3.

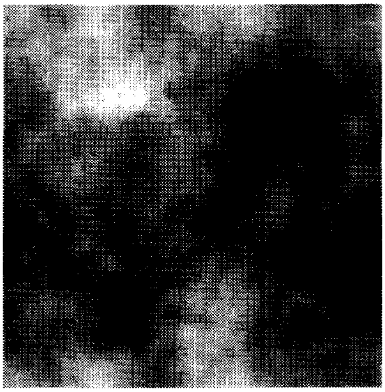

Figure 4.10: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a fractal dimension of 2.4.

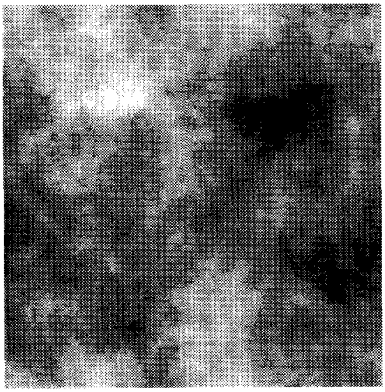

Figure 4.11: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a fractal dimension of 2.5.

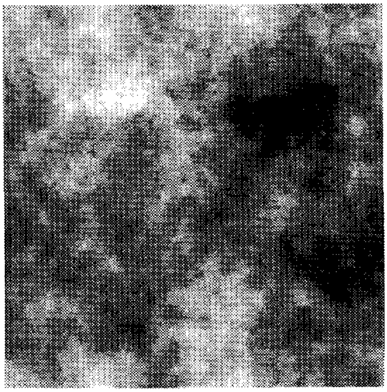

Figure 4.12: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a fractal dimension of 2.6.

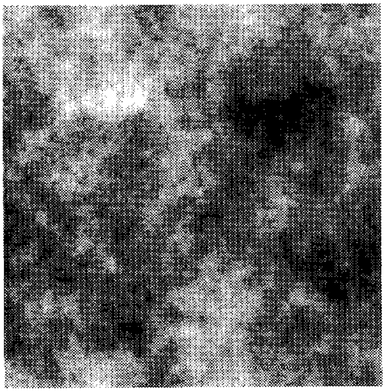

Figure 4.13: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a fractal dimension of 2.7.

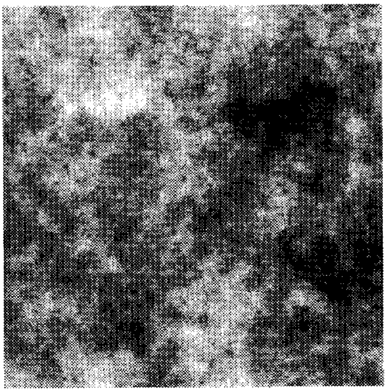

Figure 4.14: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a fractal dimension of 2.8.

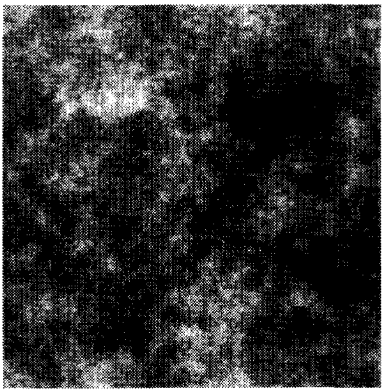

Figure 4.15: A synthesized texture generated with the Fourier filtering technique of Voss [119] with a dimension of fractal 2.9.

is discussed by Dodd. Dodd is interested in generating natural looking textures which are based on a given natural scene. This scene is analyzed before generation. First, the RGB components are decorrelated. Techniques for decomposition which can be used are based on the eigenvectors, the principal component decomposition or the Karhunen-Loeve transform. The fractal properties per component are determined, then a fractal signal per component is generated. One of the techniques as discussed in this section could be used. The resulting colored image is obtained by transforming the components back to the RGB space.

4.4 The Box-Counting Dimension

To estimate the fractal dimension of a phenomenon (a set or a signal), the box-counting dimension[7] is often recommended. Determining the box-counting dimension is based on covering the phenomenon with a mesh of boxes of size ϵ. The variable ϵ is the resolution parameter. Now, for a whole range of values for ϵ, the number of boxes $N(\epsilon)$ that are filled with at least one point of the set is counted. (see Fig. 4.16) The box-counting dimension D_B is defined by Mandelbrot as (see [78]):

---

[7] The name *box-counting dimension* is often abbreviated to *box-dimension.*

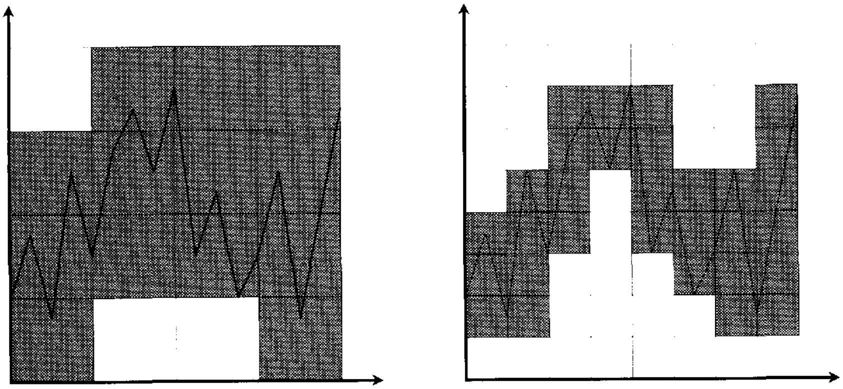

Figure 4.16: Two resolution representations for the determination of the box-counting dimension.

$$D_B = -\lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log \epsilon}, \tag{4.17}$$

which is based on the assumption that $N(\epsilon)$ behaves like:

$$N(\epsilon) \propto \epsilon^{-D_B}. \tag{4.18}$$

In practice, this means that the number of filled boxes is determined for a sufficient number of resolution values. These results are plotted in a log-log graph. Finally, the box-counting dimension is obtained by determining the slope of the line through the points. By using linear regression on the logarithmic values of ϵ and $N(\epsilon)$ we can obtain the box-counting dimension automatically. The applicability of Eq. 4.18 to the underlying study can be verified by checking whether the points in the log-log plot really have the required linear relation. Note that for a coarser resolution range, the dimension of a signal tends towards 1.

A variant to the box-counting approach has been suggested by Voss [120]. Suppose that $P(m, \epsilon)$ stands for the probability that there are m points within an E-dimensional cube of size ϵ centered about an arbitrary point in the space S. As $P(m, \epsilon)$ is a probability function, the following relation is valid:

$$\sum_{m=1}^{N} P(m, \epsilon) = 1 \text{ for all } \epsilon, \tag{4.19}$$

where N stands for the maximum number of possible points within a box. The total number of points is supposed to be equal to M. Then, the number of boxes with m points inside the box is $\frac{M}{m}P(m,\epsilon)$. To answer the question of how many boxes are filled with m points, we introduce $Q(m,\epsilon)$, which stands for the number of boxes filled with m points. Now, suppose we select an arbitrary chosen point. What is the chance that it is in a box filled with m points? We know that there are $Q(m,\epsilon)$ boxes filled with m points. Thus, there are $mQ(m,\epsilon)$ points in $Q(m,\epsilon)$ boxes which are filled with m points. Therefore, the probability for having a box filled with m points is equal to:

$$P(m,\epsilon) = \frac{mQ(m,\epsilon)}{M}. \tag{4.20}$$

Then, the expected total number of boxes $N(\epsilon)$ needed to cover the set is given by:

$$N(\epsilon) = M \sum_{m=1}^{N} \frac{1}{m} P(m,\epsilon). \tag{4.21}$$

The constant M is not of importance to the discussion and therefore the expression is simplified to:

$$N(\epsilon) = \sum_{m=1}^{N} \frac{1}{m} P(m,\epsilon). \tag{4.22}$$

This expression is proportional to ϵ^D and can be used for the estimation of the fractal dimension. A related measure is called the *lacunarity*. It is based on the moments $M^q(\epsilon)$ which are defined as followed:

$$M^q(\epsilon) = \sum_{m=1}^{N} m^q P(m,\epsilon). \tag{4.23}$$

The lacunarity $\Delta(\epsilon)$ is defined as:

$$\Delta(\epsilon) = \frac{E\left\{M^2(\epsilon)\right\} - E\left\{M^0(\epsilon)\right\}^2}{E\left\{M^0(\epsilon)\right\}^2}. \tag{4.24}$$

It has been shown that textures with equal fractal dimension, which appeared to the human observer as different, result in discriminating values for the lacunarity. In the next section we discuss dimension estimators as they have been proposed and studied in the literature.

4.5 Texture Measures based on the Fractal Dimension

The study of the use of the fractal dimension as a texture operator is in the first instance related to two names, namely Pentland and Peleg.[8] Both started at almost the same time with their studies and are considered to be the pioneers in this field.

Pentland's motivation to start this study was the natural appearance of scenes generated according to a fractal model. In his study as described in [93] [94], he focusses on the fractal Brownian motion as a model for textures. The appropriateness of the model was tested by evaluating Eq. 4.13. The dimension was measured from the Fourier power spectrum of blocks with sizes 8 × 8. A technique which we have discussed in Section 4.2.2. For the limited number and selected types of textures chosen from the Brodatz texture collection [14], Pentland considered the fractal model to be appropriate. He stresses that for the natural textures, the scaling behavior should be divided into disjunct intervals. If the interval length is sufficiently large, the use of the fractal Brownian motion model is justified. In his study he only determines one fractal dimension, which corresponds to the interval with the smallest scale. The use of the power spectrum for the estimation of the fractal dimension has also been studied by Blackledge and Fowler in [11]. Their study focused on the segmentation of Synthethic Aperture Radar Images and showed that the algorithm may be of value for target detection.

In [68], Keller and Seo describe an original use of Eq. 4.14 for texture discrimination purposes. The method is for practical reasons 1-dimensional. This gives the opportunity to study a possible directionality of the underlying texture. The authors, however, do not use the relation in its usual way, but they determine the constant in the relation. According to the authors, this method is justified if the scales of the regions of interest are approximately the same.

---

[8] Nguyen and Quinqueton published in 1982 an article [88] dealing with the use of fractal theory for texture analysis as well. Their approach, however, was based on a one-dimensional fractal. In contrast to their approach, Pentland and Peleg perform a full two-dimensional analysis.

In 1984, Peleg et al. [92] introduced the *blanket method* for the estimation of the fractal dimension. The method was inspired by a technique which was proposed by Mandelbrot for the determination of the length of the coast of Britain. The coast line problem focuses on the estimation of the length, but for the texture analysis problem we are interested in measuring the surface area. The image under study is considered as a surface. For the estimation of the fractal dimension, the authors define a blanket that varies in thickness and that covers the image surface. For reasons of symmetry, an upper and a lower blanket are defined. The latter one covers the surface below. The thicker the blanket (the lower the resolution ϵ), the more smoothly will it follow the image surface (see Fig. 4.17 for an example of the 1-dimensional case). By measuring the volume of the blankets and dividing this value by the thickness, an estimation of area A is obtained. As for the coast line problem, the area estimation is dependent on the resolution. By linear regression an estimation of the fractal dimension D can be found from the formula:

$$A(\epsilon) = F\epsilon^{2-D}. \tag{4.25}$$

Note that we have now in the exponent a constant of 2, instead of the constant 1 as mentioned in Equation 4.1. The value for this constant is determined by the topological dimension of the object. The upper blanket u_ϵ and the lower blanket b_ϵ are intially equal to the image surface I: $u_0(i,j) = b_0(i,j) = I(i,j)$ The upper blanket is defined as follows:

$$u_\epsilon(i,j) = \max\left\{u_{\epsilon-1}(i,j)+1, \max_{|(m,n)-(i,j)|\leq 1} u_{\epsilon-1}(m,n)\right\} \text{ for } \epsilon = 1,2,3,... \tag{4.26}$$

and the lower blanket is defined as:

$$b_\epsilon(i,j) = \min\left\{b_{\epsilon-1}(i,j)-1, \min_{|(m,n)-(i,j)|\leq 1} b_{\epsilon-1}(m,n)\right\} \text{ for } \epsilon = 1,2,3,... \tag{4.27}$$

This definition guarantees that the surface of each new upper, respectively lower blanket is placed above, respectively below the surface obtained in the preceding iteration step. The volume v of the blanket can

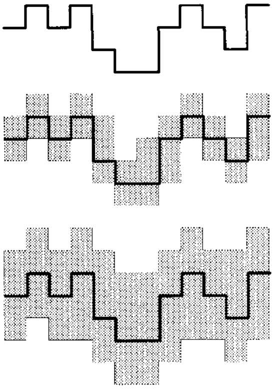

Figure 4.17: Two iteration steps of the *blanket method* suggested by Peleg et al. [92] as a technique to estimate the fractal dimension.

be obtained by:

$$v_\epsilon = \sum_{i,j} \left(u_\epsilon(i,j) - b_\epsilon(i,j)\right) \tag{4.28}$$

The area could be obtained from $\frac{v_\epsilon}{2\epsilon}$. However, the authors prefer to apply the following equation:

$$A(\epsilon) = \frac{v_\epsilon - v_{\epsilon-1}}{2}. \tag{4.29}$$

This definition possesses the advantage that it isolates those features that change from scale $\epsilon - 1$ to ϵ. For a pure fractal object, this definition makes no difference. In practice, where a texture might be fractal-like, this definition is to be prefered. Of course, it also possible to work with just the upper blanket or the lower blanket. It is this blanket technique that has still to be considered to be one of the most succesful fractal dimension estimators. In their article, the authors suggest that it might be interesting to extend the algorithm to the pyramidal data structure, because of its multiresolution appearance. Such an approach is discussed by Eijlers in [35], of which a more detailed discussion can be found in Chapter 5 of this thesis. In Figs. 4.18 - 4.21 the plots of $A(\epsilon)$ against ϵ are shown for the Figs 2.1 - 2.4. The plots are based on four windows of 16×16 for each image.

Dellepiane et al. describe in [30] a preliminary investigation on the methods suggested by Pentland and Peleg. The study shows that both

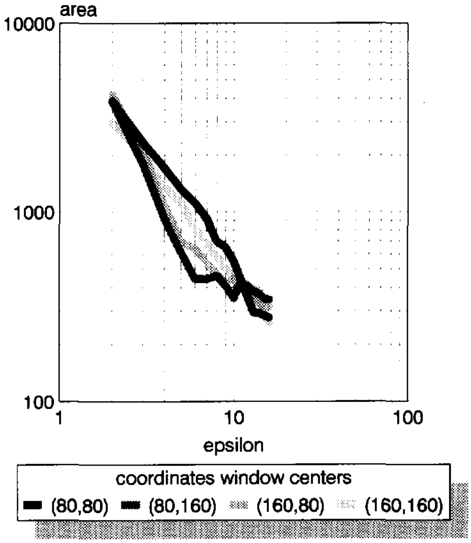

Figure 4.18: This plot shows $A(\epsilon)$ against ϵ for the image shown in Fig. 2.1.

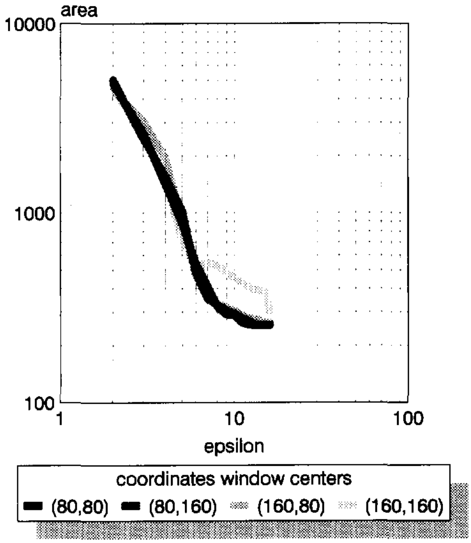

Figure 4.19: This plot shows $A(\epsilon)$ against ϵ for the image shown in Fig. 2.2.

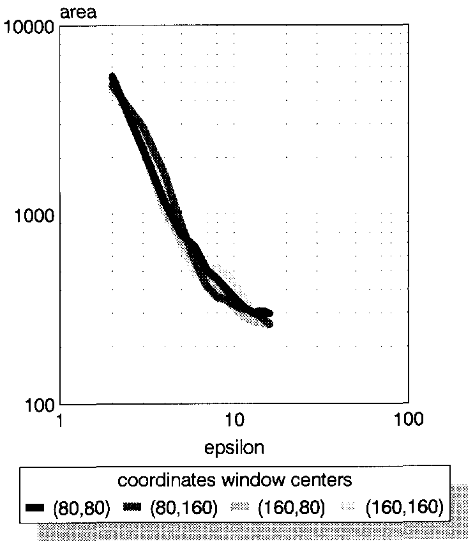

Figure 4.20: This plot shows $A(\epsilon)$ against ϵ for the image shown in Fig. 2.3.

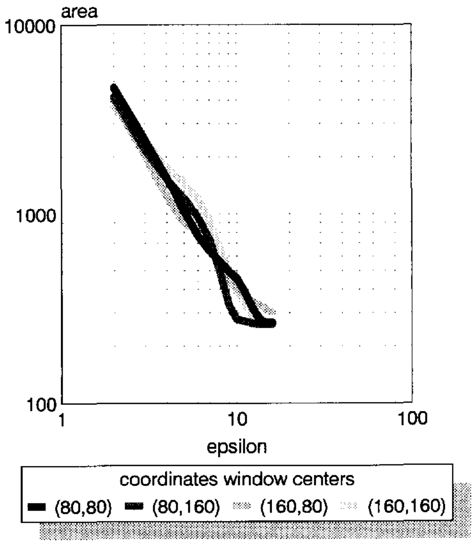

Figure 4.21: This plot shows $A(\epsilon)$ against ϵ for the image shown in Fig. 2.4.

techniques yielded interesting results for the medical application aimed at. None of the methods seems to be superior in this stage of that study. In [2], Albregtsen et al. describe an alternative elaboration of the blanket algorithm. The algorithm is also based on MAX-MIN operators. Stein introduces in [108] a blanket-like algorithm which is based on morphological operations.

In [74], Lundahl et al. describes a Maximum Likelihood Estimator which is based on the correlation matrix. This operator is derived from a discrete version of the fractal Brownian motion model. A comparison is made with an estimator which is based on the variance property of the fractal Brownian motion. It appeared that results of the Maximum Likelihood operator were better. However, because of the computational complexity of this method, the variance method could be preferable for certain types of application.

In [21], Caldwell et al. describe a method which is -just like the blanket method- based on the surface area principle of fractals. The image is considered as a collection of connected skycrapers. The area is the sum of the areas of all the roofs and the sides of the skycrapers. This gives:

$$\begin{aligned} A(\epsilon) &= \sum_{i,j} \epsilon^2 + \sum_{i,j} \epsilon \left\{ \mathrm{ABS}\left[I(i,j) - I(i+1,j)\right] + \right. \\ &\qquad \left. \mathrm{ABS}\left[I(i,j) - I(i,j+1)\right] \right\}. \end{aligned} \tag{4.30}$$

Varying the resolution, ϵ is achieved by block-wise averaging. This means that the average value for blocks of square sizes 4, 9, 16, 25 and 36 is calculated. It must be stressed that the number of resolution steps considered by the authors is not too large, this might explain the fact of a high score for the quality of fitting. However, the fact that the number of steps in this method is not very great does not invalidate the usefulness of the technique. The results obtained for the medical application aimed at could be considered interesting, however, further development is required.

In [33], Dubuc et al. describe a fast algorithm for the estimation of the fractal dimension. The method is called the *variation method.* Again a sort of volume is determined by the following formula:

$$V_\epsilon(i,j) = \max_{\mathrm{dist}((i,j),(s,t)) \leq \epsilon} I(s,t) - \min_{\mathrm{dist}((i,j),(s,t)) \leq \epsilon} I(s,t), \tag{4.31}$$

where

$$\mathrm{dist}((i,j),(s,t)) = \max(|x-s|,|y-t|). \tag{4.32}$$

The fractal dimension is again found by a linear fit in the log-log space. The method has been evaluated on a limited number of generated images. For these images it showed an accurate estimation. However, it must be stressed that the generated images were not distorted with noise or artefacts. From the variation method it might be expected that the technique is rather sensitive to shot noise.

In [67], Keller et al. describe results obtained with a fractal estimator which has been developed by Voss and which is based on the estimation of a probability function, a technique which we discussed in Section 4.4. The authors show that the operator gives the right order of values for synthetic images, but the range is strongly compressed. For instance, for the image with a fractal dimension of 2.9, the authors find a fractal dimension of 2.53. The explanation given by the authors is based on the quantization effect. Images with a higher fractal dimension are highly irregular. However, due to the limited number of grey values, a lowering of the estimated dimension is found. By extending the original algorithm with a linear estimation, the authors succeed in stretching the scale to its full extent, without influencing the ordering of the dimensions found. Although, the estimation is more accurate after the extension of the original algorithm, for automatic discrimination and classification purposes this adjustment to the original routine is superfluous. The authors also implemented a measure similar to the lacunarity measure, which was introduced in Section 4.4. According to the authors, the segmentation based on the fractal dimension in combination with the lacunarity measure performed well for the segmentation of natural textures. This performance could not be achieved by employment of the the fractal dimension only.

In [2], Albregtsen et al. test Voss' box-counting method, Keller's fractal dimension estimator and the blanket method of Peleg et al. on synthetic images and TEM (= Transmission Electron Microscopy) images of mouse liver cell nuclei. It was shown that the blanket method resulted in the best approximation of the fractal dimension and in the best discrimination results. The second-best technique is Keller's method. The authors show also that it is possible to derive a fractal dimension

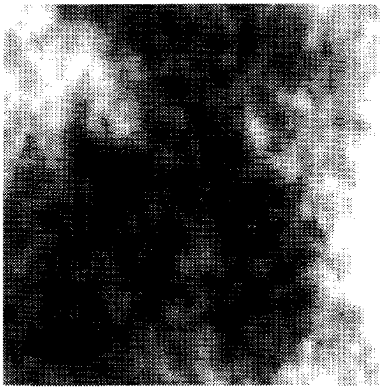

Figure 4.22: Part of a coloured biological cell. The part shown has sizes 100×100.

related texture feature from the difference in grey level runs at two different quantization widths.

In Fig. 4.22, we show an image which comes from a similar type of biological application and which is the subject of Starink's Ph.D. thesis [107]. The image shows a part of a cell which has been colored. To test the suitability of the fractal model for this type of application, we have determined the fractal dimension for $\epsilon = 1, 2..., 18$ using the blanket method. The window sizes were 32×32. We have plotted the area of the window against ϵ in Fig. 4.23 for five locations in the image. It was shown that the fractal model might be considered as being suitable if ϵ is in the range of 2 to 9, a range that by some authors is considered as being too small for accepting the fractal model. They prefer to have a range of at least 10 succeeding ϵ values before concluding that the fractal model is appropriate for the underlying study. For some applications, however, the range found can be sufficient. However, it must be stressed that the image shows a remarkable visual similarity to some of the generated images shown in Figs. 4.7 - 4.15.

In the literature, some authors experience the fractal theory as being too restricted. The application of a model based on the fractal theory is only justified if the prescribed scaling behavior is valid for a sufficiently large interval. In practical applications this constraint can be too limited. It is therefore that Kaneko suggests in [62] to study what he calls the *local fractal dimension*, an approach that has also been

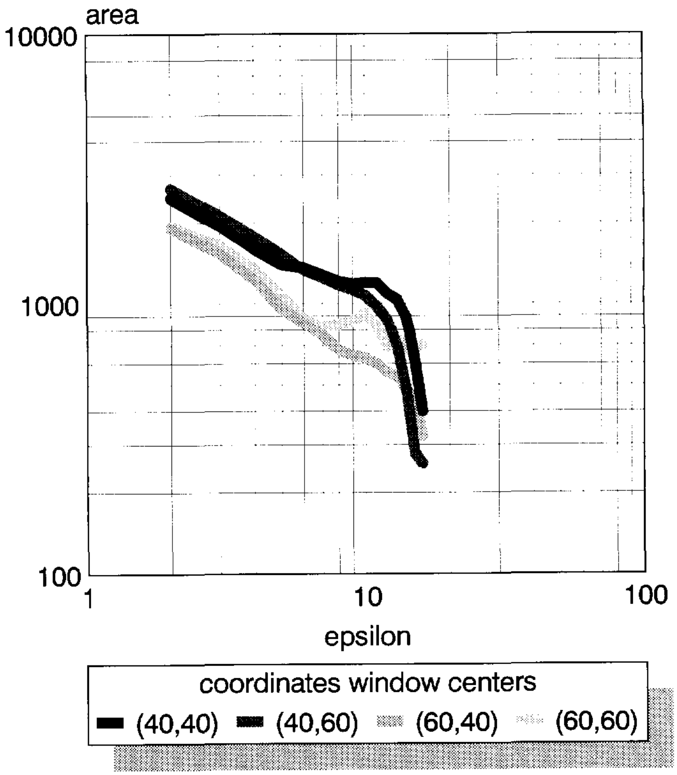

Figure 4.23: The relation of $A(\epsilon)$ against ϵ for five locations in the image shown in Fig. 4.22.

suggested by Dubuc et al. in [33]. This local fractal dimension does not consider all the resolution values for fitting a straight line through it; it only fits the line through 2 points. To proceed, Kaneko extends the original scalar fractal relation to a two-dimensional equation. This approach has the advantage that it distinguishes scaling behavior in different directions. Therefore, this approach might be considered as being more general. However, an objective study where this algorithm is compared with the most common algorithms should be carried out in the future.

In [102], Schepers et al. mention four methods for the estimation of the fractal dimension for one-dimensional signals. One of the applications mentioned in the article is the velocity of blood cells passing through a small artery. An interesting technique is based on the *relative dispersion analysis.* This analysis compares the variance of a variable as the measurement resolution increases. It is defined as the division of the standard deviation by the mean. The analysis starts at the highest resolution level. The variance for the whole signal is determined at this level . Then the signal is averaged pairwise (compare the pyramid intialization with uniform averaging on blocks of 2×2). Again the variance is determined. This scheme is repeated a number of times. Now, suppose that the signal is uncorrelated, then one expects that after the first iteration the standard deviation is reduced by a factor of $\frac{1}{2^{\frac{1}{2}}}$. In the case of having averaged n consecutive values, the standard deviation will be reduced by a factor of $\frac{1}{n^{\frac{1}{2}}}$. It should be stressed that the mean value stays equal. For the fractal Brownian case the standard deviation is proportional to n^{H-1}. It is this property that can be used for the derivation of the fractal dimension.

4.6 The Generalized Dimension Model

In Section 4.4, we discussed the box-dimension which has been defined as:

$$D_B = -\lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log \epsilon}. \tag{4.33}$$

It is a topological measure which can be used as a measure for the roughness of a set or a signal as we have seen. The box-dimension only counts the number of boxes which are filled with at least one point. We do not weight the count such that its value is determined by the number of points in the box. With the introduction of the generalized dimension concept, a weighting mechanism is obtained with which an increase in performance in discriminatory sense can be obtained.

The generalized dimension model was introduced by Hentschel and Procaccia [51].[9] Now, suppose that N_i is the number of points in the i-th box and N is the total number of points. Note that:

$$N = \sum_i N_i. \tag{4.34}$$

Now, the generalized dimension with parameter α is defined as:

$$\begin{aligned} D_\alpha &= \lim_{\epsilon \to 0} \frac{1}{\alpha - 1} \frac{\log \sum_i \left(\frac{N_i}{N}\right)^\alpha}{\log \epsilon} & (4.35) \\ &\overset{p_i = \frac{N_i}{N}}{=} \lim_{\epsilon \to 0} \frac{1}{\alpha - 1} \frac{\log \sum_i p_i^\alpha}{\log \epsilon}. & (4.36) \end{aligned}$$

Hentschel showed that for each $\alpha' > \alpha$, the following relation is valid:

$$D_\alpha \geq D_{\alpha'}. \tag{4.37}$$

With the introduced parameter α, we are able to weight boxes which are more dense, or which are less dense. This means that for small values of α we stress the less dense regions, whereas we stress the more dense regions with higher values for α. Note that for $\alpha = 0$ we obtain again the ordinary box-dimension. Two other values for α need more attention, namely $\alpha = 1$ which results in the *information dimension* and $\alpha = 2$ which results in the *correlation dimension.*

The information dimension is related to Renyi's information measures. Renyi's information measure, H_α of order α for $\alpha > 0$ and $\alpha \neq 1$ is defined as (see [73], where van der Lubbe describes Renyi's information measures in conjunction with other information measures):

$$H_\alpha(P) = \frac{1}{1 - \alpha} \; {}^2\log \left[\sum_{i=1}^{n} p_i^\alpha \right]. \tag{4.38}$$

---

[9] More information can be found in [45], [80], [99], [47], [38] and [36].

Using the proposition of 'l Hospital, one obtains for $\alpha \to 1$ Shannon's information measure, which is given by:

$$H_S(P) = -\sum_{i=1}^{n} p_i \; {}^2\log(p_i). \tag{4.39}$$

The correlation dimension is often rewritten, such that it can more easily be determined. The correlation dimension D_2 is given by:

$$D_2 \;=\; \lim_{\epsilon \to 0} \frac{\log \sum_i p_i^2}{\log \epsilon}. \tag{4.40}$$

In practice, determining the dimension with the box-counting approach can be rather cumbersome. However, the term $\sum_i p_i^2$ can be approximated by using the correlation integral $C(\epsilon)$, which has been defined as:

$$C(\epsilon) = \lim_{N \to \infty} \frac{1}{N^2} \sum_{ij} \theta \left[\epsilon - |\vec{x_i} - \vec{x_j}|\right], \tag{4.41}$$

where θ denotes the Heaviside function. The proof is given by Schuster in [103].

That the generalized dimension model might be considered as being a rather recent development is founded on the limited number of applications found in the image processing literature. The applications found are discussed in the next section.

4.7 Texture Measures based on the Generalized Dimension Model

Vepsäläinen and Ma describe in [118] the use of the fractal dimension and the correlation dimension for a number of applications. The way the fractal dimension is determined is similar to that described by Nguyen and Quinqueton in [88], i.e. that the window for which the fractal dimension has to be calculated is transformed into a 1-dimensional signal. Nguyen and Quinqueton use the Peano curve as a scanning scheme. This curve is a deterministical fractal, which is also being used for image coding purposes as discussed by Yang et al. in [124]. By scanning the image (or window) according to the Peano curve, one tries to

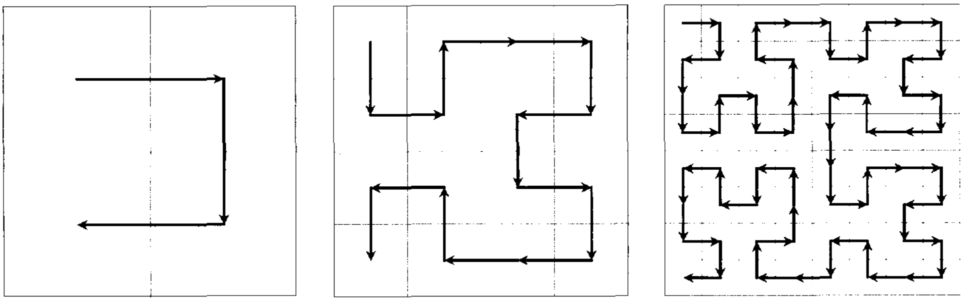

Figure 4.24: The Peano-curve.

transform the 2-dimensional signal such that the correlation behavior of the 1-dimensional resulting signal reflects the correlation behavior of its 2-dimensional origin. Therefore, the Peano curve scanning scheme is interesting for a number of image processing tasks. Three steps of the construction of a Peano curve are shown in Fig. 4.24. The scanning scheme chosen by Vepsäläinen and Ma appears to be more arbitrarily shaped. The determination of the fractal dimension is based on the length of the 1-dimensional curve. Because this determination can only be done with a limited number of yardstick lengths, the authors suggest a technique which is based on prediction. The calculation of the correlation dimension is based on the correlation integral method discussed in Section 4.6. Two of the applications mentioned by the authors are the searching for cracks in ice from satellite images and the analysis of cell images which are obtained with a confocal light microscope. The first application is of importance for directing ice breakers. With the dimensions one is able to classify the type of ice and to locate cracks in the ice. The cell images were studied on the appearance of protein fibers and small elliptical objects. The detection of the elliptical objects delivered no problems. The extraction of the fibers, however, could not be based on a simple thresholding operation on the dimension image. In their article, the authors stress that the dimension operators were used for the detection of edges (resulting in an increase of the dimension) as well as for the classification of textures.

In [4], Arduini et al. use the distribution function approach as introduced by Voss in [120] which we discussed in Section 4.4. One of their experiments showed that for the human observer, dissimilar textures can have the same fractal dimension, but different other dimensions. The authors applied a window size of 128 × 128. It should be stressed that a large window is required for the estimation of the probability function. Therefore, one could cast doubt on the applicability of this approach for texture segmentation.

In [1], Ait-Kheddache and Rajala introduce a technique which is based on *Pseudo Fractal Matrices.* Several of these matrices have to be filled for each window. Each matrix corresponds to one value of ϵ. The matrices are filled with counts, which can be compared with the number of points in the box-dimension concept. Because of the difference in meaning of the axis, the authors make a distinction between the distance measured along the grey value axis and along the topological axis. From these matrices some features are determined which are estimations of the dimensions. To prevent having too sparsely filled Pseudo Fractal Matrices, the applied window sizes should be at least 32 × 32. Experiments carried out by Kamphuis [60] showed that the algorithm was rather time-consuming and sensitive to the adjustment of the parameter which determines the resolution along the grey value axis. Although the algorithm does not calculate the dimensions fully in correspondence with the theory, and despite the fact that the applicability of the algorithm in its actual form seems to be limited, the solutions to some practical problems are certainly interesting.

4.8 Discussion

In this chapter we have dealt with the most important aspects of the fractal theory from an image processing point of view. Further, an overview has been given of the work carried out in this field. It appeared that some authors seem to exaggerate the appropriateness of the fractal model for their application. However, for certain types of applications the fractal model must certainly be considered as being suitable. In any case we may conclude that for applications for which the images under study are similar to the synthetic images shown in the figures 4.7 - 4.15, the fractal model might be of interest. For textures where the fractal model appears to be less suited, splitting up the scaling behavior in

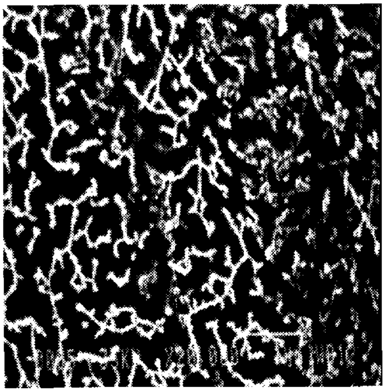

Figure 4.25: Structures obtained with a Scanning Electron Microscope after etching amorphous silicon (from [90]).

ranges for which the scaling behavior in the log-log space is linear could sometimes be suggested. In this way a number of fractal dimensions can be found.

A class of textures which have not been discussed yet, is that of textures which might be considered as (nearly) binary. A typical example of such a texture is shown in fig 4.25. This SEM (Scanning Electron Microscope) image made by Oort [90] shows structures that appeared after the etching of amorphous silicon. The whimsicality of these structures can also be characterized by the fractal dimension. The determination of the fractal dimension is in this case based on the correlation behavior. A detailed study by Koorevaar for the image shown can be found in [71]. Koorevaar evaluated fractal dimension estimators on synthetic images of which an example is shown in Fig. 4.26. Further information on the synthesis and analysis of these kinds of structures is given by Kaye in [66].

A use of fractal theory in the field of image processing not mentioned yet is that of fractal image coding, where two names should be mentioned, viz. Barnsley [8] and Jacquin [56]. The applicability of the methodology followed in this field seems still to be restricted to the field of image coding. A first attempt to break through in the field of texture coding and analysis is described by Vehel in [117]. The discussion of this type of approach, however, is outside the scope of this thesis.

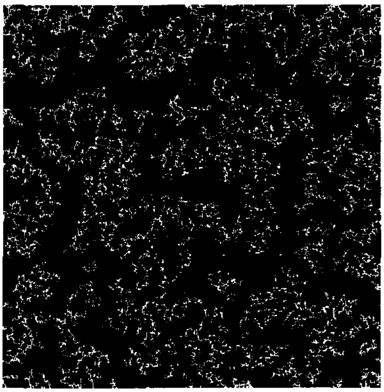

Figure 4.26: A synthetic variant to the image shown in Fig. 4.25.

In this chapter, it has been mentioned that dissimilar textures to a human observer might yield equal values for the fractal dimensions. In these cases at least, the measurements should be complemented with measures based on the generalization of the dimension model or measures such as the lacunarity. Regretfully, only a few articles have been found in the literature where the authors use these concepts. In the following chapter we attempt to combine the ideas about texture in combination with the pyramid as a multi-resolution structure and the fractal theory.

Chapter 5

Scaling and Texture: Pyramids and Fractals

5.1 Introduction

In Chapter 2 we discussed texture and the scaling aspects of textures. From this discussion it followed that there are two major arguments to give, that the analysis of textures should be based on several levels of resolution. Firstly, texture pairs might not be discriminable on a certain level of scaling, whereas they might be discriminable on another level of scaling. This behavior will be shown in Section 5.5. Secondly, some textures have an optimal level of scaling on which they should be studied, as for instance the texture shown in Fig. 2.7. This level does not need to be known in advance, therefore a study on several levels is recommended.

These arguments motivated the study of texture analysis within pyramidal data structures, which -as we have seen in Chapter 3- is a data structure which is especially designed for the study of signals on several levels of resolution. The initialization of these structures for texture analysis appeared to be not trivial. Considering the number of articles that have appeared and dealing with the use of pyramids in general, the number of published articles that deal with the use of pyramids for the study of textures might be considered limited.

Burt [16] proposes two initialization schemes for the texture pyramid. The first initialization scheme, which makes elegant use of the data structure, is called the local spectral estimator. After having initialized a Laplacian pyramid, the values are squared. Then for each window, for which the texture value has to be determined, the average value is calculated. This average value is assigned to the center pixel of the window.

The second initialization scheme as proposed by Burt starts with the initialization of a Gaussian pyramid. By applying an edge operator to the layers of the pyramid, a binary pyramid is obtained. The true-valued pixels in the pyramid correspond to pixels which lie on an edge in the original image. Now, for each window, the fraction of true-valued pixels is determined. The resulting values are called edge-density values.

Both schemes calculate the texture value for a window with given sizes. These sizes do not change for the layers higher in the pyramid (see also Section 3.4). Therefore, the effect of border artefacts -which means the effect where pixels of the window lie outside the borders of the image- increases for levels higher in the pyramid. This limits the practical use of such approaches. The reason for this disadvantage is that the texture is quantified on a window basis for each level.

Another approach is described by Eijlers [34]. For this approach the bottom level of the pyramid is filled with the texture data, which has been obtained by applying a texture operator to the original data. The other levels of the pyramid are initialized with the Gaussian reduction operator. Thus, the initialization of this type of texture pyramid is identical to that of a Gaussian grey level pyramid. Examples of segmentation results which have been obtained are shown in Figs. 3.15 and 3.16. In this case, the pyramid is only used for the reduction of the variance in the texture data. A similar kind of use of a pyramidal data structure is described by Unser and Eden [116]. The disadvantage of such an approach is that the higher levels in the pyramid do not correspond to coarser aspects of the texture. Unser and Eden solve this problem by initializing a number of pyramids, where each pyramid is filled with data that highlights a certain aspect of the texture. However, it is preferable that the lower resolution levels in the texture pyramid correspond to coarser aspects of the texture, whereas the higher resolution levels correspond to the finer details of the texture.

In case that the quantification is based on a model, the texture features are derived from the model's parameters. For each new level, the parameters result from the parameters of the level underneath. A similar type of approach is used by Bierhuizen [10] for the estimation of motion parameters, which are used for the segmentation of images. We expect that scaling behavior is an intrinsic aspect of the model to be applied. Such a model, as we have seen in Chapter 4, can be a model which is based on the fractal theory.

Fractal theory in conjunction with the study of textures is often associated with the fractal dimension. As we have seen in Chapter 4, the fractal dimension is used as a parameter for the whimsicality of the signal. For the analysis of signals other dimension definitions as, for instance, the information and correlation dimension are starting to be considered as well. These dimension definitions follow from the generalized dimension definition. In contrast to the topologically oriented fractal dimension, the other dimensions are more stochastic measures. It appears from the literature that signals might have the same fractal dimension whereas the other dimensions might deviate. Other, measures -as for instance the lacunarity (see Section 4.4)- are derived from fractal theory as well.

One of the first attempts to determine the fractal dimension of textures is described by Peleg et al. [92] and is called the blanket method (see Section 4.5). Despite the sensitivity to noise, this technique is -because of its simplicity- in wide use for the determination of the fractal dimension. In their article, the authors suggested the possiblility that the technique might be implementable within a pyramid environment.

Such an approach has been followed by Eijlers [35] of which the discussion is postponed to Section 5.3. Besides the approach of Eijlers based on pyramids, a number of articles have been found that are based on other types of multi-resolution techniques. Two types of such approaches can be distinguished. The first type of approach is based on scale space. An example of this approach is described by Müssigmann [87]. He describes a method of determining the fractal dimension which is based on the change in the behavior of the signal in scale space (see Section 3.3). The determination is based on the same ideas as that of the determination of the length of the coastline. This is done for a number of scaling values (σ). The author mentions that the required behavior in the log-log space is found for a number of textures. The second type of approach as followed by Cohen and You [23] is based on wavelet theory. Here the signal is studied with a number of filters where different aspects of the spectrum are considered.

From the preceding discussion, it is concluded that an initialization technique for the texture pyramid has to be developed. The intialization has to be based on a model, for which scaling is preferably an intrinsic aspect. Such a model could possibly be derived from fractal theory. In the next section we discuss a number of aspects with respect to such an approach.

5.2 Thoughts on the Integration

In the following sections we describe three attempts at integration of fractal concepts for the study of textures within pyramidal data structures. Before these attempts are discussed, some remarks should be made with regard to the integration. For the texture operator, we could give a list of requirements and desirable properties. Of course, the operator should discriminate properly between the textures corresponding to the application domain. If, as in our case, the application domain is not given, its general ability to discriminate pairs of textures should be satisfactory. The second requirement for the texture operator is that it is insensitive to shot noise. Further, it is desirable that the output of the operator is interpretable. This means that, for instance, high values correspond to a texture patch which is finer, whereas lower values correspond to patches with coarser textures. For certain applications, it can be desirable that the texture values can be estimated by man as well. Further, it might be of interest that the operator is sensitive in a certain direction. For a number of operators this behavior is not intrinsic to its definition, but by adapting, for instance, the window shape of the operator, its sensitivity to a certain orientation can be facilitated. Finally, it might be desirable that the computation time is restricted.

For the texture operator we aim at, we can add some further requirements. First of all, the operator should be based on a model for which scaling is an intrinsic aspect. Therefore, the operator might be based on concepts from fractal theory. It must be stressed that we do not expect an operator which fully conforms to this theory, our only objective is the design of a good texture operator that satifies the requirements and demands as stated above. Further, the functioning of the operator should be such that it can be integrated in the pyramidal data structure. As we have seen in Chapter 3, a pyramid can be used in two different ways. The first type of use is as a data structure to reduce variance in the data. The second type of use is that levels higher in the pyramid correspond to coarser aspects of the image. It is the second type of use one should aim at when applying a pyramidal data structure.

An elaboration of this concept is that in which only the changes between two layers are considered. Thus the required linear relation in the log-log space is left. Although the approach is not in agreement with the fractal theory, it does not exclude the possibility of practical use. As has been mentioned by a number of authors, the required linear relation was not found for many textures, which restricted the use of measures based on the fractal theory. If we divide the relations into log-log space, a more general type of texture operator might be obtained. An example of such an approach is also proposed by Albregtsen et al. [2]. In Section 5.5 an attempt at integration is proposed where a similar type of use of the log-log space is followed.

Finally, we should stress that if the concepts of the fractal theory are used, it is desirable that the texture operator to be designed includes the weighting abilities as prescribed by the generalized dimension model (see Section 4.6). Further, it might be of interest to study the use of such related measures as, for instance, the lacunarity.

In the next section we describe a first attempt at integration. This attempt does not fulfill all the requirements we have stated above. The method is based on the blanket algorithm, but extended to the pyramid environment.

5.3 The Blanket Technique expanded

The algorithm is based on the blanket technique [92], described in Section 4.5. This method calculates the area $A(\epsilon)$ of the image with de-

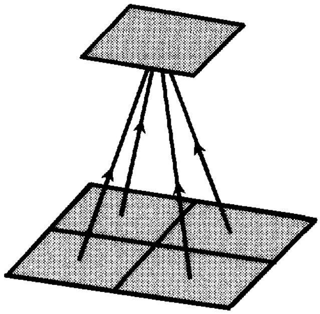

Figure 5.1: The transport of the sums required for linear regression.

creasing resolution ϵ. The values obtained can be plotted in a log-log graph as shown in Figs. 4.18 - 4.21, and 4.23. By fitting a line through the points in the log-log space, the fractal dimension can be determined from the slope of the line through these points. The slope of the line for a fractal image is equal to $2 - D$, where D stands for the fractal dimension. In practice, this curve will only be drawn for testing the appropriateness of the model for the texture under study. For the calculation of the fractal dimension, the linear regression algorithm is used, which is based on the calculation of four sums, viz.: $\sum \log \epsilon \log A(\epsilon)$, $\sum \log \epsilon$, $\sum \log A(\epsilon)$, and $\sum (\log \epsilon)^2$.

The idea behind this initialization technique is to transport these sums to the father nodes [1] as shown in Fig. 5.1. Thus a sort of "super sums" are obtained, such that the fractal dimension can be obtained for each level in the pyramid. This approach has been described in [35]. In this article, determining the variance of the fitting has also been suggested. A higher variance should possibly correspond to an edge in the underlying region.

The accuracy of this measurement has been tested on images which were generated according to the frequency filtering method as suggested by Voss, discussed in Section 4.3. The dimensions of the images lay in the range of 2.1 to 2.9 in steps of 0.1. For each dimension five realizations were generated. The resolution parameter ϵ was varied between 1 and 8 and the window sizes were set to 12×12. As discussed in Chapter 2, larger window sizes would result in ill-defined edges. Larger

---

[1] Because of the contents of the elements, the term *node* is used rather than *pixel.*

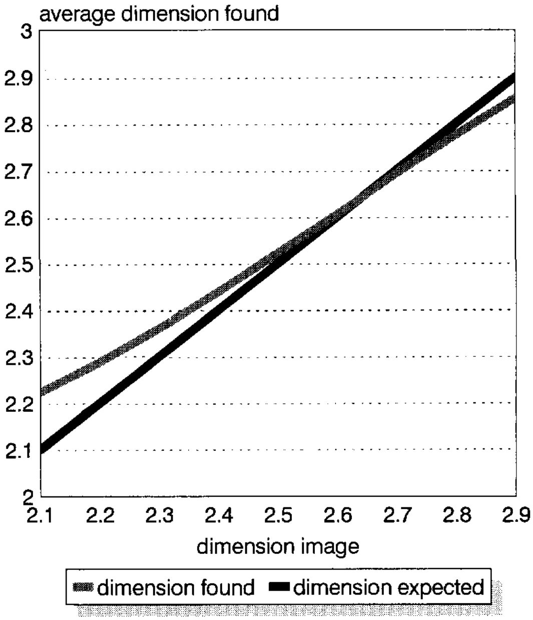

Figure 5.2: The average dimensions found for synthesized images with increasing dimension.

window sizes also give problems for smaller textural regions. Smaller window sizes result in a decrease of the statistical accuracy. For each image, the average dimension has been determined, together with the standard deviation. These values were averaged again per group of five realizations. In Fig. 5.2, the average dimensions are shown. Because of the technique used, the average values for levels higher in the pyramid do not differ from the values at the bottom level. Only the standard deviation decreases with the resolution as is shown in Fig. 5.3. These standard deviations appeared not to vary significantly within the groups of five realizations.

To illustrate the practical use of the method, we have made two images that are a composition of two Brodatz textures (see Figs. 5.4, and 5.5). From theses images the corresponding texture pyramids were initialized. The window sizes were fixed at 12×12 and the resolution was varied between 1 and 8. The resulting pyramids are shown in Figs. 5.6 and 5.7. Purely for illustration purposes, the dimension values are rescaled such that they lie in the range of $[0, 255]$. It appears

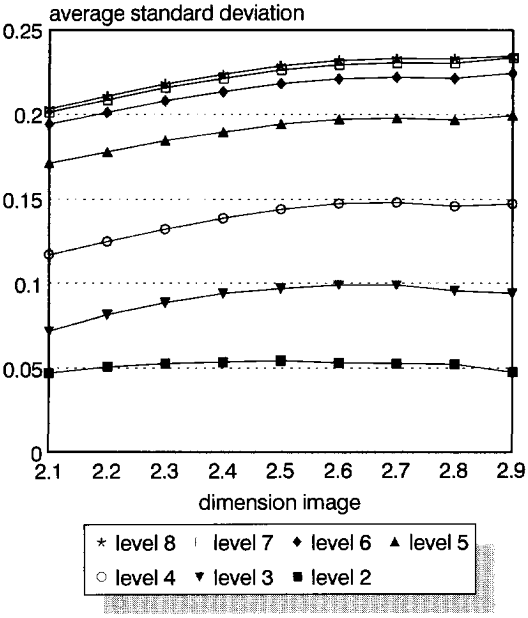

Figure 5.3: The average of the standard deviations for each level in the pyramid.

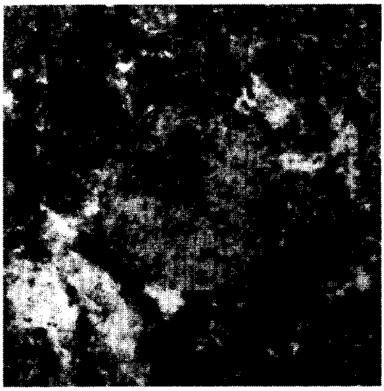

Figure 5.4: A composition of the Brodatz textures [14] d60 (European marble) and d100 (ice crystals on an automobile).

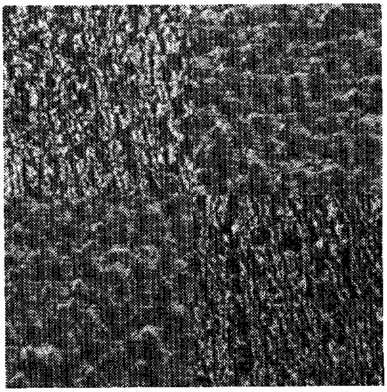

Figure 5.5: A composition of the Brodatz textures [14] d24 (pressed calf leather) and d92 (pigskin).

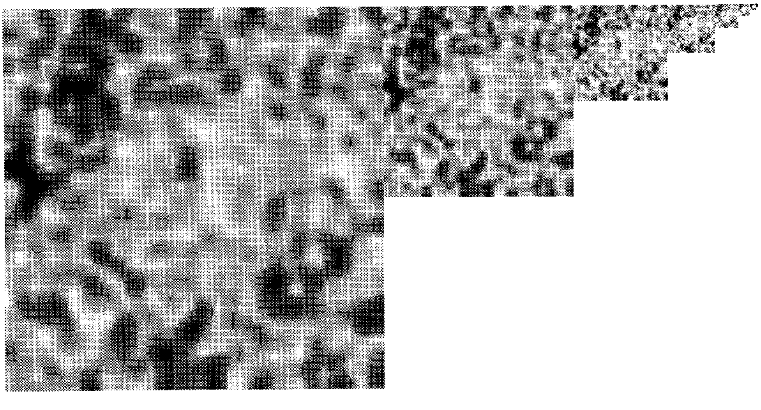

Figure 5.6: The texture pyramid of the image shown in Fig. 5.4.

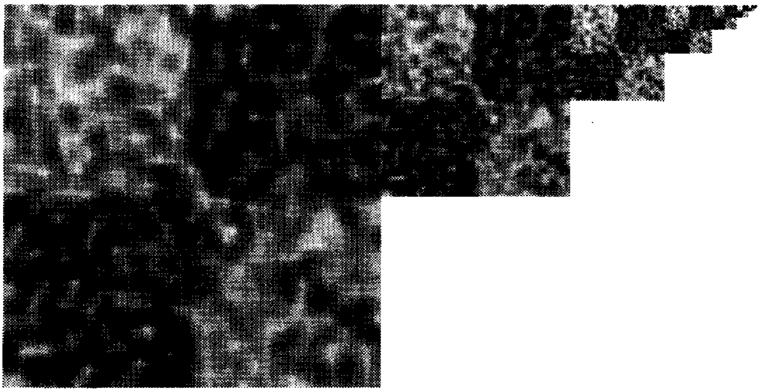

Figure 5.7: The texture pyramid of the image shown in Fig. 5.5.

that the operator is not able to discriminate between the textures d60 and d100. Even at higher levels in the pyramid, no discrimination could be obtained. Discrimination is possible for the texture pair d24 and d92. At the bottom level of the pyramid, the distributions of the two types of textures are still overlapping, as shown in Fig. 5.8. But from level 5, the distributions start to separate as is shown in Figs. 5.9, 5.10 and 5.11. In Figs. 5.12, 5.13 and 5.14, segmentation results are shown that were obtained by thresholding the levels in the pyramid at a threshold of 3.47. This choice of threshold value is based on the histograms shown in Figs. 5.9, 5.10 and 5.11.

As we mentioned at the beginning of this section, this attempt does not fulfill all the requirements that are given in Section 5.2. The main disadvantage of this technique is that it determines the fractal dimension only on one level of scaling. This means that the pyramidal data structure is only used as a technique to decrease the variance. Further, the technique is based on the blanket technique, which is sensitive to shot noise because of the way the MIN- and MAX-operators are used. In Chapter 4, we mentioned the tendency of considering not only the fractal dimension, but studying a whole range of dimensions which followed from the generalized dimension definition. In the image processing community this tendency has not begun yet; only a very few articles deal with the information and correlation dimension, besides the fractal dimension. Because of the expected increase in discrimi-

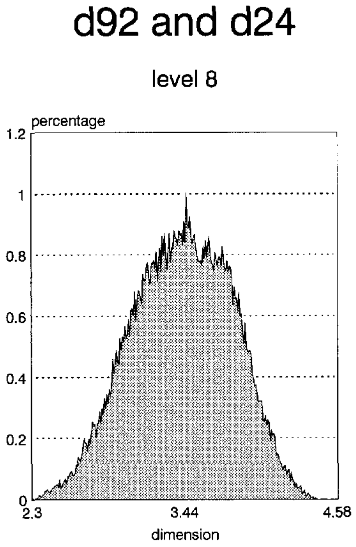

Figure 5.8: The histogram of the texture map shown in Fig. 5.7 at level 8 of the pyramid.

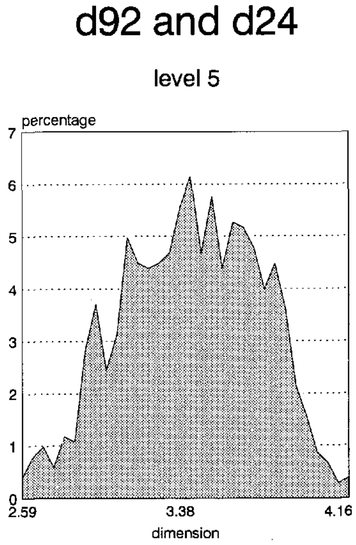

Figure 5.9: The histogram of the texture map shown in Fig. 5.7 at level 5 of the pyramid.

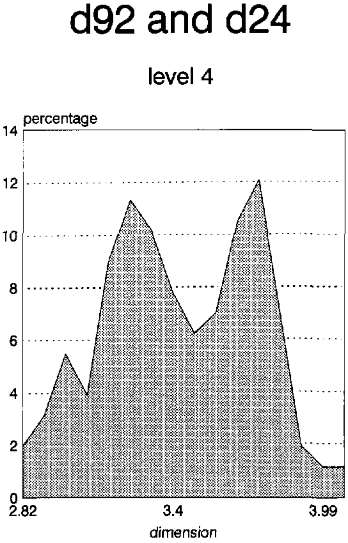

Figure 5.10: The histogram of the texture map shown in Fig. 5.7 at level 4 of the pyramid.

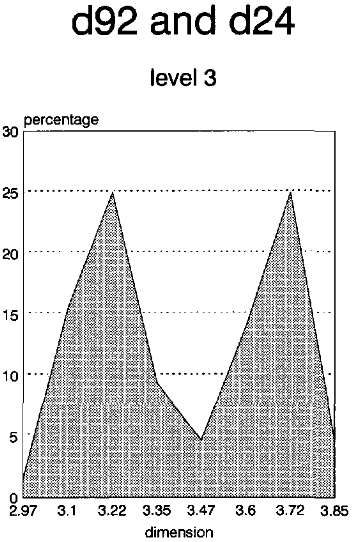

Figure 5.11: The histogram of the texture map shown in Fig. 5.7 at level 3 of the pyramid.

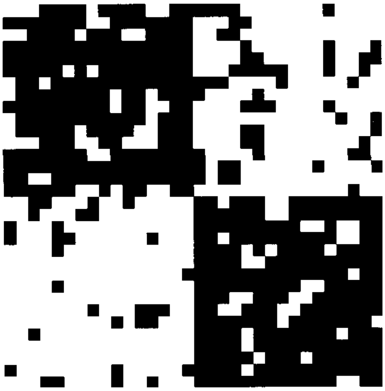

Figure 5.12: Thresholding result of level 5 in the pyramid shown in Fig. 5.7

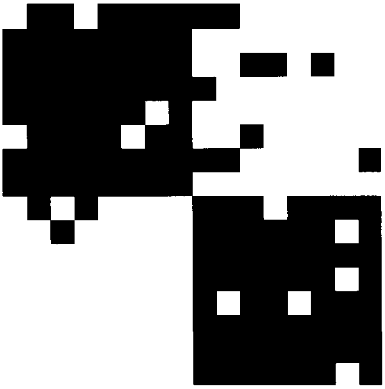

Figure 5.13: Thresholding result of level 4 in the pyramid shown in Fig. 5.7

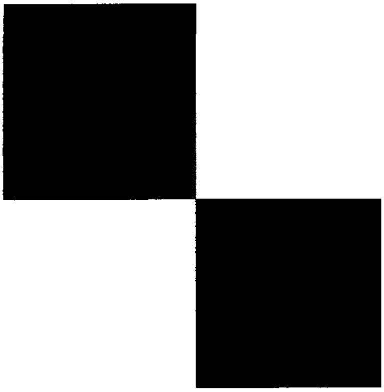

Figure 5.14: Thresholding result of level 3 in the pyramid shown in Fig. 5.7

natory power, an operator that can determine all these dimensions is recommended. Finally, it must be stressed that the initialization technique as discussed in this section, only works for textures that fulfill the requirement that there is a linear relation in the log-log space. As we mentioned in Section 5.2, the behavior in the log-log space can be divided up into a sequence of line segments. Although such an approach departs from the fractal theory, and may only be called a technique that uses fractal concepts, it might be of interest for application in a more generally oriented texture operator.

5.4 The Box-Counting Technique

The box-dimension approach is often recommended for the determination of the dimensions of a phenomenon in practical situations. This is because the implementation is straightforward, and the method is able to determine all dimensions of the generalized dimension model. A more detailed discussion of this dimension can be found in Section 4.4.

It appears that the box-counting algorithm has a lot in common with the pyramidal data structure (see Fig. 5.15). Consider a node higher in the pyramid. To such a node there is a corresponding square of 2 × 2 son nodes, which on its turn corresponds to a square of 4 × 4 at the next level. In general, we could say that for a pyramid that

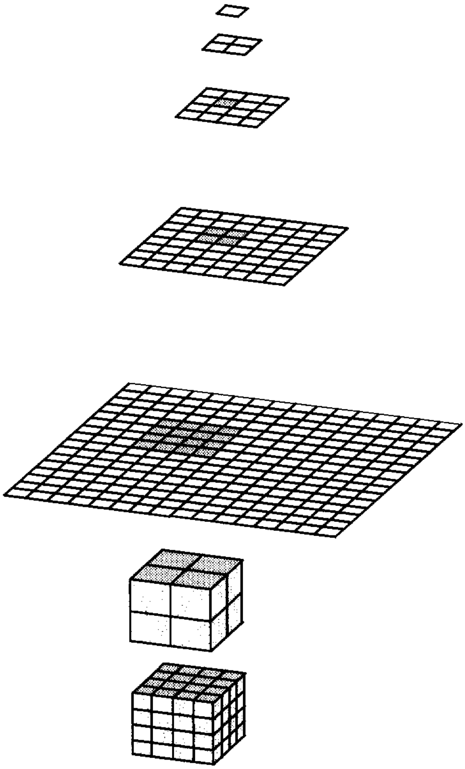

Figure 5.15: The similarity between the pyramidal data structure and the box-dimension approach.

consists of $N+1$ levels, a node at level l corresponds to an area patch of $2^{N-l} \times 2^{N-l}$, which is studied on $N-l$ levels of resolutions. Therefore this study of scaling behavior appears to be similar to that of the box-dimension approach. For the box-dimension approach, however, we need to consider the intensity axis as well. Therefore, we do not consider area patches, but cubes (see Fig. 4.16). Now, the possible number of divisions of theses cubes depends on the level of the node. This means that the dimension can be determined over a larger scaling range for higher nodes. When more resolution levels are considered, the influence of the individual resolution levels decreases. The pyramid which is obtained by following such an approach is comparable to the pyramid as discussed in the previous section, where the pyramidal data structure is used only for smooting the data. As we have stated before, we expect from a pyramidal data structure that higher levels correspond to a lower resolution. This can be achieved by only considering the two coarsest divisions of the corresponding cubes. Of course, the method is no longer in agreement with the fractal theory, which assumes a study which is based on an adequate range of resolution levels. We discuss this approach in this section.

From a computational point of view, it might be preferable to work with larger cubes, because there are more points within the cube, which increases the accuracy of the measurement. This might be obtained by working with cubes that overlap for 50% with cubes of neighboring nodes.

Using this algorithm, a number of experiments were carried out with the synthesized fractal images. The advantage of using these images is that the extent to which the dimensions have to differ before discrimination can be obtained can be easily tested. In the experiments, several values for the dimension parameter α have been used. From the resulting texture maps, a number of statistics have been determined. We found that discrimination was only possible between the texture pair with fractal dimension 2.1 and 2.9. Experiments have also been carried out with variants of this approach, where a selection of the available resolution steps could be made. The results obtained were similar.

Despite the general recommendation of the box-counting technique, we found that the method is not suited for the type of application we aim at. To obtain accurate results with the box-counting method, it is required that enough points be found in the resolution cells of the cube. Bear in mind that the number of points in the cube is fully determined by the sizes of the window, and not by the length of the intensity axis of the cube; therefore large window sizes are required. But, as we have mentioned in Chapter 2, window sizes which are too large are often not desirable from a texture analysis point of view and conflict with our search for a resolution reduction operator that is suited for the initialization of a texture pyramid. A good choice of the sizes is further hampered by the differences in the types of axes. For discrimination purposes, it is required that the resolution with which the intensity axis is studied is not too coarse, as is now the case. A finer resolution along the intensity axes would require a number of divisions, which is not always possible because of the limited dimensions of the window. Therefore, it appears that an optimal window size for a locally oriented texture study based on such an approach has not been found.

Despite the recommendations on fractal theory to estimate the dimensions of phenomena with a box-dimension type of approach found in the literature, we have found that this method is not suitable in its original definition for image processing applications. A conclusion that has also been drawn by Dubuc et al. [33]. To make the method suitable for image processing, it has to be altered substantially, which means that it will lose the elegance of its original definition. Such an approach has been followed by Ait-Kheddache and Rajala [1] and was discussed in Section 4.7. Experiments carried out by Kamphuis [60] showed that this method was time consuming and was rather sensitive to the choice of the parameters.

In the next section a more succesful attempt is discussed where the relationship with fractal theory is even further weakened.

5.5 The Mass Pyramid

In the previous section, we concluded that the cube was too sparsely filled with points to give accurate results. Further, the dimensions of cubes were such that the resolution along the intensity axis was too coarse to discriminate between significantly differing textures. Because of these disadvantages, it is interesting to think of a method that works

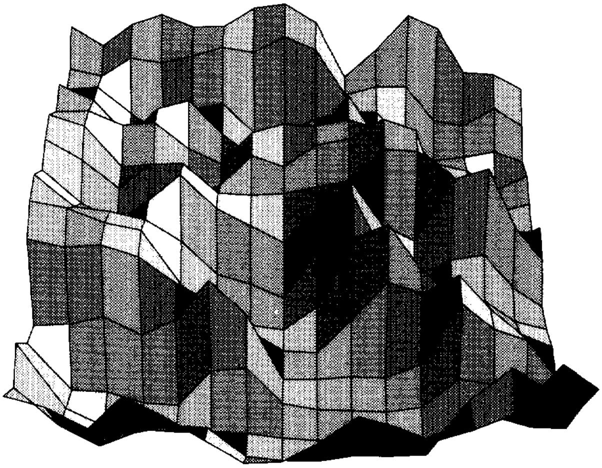

Figure 5.16: Image data within a window of 16 × 16 of the upper left corner of the image shown in Fig. 2.4.

in the image plane and which is not based on a 3-dimensional cube as is the previous method. Of course, it makes no sense to count the number of pixels in the window, for a variety of resolution values. Therefore, another type of measure has to be defined: a measure that is related to the texture aspects. Before such a measure is derived we first give three possible interpretations of the 2-dimensional image data.

The first interpretation is that the image data represents photon-densities. A high grey value corresponds to a high photon density. For a lot of images this mechanism underlies the imaging process. The second interpretation is based on the way images are printed. Grey values are obtained by printing clusters of dots, where the density depends on the grey value to be printed. The third interpretation is based on the idea that an image can be considered as a mountain landscape (see Fig. 5.16). Now, the grey value corresponds to a height in the image. As the previous interpretations were based on densities, this can be extended to this interpretation as well. Then, each pixel is considered as a column of mass in the mountain landscape. The mass is now defined as the height of the column. Density values are obtained by dividing the mass by the total mass in the window.

The method discussed in this section is based on measuring differences in the mass density values within a window. Now, suppose that the grey value at position (i, j) is given by $I(x, y)$, then the total mass M in a window is given as:

$$M = \sum_{i,j} I(i, j), \tag{5.1}$$

where the sum is taken over the whole window. Then the mass density μ at position (i, j) is given by:

$$\mu_{i,j} = \frac{I(i, j)}{M}. \tag{5.2}$$

Note that:

$$\sum_{i,j} \mu_{i,j} = 1, \tag{5.3}$$

a condition that has to be fulfilled to apply the generalized dimension model. Within this method these density values are considered as being the probability values of the generalized dimension model. Now, the formula of the generalized dimension model can be applied again. This means:

$$D_\alpha = \lim_{\epsilon \to 0} \frac{1}{\alpha - 1} \frac{\log \sum_{i,j} \mu_{i,j}^{\alpha}}{\log \epsilon}. \tag{5.4}$$

As we stated in Section 4.6 the exponent α is considered to be a weighting coefficient. With this mechanism, an increase in the discriminating performance might be obtained compared to techniques that are only based on the determination of the fractal dimension.

The extension of this approach to the pyramidal data structure is straightforward. In fact, this operator is especially suited for this data structure. The initialization of the texture pyramid starts by a summing process. Each father node is given the total mass of the corresponding son nodes. Density values are obtained by dividing the son values by the value of the father node. As for the initialization technique discussed in the previous section, 50%-overlapping windows are applied. Bear in mind that the total sum value has to be corrected on this extension. For the determination of the dimension values the two levels underneath the level of the father node are considered.

Of course, initialization of the two lowest levels is not possible. This is not really a disadvantage of this technique; actually, it is fully in

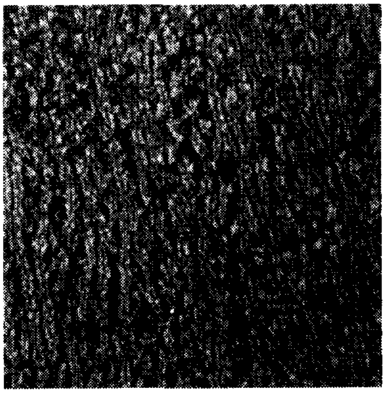

Figure 5.17: Pressed calf leather. Picture d24 from [14].

agreement with the idea that the level in the pyramid corresponds with the scale at which the texture is studied. Thus, the two lowest levels correspond to scales where the recognition of the texture is impossible. Segmentation on these levels can only be based on grey value information. Already having a coarse segmentation result from at least one of the levels higher in the pyramid, an adjustment on the borders of the textural regions could take place at these lower levels. The grey values of the border pixels are, for instance, compared with the local averages of the neighboring regions. The segmentation results that are obtained by following such an approach will probably be more irregular than the obtained results as shown in Figs. 3.15, 3.16. However, the results shown in these figures do not need to be better, because irregular region borders can be smoothed out by the technique followed.

The experiments carried out using this algorithm are based on a selection of five images taken from the Brodatz collection. The selection consists of (for the human observer) similar types of textures and of dissimilar types of textures. The average values of the resulting images were equalized. The resulting images of the selection are shown in the Figs. 5.17 - 5.21.

The Brodatz images have also been used to make composition images, where each composition consists of two textures taken from the selection of Brodatz textures. The composition schemes are shown in the Figs. 5.22, and 5.23. Based on these composition schemes, 20 com-

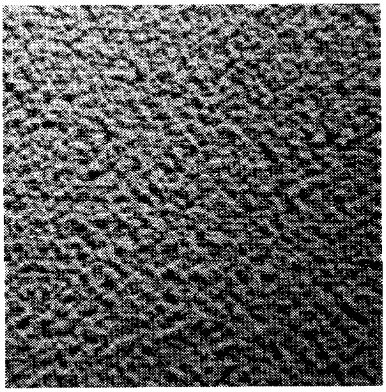

Figure 5.18: Handmade paper. Picture d57 from [14].

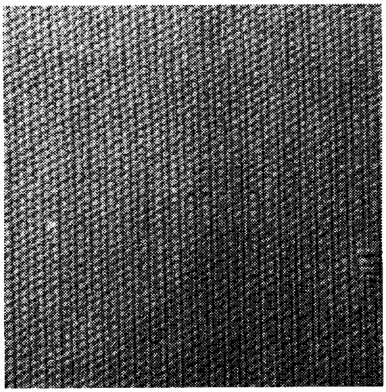

Figure 5.19: Cotton canvas. Picture d77 from [14].

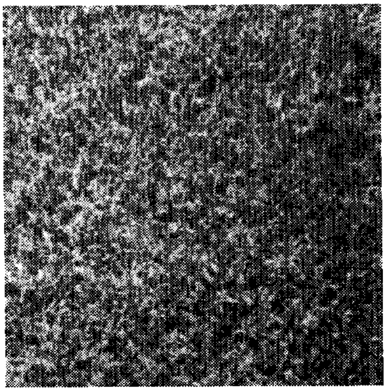

Figure 5.20: Grass lawn. Picture d9 from [14].

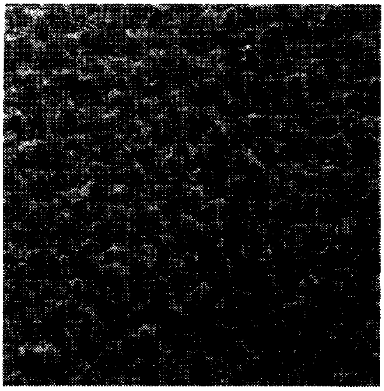

Figure 5.21: Pigskin. Picture d92 from [14].

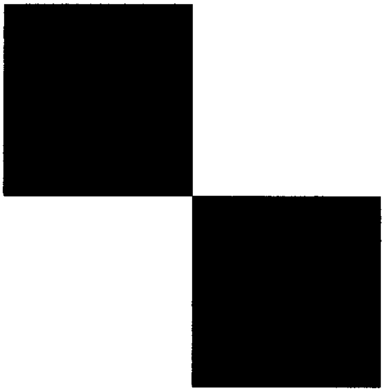

Figure 5.22: The block composition scheme.

position images were obtained. Two of them are shown in Figs. 5.24, and 5.25. By applying these two composition schemes, the influence of the shape and the positioning of the borders in the image could be studied as well.

The evaluation of the performance of the mass pyramid technique is characterized by two stages. First, a selection of values for α has to be made with which a maximum discrimination performance is obtained. Second, the performance on discrimination has to be tested.

Initially, the following values for α have been tested: $\alpha = -10, -2, -1, 1, 2, 10$. Per single texture image (see Figs. 5.17 - 5.21) we made a pyramid for each value of α. The histograms made were

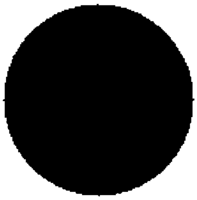

Figure 5.23: As Fig. 5.22, but now circular. The center of the circle is at position $(128, 128)$ and the radius is 64.

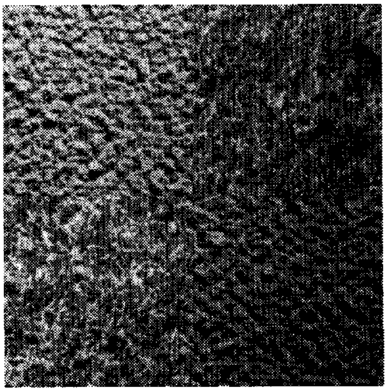

Figure 5.24: An example of a block composition corresponding to Fig. 5.22.

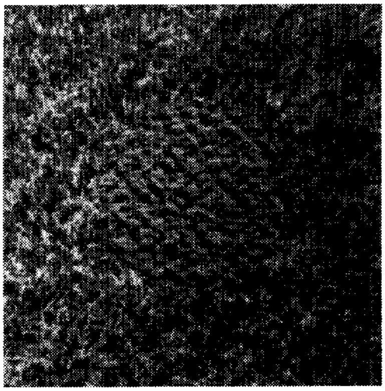

Figure 5.25: An example of a circular composition corresponding to Fig. 5.23.

for the levels 6, 5, and 4. Now, distances between the histograms had to be determined per level and per value for α . The most interesting values for α had to be selected from these distances . The distances were measured with the Matusita distance, that is given by:

$$d_M = \sqrt{\sum_i (h_1(i) - h_2(i))^2}, \tag{5.5}$$

where $h_1(x)$ and $h_2(x)$ stand for the two histograms.

From the resulting 240 distance values, the average and standard deviation were determined per level and per value for α. The results are shown in Table 5.1. It must be stressed that the distances between the levels cannot be compared, because the histograms were made with different lengths of intervals. From Table 5.1, it was concluded that the values -1.0 and 1.0 for α appeared to be the most interesting values for discrimination purposes. A selection of more values would significantly increase the complexity of the analysis of this technique. In case where only one value of α is selected, there is no longer any advantage anymore over an approach that is based on the fractal dimension.

In order to evaluate the possible discrimination of the texture pairs in the composition images, scatter plots were made. Therefore, the data points are plotted in a space that has been spanned by axes that correspond to $\alpha = -1.0$, respectively $\alpha = 1.0$. These plots were made for the levels 6, 5, and 4 of the pyramids.

level	measure	-10.0	-2.0	-1.0	1.0	2.0	10.0
6	μ	0.085	0.180	0.252	0.313	0.145	0.079
	σ	0.028	0.096	0.134	0.209	0.102	0.034
5	μ	0.189	0.366	0.502	0.499	0.346	0.159
	σ	0.061	0.562	0.223	0.205	0.160	0.062
4	μ	0.276	0.562	0.718	0.606	0.564	0.258
	σ	0.086	0.196	0.256	0.243	0.223	0.086

Table 5.1: The averages (μ) and the standard deviations (σ) of the histogram distances.

From these plots it appeared that only the texture pair consisting of the textures d24 and d9 (see Figs. 5.17, and 5.20) did not result in separable clusters. The scatter plot for level 6 of this combination his shown in Fig. 5.26. In the plot, only 50 data points per texture have been plotted. The scatter plots for the levels 5 and 4 looked similar. The other texture pairs did result in separable clusters. Discrimination was for some texture pairs only possible at a specific level in the pyramid. For instance, the combination d9 vs. d92 was only separable at level 6 of the pyramid, whereas the combination d9 vs. d77 was only separable at level 4 (see Figs. 5.27 - 5.29). It was this behavior we were aiming at. By applying a texture operator in a single plane, one might miss a scaling level that is significant for discrimination. The pyramidal data structure ables us to study a whole range of scaling levels simultaneously.

Other combinations -as for instance the combination d57 vs d92- were almost completely separable at all levels, although a certain preferable level was found for which the separability was maximum. The texture pair d57 vs. d77 appeared to be separable at levels 6 and 5 by using the combination of $\alpha = -1.0$ and $\alpha = 1.0$, whereas at level 4 only one α was required.

The combination of d57 and d92 was rather well separable at all levels with one value for α. By manually thresholding, we obtained some preliminary segmentation results, which are shown in Figs. 5.30 - 5.35. It must be stressed that a more advanced type of segmentation procedure may result in an increase of the quality of the segmentation

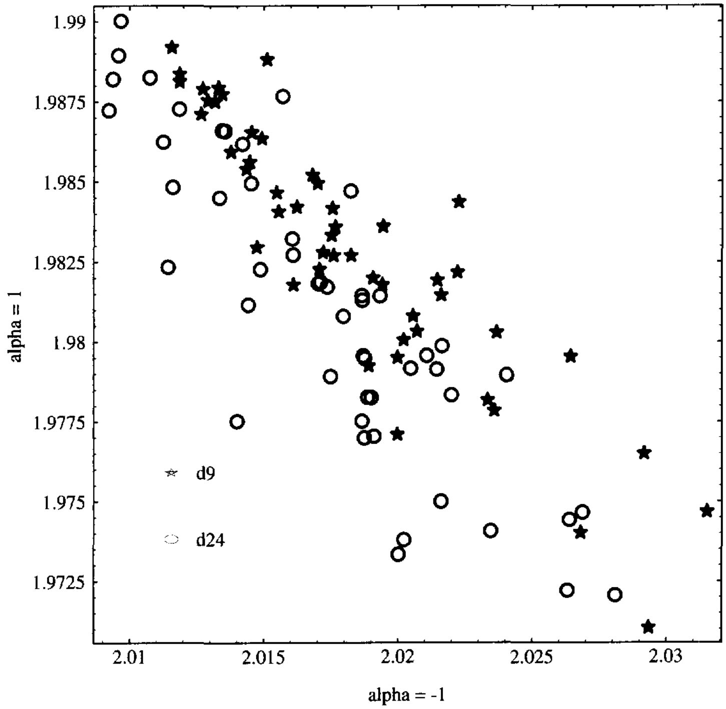

Figure 5.26: The scatter plot at level 6 for the textures d24 and d9. This was the only combination which did not result in separable clusters.

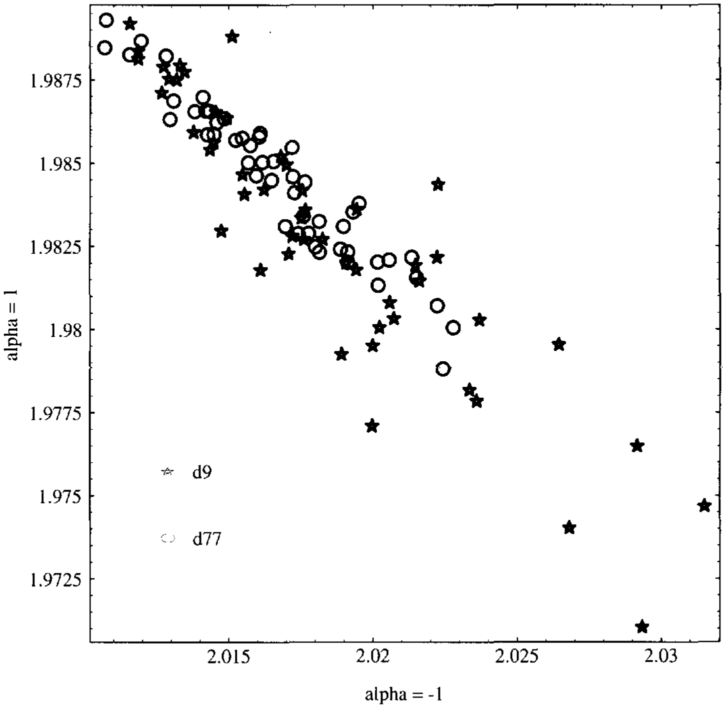

Figure 5.27: The scatter plot at level 6 for the textures d9 and d77. This texture pair is only separable at level 4 of the pyramid.

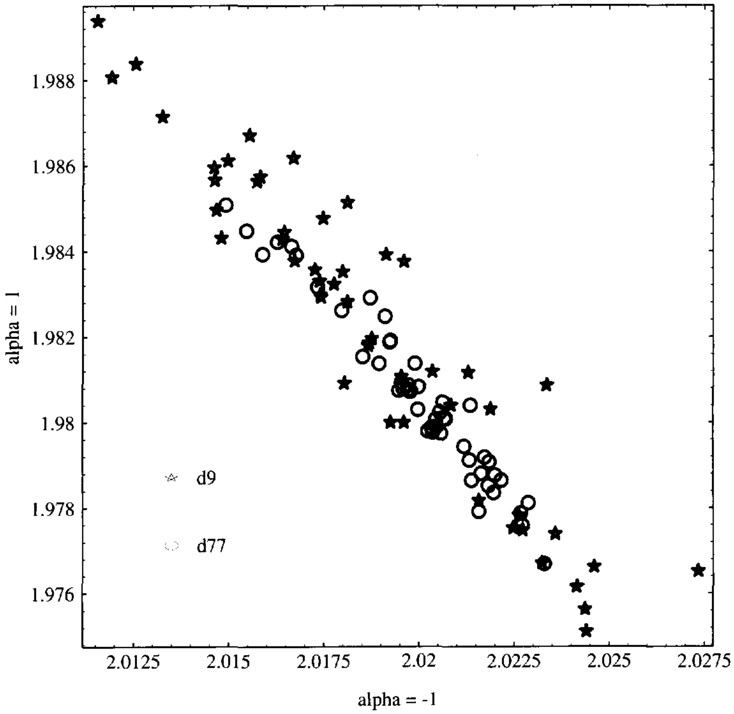

Figure 5.28: As Fig. 5.27 but then for level 5.

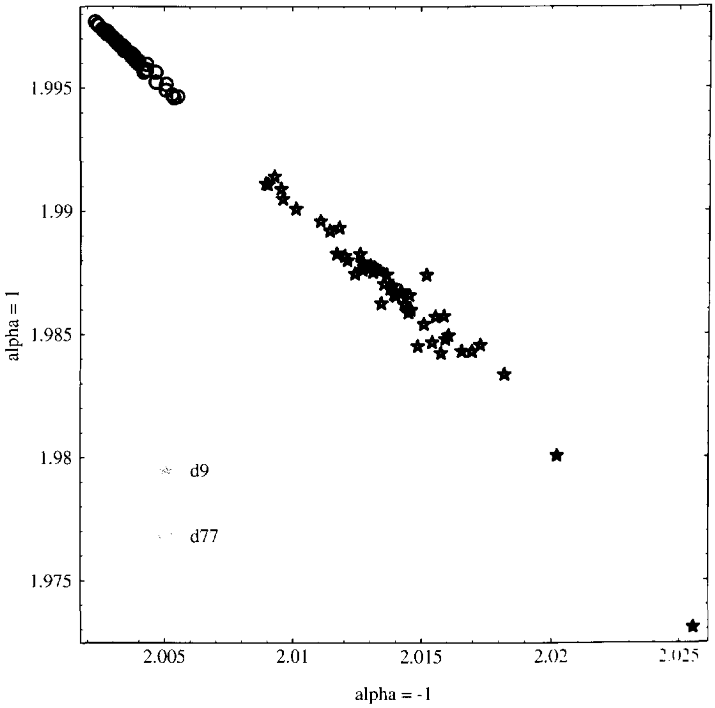

Figure 5.29: As Fig. 5.27 but then for level 4.

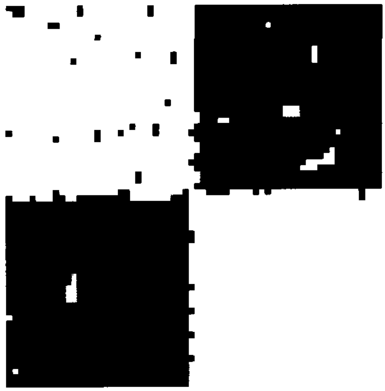

Figure 5.30: Thresholding result of the block composition of texture pair d57 and d92 at level 6 of the pyramid.

result. Further experiments carried out with composition images that contain other textures than the five textures studied in this section showed similar behavior.

Considering the fact that the method shows a good discrimination performance, and the fact that separability for some texture pairs only occurred at particular levels in the pyramid, it is concluded that we have succeeded in the task of finding a resolution reduction operator for the texture pyramid. Further, it must be stressed that despite the straightforward implementation, the computation is fast. When running the algorithm on a SUN Sparc station IPC, where the algorithm has been implemented in the SCIL image processing package, the computation took about 30 seconds.

5.6 Discussion

With the mass pyramid, we have found a well-performing initialization technique for the texture pyramid. The design of the initialization technique originates from the fractal theory, but finally only the concept of quantifying scaling behaviour as prescribed by the fractal theory has been used. The operator that has been developed has the weighting mechanism as prescribed by the generalized dimension model. Because of the distance that the initialization technique takes from the fractal

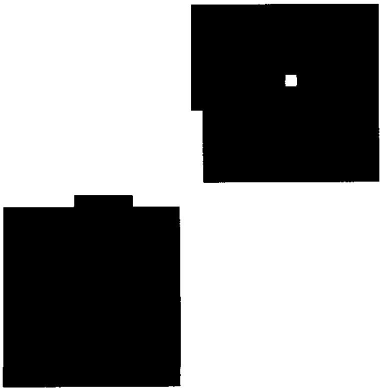

Figure 5.31: As Fig. 5.30, but for level 5.

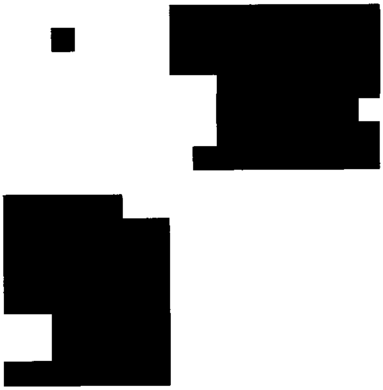

Figure 5.32: As Fig. 5.30, but for level 4.

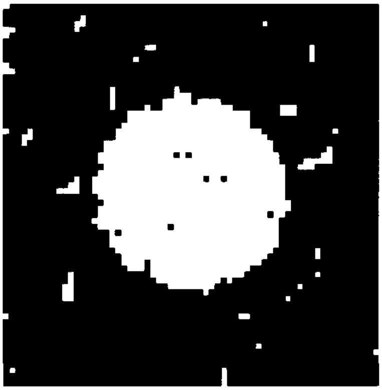

Figure 5.33: Thresholding result of the circle composition of texture pair d57 and d92 at level 6 of the pyramid.

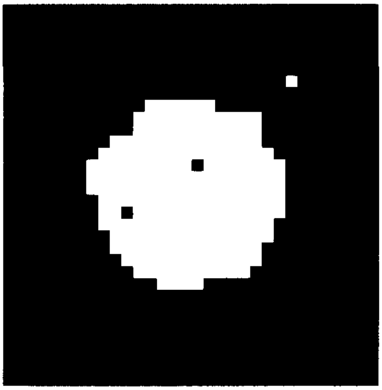

Figure 5.34: As Fig. 5.33, but for level 5.

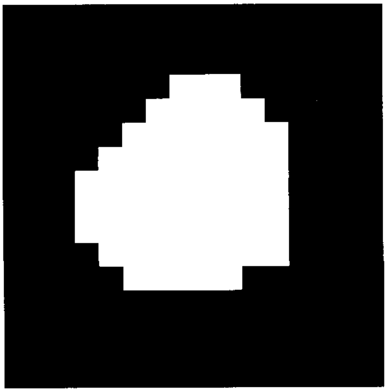

Figure 5.35: As Fig. 5.33, but for level 4.

theory, one might wonder if it makes sense to continue this relationship. For practical reasons -for instance if the applications have to be implemented into hardware-, it might be interesting to design a similar type of algorithm that leaves the generalized dimension model. It is important that the algorithm posseses a weighting mechanism as prescribed by the generalized dimension model. Further, the idea that the level of the pyramid corresponds to the scale at which the texture is quantified should remain.

Until now, we have only made some remarks in Chapter 3 regarding the suitability of the pyramidal data structure for segmentation. Besides the linked pyramid approach as discussed in Section 3.2.4, reference has been made to an approach described by Eijlers [34], which is based on an extension to the region-growing algorithm. The information in a pyramid can be used in several ways for image segmentation purposes. An advanced technique might integrate all information in the pyramid(s) in a such way that a hierarchical segmentation is obtained. This means that the segmentation algorithm starts at a higher level in the pyramid on which a preliminary segmentation result is obtained. At higher resolution levels, the positioning of the borders may be more exact. Further, on these higher resolution levels, segmentation within the regions might take place and in this way a hierarchical segmentation result be obtained. The attention paid in general to this kind of use of the information in pyramids is still limited and, therefore, it is recommended that more attention be paid to this subject.

Chapter 6

Conclusions and recommendations

Texture plays a significant role in the processing and analysis of images. Despite the amount of research carried out on this subject, texture as an image processing phenomenon is still an undefined phenomenon. However, in the literature, several attempts at describing textures can be found. Most attempts have in common the awareness that the appearance is most often dependent on the level of scaling at which the texture is studied.

Studying image processing phenomena on several levels of resolution is often done in the pyramidal data structure. Many applications of this data structure are to be found in the literature. One of these is the use of the pyramid for the analysis of textures. However, the number of studies on this application is limited when compared to other types of applications.

In general, we can distinguish two ways in which the pyramids are used. The first type of use aims at the decrease in the variance of the data. This means that the meaning of the data does not change with the level in the pyramid, but only that the variance is averaged out. The second type of use is based on the idea that the levels higher in the pyramid should correspond to a more coarsely oriented description of the original image.

Both types of use can be found in the literature. Regretfully, it is the first type of use that is often (consciously or unconsciously) implemented. This usage can be explained by the fact that resolution reduction for this type of use is straightforward. This reduction, however, does not match with the meaning of the data. In contrast to the first type of use, the second type of use is in agreement with the concept of pyramidal data structures. This means that for the design of the resolution reduction operator, one needs a description of the phenomenon, of which scaling is an intrinsic aspect.

Both types of use are also found in the study of textures. Following the second type of use meant originally that the effect of borders increases with the height in the pyramid. By applying a model for the initialization, the derived features might automatically bubble up in the pyramid. As mentioned before, scaling should be an intrinsic aspect of such a model. Such a model might be based on the fractal theory, which describes phenomena on different levels of scaling.

Since the popularization of the fractal theory, articles have started to appear on its use in image processing applications. Most of the articles focus on the use of the fractal dimension, as a measure for the whimsicality of signals. Fractal theory, however, describes also the use of such derived measures as the lacunarity, and the information and the correlation dimension. These latter types of dimension definitions are based on the *generalized dimension model.* Using the fractal dimension in conjunction with at least one of these other measures might increase the discrimination performance.

Despite the fact that texture, pyramids and fractal theory have scaling as a concept in common, the study on this relationship appears to be a step-child. In this thesis, we have studied the applicability of the fractal theory for the initialization of texture pyramids.

By making use of the above-mentioned relationship we succeeded in the design of an initialization algorithm for the texture pyramid which is based on the concept of the generalized dimension model. As the fractal theory describes a phenomenon on a broad range of scaling, for the pyramidal data structure we had to depart from this central concept of fractal theory. Our method is based on the idea of dividing the scaling range into a number of smaller intervals, where the number of intervals is given by the number of layers in the pyramid. For each interval, one or more dimension features from the generalized dimension model are determined. Although this means that the relationship with the fractal theory is weakened, we win in generality. As the use of the fractal theory *en pure sang* is restricted to a rather restricted class of textures, our method is more general.

For our technique we have introduced the *mass pyramid*. The initialization of this pyramid is straightforward. In a discriminatory sense the operator has shown to perform good. For some of the texture pairs we have studied, we found that discrimination was only possible at a higher level in the pyramid, whereas the discrimination between other texture pairs was only possible at lower levels. Some other texture pairs could be discriminated on a number of layers. The scaling behavior found was what we were aiming at, when starting with the study of the use of pyramidal data structures for texture analysis and segmentation. Further, the implementation of the method is -because of its elegance- straightforward and only a limited processing time is required.

Because of the weakened relationship with the fractal theory, it is recommended to study whether a further weakening might lead to an operator that performs equally well in discriminatory sense, but where the methodology is even more simple. Such an algorithm would require possibly even less processing time and its suitability for implementation in a pyramidal hardware architecture is not excluded. It is of importance, however, to maintain a weighting mechanism as prescribed by the generalized dimension model. A different type of approach could be based on the integration of wavelet theory and fractal theory for the study of textures. This relationship is already the subject of some articles that can be found in the literature.

References

[1] A. Ait-Kheddache and S. A. Rajala. "Texture classification based on higher-order fractals". In: *Proceedings 1988 International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1112–1115, New York, 1988.

[2] F. Albregtsen, B. Nielsen, and K. Yogesan. "Fractal texture signature estimated by multiscale lit-snn and max-min operators on landsat-5 mss images of the antarctic". In: M. Pietikäinen and J. Röning, editors, *Proceedings of the 6th Scandinavian Conference on Image Analysis*, pages 995–1002, Oulu, June 1989.

[3] H. J. Antonisse. "Image segmentation in pyramids". *Computer Graphics and Image Processing*, vol. 19, pp. 367–383, 1982.

[4] F. Arduini, S. Fioravanti, D. D. Giusto, and F. Inzirillo. "Multifractals and texture classification". In: *Proceedings IEE 4th International Conference on Image Processing and its applications*, pages 454–457, Maastricht, April 1992. Conference publication number 354.

[5] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. "Uniqueness of the Gaussian kernel for scale-space filtering". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8(1), pp. 26–33, January 1986.

[6] E. Backer, J. J. Gerbrands, and E. J. Eijlers. "Intelligent control of multi-dimensional binary image measurements". In: J. C. Simon, editor, *From Pixels to Features*, pages 361–371. North-Holland, August 1989.

[7] J. W. Bacus and E. E. Gose. "Leukocyte pattern recognition". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2(4), pp. 512–526, September 1972.

[8] M. Barnsley. *Fractal Everywhere.* Academic Press, Inc., 1988.

[9] J. Besuijen. *Statistische textuurmaten voor beeldsegmentatie.* M.Sc. thesis, Delft University of Technology, July 1984. In Dutch.

[10] D. J. W. Bierhuizen. *Segmentatie op basis van bewegingsinformatie.* M.Sc. thesis, Delft University of Technology, Department of Electrical Engineering, Information Theory Group, May 1988. In Dutch.

[11] J. M. Blackledge and E. Fowler. "Fractal dimension segmentation of synthetic aperture radar images". In: *Proceedings IEE 4th International Conference on Image Processing and its applications*, pages 445–449, Maastricht, April 1992. Conference publication number 354.

[12] R. P. Blanford and S. Tanimoto. "Bright-spot detection in pyramids". *Computer Vision, Graphics, and Image Processing*, vol. 43, pp. 133–149, 1988.

[13] A. C. Bovik. "Analysis of multichannel narrow-band filters for image texture segmentation". *IEEE Transactions on Signal Processing*, vol. 39(9), pp. 2025–2043, September 1991.

[14] P. Brodatz. *TEXTURES, a photographic album for artists & designers.* Dover Publications, Inc., 1966.

[15] P. J. Burt. "Fast filter transforms for image processing". *Computer Graphics and Image Processing*, vol. 16, pp. 20–51, 1981.

[16] P. J. Burt. "The pyramid as a structure for efficient computation". In: A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, chapter 2, pages 6–35. Springer Verlag, Berlin, 1984.

[17] P. J. Burt and E. H. Adelson. "The Laplacian pyramid as a compact image code". *Transactions on Communications*, vol. COM-31(4), pp. 532–540, April 1983.

[18] P. J. Burt and E. H. Adelson. "Merging images through pattern decomposition". In: A. G. Tescher, editor, *Applications of Digital Image Processing 8*, volume 575, pages 173–181, San Diego, California, August 1985. The International Society for Optical Engineering, SPIE.

[19] P. J. Burt, C. H. Anderson, J. O. Sinniger, and G. van der Wal. "A pipelined pyramid machine". In: *Pyramidal Systems for Computer Vision*, volume 25 of *Series F: Computer and Systems Sciences*, pages 133–152. NATO ASI Series, 1986.

[20] P. J. Burt, T. H. Hong, and A. Rosenfeld. "Segmentation and estimation of image region properties through cooperative hierarchical computation". *IEEE Transaction on Systems, Man, and Cybernetics*, vol. SMC-11(12), pp. 802–809, December 1981.

[21] C. Caldwell, S. J. Stapleton, D. W. Holdsworth, R. A. Jong, W. J. Weiser, G. Cooke, and M. J. Yaffe. "Characterisation of mammographic parenchymal pattern by fractal dimension". *Physics in Medicine Biology*, vol. 35(2), pp. 235–247, 1990.

[22] R. Chellapa and R. L. Kashyap. "Texture synthesis using 2-d noncausal autoregressive models". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33(1), pp. 194–203, February 1985.

[23] H. A. Cohen and J. You. "A multi-scale texture classifier based on multi-resolution 'tuned' mask". *Pattern Recognition Letters*, vol. 13, pp. 599–604, September 1992.

[24] R. W. Conners and C. A. Harlow. "A theoretical comparison of texture algorithms". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2(3), pp. 204–222., May 1980.

[25] J. Crowley and A. C. Parker. "A representation for shape based on peaks and ridges in the difference of low-pass transform".

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-6(2), pp. 156–170, March 1984.

[26] J. Crowley and A. C. Sanderson. "Multiple resolution representation and probabilistic matching of 2-d gray scale shape". In: *Proceedings IEEE 1984 Workshop on Computer Vision*, Anniapolis, Maryland, May 1984.

[27] J. L. Crowley and A. C. Sanderson. "Multiple resolution representation and probabilistic matching of 2-d gray-scale shape". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9(1), pp. 113–121, January 1987.

[28] J. L. Crowley and R. M. Stern. "Fast computation of the difference of low-pass transform". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6(2), pp. 212–222, March 1984.

[29] L. S. Davis, M. Clearman, and J. K. Aggarwal. "An empirical evaluation of generalized co-occurrence matrices". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3(2), pp. 214–221, March 1981.

[30] S. Dellepiane, S. B. Serpico, G. Vernazza, and R. Viviani. "Fractal-based image analysis in radiological applications". *SPIE Visual Communications and Image Processing*, vol. 845(2), pp. 396–403, 1987.

[31] N. Dodd. "Multispectral texture synthesis using fractal concepts". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9(5), pp. 703–707, September 1987.

[32] N. A. Dodd. "Texture generation using fractal concepts". In: *Proceedings Second International Conference on Image Processing and its Applications*, pages 253–257, London, June 1986.

[33] B. Dubuc, C. Roques-Carmes, C. Tricot, and S. W. Zucker. "The variation method: a technique to estimate the fractal dimension of surfaces". *SPIE Visual Communications and Image Processing*, vol. 845(2), pp. 241–248, 1987.

[34] E. J. Eijlers. *Pyramidal Structures for Image Understanding Systems.* M.Sc. thesis, Delft University of Technology, Department of Electrical Engineerig, Information Theory Group, June 1987.

[35] E. J. Eijlers, J. J. Gerbrands, and E. Backer. "An improved linked pyramid for texture segmentation using the fractal Brownian model". In: *Proceedings 10th International Conference on Pattern Recognition*, volume 1, pages 687–689, Atlantic City, New Jersey, USA, 1990.

[36] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications.* John Wiley & Sons, 1990.

[37] F. Farrokhnia and A. K. Jain. "A multi-channel filtering approach to texture segmentation". In: *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Patter Recognition*, Lahaina, Maui, Hawaii, June 1991. IEEE Computer Society Press.

[38] J. Feder. *Fractals.* Plenum, 1988.

[39] A. Fournier, D. Fussell, and L. Carpenter. "Computer rendering of stochastic models". *Communications of the ACM*, vol. 25(6), pp. 371–384, June 1982.

[40] A. Gagalowicz and S. D. Ma. "Sequential synthesis of natural textures". *Computer Graphics and Image Processing*, vol. 30, pp. 289–315, 1985.

[41] M. M. Galloway. "Texture analysis using gray level run lengths". *Computer Graphics and Image Processing*, vol. 4, pp. 172–179, 1975.

[42] F. Glazer. "Scene matching by hierarchical correlation". In: *Proceedings 1983 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1983*, Washington D.C., June 1983.

[43] X. Gong and N. K. Huang. "Textured image recognition using hidden Markov model". In: *Proceedings International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1128–1131, New York, USA, 1988.

[44] L. Gool, P. Dewaele, and A. Oosterlinck. "Survey: Texture analysis anno 1983". *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 336–357, 1985.

[45] P. Grassberger and I. Procaccia. "Dimensions and entropies of strange attractors from a fluctuating dynamics approach". *Physica 13D*, pages 34–54, 1984.

[46] E. L. Hall, R. Y. Wong, and L. J. Rouge. "Hierarchical search for image matching". In: *Proceedings 1976 IEEE Conference on Decision and Control*, pages 791–796, Clearwater, 1976.

[47] T. C. Halsey, M. H. J. L. P. Kadanoff, I. Procaccia, and B. I. Shraiman. "Fractal measures and their singularities: The characterization of strange sets". *Physical Review A*, vol. 33(2), pp. 1141–1151, February 1986.

[48] R. M. Haralick. "Statistical and structural approaches to texture". In: *Proc. 4th International Conference on Pattern Recognition*, Kyoto, Japan, 1978. pp. 45-69.

[49] R. M. Haralick, K. Shanmugam, and I. Dinstein. "Textural features for image calssification". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3(6), pp. 610–621, November 1973.

[50] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley Publishing Company, 1992.

[51] H. G. E. Hentschel and I. Procaccia. "The infinite number of generalized dimensions of fractals and strange attractors". *Physica*, vol. 8D, pp. 435–444, 1983.

[52] T. H. Hong and M. Shneier. "Extracting compact objects using linked pyramids". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6(2), pp. 229–237, March 1984.

[53] T. Hong, K. A. Narayanan, S. Peleg, A. Rosenfeld, and T. Silberberg. "Image smoothing and segmentation by multiresolution pixel linking: Further experiments and extensions". *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-12(5), pp. 611–622, September 1982.

[54] T. Hong and A. Rosenfeld. "Compact region extraction using weighted pixel linking in a pyramid". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6(2), pp. 222–237, March 1984.

[55] S. L. Horowitz and T. Pavlidis. "Picture segmentation by a directed split-and-merge procedure". In: *Proceedings Second International Joint Conference on Pattern Recognition*, pages 424–433, Copenhagen, Danmark, 1974.

[56] A. Jacquin. "Fractal image coding based on a theory of iterated contractive image transformations". In: M. Kunt, editor, *Visual Communications and Image Processing*, volume 1360, pages 227–239, Lausanne, Switzerland, October 1990. SPIE.

[57] A. K. Jain and F. Farshid. "Unsupervised texture segmentation using Gabor filters". *Pattern Recognition*, vol. 24(12), pp. 1167–1186, 1991.

[58] T. Jeffery. "Mimicking mountains". *Byte*, pages 337–344, December 1987.

[59] B. Julesz. "Textons, the elements of texture perception, and their interactions". *Nature*, vol. 290, pp. 91–97, March 1981.

[60] R. E. H. Kamphuis. *Bepaling van hogere orde fractal dimensies.* Undergraduate report, Delft University of Technology, June 1991. In Dutch.

[61] K. Kanatani and T. C. Chou. "Shape from texture: General principle". *Artificial Intelligence*, pages 1–48, 1989.

[62] H. Kaneko. "Fractal matrix model and its application to texture analysis". *The Transactions of the IEICI*, vol. E 71(12), pp. 1221–1228, December 1988. Special Issue on CAS Karuizawa Workshop.

[63] R. L. Kashyap and K. B. Eom. "Texture boundary detection based on the long correlation model". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-11(1), pp. 58–67, January 1989.

[64] R. L. Kashyap and P. M. Lapsa. "Synthesis and estimation of random fields using long-correlation models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6(6), pp. 800–809, November 1984.

[65] S. Kasif and A. Rosenfeld. "Pyramid linking is a special case of isodata". *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-13(1), pp. 84–85, January 1983.

[66] B. H. Kaye. *A Random Walk Through Fractal Dimensions.* VCH Verlagsgesellschaft, Weinheim, 1989.

[67] J. M. Keller and S. Chen. "Texture description and segmentation through fractal geometry". *Computer Vision, Graphics, and Image Processing*, vol. 45, pp. 150–166, 1989.

[68] J. M. Keller and Y. B. Seo. "Local fractal geometric features for image segmentation". *International Journal of Imaging Systems and Technology*, vol. 2, pp. 267–284, October 1990.

[69] M. D. Kelly. "Edge detection in pictures by computer using planning". *Machine Intelligence*, vol. 6, pp. 397–409, 1971.

[70] J. J. Koenderink. "The structure of images". *Biological Cybernetics*, vol. 50, pp. 363–370, 1984.

[71] J. P. Koorevaar. *De fractaldimensie van fotostructuren in amorf silicium.* Undergraduate report, Delft University of Technology, 1990. In Dutch.

[72] B. J. A. Kröse. *A Description of Visual Structure.* Ph.D. thesis, Delft University of Technology, 1986.

[73] J. Lubbe. *A Generalized Probabilistic Theory of the Measurement of Certainty and Information.* Ph.D. thesis, Delft University of Technology, June 1981.

[74] T. Lundahl, W. J. Ohley, S. M. Kay, and R. Siffert. "Fractional Brownian motion: A maximum likelihood estimator and its application to image texture". *IEEE Transactions on Medical Imaging,* vol. MI-5(3), pp. 152–161, September 1986.

[75] B. B. Mandelbrot. *Fractals: Form, Chance, and Dimension.* W. H. Freeman and Company, 1977.

[76] B. B. Mandelbrot. "Comment on computer rendering of stochastic models". *Communications of the ACM,* vol. 25(8), pp. 581–584, August 1982. Authors' Reply to comment included.

[77] B. B. Mandelbrot. *The Fractal Geometry of Nature.* W. H. Freeman and Company, 1983.

[78] B. B. Mandelbrot. "Fractal measures (their infinite moment sequences and dimensions) and multiplicative chaos: Early works and open problems". In: G. Mayer-Kress, editor, *Dimensions and Entropies in Chaotic Systems: Quantification of Complex Behavior,* pages 19–27, Pecos River Ranch, New Mexico, September 1985.

[79] B. B. Mandelbrot. "Self-affine fractals and fractal dimension". *Physica Scripta,* vol. 32, pp. 257–260, 1985.

[80] B. B. Mandelbrot. "An introduction to multifractal distribution functions". In: H. E. Stanley and N. Ostrowsky, editors, *Random*

Fluctuations and Pattern Growth: Experiments and Growth, volume 157 of *NATO ASI Series, Series E, Applied Sciences*, pages 279–291. Kluwer Academic Publishers, Cargese, July 1988.

[81] B. B. Mandelbrot and J. W. van Ness. "Fractional Brownian motions, fractional noises and applications". *SIAM Review*, vol. 10(4), pp. 422–437, 1968.

[82] D. Marr. *VISION A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman and Company, 1982.

[83] B. H. McCormick and S. N. Jayaramamurthy. "Time series model for texture synthesis". *International Journal of Computer and Information Sciences*, vol. 3(4), pp. 329–343, 1974.

[84] M. Minsky. "Steps toward artificial intelligence". In: E. A. Feigenbaum and J. Feldman, editors, *Computer Thoughts*, pages 406–450, New York, 1963. McGraw-Hill.

[85] O. R. Mitchell, C. R. Myers, and W. Boyne. "A max-min measure for image texture analysis". *IEEE Transactions on Computers*, pages 408–414, April 1977.

[86] A. Morales and R. Acharya. "An image pyramid with morphological operators". In: *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 526–531, Lahaina, Maui, Hawaii, June 1991.

[87] U. Müssigmann. "Texture analysis, fractals and scale space filtering". In: M. Pietikäinen and J. Röning, editors, *Proceedings of the 6th Scandinavian Conference on Image Analysis*, pages 987–994, Oulu, Finland, June 1989.

[88] P. T. Nguyen and J. Quinqueton. "Space filling curves and texture analysis". In: *Proceedings 6th International Conference on Pattern Recognition*, pages 282–285, Munich, Germany, October 1982.

[89] W. Ohley and T. Lundahl. "Discrete 2-dimensional fractional Brownian motion as a model for medical images". *SPIE Visual Communications and Image Processing*, vol. 845(2), pp. 227–232, 1987.

[90] R. Oort. *Hydrogenated amorphous and microcrystalline silicon deposited from silane-hydrogen mixtures.* Ph.D. thesis, Delft University of Technology, April 1989.

[91] S. Peleg and O. Federbusch. "Custom made pyramids". In: *Pyramidal Systems for Computer Vision*, volume 25 of *Series F: Computer and Systems Sciences*, pages 165–171. NATO ASI Series, 1986.

[92] S. Peleg, J. Naor, R. Hartley, and D. Avnir. "Muliple resolution texture analysis and classification". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6(4), pp. 518–523, 1984.

[93] A. P. Pentland. "Fractal-based description". In: *Proceedings International Joint Conference Artificial Intelligence*, pages 973–981, Karlsruhe, Germany, August 1983.

[94] A. P. Pentland. "Fractal-based description of natural scenes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6(6), pp. 661–674, November 1984.

[95] A. P. Pentland. "Fractal surface models for communication about terrain". *SPIE Visual Communications and Image Processing*, vol. 845(2), pp. 301–306, 1987.

[96] M. Pietikäinen and A. Rosenfeld. "Image segmentation by texture using pyramid node linking". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11(12), pp. 822– 825, December 1981.

[97] W. K. Pratt, O. D. Faugeras, and A. Gagalowicz. "Visual discrimination of stochastic texture fields". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-8(11), pp. 796–804, November 1978.

[98] W. K. Pratt, O. D. Faugeras, and A. Gagalowicz. "Applications of stochastic texture field models to image processing". *Proceedings of the IEEE*, vol. 69(5), pp. 542–551, May 1981.

[99] I. Procaccia. "The characterization of fractal measures as interwoven sets of singularities: Global universality at the transition to chaos". In: G. Mayer-Kress, editor, *Dimensions and Entropies in Chaotic Systems: Quantification of Complex Behavior*, pages 8–18, Berlin, September 1985. Springer-Verlag. Proceedings of an International Workshop at the Pecos River Ranch, New Mexico.

[100] A. Rosenfeld and M. Thurston. "Edge and curve detection for visual scene analysis". *IEEE Transactions on Computers*, vol. C-20(5), pp. 562–569, May 1971.

[101] A. Rosenfeld and G. J. Vanderbrug. "Coarse-fine template matching". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7, pp. 104–107, February 1977.

[102] H. E. Schepers, J. H. G. M. van Beek, and J. B. Bassingthwaighte. "Four methods to estimate the fractal dimension from self-affine signals". *IEEE Engineering in Medicine and Biology*, pages 57–64, June 1992.

[103] H. G. Schuster. *Deterministic Chaos.* Physik Verlag, 1984.

[104] M. Shneier. "Using pyramids to define local thresholds for blob detection". *IEEE Transactions on Patterns Analysis and Machine Intelligence*, vol. PAMI-5(3), pp. 345–349, May 1983.

[105] M. O. Shneier. "Extracting linear features from images using pyramids". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-12(4), pp. 569–572, July 1982.

[106] M. Spann, C. Horne, and J. M. H. du Buf. "The detection of thin structures in images". *Pattern Recognition Letters*, vol. 10, pp. 175–179, 1989.

[107] J. P. P. Starink. *Analysis of Electron Microscope Images: 3D Localization of Immuno Markers.* Ph.D. thesis, Delft University of Technology, April 1993.

[108] M. C. Stein. "Fractal image models and object detection". *SPIE Visual Communications and Image Processing*, vol. 845(2), pp. 293–300, 1987.

[109] S. Tanimoto and T. Pavlidis. "A hierarchical data structure for picture processing". *Computer Graphics and Image Processing*, vol. 4, pp. 104–119, 1975.

[110] S. L. Tanimoto. "Pictorial feature distortion in a pyramid". *Computer Graphics and Image Processing*, vol. 5, pp. 333–352, 1976.

[111] A. Toet. "Image fusion by a ratio of low-pass pyramid". *Pattern Recognition Letters*, vol. 9, pp. 245–253, May 1989.

[112] F. Tomita and S. Tsuji. *Computer Analysis of Visual Textures.* Kluwer Academic Publishers, 1990.

[113] M. Tuceryan and A. K. Jain. "Texture analysis". In: C. Chen, L. Pau, and P. Wang, editors, *The Handbook of Pattern Recognition and Computer Vision*, chapter 11. World Scientific Publishing Co., 1992.

[114] L. Uhr. "Layered "recognition cone" networks that preprocess, classify, and describe". *IEEE Transactions on Computers*, vol. C-21, pp. 758–768, July 1972.

[115] L. Uhr. "Psychological motivation and underlying concepts". In: S. Tanimoto and A. Klinger, editors, *Structured Computer Vision Machine Perception through Hierarchical Computation Structures*, pages 1–30. Academic Press, 1980.

[116] M. Unser and M. Eden. "Multiresolution feature extraction and selection for texture segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11(7), pp. 717–728, July 1989.

[117] J. L. Vehel. "Fractal probability functions: An application to image analysis". In: *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 378–383, Laihana, Maui, Hawaii, 1991.

[118] A. M. Vepsäläinen and J. Ma. "Estimating of fractal and correlation dimension from 2d- and 3d-images". *SPIE Visual Communications and Image Processing*, vol. 1199(4), pp. 431–438, 1989.

[119] R. F. Voss. "Random fractal forgeries". In: *Fundamental Algorithms for Computer Graphics*, volume F17 of *NATO ASI Series*, pages 805–835, Ilkley, Yorkshire, England, March 1985. Springer-Verlag, Berlin, Heidelberg.

[120] R. F. Voss. "Random fractals: characterization and measurement". In: R. Pynn and A. Skjeltorp, editors, *Scaling Phenomena in Disordered Systems*, volume 133 of *NATO ASI Series, Series B: Physics*, pages 1–11. NATOGeilo, Norway, April 1985.

[121] J. S. Weszka, C. R. Dyer, and A. Rosenfeld. "A comparative study of texture measures for terrain classification". *IEEE Transaction on Systems, Man, and Cybernetics*, vol. SMC-6(4), pp. 269–285, April 1976.

[122] H. Weyl. "Bemerkungen zum begriff des differentialquotienten gebrochener ordnung". In: K. Chandrasekharan, editor, *Gesammelte Abhandlungen*, volume 1, chapter 28, pages 663–698. Springer-Verlag, Berlin, 1968. In German.

[123] A. P. Witkin. "Scale space filtering". In: *Proceedings International Conference on Artificial Intelligence*, pages 1019–1021, Karlsruhe, 1983.

[124] K. M. Yang, L. Wu, and M. Mills. "Fractal based image coding scheme using Peano scan". In: *Proceedings 1988 IEEE International Symposium on Circuits and Systems*, volume 1, pages 2301–2304, Helsinki University of Technology, Espoo, Finland, June 1988.

[125] R. Yokoyama and R. M. Haralick. "Texture synthesis using a growth model". *Computer Graphics and Image Processing*, vol. 8, pp. 369–381, 1978.

[126] I. T. Young, R. L. Peverini, P. W. Verbeek, and P. J. van Otterloo. "A new implementation for the binary and minowski operators". *Computer Graphics and Image Processing*, vol. 17, pp. 189–210, 1981.

[127] S. W. Zucker. "Toward a model of texture". *Computer Graphics and Image Processing*, vol. 5, pp. 190–202, 1976.

Samenvatting

Bij de ontwikkeling van systemen voor digitale beeldverwerking en -analyse wordt men geconfronteerd met de mogelijkheid dat er zich texturele gebieden kunnen bevinden in de te verwerken beelden. Ondanks de rol die textuur speelt binnen de beeldverwerking, is textuur als beeldverwerkings fenomeen nog steeds ongedefinieerd. Zelfs het beschrijven van het begrip textuur blijkt een moeilijke zaak te zijn. Vaak wordt de beschrijving aangevuld met voorbeelden om het te verduidelijken. Als voorbeelden van texturen zijn te noemen: tapijt, wolken, textiel, leer, enz. Het begrip textuur wordt nog weleens in verband gebracht met structuur. Echter, een textuur hoeft zeker niet gestructureerd te zijn om door de mens als homogeen ervaren te worden. Het is juist deze eigenschap die de definitie van het begrip bemoeilijkt, en dientengevolge de analyse.

Analyse vindt plaats met een zogenaamde textuuroperator. Meestal kwantificeert een dergelijke operator de textuur binnen een venster waarvan de afmetingen gegeven zijn. Het oorpronkelijke beeld wordt getransformeerd door deze operator te schuiven over het gehele beeldvlak, waarbij de verkregen waarde meestal wordt toegekend aan het centrale beeldelement. Reeds tijdens de segmentatiefase -één van de eerste fasen binnen het hele beeldverwerkingsproces- bestaat er de behoefte aan een dergelijke operator. Gedurende deze fase wordt het beeld opgedeeld in gebieden die volgens een bepaald criterium homogeen zijn. Dat betekent dat van een textuuroperator wordt verwacht dat deze een konstante waarde afgeeft indien de onderliggende textuur gelijk is.

Een aantal van de omschrijvingen van textuur wijst op de schalingsaspecten van textuur. De verschijningsvorm van een textuur kan namelijk drastisch veranderen indien de waarnemingsschaal wordt gewijzigd. Van de vele textuuroperators die in de literatuur te vinden zijn, zijn er een aantal die de direkte mogelijkheid bieden om het schalingsniveau waarop de textuur wordt bestudeerd in te stellen. Echter, een dergelijke mogelijkheid verwacht voorkennis van de schaal (of het schalingsinterval) waarop de textuur bestudeerd dient te worden. In de praktijk hoeft deze voorkennis niet aanwezig te zijn. Het is daarom wenselijk om een analyse methode te volgen waarbij het beeld op een breed schalingsinterval wordt bestudeerd. Bij het bestuderen van beeldaspecten op verschillende niveaus van schaling wordt binnen de beeldverwerking vaak gebruik gemaakt van de zogenaamde piramidale datastructuur; ook wel aangeduid met piramide. Een dergelijke structuur bestaat uit een aantal lagen die het beeld op verschillende niveaus van resolutie beschrijven. Stel dat de afmetingen van het originele beeld 256×256 bedragen, dan bestaat de piramide uit 9 lagen, met de afmetingen: 256×256, 128×128, 64×64, 32×32, 16×16, 8×8, 4×4, 2×2, en 1×1. Indien we nu de lagen boven elkaar geplaatst denken, herkennen we hierin de piramide vorm. Cruciaal bij de initialisatie van een piramide is de resolutiereductie. In het geval van bijvoorbeeld de Gaussische piramide bestaat deze uit een Gaussisch filter en een herbemonsteringsstap. Met de Gaussische piramide beschikt men over 9 laag-doorlaat gefilterde versies van het oorspronkelijke beeld.

Het scala aan piramidesoorten binnen de beeldverwerking is groot. Dit wordt zowel veroorzaakt door de aard van de informatie in de piramide als de wijze waarop de resolutiereductie plaatsvindt. Andere voorbeelden van piramides zijn die van de binaire piramide en de Laplace piramide. Deze laatste bestaat uit banddoorlaat gefilterde versies van het oorspronkelijk beeld.

In de literatuur zijn twee soorten van gebruik van de piramide te vinden. Bij het eerste type van gebruik wordt de structuur gebruikt om de variantie in de data te reduceren. Dat wil zeggen dat de variantie in de data afneemt met de hoogte in de piramide. In de praktijk blijkt dat deze wijze van resolutiereductie meestal de eenvoudigste is. Het tweede type van gebruik is gebaseerd op het idee dat hogere niveaus in de piramide corresponderen met een beschrijving van de beeldaspecten op een grover schalingsniveau. Dit type van gebruik wordt geacht meer in overeenkomst te zijn met het concept van piramidale datastructuren. Echter, het ontwerp van een resolutiereductie operator is voor dit type gebruik niet triviaal. Dit geldt zeker ook voor de textuurpiramide, die de onderliggende textuur beschrijft op meerdere niveaus van schaling. Het ontwerp van een resolutiereductie operator voor de textuurpiramide staat daarom centraal binnen dit proefschrift.

Zoals we reeds hebben opgemerkt zijn er een aantal textuuroperators, welke de mogelijkheid bieden om het schalingsniveau in te stellen waarop de textuur wordt bestudeerd. Echter deze mogelijkheid garandeert nog niet dat de initialisatie van de textuurpiramide triviaal is. Met name randeffecten als gevolg van de verhouding tussen de vensterafmetingen en de afmetingen van de te initialiseren laag in de piramide gaan steeds meer overheersen naarmate men hoger in de piramide komt. Feitelijk is de optimale wijze van resolutiereductie, die waarbij de textuureigenschappen opborrelen in de piramide, en wel zodanig dat naarmate de laag hoger in de piramide is, de beschrijving een grover schalingsaspect van de textuur belicht. Een dergelijke methode dient gebaseerd te zijn op een model, waarbij het schalingsaspect is verdisconteerd. Een dergelijk model zou gebaseerd kunnen zijn op de fractal theorie. Centraal binnen deze theorie is de beschrijving van fenomenen op verschillende niveaus van schaling.

Het idee van de fractaltheorie wordt vaak geïllustreerd met de vraag: "Hoe lang is de kustlijn van Engeland?" Het blijkt dat op deze vraag geen consistent antwoord is te geven. Immers naarmate de kustlijn met een kleinere meetlat wordt gemeten, zal de nauwkeurigheid waarmee deze wordt gevolgd toenemen. Als gevolg hiervan zal de kustlengte toenemen met de nauwkeurigheid waarop men meet. Om uiteindelijk een consistent antwoord te kunnen geven op deze vraag, wordt de fractal dimensie geïntroduceerd. De fractal dimensie kan worden beschouwd als een maat welke de grilligheid van een fenomeen kwantificeert. Een eigenschap die dan ook overeenkomst vertoont met onze wens om textuur te kwantificeren.

De fractal dimensie maakt deel uit van een oneindig aantal dimensies, welke zijn gedefinieerd in het algemene dimensiemodel. Dit model heeft een parameter die wordt gebruikt als een wegingsmechanisme. Voor elke waarde van de parameter wordt een nieuwe dimensiemaat verkregen, welke een ander aspect benadrukt. De theorie beveelt dan ook aan om naast de fractal dimensie nog tenminste één andere dimensiemaat in beschouwing te nemen. Het is bijvoorbeeld denkbaar dat twee duidelijk verschillende signalen een gelijke fractal dimensie opleveren. Juist in die gevallen wordt het gebruik van een aanvullende dimensiemaat aanbevolen. Bij het ontwerp van een resolutiereductie operator dienen we daarom uit te gaan van dit wegingsmechanisme.

Uitgaande van de wens om een resolutiereductie operator te ontwerpen welke de textuur daadwerkelijk op verschillende schalingsniveaus beschrijft, dient de relatie met de fractaltheorie verzwakt te worden. De theorie schrijft een bepaald schalingsgedrag voor over een breed schalingsinterval. Juist bij de piramide wordt de schalingsruimte opgedeeld in kleine intervallen. Dit betekent dat we nu de fractalmaten zullen loslaten op slechts zeer kleine intervallen.

In dit proefschrift worden een aantal pogingen beschreven voor het ontwerp van een dergelijke operator. De uiteindelijke voorgestelde operator is gebaseerd op het idee dat de grijswaarde wordt opgevat als een massa. Door de massa te delen door de totale massa binnen het venster wordt de massadichtheid verkregen. De afgeleide textuurmaten zijn gebaseerd op het kwantificeren van de distributie van de massa dichtheden, waarbij gebruik wordt gemaakt van concepten uit de fractaltheorie.

Experimenten hebben aangetoond dat de methode goede resultaten oplevert met betrekking tot discrimineerbaarheid. Het bleek dat voor een aantal gevallen inderdaad twee dimensiematen nodig waren om te kunnen discrimineren. Echter ook waren er gevallen waarbij slechts één dimensiemaat voldoende was. Verder zijn er textuurparen gevonden, welke slechts op hogere niveaus in de piramide te discrimineren waren. Terwijl er tevens textuurparen waren die slechts op lagere niveaus in de piramide te discrimineren waren. Met deze resultaten is het nut van het gebruik van de piramide bij textuuranalyse aangetoond. Door de relatie met de fractaltheorie te verzwakken heeft de methode in toepasbaarheid gewonnen. Voorts bleek uit de experimenten dat slechts een beperkte verwerkingstijd nodig was.

Acknowledgements

To carry out a Ph.D. study is a very unsocial task. The more suprising it therefore is to find so many people prepared to help you to fulfil this task. It is thus a pleasure to mention here those people who were of particular importance during this work and to express my gratitude for their support.

First I would like to mention prof.dr.ir. E. Backer and dr.ir. J.J. Gerbrands. They gave me the opportunity to carry out this study and the freedom to explore the subject. With dr.ir. J.C.A. van der Lubbe, I had many discussions about several possible types of use of the fractal and chaos theory within the field of applications that are within focus of the Information Theory Group. At an early stage of my research, our common interests flowed into cooperative effort in which we involved a number of students. These students were vital for our sparring sessions behind the blackboard. I certainly will miss these sparring sessions.

Before experiments could be carried out, a number of images had to be digitized. As I am used to wanting what is impossible, I also experienced difficulties in finding an image acquisition facility that fully met my requirements. It was ing. R.J. Ekkers who found such a facility somewhere in the cellars of the Applied Physics building. A cell image from a biological application was obtained from ir. J.J.P. Starink. As this image shows an amazing similarity to the synthesized images (or vice versa), I still consider this image as the eye-opener for my research. The amorphous silicon image to be found in this thesis was obtained from dr.ir. R.C. van Oort and prof.dr. M. Kleefstra, when they asked me to study whether or not it was possible to model the structures with the fractal model. I found one of the students, J.P. Koorevaar, prepared to analyze these structures. Another contact I enjoyed was that with

prof.dr. H. Koppelaar. He was always interested in my work and some of his students carried out some of their tasks on my project. Of the many students that I have guided, I would like to mention ir. R.E.H. Kamphuis. He worked on the implementation of a fractal dimension estimator suggested in the literature.

During the writing stage of this thesis, the need for assistence grew. Difficulties I experienced with the LaTeX-package were smoothly solved by ir. R.J. van der Vleuten and ir. F. Odijk. For an increased quality of the plotting style of the Mathematica package, I found ir. H.J. Barnard immediately prepared to help me. As the writing had to be done in the English language, I needed professional help. I found Mrs. J.B. Zaat-Jones prepared to edit the text.

Leaving the Information Theory Group after having been there since 1985 -when I came there as a student- means that I leave a lot of colleagues with whom I had a pleasant time. I certainly will miss them and hope to have such pleasant professional relationships in my new job.

Not only did I made appeals to a lot of people, I also expected a great deal of patience of my friends. Our contact was minimized due to this study, but I hope it will become more regular again. Travelling by train, I also made new friends. They sometimes had to stand a stressed passenger who could only talk about the problems encountered in his research. Sometimes I tried to hide the problems and stress by completely focussing on one of my other interests such as, for instance, ham radio, cooking, the culture of the Nordic countries, and photography. And I bored them with all my discussions on these subjects.

Definitely the work would not have been finished -at least in this order- without the tremendous support of Dorine. She did not want me to include the well-known page, mentioning ”To ....”, but she certainly deserves such a page. She created an atmosphere at home in which I could fully concentrate on my study, which means that it, regretfully, too often dominated daily life. In the final stage, I even started to listen to operas where singers desperately try to express their emotions. She stood it!

Curriculum Vitae

Ernst Jan Eijlers was born in Woerden, the Netherlands, on April 10, 1963. In June 1982, he obtained his Atheneum B diploma. He continued his studies in the same year at the Delft University of Technology, Department of Electrical Engineering. In 1987, he obtained his M.Sc. degree, and his Master's project was carried out at the Information Theory Group. His thesis was entitled: "Pyramidal Structures for Image Understanding Systems". At the same group he started to work on his Ph.D. study, where the subject was a continuation of the work already carried out during his Master's project. In January 1989, he was appointed for three years as assistent professor. During his time at the Information Theory Group, he has supervised students on a variety of projects. Apart from his work, he is interested in ham radio (in particularly low-power transmission techniques and telegraphy), cooking, photography and the culture of the Nordic countries.