Structure Characterization Using Mathematical Morphology

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 20 januari 2004 om 13.00 uur

door

Cristian Luis LUENGO HENDRIKS

natuurkundig ingenieur
geboren te Leiden

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. L.J. van Vliet

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. L.J. van Vliet,	Technische Universiteit Delft, promotor
Prof. dr. ir. J. Biemond,	Technische Universiteit Delft
Prof. dr. S.M. Luthi,	Technische Universiteit Delft
Prof. dr. E. Pirard,	Université de Liège (Belgique)
Prof. dr. J.B.T.M. Roerdink,	Rijksuniversiteit Groningen
Dr. P. Soille,	EC JRC Space Applications Institute (Italy)
Dr. ir. R. van den Boomgaard,	Universiteit van Amsterdan
Prof. dr. I.T. Young,	Technische Universiteit Delft, reservelid

This work was partially supported by the Dutch Ministry of Economic Affairs, through their Innovation-Driven Research Programme (IOP Beeldverwerking), project number IBV98006.

This work was carried out in graduate school ASCI.
ASCI dissertation series number 96.

ISBN: 90-75691-10-6

Contents

Summary

This thesis deals with the application of mathematical morphology to images of some kind of structure, with the intention of characterizing (or describing) that structure. The emphasis is placed on measuring properties of the real-world scene, rather than measuring properties of the digital image. That is, we require that the measurement tools are sampling-invariant, or at least produce a sampling-related error that is as small as possible. Filters defined by mathematical morphology can be defined both in the continuous space and the sampled space, but will produce different results in both spaces. We term these differences "discretization errors". Many of the results presented in this thesis decrease the discretization errors of morphological filters.

The size distribution is the main tool used in this thesis to characterize structures. We estimate it using a granulometry, which is the projection of a morphological scale-space on the scale axis. This morphological scale-space is built with a sieve: an operation that is extensive (or anti-extensive), increasing and absorbing. The volume-weighted, cumulative size distribution of the objects in the image follows by normalization of the granulometry. Two variants of this granulometry receive the most attention: one based on isotropic, structural openings or closings, and one based on Rotation-Invariant Anisotropic (RIA) morphology. RIA openings and closings complement the isotropic ones, in that the latter remove objects based on their smallest diameter, whereas the former remove objects based on any of the other diameters (such as the length).

Isotropic structural openings and closings use a disk (or an n-ball in n-D) as structuring element. The RIA openings and closings we are interested in use line segments as structuring elements. These two shapes are extensively studied in this thesis, and we propose various improvements to the classical algorithms that decrease the discretization errors (that is, they improve the discrete approximation to the continuous operation). For any shape, interpolation directly reduces discretization errors by reducing the relative sampling error of that shape. In addition to that, for disks and balls we propose a small shift with respect to the sampling grid to further reduce discretization errors. For the line segment we propose an algorithm based on skews (with interpolation) of the image. Both these shapes can also be improved by using gray-value structuring elements.

The only way of completely avoiding discretization errors in mathematical morphology is using an alternative image representation. For one-dimensional images we propose to use a piece-wise polynomial representation, based on spline interpolation. Due to the continuous nature of this representation, discretization effects are no longer relevant.

We also study the selection of the morphological operation for the granulometry, and some useful pre-processing steps to prepare the image so that the estimated size distribution is more accurate. Among other things, we look at noise-reduction filters and their effect on the estimated granulometry.

On a somewhat different note, the Radon transform (also known as Hough transform) is studied. It detects parameterized shapes in an image, and can therefore also be used to construct a size distribution. The most important difference between the Radon transform and the granulometry is that the former is linear, whereas the latter is strongly non-linear. Both methods do not require any form of segmentation, although they can benefit from pre-processing.

We show how the Radon transform can be defined such that the resulting parameter response function is band-limited. This makes it possible to define a minimal sampling rate for this function, avoiding aliasing. The parameters can therefore be estimated with sub-pixel accuracy. Secondly, the accuracy and precision of the Radon transform for spheres is examined. In particular, we derive a theoretical approximation for the bias in the estimated radii, and propose a way to modify the transform to reduce this bias. Finally, a memory-efficient algorithm for the Radon transform is proposed.

Samenvatting

Dit proefschrift behandeld het toepassen van mathematische morfologie, op beelden van een structuur, met de bedoeling deze structuur te karakteriseren (of te beschrijven). De nadruk ligt op het meten van eigenschappen in de werkelijke wereld, in tegenstelling tot het meten van eigenschappen van het digitale beeld. Dat is, we eisen dat de meetinstrumenten bemonstering-invariant zijn, of tenminste een minimale bemonstering-gerelateerde fout maken. Filters gedefinieerd in de mathematische morfologie kunnen zowel in de continue als in de discrete ruimte worden gedefinieerd, maar zullen verschillende resultaten geven in beide ruimten. We noemen deze verschillen “discretisatiefouten”. Veel van de resultaten die in dit proefschrift gepresenteerd worden reduceren de discretisatiefouten van morfologische filters.

De grootte-distributie is het belangrijkste gereedschap dat in dit proefschrift gebruikt wordt om structuren te karakteriseren. We schatten deze distributie door middel van een granulometrie, wat de projectie is van een morfologische schaal-ruimte op de schaal-as. Deze morfologische schaal-ruimte wordt opgebouwd met een zeef: een operatie die extensief (of anti-extensief), monotoon stijgend en absorberend is. De volume-gewogen, cumulatieve grootte-distributie van de objecten in het beeld wordt verkregen door normalisatie van de granulometrie. Twee varianten van deze granulometrie krijgen de meeste aandacht: één gebaseerd op isotrope structurele openingen of sluitingen, en één gebaseerd op Rotatie-Invariante Anisotrope (RIA) morfologie. RIA openingen en sluitingen complementeren de isotrope, daar deze objecten selecteren op de kleinste diameter, waar RIA morfologie dat doet op een willekeurige andere diameter (zoals de lengte).

Isotrope structurele openingen en sluitingen gebruiken een schijf (of een *n*-bol in *n*-D) als structurerend element. De RIA openingen en sluitingen waar we geïnteresseerd in zijn gebruiken lijnsegmenten als structurerende elementen. Deze twee vormen worden uitgebreid bestudeerd in dit proefschrift. We stellen verschillende verbeteringen voor die de discretisatiefouten in de klassieke algoritmen verminderen (dat is, ze verbeteren

de discrete benadering van de continue operatie). Voor een willekeurige vorm zal interpolatie rechtstreeks de discretisatiefouten verminderen door de relatieve bemonsteringsfout van de vorm te verminderen. Daarnaast, voor schijven en bollen stellen we een kleine verschuiving voor ten opzichte van het raster, om verder de discretisatiefouten te verminderen. Voor het lijnsegment stellen we een algoritme voor dat gebaseerd is op afschuivingen (met interpolatie) van het beeld. Beide vormen kunnen ook verbeterd worden door het gebruik van grijswaarde structurerende elementen.

De enige manier om discretisatiefouten volledig te vermijden in mathematische morfologie is het gebruiken van een alternatieve representatie van het beeld. Voor één-dimensionale beelden stellen we een representatie voor, bestaand uit stuksgewijs interpolerende polynomen verkregen via splines. Omdat dit een continue representatie is, zijn discretisatiefouten niet meer relevant.

We bestuderen ook de selectie van de morfologische operatie voor de granulometrie, en sommige nuttige voorbewerking-stappen om het beeld zodanig te transformeren dat de geschatte grootte-distributie zuiverder is. Onder andere kijken we naar ruis-onderdrukkende filters en hun effect op de geschatte granulometrie.

We bestuderen verder de Radon transformatie (ook bekend als Hough transformatie). Deze transformatie detecteert geparametriseerde vormen in een beeld, en kan dus ook gebruikt worden om een grootte-distributie te schatten. Het belangrijkste verschil tussen de Radon transformatie en de granulometrie is dat de eerste lineair is, terwijl de laatste zeer niet-lineair is. Beide methoden behoeven geen enkele vorm van segmentatie, maar ze kunnen wel voordeel uit voorbewerking halen.

We laten zien hoe de Radon transformatie gedefinieerd dient te worden zodat de verkregen parameter-respons-functie band-gelimiteerd is. Dit maakt het mogelijk een minimale bemonstering-dichtheid te definiëren voor deze functie, en zo aliasing te voorkomen. De parameters kunnen daardoor met sub-pixel nauwkeurigheid geschat worden. Daarna wordt de zuiverheid en nauwkeurigheid van de Radon transformatie voor bolschillen onderzocht. We leiden een theoretische benadering af voor de afwijking van de geschatte diameters, en stellen een aanpassing voor om deze afwijking te verminderen. Als laatste stellen we een geheugen-efficiënt Radon algoritme voor.

Resumen

Esta tesis trata sobre la aplicación de Morfología Matemática a imágenes de algún tipo de estructura, con la intención de caracterizar (o describir) esa estructura. El énfasis lo ponemos en medir propiedades de la escena real, en vez de medir propiedades de la imagen digital. Esto es, requerimos que las herramientas de medir sean invariantes al muestreo, o al menos produzcan un error relacionado al muestreo que sea lo menor posible. Filtros definidos en la Morfología Matemática pueden ser definidos tanto en el espacio continuo como en el discreto, pero producen resultados diferentes en ambos espacios. A estas diferencias les llamamos “errores de discretización”. Muchos de los resultados presentados en esta tesis disminuyen los errores de discretización de los filtros morfológicos.

La distribución de tamaños es la principal herramienta usada en esta tesis para caracterizar estructuras. La estimamos usando una granulometría, que es la proyección de un espacio de escalas morfológico sobre el eje de escala. Este espacio de escalas morfológico se construye con una “coladera”: una operación que es extensiva (o anti-extensiva), creciente y absorbente. La distribución, cumulativa y pesada por volumen, de los tamaños de los objetos en la imagen se obtiene normalizando la granulometría. Dos variedades de esta granulometría reciben la mayor atención: una basada en aperturas o clausuras estructurales isotrópicas, y otra basada en morfología Anisotrópica Invariante a la Rotación (RIA). Aperturas y clausuras RIA complementan las isotrópicas, en que las últimas quitan objetos según el diámetro menor, mientras que las anteriores quitan objetos según otros diámetros (como la longitud).

Aperturas y clausuras estructurales isotrópicas usan un disco (o *n*-bola en *n*-D) como elemento estructural. Las aperturas y clausuras RIA en las que estamos interesados usan segmentos de línea como elementos estructurales. Estas dos formas son estudiadas extensivamente en esta tesis, y proponemos varias mejoras a los algoritmos clásicos que aminoran los errores de discretización (eso es, mejoran la aproximación discreta a la operación continua). Para cualquier forma, interpolación directamente reduce los errores de discretización, porque reduce el error de muestreo de esa forma. Adicionalmente, para discos y bolas proponemos una pequeña traslación con respecto al retículo de muestreo para reducir aun más los errores de discretización. Para el segmento de línea proponemos un algoritmo basado en sesgar (con interpolación) la imagen. Ambas formas también pueden ser mejorados usando elementos estructurales con niveles de gris.

La única forma de eludir completamente los errores de discretización en la Morfología Matemática es usando una representación alternativa de la imagen. Para imágenes unidimensionales proponemos usar una representación con segmentos de polinomio, basado en interpolación con splines. Como esto es una representación continua, los efectos de discretización ya no son relevantes.

También estudiamos la selección de operaciones morfológicas para la granulometría, y algunos pasos de pre-proceso útiles para preparar la imagen de tal forma que la distribución de tamaños estimada sea más exacta. Entre otras cosas, echamos una mirada a filtros que reducen el ruido y el efecto que tienen en la granulometría.

En un tono algo distinto, la transformada de Radon (también conocida como transformada de Hough) es estudiada. Detecta formas parametrizadas en una imagen, y por ello también puede ser usada para construir una distribución de tamaños. La diferencia más importante entre la transformada de Radon y la granulometría es que la anterior es linear, mientras que última es muy no linear. Ambos métodos no requieren ninguna forma de segmentación, aunque pueden beneficiarse de pre-proceso.

Enseñamos como la transformada de Radon se puede definir de forma que la función de respuesta a parámetros resultante sea limitada en banda. Esto hace posible definir una tasa de muestreo mínima para esta función, evitando aliasing. Los parámetros se pueden, por tanto, obtener con exactitud sub-pixel. Después examinamos la exactitud y precisión de la transformada de Radon para esferas. En particular, derivamos una aproximación teorética para el error sistemático en el radio estimado, y proponemos una manera de cambiar la transformación para reducir este error. Por último proponemos un algoritmo para la transformada de Radon eficiente en memoria.

Research is what I'm doing when I don't know what I'm doing
(Wernher von Braun)

Introduction

In this thesis, methods from mathematical morphology will be described, with the purpose of characterizing structures. Three things need to be clarified: what mathematical morphology is, what structures are, and how methods from mathematical morphology can be applied to characterize these structures.

Mathematical morphology is a collection of algorithmic tools that can be executed by a digital computer and, when applied to an image, yield a transformed image. Transforming images is referred to as *image processing*.[1] The goal of applying such algorithms to images might be improving the appearance of these images, creating art, performing measurements, or understanding what is imaged. In these last two cases we speak of *image analysis*.[2] Mathematical morphology was born in the mid 1960's from work by Georges Matheron and Jean Serra. At that time they heavily stressed the mathematical formalisms (probably because computers took a long time to compute the complex transforms they were describing, and they recognized the importance of a strong mathematical base). Many authors since have extended this set of tools, mostly working on the mathematical base (definitions, propositions and theorems). Nonetheless, mathematical morphology is a relatively simple and powerful tool to solve a wide variety of problems in image processing and analysis.

We might distinguish objects, *structure* and texture. In principle, these are the same phenomena but at different scales: individual objects, when repeated in either a random or a predictable way, form structure. When the individual objects are not distinguishable, but the repeating pattern still is, one refers to it as texture. A good example is cloth.

[1] Mathematical morphology can also be applied to other things besides images, and there are many more methods next to mathematical morphology that are used in image processing.

[2] We use the term "image processing" for any operation that has an image as input and produces an image as output, and "image analysis" for any operation that uses an image as input and produces numbers, such as measurements, as output.

Because it is woven, a cloth has texture, that is, its surface is not smooth but is formed by a repetitive change in height. A closer look reveals the way the threads are intertwined in a repeating fashion. This is structure. An even closer look reveals the individual threads. These are the objects that form the structure. If you look even closer, you might discover new texture, structure and objects, down to the atomic level. It all depends on the scale at which you look at things...

This thesis focuses on how to apply methods from mathematical morphology to characterize structures.

Characterizing Structure

We stated before that structure is a pseudo-regular repetition of basic building blocks. Usually these building blocks are not exactly alike, and the repetition is not precisely uniform. Some structure can be described well by random processes such as the Poisson point process, meaning that the repetition is far from uniform. No matter how it was generated, though, the parameters governing the construction of the structure (which are also the parameters that might be used to characterize it) are not fixed values but random variables. Thus, to characterize a structure, one should characterize the random variables that generated it. These can be described by an average or median value, in more detail by adding a variance, or completely by specifying a *probability density function*. This is a function expressing the probability that the random variable assumes a specific value.

Two instances of a structure, generated with identical random variables as input parameters, will yield different estimates of the underlying probability density function. This is because of the finite extent of the structure. The larger the extent, the more the estimates will approximate the true distribution, and thus each other. If it is known how good the estimates are (usually with a confidence interval), it is possible to compare the estimated values for two instances. If the confidence intervals do not overlap, the two instances were not generated by the same random variables. However, if they do overlap it is not proven that the structures were the same: the measurements used to characterize the structure might not be discriminatory enough.

Furthermore, the estimates of these random variables are obtained through measurements, which are inherently noisy. This noise can be analyzed to determine how much it adds to the uncertainty of the estimate. The measurement error can be either *systematic* (i.e. the error is the same for all measurements: the measurement is biased) or *stochastic* (each time the measurement is repeated, the result is slightly different). The systematic error is inconsequential if the obtained estimates are compared to each other: they will all contain the same bias. However, when measurements are used as physical properties (for example to derive other properties from) or to compare with measurements made with a different method, systematic errors are important. In this case, both types of error must be analyzed.

In this thesis we will mainly use *size distributions* to characterize structures. The size of the elements that compose a structure is a random variable, and the size distribution is the probability density function for this random variable. A measured size distribution will deviate from the true size distribution because of two reasons:

- the error in the measurements (both systematic and stochastic), and
- the stochastic deviation of the sample these measurements were taken from (i.e. because of the limited extent of the instance).

From a size distribution it is possible to derive other statistical parameters related to size, such as the minimum, maximum, average, median, variance, etc.

Digital Images

The structures that we study are *continuous*.[3] To be able to use the computer as a measurement instrument, we will image them (through e.g. lenses), and digitize the resulting image. *Digitizing* requires *sampling* (recording the value of the image at regular intervals) and *quantization* (due to storing these values in a digital computer with a finite precision). The obtained data is called a *digital image*.

Quantization means that an arbitrary real value is being represented by a quantized value that can be stored in a finite number of bits. Two values that differ by a small amount might end up being identical in the computer's representation, depending on the result of the rounding operation. This quantization can be regarded as a source of noise, and will contribute as such to the signal-to-noise ratio (SNR) of the image. As we regard all images as noisy, this quantization does not need special consideration (we will discuss noise sources later on).

However, the sampling does require special consideration. If the image being sampled is band-limited (meaning the change in intensity from one location to the next is smooth enough), it can be sampled without loss of information. To do so, the distance between the samples must be small enough to follow the highest frequency in the image (i.e. the strongest change in intensity). That is, the sampling frequency should be higher than

[3] We use the term *continuous* to describe a non-sampled image, irrespective of its differentiability.

two times the highest image frequency. This condition is called the *sampling criterion* or the *Nyquist sampling theorem* [88, 108]. If this condition is met no information is lost, and it is possible to reconstruct the original image from the set of samples by a linear, shift-invariant (LSI) filter. Actually, this is only true for images with an infinite extent. For digital images this is never the case, so some information is lost, especially near the edges of the image. A sample is called a *pixel* (for picture element); in three-dimensional images a sample is often referred to as *voxel* (for volume element).

There are various noise sources that come into play in the imaging process [140]. First of all, if the physical signal we observe is derived from light, the quantum nature of light will influence this signal (*photon noise*). That is, the signal strength must be connected in some way to a photon count. The number of photons arriving at the light-sensitive device is governed by a Poisson distribution. This is especially important for small photon counts. The light-sensitive device itself will also introduce some error in this photon counting. For example, a CCD (charge-coupled device, common in modern cameras) will produce a *dark current*, a signal that is not related to the amount of light falling on it, but to the temperature that it is operating at. This is referred to as *thermal noise*, and is also governed by a Poisson distribution. When the signal is transferred from the CCD, a frequency-dependent noise is added, called *read-out noise*, which becomes more important with increasing read-out rate. The electronics used to process these signals (amplifier, analog to digital converter, etc.) also contribute noise. Other imaging techniques have similar noise sources. If the SNR is not high enough, noise will significantly influence any measurement performed on the image, sometimes in very complex ways.

Digital images of a structure can be obtained by any number of means. To process these images, it is not important how they were obtained as long as certain conditions are met (such as the sampling criterion). That is, a specific filter can be applied to the image no matter what it contains or where it came from; whether it produces a useful result or not is another matter. However, to interpret the image (as in measuring statistical properties), understanding the recording method is fundamental. For example, some techniques such as Confocal Laser Scanning Microscopy (CLSM) and Magnetic Resonance Imaging (MRI) blur the image more in one direction than in the others. Other techniques produce shadowing effects that make the analysis very difficult, such as Scanning Electron Microscopy (SEM). These images can all be filtered in the same way to, for example, reduce noise, but they must be handled differently when the goal is to obtain a size distribution. One way of accomplishing this is appropriate pre-processing. The goal here is to convert the acquired image into a function that bears a good resemblance to the actual structure. Examples are restoration of the CLSM image, and some shape-from-shading technique in case of a SEM image.

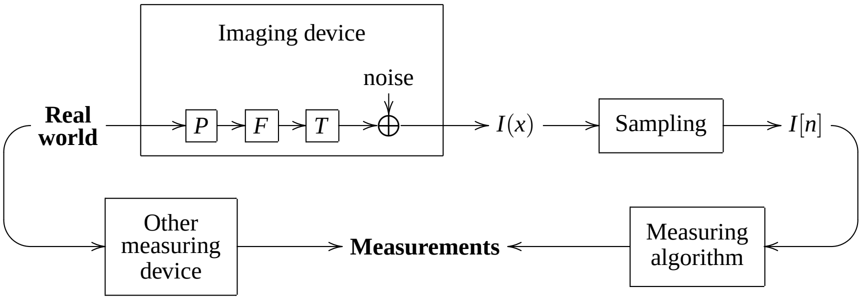


Figure 1: *From a real-world scene to a measurement through an imaging device that produces discrete images. The real-world scene is projected (P), filtered (F) and geometrically transformed (T) by the imaging device. The resulting image $I(x)$ is sampled, and some measurements are obtained from these samples. Our goal is to design the measurement algorithm in such a way that its results are comparable to measurements obtained directly from the physical world.*

For this thesis we used images obtained with many different techniques. Each of these produces band-limited images that were digitized satisfying the Nyquist sampling theorem. This is very important as it allows us to relate the measurements obtained from the digital image to the continuous image. Note that the continuous band-limited image is a filtered, geometrically transformed projection of the physical world. Therefore, understanding the imaging technique used is vital to relate the measurements obtained from the digital image to the real world (see Figure 1).

When using digital images, there is the possibility of analyzing the samples instead of the physical objects being imaged. One can find this in the literature very often. In this case, the results on two instances of the same physical object can be very different. In this thesis we try to avoid this, and propose algorithms for mathematical morphology that are as independent as possible of the sampling process. By using the sampled image in specific ways, it is possible to analyze the continuous image, and avoid characterizing the samples themselves instead. Since the relation between the continuous image and the real-world scene is known, the measurements obtained have a physical meaning.

All techniques studied in this thesis are applied to *gray-value images*. These are images where each sample is represented by a scalar value (e.g. intensity, height, density, concentration). We do not consider samples composed of more than one value (such as color images), since that introduces complications that are unnecessary for the applications presented. Nor do we use *binary images* (images that have been thresholded or converted otherwise into an image where for each pixel we only know whether it belongs to the foreground or the background). Such an image has lost a lot of information present in gray-value images, among others the exact location of object boundaries.

Invariance and Isotropy

One important property for a good characterizing measurement is *invariance* to a set of transformations. A measurement is invariant with respect to a certain transformation if its result is independent of the free parameters of the operation (note that this condition is necessary and sufficient). Examples are translation, rotation, scale, contrast, background, illumination, etc. A filter, on the other hand, is invariant to a certain transformation if the order in which the transformation and the filter are applied can be changed without affecting the result. That is, if filtering the image and rotating its output yields the same result as filtering the rotated image, it is said that the filter is invariant under rotation (rotation-invariant filter).

Additionally, a filter or measurement can be said to be *compatible* under certain transformation if it must be changed in a pre-defined way to account for the transformation. For example, a filter is compatible under rotation if it yields the same result when the output is rotated or when both the input and the filter itself are rotated.

Mathematical morphology only considers invariance within one domain. This means that in the case of discrete data, the operation under which the measurement is invariant is discrete as well: only operations that map the sample values onto new locations of the sampling grid are considered (e.g. translation of the image by an integer multiple of the grid spacing). Throughout this thesis we consider only continuous-domain invariances, even though the data we use is discrete. That is, we are interested in measurements that are invariant under translations, rotations, scalings, etc. of the image *before* sampling. We will use the term *sampling-invariant* to refer to an operation that is independent of the chosen sampling grid. A sampling-invariant discrete measurement will produce the same result as its continuous counterpart, which means that, if the continuous measurement is rotation invariant, the discrete measurement will be too.

With linear filters, translation invariance and rotation invariance (in the continuous-domain sense) are easily accomplished: as long as both the filter kernel and the image are properly sampled, the convolution is invariant to the sampling grid [139]. With mathematical morphology things are not this simple. Morphological operators produce images that are not band-limited (they cannot be sampled, without loss, using a regular sampling grid).[4] Therefore the discrete implementation of a morphological operation cannot produce an image that represents the results on the continuous image. So even though an operation in the continuous domain is invariant to translation and rotation, the discrete version is not. A large part of this thesis deals with this specific problem.

An *isotropic* measure is rotation invariant. It produces the same result, no matter what the orientation of the objects in the scene is. It accomplishes this by weighing all directions equally. For example, one could measure the average distance of the surface of an American football to its center. Such an isotropic measure does not say anything about the eccentricity or ellipticity. In contrast, an *anisotropic* measure is one that favors one direction above the others. For example, the length of the projection of an object on the x-axis: an American football has a different size shadow depending on its orientation. However, using a measure that is anisotropic, it is also possible to construct a rotation-invariant measure. The key to this problem is aligning the isotropic measure with some (rotation-invariant) property of the object. For example, the largest distance between two points on the surface of an American football is both anisotropic and rotation-invariant.

A *scaling-invariant* measure is independent of the magnification of the image. This magnification is influenced by the projection in the imaging system as well as by the density of the sampling grid. As long as this magnification is known, a measure can be related to a physical quantity, thus being independent of the magnification. This, of course, breaks down if the change in magnification is such that relevant features of the scene are below the resolution of the imaging system (such that the structure becomes a texture), or larger than the image field-of-view (such that the structure becomes a small collection of objects).

Contrast invariance is invariance to the scaling of the intensity axis. Though it seems similar, it is unrelated to *background invariance* and *illumination invariance*, which are insensitivities to uneven or changing light sources. The background is produced by stray light, also called ambient light (additive light source); the illumination light source is multiplicative. Both types of light sources are usually not uniform, and must often be corrected for to avoid erroneous measurements.

[4] Using a non-regular grid such an image might be sampled and reconstructed without loss (using some pre-defined non-LSI filter). We will, however, not consider such a sampling scheme.

This Thesis

The main contribution of this thesis is the creation of a link between discrete and continuous mathematical morphology. That is, we propose discrete methods that, when applied to discrete images, approximate their continuous counterparts as applied to the image before sampling. The reason to develop such methods is that they yield measurements that relate to measurements of the physical world, as in Figure 1. An important, beneficial side-effect of such sampling-invariant measurements is that they typically have a smaller stochastic error because the contribution of the sampling to the error becomes negligible.

To develop these sampling-invariant discrete methods we need to explore changes to classical discrete mathematical morphology. This is done in Chapters 3, 4 and 5. Chapter 3 gives an overview of granulometries and then proceeds to define something that is not a granulometry in the strict sense, but approximates the continuous-domain isotropic granulometry much better than any discrete granulometry[5]. Chapter 4 improves on the basic operators of discrete mathematical morphology, so as to obtain operators that produce the same results as their continuous-domain counterparts[6]. We have only been able to implement this for one-dimensional images, though. The n-dimensional version is possible, but very complex to implement. Chapter 5 takes a closer look at line segments as structuring elements. Line segments are notoriously difficult to represent discretely, and this chapter examines various approaches, comparing their performance with the continuous-domain line segment[7].

To make this thesis self-contained to some extent, the first chapter introduces mathematical morphology. It introduces all the operators, as well as the notation, used in this thesis.

Chapter 2 introduces an alternative to the isotropic structuring element that is also rotation invariant[8]. The closing and opening in this framework were already described by Matheron [79] from a different point of view.

Chapter 6 examines solutions to some common problems encountered when applying the granulometry in practice: noise, the image border, the large diversity of shapes composing the structure, etc. These solutions involve pre-processing the input image and

[5] A large part of Chapter 3 has been submitted for publication [74].

[6] Chapter 4 has been adapted from [72].

[7] A portion of Chapter 5 has been published in [73].

[8] Chapter 2 has been published in a reduced form in [70].

adapting the operations used in the granulometry itself.

Chapter 7 presents various applications of the granulometry[9]. These constitute the goal of this thesis, illustrate the use of the techniques discussed in it, and prove the validity and usefulness of the proposed methods.

Finally, and on a different note, Chapter 8 is a paper about the Radon transform, yet to be published [67][10]. Although the Radon transform is unrelated to mathematical morphology, it can also be used to characterize structures: it detects a shape in an image, without the need for segmentation, yielding the parameters for each instance of that shape. In fact, the Radon transform has quite a lot in common with the morphological sieve; both transform an image by fitting a pre-defined shape family to it. The difference is the way the similarity of a shape with the image is computed: the Radon transform is a linear transformation, whereas the sieve is strongly non-linear. The method to extract relevant information from these two transforms is also different. Appendix A is a technical report, published in [69], that contains the derivation of an equation used in this paper.

[9] Chapter 7 contains results published or submitted for publication in [18, 71, 74]

[10] An earlier, shorter version of this paper has been published in [68].

The important thing is never to stop questioning
(Albert Einstein)

Chapter 1
Mathematical Morphology

This chapter reviews the basics of mathematical morphology. It is meant to make this thesis self-contained to some extent, and to introduce the notation used in the following chapters.

Over the years, different notations for mathematical morphology have emerged. Each one has its advantages and disadvantages. We use the monadic operator style as used by e.g. Soille [114], because most of the equations in this thesis are simpler in this notation than when using the dyadic operators used by authors such as Matheron [78] or Serra [103].

There are also two different definitions for the dilation, which differ in the mirroring of the structuring element. The definition used by e.g. Haralick [43] or Heijmans [44] is chosen such that the dilation and erosion form an adjunction [45]. This is interesting if one uses theoretical concepts such as complete lattices. The definition used by e.g. Serra and Soille, on the other hand, makes the dilation and erosion symmetric. Using this definition, these operations correspond directly to the local maximum and minimum filter, respectively. It also makes both of them a special case of the more general rank filter [117]. We feel that this latter definition fits better with our filtering approach.

History

Mathematical morphology was born in 1964 from work by Matheron and Serra, the former working for the French Geological Survey (BRGM) on the hydrodynamics of porous media, the latter at the French Steel Institute (IRSID) on the petrography of an iron ore body [80]. In that year, both authors published their first reports concerning some of the methods that now are part of mathematical morphology. Right away they started building on each other's results. In January of 1966, J.C. Klein and J. Serra finished the *Texture Analyser* [61] (TAS). It allowed the analysis of binary images using any structuring element based on straight lines along the three axes of the hexagonal grid. Later that year Matheron and Serra coined the term “mathematical morphology”, as well as other terms commonly used in the field now. The Centre de Morphologie Mathématique was created on April 20, 1968, by the Ecole de Mines de Paris at Fontainebleau, and manned by Matheron and Serra. This center is considered the birthplace of mathematical morphology, and to this day is an important center for research in the field.

In 1964 Matheron had rediscovered *Minkowski algebra*. Minkowski defined the set addition (dilation) in 1903, which Hadwiger studied in detail in a work published in 1957 [42]. This work also introduces the set subtraction (erosion), as well as their combinations (opening and closing). Hadwiger’s field is the Integral Geometry, which spawned a field of study called *stereology*. Stereology is therefore closely related to mathematical morphology, although both fields come with a very different way of thinking. In stereology, a statistic is computed from the image[1] (for example, the number of points of a regular grid that fall on a particular phase of the structure, or the number of phase boundaries crossed by a line), resulting in a number that characterizes the structure in some way (volume fraction and surface area, respectively, for the two examples given above). What was new about Matheron and Serra’s way of thinking was the transformation of the image into a new one. The difference between the images is limited to structures with a specific shape and size. These fields have grown closer together, but are still considered separate fields by many.

In the rest of this thesis, we will use the term “morphology” for “mathematical morphology”. Statements in the rest of this chapter that are not followed by a reference implicitly refer to Soille [114].

Basics

When morphology was first introduced, it was based on set theory. In practice, this means it was limited to binary images (pixels having one of only two values). Nakagawa and Rosenfeld [86], in 1978, were the first to link the binary dilation and erosion operations to the maximum and minimum filters applied to gray-value images. Sternberg [121] introduced the notion of umbra around 1978, which made it possible to apply all the mathematics developed for binary images to gray-value images. We will consider

[1] George Leclerc Buffon presented his needle problem to the Royal Academy of Sciences in Paris, 1777 [84]. He showed that a needle, falling randomly on a grid of lines, intersects each line with a probability directly proportional to the length of the needle, without any further assumptions. This is one of the basic principles of stereology.

only gray-value morphology because it is more general and therefore can be applied to binary images producing the same results that would be produced by set operations. The morphological operations we will review in the next sections use *structuring elements* as probes to analyze the image. We believe that flat structuring elements (i.e. defined by a set) make morphology easier to explain. Nonetheless, it is also possible to define structuring elements as non-flat functions. This is an even more general case, which we will also exploit in this thesis. By defining the gray-value structuring element to have only values of 0 and $-\infty$, one mimics a structuring element defined by a set. We will explain the morphological operations using flat structuring elements, and then give the modified definitions for gray-value structuring elements.

In this chapter, the function $f:\mathbb{R}^n \to \mathbb{R}$ is a continuous, gray-value image (defined on an infinite domain). We will not consider the operations as applied to digitized images yet. The set $B \subset \mathbb{R}^n$ will represent the structuring element.

$$\check{B} = \{-x \mid x \in B\} \tag{1.1}$$

is the mirrored set,

$$T_t B = \{x+t \mid x \in B\} \tag{1.2}$$

is the translated set, and

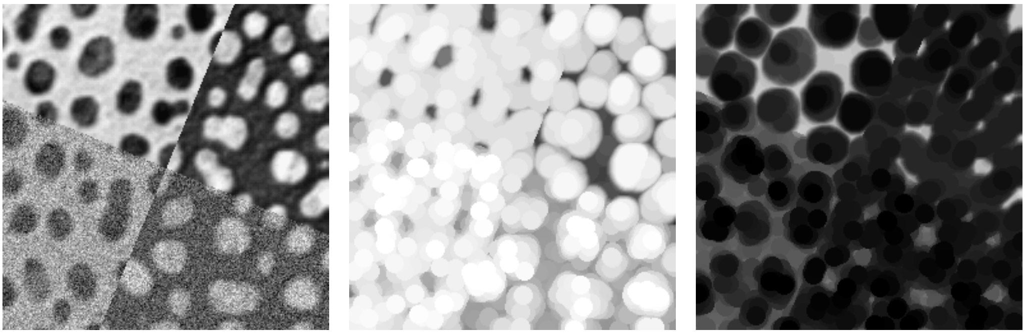
$$S_s B = \{sx \mid x \in B\} \tag{1.3}$$

is the scaled set. The operators T_t and S_s are also applied to functions. Finally, $\bigwedge$ and $\bigvee$ represent the infimum and supremum respectively. We consider all sets closed,[2] and we only concern ourselves with smooth, real images. Therefore, we can use the minimum value as the infimum, and the maximum value as the supremum. We use these terms interchangeably.

1.1 Dilation and Erosion

The two basic morphological operators are the *dilation*, denoted by δ, and the *erosion*, denoted by ε. They are defined respectively as the maximum and minimum value of a

[2] The difference between open and closed sets is not relevant when processing continuous, band-limited functions.



a: *Input* b: *Dilation* c: *Erosion*

Figure 1.1: *Demonstration of the dilation and erosion on a test image. The structuring element used was a disk with a radius of* 7.5 *pixels, which is the amount by which the dark objects have shrunk and expanded, respectively (image size is 256 by 256 pixels).*

function f in a neighborhood defined by the structuring element B,

$$[\delta_B(f)](x) = \bigvee_{b \in B} f(x+b) \tag{1.4}$$

$$[\varepsilon_B(f)](x) = \bigwedge_{b \in B} f(x+b) \quad . \tag{1.5}$$

The dilation expands the light objects in the image, and shrinks the dark ones. The erosion has the opposite effect. Figure 1.1 shows this behavior on a test image. These two operators are related by a simple and obvious property, *duality*,

$$\delta(-f) = -\varepsilon(f) \quad . \tag{1.6}$$

This means that applying one to the ‘foreground’ produces the same result as applying the other to the ‘background’. That is, the dilation expands the light objects in the same way as the erosion expands the dark ones.

Properties

The dilation and the erosion have some interesting properties that are rather trivial to prove (for these proofs we refer to Soille [114]). For example, both operators preserve the ordering relation (*increasingness*),

$$g \geq f \Rightarrow \begin{cases} \delta(g) \geq \delta(f) \\ \varepsilon(g) \geq \varepsilon(f) \quad . \end{cases} \tag{1.7}$$

If the structuring element contains the origin, there is also the *extensivity* of the dilation and the *anti-extensivity* of the erosion,

$$0 \in B \Rightarrow \delta_B(f) \geq f \geq \varepsilon_B(f) \quad . \tag{1.8}$$

This implies that, as long as the origin is included in the structuring element, the dilation never lowers the value of the image at any point. The light objects in the image are enlarged. If the origin is not included in *B*, the objects are enlarged and translated:

$$y \in B \Rightarrow [\delta_B(f)](x) \geq f(x+y) \geq [\varepsilon_B(f)](x) \quad . \tag{1.9}$$

Obviously, the result of the dilation at a point *x* is only affected by a region around *x* in input image. This region is defined by the structuring element *B*. This property is called *local knowledge*. The same holds for the erosion.

Also, both the dilation and the erosion are invariant to translation and contrast change, and compatible under scaling:

$$\delta_B T_t = T_t \delta_B \tag{1.10}$$

$$\delta_B(af) = a\delta_B(f) \tag{1.11}$$

$$\delta_{S_s B} S_s = S_s \delta_B \quad . \tag{1.12}$$

But there are also some less obvious properties. For example, the dilation and the erosion are defined by their *distributivity,*

$$\delta(f \vee g) = \delta(f) \vee \delta(g) \tag{1.13a}$$

$$\varepsilon(f \wedge g) = \varepsilon(f) \wedge \varepsilon(g) \quad . \tag{1.13b}$$

The dilation distributes with the supremum, and the erosion with the infimum. This is a very important property when dealing with sets, but it is not used much when working with gray-value images.

Also, the structuring element can be decomposed because of the *associativity* of the operator,

$$\delta_{\delta_B(C)} = \delta_B \delta_C \tag{1.14a}$$

$$\varepsilon_{\delta_B(C)} = \varepsilon_B \varepsilon_C \quad . \tag{1.14b}$$

This property looks a little curious in this notation, but using Minkowski algebra it looks familiar. $\delta_B(C)$ is then written as $C \oplus B$; $\varepsilon_B(C)$ as $C \ominus B$. The associativity property now reads

$$f \oplus (C \oplus B) = (f \oplus C) \oplus B \tag{1.15a}$$

$$f \ominus (C \oplus B) = (f \ominus C) \ominus B \quad . \tag{1.15b}$$

This means that if a structuring element can be written as the Minkowski addition of two or more simpler structuring elements, the dilation (or erosion) operation becomes much simpler to implement and cheaper to compute. We will come back to this later in this section.

Furthermore, it is possible to change the order of the operands and operations (*commutativity*):

$$\delta_B \delta_C = \delta_C \delta_B \quad . \tag{1.16}$$

Gray-Value Structuring Elements

Using gray-value structuring elements, the dilation and the erosion bear a great resemblance to the convolution integral. Let us define the structuring element as a function $b : \mathbb{R}^n \to \mathbb{R}$. The dilation is written as

$$[\delta_b(f)](x) = \bigvee_{y \in \mathbb{R}^n} f(x+y) + b(y) \quad , \tag{1.17}$$

and the erosion as

$$[\varepsilon_b(f)](x) = \bigwedge_{y \in \mathbb{R}^n} f(x+y) - b(y) \quad . \tag{1.18}$$

Comparing these equations to the convolution integral we observe that the integral is substituted by a supremum (or infimum) and the multiplication is substituted by an addition (or subtraction). Another difference is that the structuring element is not mirrored, whereas the convolution kernel is. As mentioned earlier, some authors mirror the structuring element for the dilation, but not for the erosion. Also note that, when defining

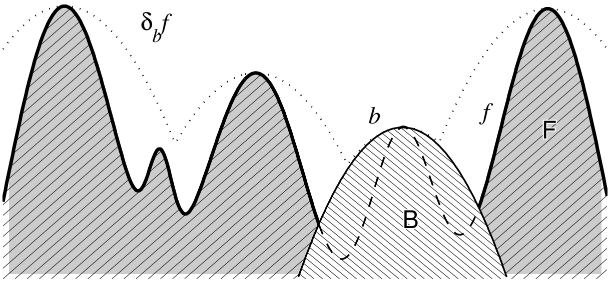


Figure 1.2: *The notion of umbra serves to apply set morphology to functions. Dilating F, the umbra of f, with B, the umbra of b, yields the umbra of the result obtained when dilating the function f with b. The function b is a parabola, with the origin at its maximum.*

b only with values 0 and $-\infty$, one obtains the operators as defined previously for flat structuring elements, with

$$B = \{x \mid b(x) = 0\} \quad . \tag{1.19}$$

An intuitive understanding of this operation can be obtained through the notion of *umbra*. The function f is converted into a set F by adding a new dimension,[3]

$$F = \{(x,y) \mid y \leq \lambda f(x)\} \quad , \tag{1.20}$$

where λ is a shape parameter [51] (this value is undefined by the problem; there is usually no relation between the intensity and the spatial axes). The structuring element b is treated in the same way. As can be seen in Figure 1.2, the set dilation $\delta_B(F)$ yields the umbra of the dilated function $\delta_b(f)$. When using flat structuring elements, the dilation translates the slopes, that is, the umbra is extended only horizontally. Non-flat structuring elements also extend the umbra in the newly defined direction, the y-axis, thereby introducing new gray-values in the function f. Also, flat structuring elements are insensitive to the shape parameter λ.

For all points y where $b(y) < -I + b(0)$ (with $I = \bigvee f - \bigwedge f$, the total gray-value range in the input image), the sum $f(x+y) + b(y)$ can never influence the result of the supremum. Therefore, only the points y for which $b(y) - b(0) \geq -I$ need to be considered when

[3] The function f represents a surface, the set F is the solid body whose surface is represented by f. If f is an n-dimensional function, F has $n+1$ dimensions.

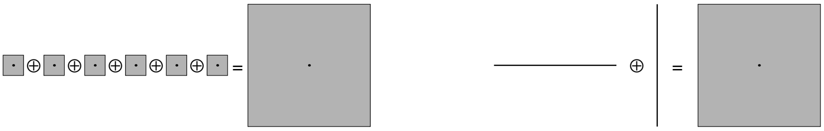


Figure 1.3: Left: *Successive dilation with a small square structuring element produces the same result as dilation with a large one.* Right: *Two one-dimensional dilations in orthogonal directions produce the same result as one dilation with a large square structuring element.*

computing the dilation. This limits the support of the structuring element used.

Using this definition, the operations are no longer contrast invariant. It is necessary to change the structuring element accordingly: it is compatible under scaling of the intensity-axis,

$$\delta_{ab}(af) = a\delta_b(f) \quad . \tag{1.21}$$

Decomposition of the Structuring Element

As stated before, a structuring element can be decomposed into smaller structuring elements, which makes the computation of the dilation and the erosion faster. Equations (1.14a) and (1.14b) show that, if a structuring element B can be written as the dilation of B_1 with B_2, then one might as well use two dilations with the composing sets B_1 and B_2, instead of the one with B. In many cases this is computationally more efficient.

For example, instead of applying a dilation with a square of size n units, one can apply n iterations of a dilation with a square of 1 unit in size (see Figure 1.3). Moreover, this same square can be decomposed into two orthogonal line segments. A dilation or an erosion with a discrete line segment can be accomplished with only three comparisons per image pixel, irrespective of the length of the line segment [46, 118], thus making a dilation with a square a computationally cheap operation.

It is also possible to approximate a disk by dilation with a series of line segments. The more segments used, the better the disk is approximated. For example, three line segments form a hexagonal structuring element, and four an octagonal one.

1.2 Closing and Opening

When combining the dilation and the erosion one obtains the *closing*, denoted by ϕ, and the *opening*, denoted by γ,

$$\phi_B = \varepsilon_{\check{B}} \delta_B \tag{1.22}$$

$$\gamma_B = \delta_{\check{B}} \varepsilon_B \quad . \tag{1.23}$$

The dilation can diminish the number of local minima, which cannot be restored by a subsequent erosion. Thus, the closing produces a simplification of the image. Note the mirroring of the structuring element in the second operation. It is required to give the closing and opening some of their defining properties (idempotence and (anti-)extensivity). It also causes the operation to be independent of the placing of the structuring element with respect to the origin. When looking at an image as composed of objects, a closing removes dark objects that are smaller than the structuring element used (i.e. the ones in which the structuring element does not fit). See Figure 1.4 for an example. In this light, the closing at a point x can be written as the maximum value of the image f over the structuring element B, shifted in such a way that the obtained value is minimal. This shift is constrained by the requirement that the structuring element hit the point x under consideration. In other words,

$$[\phi_B(f)](x) = [\varepsilon_{\check{B}} \delta_B(f)](x) = \bigwedge_{z \in \check{B}} \bigvee_{y \in T_z B} f(x+y) \quad . \tag{1.24}$$

Note that, because $z \in \check{B}$, $T_z B$ always contains the origin. Of course, the same notation can be used for the opening. A closing or opening with a gray-value structuring element can be constructed with a dilation and erosion using the gray-value structuring element.

Properties

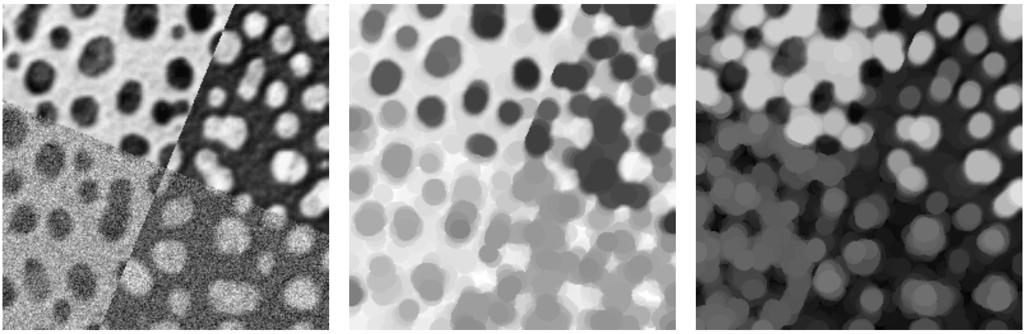
The closing and the opening share quite a few properties with the dilation and the erosion. First of all, the two operators are dual,

$$\phi(-f) = -\gamma(f) \quad , \tag{1.25}$$

because the operators from which they are constructed are dual as well.

Like the dilation and the erosion, these operators are increasing,

$$g \geq f \Rightarrow \begin{cases} \phi(g) \geq \phi(f) \\ \gamma(g) \geq \gamma(f) \quad . \end{cases} \tag{1.26}$$



a: *Input* b: *Closing* c: *Opening*

Figure 1.4: *Demonstration of the closing and opening on a test image. The structuring element used was a disk with a diameter of* 15 *pixels, which is the minimal size of the remaining dark and light features, respectively.*

As we saw earlier, increasingness can be interpreted as a preservation of the ordering relation between images.

The closing and the opening are also extensive and anti-extensive, respectively:

$$\phi(f) \geq f \geq \gamma(f) \quad . \tag{1.27}$$

However, unlike the dilation and the erosion, this property holds for all closings and openings, irrespective of the structuring element used. Because the structuring element is mirrored, the origin is always included in the region from which the maximum (or minimum) is selected. Adding this relation to Equation (1.8) results in an interesting *ordering relation*:

$$0 \in B \Rightarrow \delta_B(f) \geq \phi_B(f) \geq f \geq \gamma_B(f) \geq \varepsilon_B(f) \quad . \tag{1.28}$$

The translation and contrast invariances of the dilation and the erosion, as well as the compatability under scaling, are also inherited by the closing and the opening.

$$\phi_B T_x = T_x \phi_B \tag{1.29}$$

$$\phi_B(af) = a\phi_B(f) \tag{1.30}$$

$$\phi_{S_s B} S_s = S_s \phi_B \quad . \tag{1.31}$$

However, there are some other properties not shared with the dilation and the erosion. A very important property is *idempotence*,

$$\phi_B \phi_B = \phi_B \tag{1.32a}$$

$$\gamma_B \gamma_B = \gamma_B \quad , \tag{1.32b}$$

which means that applying a closing to an image more than once with the same structuring element is futile.

Finally, the *absorption* property,

$$B = \gamma_C B \Rightarrow \begin{cases} \phi_C \phi_B = \phi_B = \phi_B \phi_C \\ \gamma_C \gamma_B = \gamma_B = \gamma_B \gamma_C \quad , \end{cases} \tag{1.33}$$

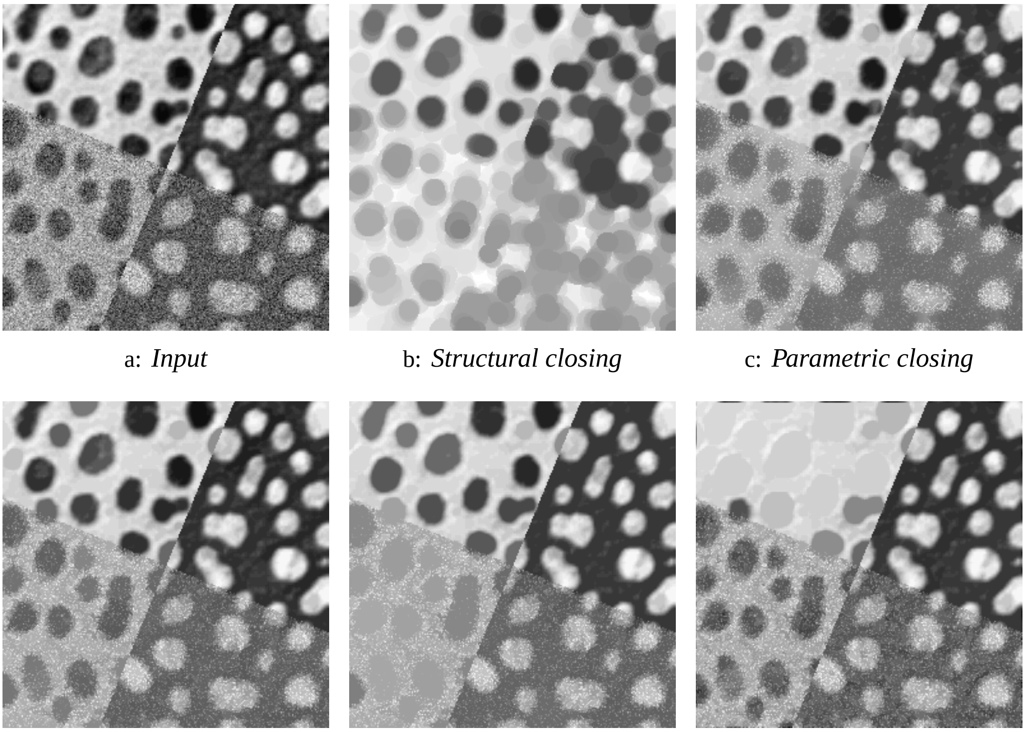
is satisfied only if B is open with respect to C, that is, if applying the opening with set C to the set B, the set B does not change. This condition usually holds when B and C are scaled versions of the same convex set (e.g. a rectangle or a disk). This property will prove useful in Chapter 3. Note that idempotence is a special case of absorption (the case $B = C$, since $\gamma_B(B) = B$). Also note that the closing generally does not commute with itself. It is only commutative when one of the structuring elements is open with respect to the other one.

Algebraic Closing and Opening

Of all the properties defined above, extensivity, increasingness and idempotence define the closing operation. This means that any operation that satisfies these three properties is called a closing (the same is true for the opening, substituting anti-extensivity for extensivity). However, there are operations that satisfy these properties but cannot be written in terms of a dilation followed by an erosion.

To distinguish these two types of closings, we call *structural* or *morphological closing* the closing constructed with a dilation and an erosion. If a closing is not a morphological closing, it is called an *algebraic closing*, and can be written as the infimum of a family of morphological closings [79],

$$\Psi(f) = \bigwedge_i \phi_{B_i}(f) \quad . \tag{1.34}$$



a: *Input*

b: *Structural closing*

c: *Parametric closing*

d: *Area closing*

e: *Closing by reconstruction*

f: *Flooding*

Figure 1.5: *Demonstration of the various algebraic closings.* a: *Input image.* b: *The structural closing (same images as in Figure 1.4).* c: *The rank-min or parametric closing; the structuring element is a disk with diameter 15 pixels, with 20% of the pixels ignored.* d: *The area closing, using an area of 177 pixels, which is the area of a disk with a diameter of 15 pixels.* e: *The closing by reconstruction, again using a disk with a diameter of 15 pixels.* f: *The flooding, implemented with the UpperEnvelope algorithm (see Section 6.3), with a connectivity of* 4*, a maximum depth of* 40 *and no limit on the size (these parameters handle the merging of the regions).*

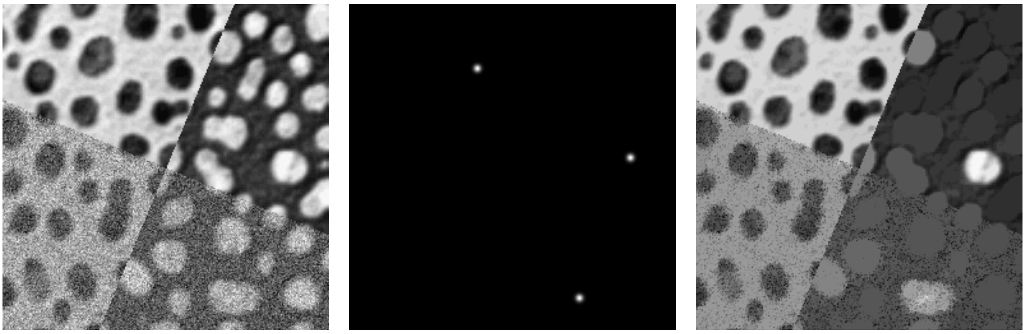
Among algebraic closings are (see an example of each in Figure 1.5):

- *Parametric closings*: much like a structural closing, a feature is removed if the structuring element does not fit in it. The difference is that here a specified number of mismatched points are ignored. That is, the fitting condition is relaxed somewhat. It can be written as the infimum of all closings with a structuring element of which the specified number of points has been removed. It is also called *rank-min closing* because it is usually implemented with a *rank filter* followed by an erosion [100]. The rank filter returns the n^{th} largest value of a neighborhood of the input image, n being the rank of the filter [117].
- *Attribute closings*: features are selected by some attribute, such as size. It can be interpreted as the infimum of all closings with a structuring element with that attribute (of which there might be an infinite number, of course). The most frequently used attribute closing is the *area closing* [1, 29], which removes dark features with an area (number of pixels) smaller than some threshold. Note that it is required that the attribute be increasing for this to be a closing [22, 23].
- *Closing by reconstruction*: implemented by a dilation followed by an erosion by reconstruction (see Section 1.3). It is actually an attribute closing, the attribute being the ability of the structuring element to fit the feature. Note that it removes the same 'holes' as the structural closing with the same structuring element, but it does not change the shape of the holes it does not remove.
- *Floodings*: based on the watershed transform (see Section 1.3), the image is transformed by filling up the minima (catchment basins in the watershed transform) [83]. The rate of filling and the stopping criterion define the result. This transform is also referred to as *upper leveling*. Its dual operation is the *lower leveling*. The flooding is a form of closing by reconstruction [82], but the seed image used in the dilation by reconstruction is not defined by an erosion. Section 6.3 gives the details of the UpperEnvelope algorithm, which implements a flooding.

The dual transform of the algebraic closing is the algebraic opening. For each of the examples above, an opening counterpart exists.

1.3 Other Morphological Tools

Besides dilations and erosions with structuring elements, and their combinations, there are many other morphological tools available to the image analyst. Here we summarize some we will be using later.



a: *Mask* b: *Seed* c: *Result*

Figure 1.6: *Demonstration of the reconstruction by dilation.*

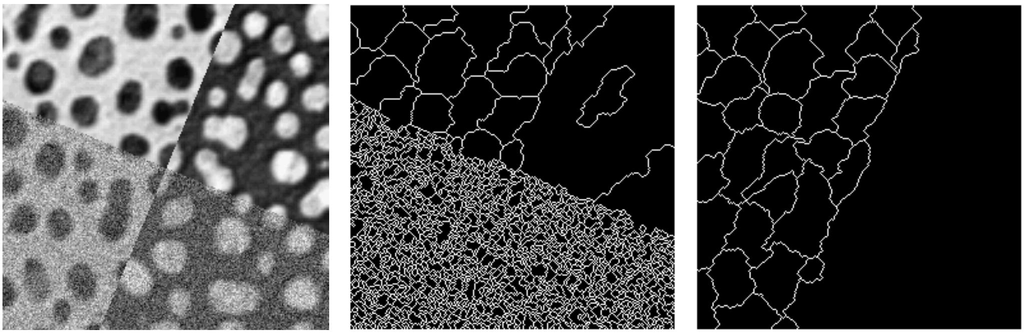
Reconstruction by Dilation

Reconstruction by dilation, *infimum reconstruction* or *propagation* is an iterative constrained dilation [39] applied to an image, called seed image. The constraint is given by another image, called mask image. The seed image is dilated in such a way that the result at any point is never larger than the mask. The procedure is iterated until convergence (i.e. no more changes are possible). Figure 1.6 shows an example. A constrained dilation is also called *geodesic dilation*.

Such an iterative dilation can be accomplished through various algorithms [137], but the most elementary implementation uses dilations with a very small structuring element, and after each dilation the point-wise minimum of the result with the mask is taken. The algorithm terminates when an iteration introduces no changes in the result.

The Watershed Transform

The *watershed transform* [31] is an operation (only defined for gray-value images) that can be used for *segmentation* (dividing the image into regions of similar properties) [16]. Like the reconstruction by dilation and the skeletonization, the watershed transform is an iterative algorithm. It can be visualized as a gradual flooding of the surface defined by the image (see Figure 6.10 on page 119). The lowest points are filled first, and are called the catchment basins. As the water level rises, the catchment basins grow in size. At the points where two of these basins meet, dams are raised, to avoid merging. These dams sit at the watershed lines. When the whole image has been flooded, the dams form



a: *Input* b: *Watershed* c: *Result on smoothed input*

Figure 1.7: *Demonstration of the watershed transform. In the middle is the result of the watershed (this includes some simple region merging). Note the poor performance, especially in the noisy region. By applying a simple noise-reduction filter, the median filter on a 7-by-7 window in this case, the performance increases drastically.*

a network of connected segments that separate the different regions in the image. See Figure 1.7 for an example.

For this to be a correct segmentation, some pre-processing of the image is essential. Smoothing is required to reduce the number of local minima, each of which forms a catchment basin. Figure 1.7 shows the importance of this. Depending on the contents of the image, other operations like edge detection might prove useful pre-processing steps.

The watershed transform is usually followed by a region merging algorithm. This merging step can be performed simultaneously with the transform by building the watersheds selectively. Section 6.3 discusses a method to accomplish this.

Drawing on my fine command of language, I said nothing
(Mark Twain)

Chapter 2

Rotation-Invariant Anisotropic Morphology

When analyzing images without a preferred orientation, or with an unknown one (as is the case, for example, of an image acquired after placing a sample randomly under a microscope), it is desirable to use rotation-invariant operations. As stated in the introduction, a rotation-invariant operation yields an output that is independent of the orientation of the scene with respect to the sampling grid. We also stated that it is possible to construct a rotation-invariant operator with anisotropic operators. This can be accomplished in one of two ways:

- Using a single anisotropic operator whose orientation depends on the (local) image content (e.g. estimate the orientation of the structure under study and create an operator that takes this orientation into account). This can be done for the image as a whole or on a point-by-point basis. In the latter case this is referred to as *steered filtering* or *adaptive filtering*. Adaptive filtering also allows changing the shape and size of the filter, not only its orientation. See [10] for an example of adaptive filtering.
- Using an infinite amount of these anisotropic operators, and selecting a result that satisfies certain properties, e.g. the maximum, minimum or median value. The selection can, again, be done on a point-by-point basis. This is the approach we follow in this chapter.

In this chapter we develop a new morphological framework that we will call RIA *morphology*. RIA stands for “Rotation-Invariant Anisotropic”. Section 2.1 introduces the RIA counterpart of the dilation and erosion operators. These are not a dilation and erosion in the strict morphological sense, because they do not commute with the supremum and infimum, respectively. Therefore, we call them *sedimentation* and *wear*.[1] Section 2.2 introduces the RIA closing and opening. These do satisfy all three properties required for the algebraic closing and opening, but are not constructed using the sedimentation and wear operators. The RIA closing and opening were already defined by Matheron [79], but from a different point of view. These operators also fit within the larger theoretical framework of group morphology [96, 97, 98].

RIA morphology is based on an isotropic structuring element in a lower-dimensional space than the image space. In the image space, therefore, it is anisotropic. By giving this structuring element rotational freedom (i.e. aligning it with some feature of the image), it becomes rotation invariant.

This chapter is limited to RIA morphology in the continuous domain. Actual implementation details are deferred to Section 7.3, where the RIA opening is used, and to Chapter 5, were implementation details of linear structuring elements are discussed.

Notation

As in Chapter 1, the function $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous, gray-value image.

The set D is a flat, isotropic structuring element of radius r,

$$D = \{x \mid x \in \mathbb{R}^n, \|x\| \leq r\} \quad , \tag{2.1}$$

and L is a flat isotropic structuring element with less dimensions than f,

$$L = \{x \mid x \in \mathbb{R}^m, \|x\| \leq r\} \quad , \tag{2.2}$$

where $0 < m < n$, so that $\mathbb{R}^m$ is the m-dimensional subspace of $\mathbb{R}^n$. Finally, $L_\varphi = R_\varphi L$ is L rotated over an angle φ, and $L_{x,\varphi} = R_\varphi T_x L$ is L translated over x and then rotated over an angle φ. For two-dimensional images, L_φ is a rotated line segment. For three-dimensional images it can be either a rotated line segment or a rotated disk. In this case, φ represents the Euler angles. RIA morphology operations are demonstrated with line segments only, but it is possible to substitute e.g. the rotated disk for this line segment without changing any of the equations and proofs. Since both D and L_φ are isotropic (in some subspace), they are point-symmetric around the origin. Therefore, mirroring these structuring elements does not introduce a change, $D = \check{D}$ and $L_\varphi = L_{-\varphi} = \check{L}_\varphi$. In this

[1] We use these terms because they have similar meanings to dilation and erosion, but do not posses the morphological connotations.

chapter, we refrain from using the superfluous set mirror operation with these structuring elements.

2.1 RIA Sedimentation and Wear

A flat, isotropic structuring element D of radius r can be decomposed into (an infinite number of) rotated line segments L_φ of length $2r$. These segments are the diameters of the disk. The dilation with this disk then becomes

$$\delta_D(f) = f \oplus D = f \oplus \bigcup_\varphi L_\varphi = \bigvee_\varphi \left[f \oplus L_\varphi \right] = \bigvee_\varphi \delta_{L_\varphi}(f) \quad . \tag{2.3}$$

Instead of taking the supremum over the orientation, we take the infimum. This defines a new morphological operator, which we will call RIA *sedimentation*, and denote with the symbol $\delta^\triangleleft$,

$$[\delta_L^\triangleleft(f)](x) = \bigwedge_\varphi [\delta_{L_\varphi}(f)](x) = \bigwedge_\varphi \bigvee_{y \in L_\varphi} f(x+y) \quad . \tag{2.4}$$

This operator takes the maximum of the image over a line segment rotated in such a way as to minimize this value. In contrast, an isotropic dilation uses the maximum over a line segment rotated in such a way as to maximize this value, (see Figure 2.1).

Using the terminology of binary morphology, the RIA sedimentation differs from the isotropic dilation in that a point belongs to the sedimented object if all rotated line segments (the diameters) hit the object. For the isotropic dilation, just one of the diameters needs to hit the object.

Figure 2.2 shows the effect that the operator has on an object boundary. Note that a convex object boundary is not changed, but a concave one is. We like to compare this sedimentation operator with a train running along a track.[2] The train wagons (which are constrained at both ends to the track) require some extra space at the inside of the curves. This operator, applied to a train track, and using a structuring element with the length of the wagons, reproduces the area required by them. This, of course, only holds when the curvature is small enough. Figure 2.3 shows the effect of the RIA sedimentation on a test image.

[2] The behavior of a train track operator was first introduced by P.W. Verbeek in the early nineties. RIA morphology and the sedimentation operator were not developed with this idea in mind.

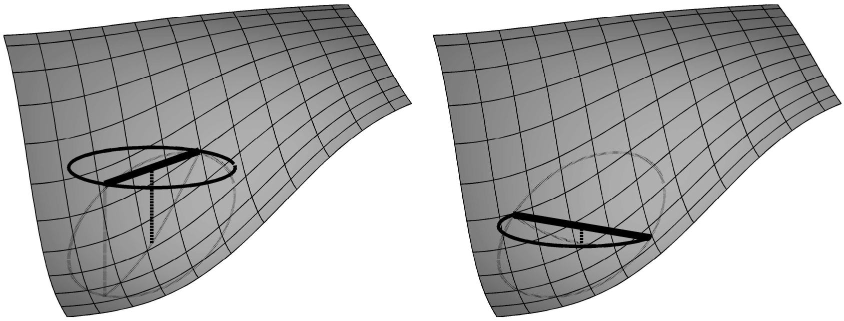


Figure 2.1: *Difference between an isotropic dilation and the* RIA *sedimentation.* Left: *The isotropic dilation chooses the orientation of the linear structuring element so that the maximum value over it is maximized.* Right: *The* RIA *sedimentation chooses the orientation to minimize this value.*

RIA *wear* is defined as the dual of the RIA sedimentation, and will be denoted with the symbol $\varepsilon^{\triangleleft}$,

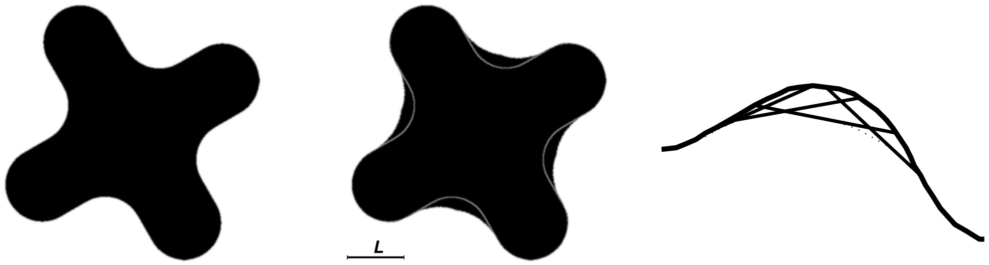
$$\varepsilon_L^{\triangleleft}(f) = -\delta_L^{\triangleleft}(-f) = -\bigwedge_{\varphi} \delta_{L_\varphi}(-f) = \bigvee_{\varphi} \left[-\delta_{L_\varphi}(-f)\right] = \bigvee_{\varphi} \varepsilon_{L_\varphi}(f) \quad . \tag{2.5}$$

Properties

Below, we present and prove the properties satisfied by both the RIA sedimentation and wear, as well as some properties of the dilation and the erosion that are not satisfied by these new operators. When properties are identical for both operators, we only mention them for the RIA sedimentation. In this case, the property and its proof for the RIA wear can be obtained by substituting the corresponding symbols and inverting the infimum and supremum operators.

Property 1 *Invariance under translation and change of contrast, and compatibility under scaling:*

$$\delta_L^{\triangleleft} T_t = T_t \delta_L^{\triangleleft}$$
$$\delta_L^{\triangleleft}(af) = a\delta_L^{\triangleleft}(f)$$
$$\delta_{S_s L}^{\triangleleft} S_s = S_s \delta_L^{\triangleleft}$$



a: *Original* b: *Sedimentation* c: *Construction*

Figure 2.2: *Effect of the* RIA *sedimentation on an object boundary.* c: *The sedimented object boundary is constructed by drawing line segments L of fixed length between all pairs of points on the original boundary that are at the correct distance from one another.*

Proof: because both the dilation and the minimum are translation invariant,

$$\delta_L^{\triangleleft} T_t(f) = \bigwedge_{\varphi} \delta_{L_\varphi} T_t(f) = \bigwedge_{\varphi} T_t \delta_{L_\varphi}(f) = T_t \bigwedge_{\varphi} \delta_{L_\varphi}(f) = T_t \delta_L^{\triangleleft}(f) \quad . \tag{2.6}$$

Because both the dilation with a flat structuring element and the minimum are contrast invariant,

$$\delta_L^{\triangleleft}(af) = \bigwedge_{\varphi} \delta_{L_\varphi}(af) = \bigwedge_{\varphi} a\delta_{L_\varphi}(f) = a \bigwedge_{\varphi} \delta_{L_\varphi}(f) = a\delta_L^{\triangleleft}(f) \quad . \tag{2.7}$$

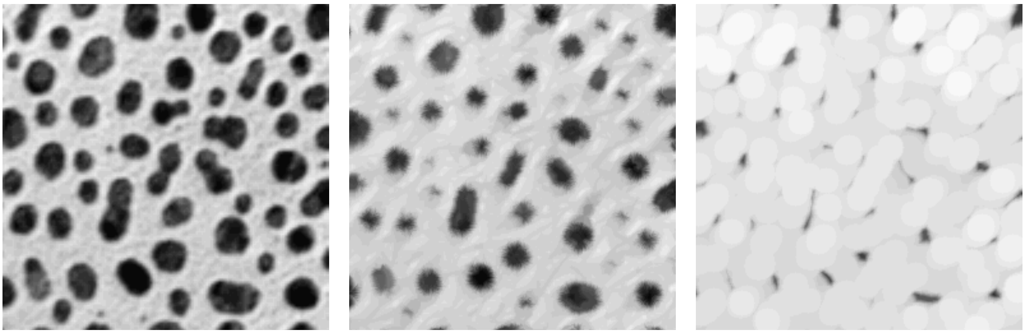
Finally, because the dilation is compatible under scaling, and the infimum is a point operation,

$$\delta_{S_s L}^{\triangleleft} S_s(f) = \bigwedge_{\varphi} \delta_{S_s L_\varphi} S_s(f) = \bigwedge_{\varphi} S_s \delta_{L_\varphi}(f) = S_s \bigwedge_{\varphi} \delta_{L_\varphi}(f) = S_s \delta_L^{\triangleleft}(f) \quad . \tag{2.8}$$

■

Property 2 *Rotation invariance:*

$$\delta_L^{\triangleleft} R_\theta = R_\theta \delta_L^{\triangleleft}$$



a: *Input* b: RIA *sedimentation* c: *Isotropic dilation*

Figure 2.3: *Demonstration of the* RIA *sedimentation on a test image. The structuring element used was a line with a length of* 20 *pixels. Compare to the isotropic dilation; the disk has a diameter of* 20 *pixels.*

Proof: It is easy to see that the dilation is compatible under rotation, meaning that the operation is invariant under rotation if the structuring element is rotated by the same amount as the image. This gives us

$$\delta_L^{\triangleleft} R_\theta(f) = \bigwedge_\varphi \delta_{L_\varphi} R_\theta(f) = \bigwedge_\varphi R_\theta \delta_{L_{\varphi-\theta}}(f) \quad , \tag{2.9}$$

taking $L_{\varphi-\theta}$ as the set L rotated by φ and then by $-\theta$. Since it does not matter if the infimum is taken before or after rotating the image,

$$\bigwedge_\varphi R_\theta \delta_{L_{\varphi-\theta}}(f) = R_\theta \bigwedge_\varphi \delta_{L_{\varphi-\theta}}(f) \quad . \tag{2.10}$$

But the infimum over all angles φ does not change by the offset θ because of the periodicity of the orientation, thus

$$R_\theta \bigwedge_\varphi \delta_{L_{\varphi-\theta}}(f) = R_\theta \bigwedge_\varphi \delta_{L_\varphi}(f) = R_\theta \delta_L^{\triangleleft}(f) \quad . \tag{2.11}$$

■

Rotation invariance of the RIA morphology is a key property, necessary for the correct analysis of images with an unknown orientation, or images without a single dominant orientation.

Property 3 *Increasingness:*

$$g \geq f \Rightarrow \begin{cases} \delta_L^{\lhd}(g) \geq \delta_L^{\lhd}(f) \\ \varepsilon_L^{\lhd}(g) \geq \varepsilon_L^{\lhd}(f) \end{cases}$$

Proof: for any structuring element B, the dilation is increasing:

$$g \geq f \Rightarrow \delta_B(g) \geq \delta_B(f) \quad . \tag{2.12}$$

It follows that

$$\delta_L^{\lhd}(g) = \bigwedge_{\varphi} \delta_{L_\varphi}(g) \geq \bigwedge_{\varphi} \delta_{L_\varphi}(f) = \delta_L^{\lhd}(f) \quad . \tag{2.13}$$

The same argument proves the other inequality. ■

Property 4 *Extensivity / anti-extensivity:*

$$\delta_L^{\lhd}(f) \geq f \geq \varepsilon_L^{\lhd}(f)$$

Proof: because of the extensivity of the dilation (if the structuring element contains the origin, which it does for the rotated line segments), we know that

$$\delta_{L_\varphi}(f) \geq f \quad , \quad \forall \varphi \quad ; \tag{2.14}$$

it follows that

$$\delta_L^{\lhd}(f) = \bigwedge_{\varphi} \delta_{L_\varphi}(f) \geq f \quad . \tag{2.15}$$

The same argument proves the other inequality. ■

Property 5 *Comparison with isotropic morphology:*

$$\delta_D(f) \geq \delta_L^{\lhd}(f) \geq f \geq \varepsilon_L^{\lhd}(f) \geq \varepsilon_D(f)$$

Proof: together with the extensivity of the RIA sedimentation (Property 4), the relation

$$\delta_L^{\lhd}(f) = \bigwedge_{\varphi} \delta_{L_\varphi}(f) \leq \bigvee_{\varphi} \delta_{L_\varphi}(f) = \delta_D(f) \tag{2.16}$$

proves the first statement. The other one is proven in a similar way. ■

Applications

As mentioned above, the RIA sedimentation affects only concave boundaries. By applying the RIA sedimentation iteratively, the boundary of the objects will expand outwards until it becomes convex. Likewise, the RIA wear shrinks convex boundaries until the object disappears. This process is similar to a curve evolution [106]. One might apply the RIA sedimentation and wear alternatively to see the object become convex and shrink. This has similarities in the differential equations of non-linear diffusion [41, 144] and the related field of level-set methods [107]. Pursuing these applications would lead us away from the theme of this thesis.

2.2 RIA Closing and Opening

Following the second definition of the closing, as in Equation (1.24), the closing at point x is given by the maximum over the structuring element after shifting it in such a way that the result is minimized. The constraint for this shift is that the structuring element must contain the point x (see Figure 2.4a).

To define the RIA *closing* we need again to decompose the disk D into its diameters. However, when we do this, we can only use those diameters that actually hit the origin, so that the point x is always included in the structuring elements used. In accordance to this, we define a new morphological operation as the maximum of the image f over the linear structuring element L after shifting and rotating it in such a way that the result is minimized, while keeping the point x within L (see Figure 2.4b). It will be denoted by $\phi^{\triangleleft}$, and defined by

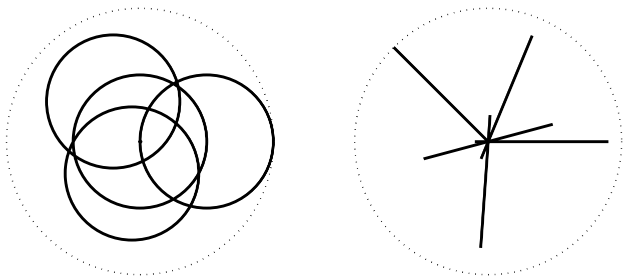
$$[\phi_L^{\triangleleft}(f)](x) = \bigwedge_{\varphi} \bigwedge_{y \in L_\varphi} \bigvee_{z \in L_{y,\varphi}} f(x+z) \quad . \tag{2.17}$$

This is the same as the minimum of the closings, at all orientations, with a line segment as structuring element,

$$\phi_L^{\triangleleft}(f) = \bigwedge_{\varphi} \bigwedge_{y \in L_\varphi} T_y \bigvee_{z \in L_\varphi} T_z(f) = \bigwedge_{\varphi} \varepsilon_{L_\varphi} \delta_{L_\varphi}(f) = \bigwedge_{\varphi} \phi_{L_\varphi}(f) \quad , \tag{2.18}$$

but not equal to an RIA sedimentation followed by an RIA wear. Intersections of closings with rotated line segments are not new (see for example [103, 115]).

Again using the terminology of binary morphology, the RIA closing differs from the isotropic closing in that a point belongs to the closed object if all of the diameters hit the object. For the isotropic closing, just one of the diameters needs to hit the object.



a: *Isotropic closing* b: RIA *closing*

Figure 2.4: *The closing with an isotropic structuring element (disk) is determined by shifting the disk in such a way that it minimizes the supremum of the image over its support, but still hits the point being evaluated. The* RIA *closing is determined by shifting and rotating the line segment in such a way that it minimizes the supremum of the image over its support, but still hits the point being evaluated.*

It is possible to imagine this operation creating a two-dimensional space, with the rotation angle φ as one axis, and the shift y as the other (see Figure 2.5). Each location in this space represents a possible position of L over the image, relative to the point x being evaluated. The value at each point (y, φ) is the maximum of the image f over the support of the rotated and translated structuring element $L_{y,\varphi}$. The absolute minimum in this space is the result of the operation. This notion will return when we prove some properties of this operator.

The RIA *opening* is defined as the dual of the RIA closing, and denoted by the symbol $\gamma^{\triangleleft}$,

$$\gamma_L^{\triangleleft}(f) = -\phi_L^{\triangleleft}(-f) = -\bigwedge_{\varphi} \phi_{L_\varphi}(-f) = \bigvee_{\varphi} \left[-\phi_{L_\varphi}(-f)\right] = \bigvee_{\varphi} \gamma_{L_\varphi}(f) \quad . \qquad (2.19)$$

Figure 2.6 shows the effect of both the RIA closing and opening on a test image.

Properties

Since RIA closing is an infimum of structural closings, it is an algebraic closing. This means it is an increasing, idempotent and extensive transform, as discussed in Chapter 1. Below, we present and prove the properties satisfied by both the RIA closing and opening. As before, we do not explicitly write most of these down for the RIA opening,

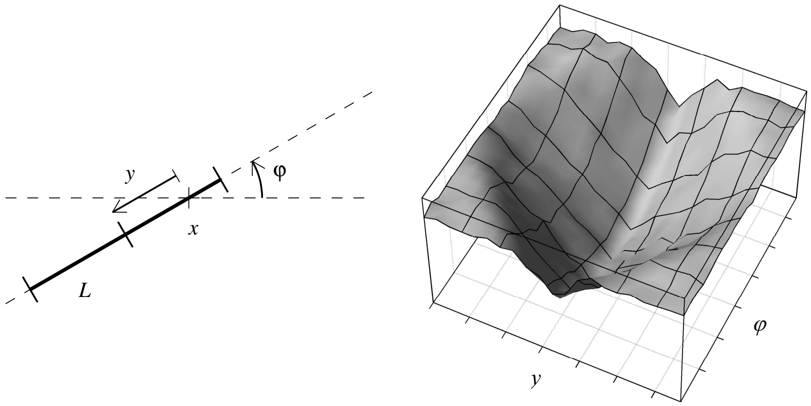


Figure 2.5: *The construction of the* RIA *closing.* Left: *A line segment $L_{y,\varphi}$.* Right: *The two-dimensional space spanned by the variables φ and y. The absolute minimum in this space is the result chosen by the* RIA *closing.*

especially when they are identical for both operators. In this case, the property and its proof for the RIA opening can be obtained by substituting the corresponding symbols and inverting the infimum and supremum operations.

Property 6 *Invariance under translation and change of contrast, and compatibility under scaling of the coordinate-axes:*

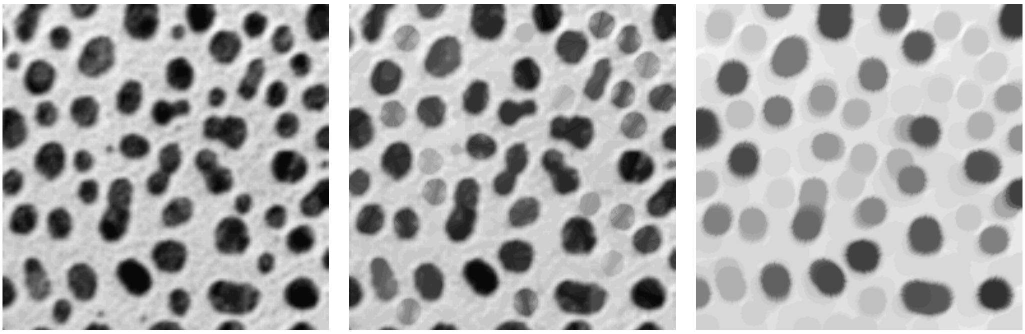
$$\begin{aligned} \phi_L^{\lhd} T_t &= T_t \phi_L^{\lhd} \\ \phi_L^{\lhd}(af) &= a\phi_L^{\lhd}(f) \\ \phi_{S_s L}^{\lhd} S_s &= S_s \phi_L^{\lhd} \end{aligned}$$

These properties can be proven in the same way as Property 1.

Property 7 *Rotation invariance:*

$$\phi_L^{\lhd} R_\theta = R_\theta \phi_L^{\lhd}$$

This property can be proven in the same way as Property 2.



a: *Input* b: RIA *closing* c: *Isotropic closing*

Figure 2.6: *Demonstration of the* RIA *closing on a test image. The structuring element used was a line with a length of* 20 *pixels, which is the minimal length of the remaining dark features. Compare to the isotropic closing; the disk has a diameter of* 20 *pixels.*

Property 8 *Increasingness:*

$$g \geq f \Rightarrow \begin{cases} \phi_L^{\triangleleft}(g) \geq \phi_L^{\triangleleft}(f) \\ \gamma_L^{\triangleleft}(g) \geq \gamma_L^{\triangleleft}(f) \end{cases}$$

This property can be proven in the same way as Property 3. Also, this property is proven by Matheron [79] because the RIA closing is an infimum of closings (i.e. an algebraic closing).

Property 9 *Extensivity / anti-extensivity:*

$$\phi_L^{\triangleleft}(f) \geq f \geq \gamma_L^{\triangleleft}(f)$$

This property can be proven in the same way as Property 4. As with the increasingness, this property was proven by Matheron.

Property 10 *Extended extensivity:*

$$\delta_L^{\triangleleft}(f) \geq \phi_L^{\triangleleft}(f) \geq f \geq \gamma_L^{\triangleleft}(f) \geq \varepsilon_L^{\triangleleft}(f)$$

Proof: By definition,

$$[\phi_L^{\lhd}(f)](x) = \bigwedge_{\varphi} \bigwedge_{y \in L_\varphi} \bigvee_{z \in L_{y,\varphi}} f(x+z) \tag{2.20}$$

and

$$[\delta_L^{\lhd}(f)](x) = \bigwedge_{\varphi} \bigvee_{z \in L_\varphi} f(x+z) \quad . \tag{2.21}$$

If $y = 0$ (which is always a part of L_φ) minimizes Equation (2.20), it becomes identical to Equation (2.21). Any other value decreases the result of the RIA closing. We conclude

$$[\phi_L^{\lhd}(f)](x) = \bigwedge_{\varphi} \bigwedge_{y \in L_\varphi} \bigvee_{z \in L_{y,\varphi}} f(x+z) \leq \bigwedge_{\varphi} \bigvee_{z \in L_\varphi} f(x+z) = [\delta_L^{\lhd}(f)](x) \quad . \tag{2.22}$$

The same argument shows that

$$\gamma_L^{\lhd}(f) \geq \varepsilon_L^{\lhd}(f) \quad . \tag{2.23}$$

Adding Property 9, the proof is completed. ■

Property 11 *Absorption:*

$$L^{(1)} \supseteq L^{(2)} \Rightarrow \begin{cases} \phi_{L^{(2)}}^{\lhd} \phi_{L^{(1)}}^{\lhd} = \phi_{L^{(1)}}^{\lhd} \\ \phi_{L^{(1)}}^{\lhd} \phi_{L^{(2)}}^{\lhd} = \phi_{L^{(1)}}^{\lhd} \end{cases}$$

Proof: To prove the first equality, let $g = \phi_{L^{(1)}}^{\lhd}(f)$. By definition, there exists a $\varphi = \varphi_1(x)$ and an $y = y_1(x)$ for each location x such that

$$g(x) = \bigwedge_{\varphi} \bigwedge_{y \in L_\varphi^{(1)}} \bigvee_{z \in L_{y,\varphi}^{(1)}} f(x+z) = \bigvee_{z \in L_{y_1(x),\varphi_1(x)}^{(1)}} f(x+z) \quad . \tag{2.24}$$

These are the values for the rotation and shift of $L^{(1)}$ that minimize the supremum over its support, and are the ones selected by the operator (see Figure 2.5).

The result of the operator for all locations $x+z$, with $z \in L_{y_1(x),\varphi_1(x)}^{(1)}$ is equal or smaller to the result at the point x,

$$g(x) \geq g(x+z) \quad , \quad z \in L_{x_1(x),\varphi_1(x)}^{(1)} \quad . \tag{2.25}$$

This is seen by the fact that, for the locations $x+z$, the value of $g(x)$ is also a part of the (y,φ) space in which the minimum is taken. This inequality also holds for $z \in L^{(2)}_{x_1(x),\varphi_1(x)}$, because $L^{(2)} \subseteq L^{(1)}$.

We have

$$[\phi^{\lhd}_{L^{(2)}}(g)](x) = \bigwedge_{\varphi} \bigwedge_{y \in L^{(2)}_{\varphi}} \bigvee_{z \in L^{(2)}_{y,\varphi}} g(x+z) \leq \bigvee_{z \in L^{(2)}_{y_1(x),\varphi_1(x)}} g(x+z) \quad , \tag{2.26}$$

because $y_1(x)$ and $\varphi_1(x)$ are not necessarily the optimal values (those that minimize the expression) for y and φ. Furthermore,

$$\bigvee_{z \in L^{(2)}_{y_1(x),\varphi_1(x)}} g(x+z) \leq g(x) \quad , \tag{2.27}$$

using Equation (2.25). This means that $\phi^{\lhd}_{L^{(2)}}(g) \leq g$. From the extensivity property (Property 9), we have $\phi^{\lhd}_{L^{(2)}}(g) \geq g$. Therefore we must conclude that $g = \phi^{\lhd}_{L^{(2)}}(g)$, which proves the first equality.

To proof the second equality, let $g = \phi^{\lhd}_{L^{(2)}}(f)$. In the spirit of Equation (2.24), the values $\varphi_1(x)$ and $y_1(x)$ minimize the expression

$$[\phi^{\lhd}_{L^{(1)}}(f)](x) = \bigwedge_{\varphi} \bigwedge_{y \in L^{(1)}_{\varphi}} \bigvee_{z \in L^{(1)}_{y,\varphi}} f(x+z) = \bigvee_{z \in L^{(1)}_{y_1(x),\varphi_1(x)}} f(x+z) \quad . \tag{2.28}$$

We know that

$$[\phi^{\lhd}_{L^{(1)}}(g)](x) = \bigwedge_{\varphi} \bigwedge_{y \in L^{(1)}_{\varphi}} \bigvee_{z \in L^{(1)}_{y,\varphi}} g(x+z) \leq \bigvee_{z \in L^{(1)}_{y_1(x),\varphi_1(x)}} g(x+z) \quad , \tag{2.29}$$

because the choice of $\varphi_1(x)$ and $y_1(x)$ need not be optimal for this function.

We can choose $\varphi_2(x)$ and $y_2(x)$ such that $L^{(2)}_{y_2(x+z),\varphi_2(x+z)} \subseteq L^{(1)}_{y_1(x),\varphi_1(x)}$, for each $z \in L^{(1)}_{y_1(x),\varphi_1(x)}$. Using these values of $\varphi_2(x)$ and $y_2(x)$,

$$[\phi^{\lhd}_{L^{(1)}}(f)](x) = \bigvee_{z \in L^{(1)}_{y_1(x),\varphi_1(x)}} f(x+z) = \bigvee_{z \in L^{(1)}_{y_1(x),\varphi_1(x)}} \bigvee_{t \in L^{(2)}_{y_2(x+z),\varphi_2(x+z)}} f(x+z+t) \quad . \tag{2.30}$$

Because $\varphi_2(x)$ and $y_2(x)$ need not be optimal,

$$\bigvee_{z \in L^{(2)}_{y_2(x),\varphi_2(x)}} f(x+z) \geq g(x) \quad . \tag{2.31}$$

Taking Equations (2.29), (2.30) and (2.31) together, we have $[\phi^{\lhd}_{L^{(1)}}(f)] \geq \phi^{\lhd}_{L^{(1)}}(g)$. However, because $f \leq g$ (Property 9, extensivity), and using the increasingness property (Property 8), $\phi^{\lhd}_{L^{(1)}}(f) \leq \phi^{\lhd}_{L^{(1)}}(g)$. Thus, we can conclude that $\phi^{\lhd}_{L^{(1)}}(f) = \phi^{\lhd}_{L^{(1)}}(g)$, which proves the second equality. ■

Property 12 *Idempotence:*

$$\phi^{\lhd}_{L}\phi^{\lhd}_{L} = \phi^{\lhd}_{L}$$

This is a special case of Property 11 (absorption), setting $L^{(1)} = L^{(2)}$, and thus is proven. This property is also proven by Matheron [79] because the RIA closing is an infimum of closings.

Property 13 *Sieving:*

$$L^{(1)} \supseteq L^{(2)} \Rightarrow \phi^{\lhd}_{L^{(1)}} \geq \phi^{\lhd}_{L^{(2)}}$$

Proof: because of the extensivity (Property 9),

$$\phi^{\lhd}_{L^{(1)}}(f) \geq f \quad . \tag{2.32}$$

Applying the RIA closing with structuring element $L^{(2)}$ to both sides we get

$$\phi^{\lhd}_{L^{(2)}}\phi^{\lhd}_{L^{(1)}}(f) \geq \phi^{\lhd}_{L^{(2)}}(f) \quad , \tag{2.33}$$

since the operation is increasing (Property 8). The left-hand side can be simplified to

$$\phi^{\lhd}_{L^{(1)}}(f) \geq \phi^{\lhd}_{L^{(2)}}(f) \quad , \tag{2.34}$$

by absorption (Property 11). ■

The sieving property is a requirement for granulometric applications (see Chapter 3). Basically, it states that the RIA closing can be applied with different size parameters to an image, and the larger this parameter, the more dark features will be removed. All features removed at a smaller scale will also be removed at a larger scale. This allows such a sequence of operators to ‘sieve’ the features in an image and classify them according to size.

Property 14 *Commutativity:*

$$\phi^{\lhd}_{L^{(1)}}\phi^{\lhd}_{L^{(2)}} = \phi^{\lhd}_{L^{(2)}}\phi^{\lhd}_{L^{(1)}}$$

Proof: if $L^{(1)} \supseteq L^{(2)}$, by absorption (Property 11),

$$\phi^{\lhd}_{L^{(1)}}\phi^{\lhd}_{L^{(2)}} = \phi^{\lhd}_{L^{(1)}} = \phi^{\lhd}_{L^{(2)}}\phi^{\lhd}_{L^{(1)}} \quad . \tag{2.35}$$

Else, if $L^{(1)} \subseteq L^{(2)}$,

$$\phi^{\lhd}_{L^{(1)}}\phi^{\lhd}_{L^{(2)}} = \phi^{\lhd}_{L^{(2)}} = \phi^{\lhd}_{L^{(2)}}\phi^{\lhd}_{L^{(1)}} \quad , \tag{2.36}$$

by that same property. ■

Property 15 *Comparison with isotropic morphology:*

$$\phi_D(f) \geq \phi^{\lhd}_L(f) \geq f \geq \gamma^{\lhd}_L(f) \geq \gamma_D(f)$$

This property can be proven in the same way as Property 5.

Applications

A structural closing removes dark objects in which the structuring element does not fit. This operation is only rotation invariant if the structuring element is isotropic (i.e. a disk in 2D or a ball in 3D). When using an isotropic set with radius r, a feature in the image is not removed if the sizes of that feature in all directions are larger or equal to $2r$. That is, the smallest diameter of the feature is discriminating.

The RIA closing also removes dark objects in which the structuring element does not fit. Because it is allowed to rotate, rotation invariance is achieved. However, the set has no extension in one or more directions. This means that it will fit inside features where the isotropic set would not fit. A one-dimensional set (line segment) will therefore discriminate on length (or largest diameter). A two-dimensional set (a disk) in a three or higher-dimensional image discriminates on the second largest diameter of the features. This way, it is possible to construct structuring elements that will discriminate on any of the dimensions.

We will use the RIA closing in an application to measure the length distribution of rice grains, in Section 7.3.

2.3 Morphological Orientation-Space and RIA Morphology

A morphological operation Ψ that uses rotated versions of an anisotropic structuring element L can be used to construct a morphological *orientation-space*,

$$F(x,\varphi) = [\Psi_{L_\varphi}(f)](x) \quad . \tag{2.37}$$

Much like a scale-space (see footnote 1 on page 57), in which the image is extended with a new dimension, an orientation-space adds one or more dimensions representing orientation. Within this space it is possible e.g. to separate oriented structures [37]. The RIA sedimentation and closing result from a minimum projection along the orientation-axes,

$$[\Psi_L^{\triangleleft}(f)](x) = \bigwedge_{\varphi} F(x,\varphi) \quad . \tag{2.38}$$

The RIA wear and opening result from a maximum projection. The argument-minimum projection (or the argument-maximum projection in the case of the erosion or opening) results in an estimation of local orientation [119, 120]. In a sieve, this orientation-space is extended with a scale dimension, resulting in a joint orientation-scale-space.

The idea of the orientation-space is used later when studying implementation aspects of RIA Morphology. Because it is not possible to compute the pertinent operation for an infinite set of orientations, it is obvious that some error will be made. By examining what happens in orientation-space, we can determine how the number of orientations relates to the size of the structuring element. This is discussed in Section 5.7.

All exact science is dominated by the idea of approximation
(Bertrand Russell)

Chapter 3

Granulometries

This chapter introduces the morphological *sieve*, the tool that we use to characterize structures in this thesis. The sieve is a multi-scale closing or opening, and which results in a *granulometry* or size distribution.

Sieves were first proposed by Matheron [79]. They have been used in both binary morphology and gray-value morphology to measure particle-size distributions [126], as well as characterize textures [8, 13, 116] or shapes [75]. Because a sieve has an increasing scale parameter, it results in a *scale-space*.[1] Many theoretical studies have been made, linking it with linear scale-space theory and other non-linear scale-spaces [7, 52, 92]. See also [14, 28].

A comparison can be made between morphological sieves and sifting grains in a heap of sand [114]. The grains are sifted through meshes of decreasing size, extracting grains from the collection. Each mesh removes the set of grains that fall through it, but did not fall through the ones before. Thus, each set contains grains in a given size range. The weight of each of the sets of grains provides a point in a size distribution, which gives information on the sand heap. As seen in Chapter 1, the opening and closing operations perform a similar function to that of the mesh, removing from an image those features that are smaller than the structuring element.

A sieve is defined in mathematical terms by a transformation having a size parameter λ and satisfying the three following axioms, enumerated by Matheron [79]:

- (Anti-)extensivity: the grains that remain in the sieve are a subset of the initial grains.
- Increasingness: adding grains to the heap does not diminish the number of grains that remain in the sieve.

[1] A scale-space is an extension to an image obtained by adding a scale dimension [62, 146]. Such an image has many interesting properties related to human vision, and is used extensively to solve problems in image analysis and robot vision.

– Absorption: sifting at two different sizes λ and ν will give the same result regardless of the order of the sieves; the size of the largest sieve determines the result (although in the sand heap analogy it would be the finest sieve that determines the result).

By definition (see Section 1.2), all closings and openings satisfy the first two properties. However, as shown by Matheron, not all closings or openings with structuring elements of increasing size satisfy the absorption property. For example, an Euclidean disk satisfies this property, but a sampled Euclidean disk does not because of the discretization errors [142].

We start by introducing the sieve (Section 3.1) and deriving a size distribution from it (Section 3.2). All of this is examined in the continuous domain. Then we go into the implementation details (Sections 3.3, 3.4 and 3.5). Specifically, two improvements to the traditional implementation are discussed. Section 3.6 evaluates these improvements, comparing the various possible implementations of the discrete disk.

3.1 The Sieve and the Pattern Spectrum

We illustrate here the notion of sieving using the structural closing (i.e. closing with a structuring element) as the sieving operation. However, by substituting it for the opening, an equivalent sieving operation is obtained. The difference is that from the closing sieve one can derive a size distribution of dark objects, whereas from the opening sieve a size distribution of the light objects would be obtained. None of the properties or equations in this and the next sections change from the one type of sieve to the other. Section 6.1 presents other types of sieves from which distributions with other properties are obtained.

We will use an isotropic structuring element (e.g. a disk in two dimensions), expressed as $D(x,r) = \{x \mid \|x\| \leq r\}$. In this and the next section, it is assumed that the image $f(x)$ is continuous and does not have a boundary ($f : \mathbb{R}^n \to \mathbb{R}$). We construct a scale-space $F(x,r)$ by closing the image $f(x)$ at all scales $r \in (0,\infty)$:

$$F(x,r) = [\phi_{D(y,r)}(f)](x) \quad . \tag{3.1}$$

Each image $F(x,r_0)$ now contains only dark objects larger than r_0. We define $F(x,0) = f(x)$. Note that $F(x,r)$ is an image with one more dimension than $f(x)$: if x is an n-dimensional vector, $F(x,r)$ has $n+1$ dimensions.

A sum projection on the scale-axis of $F(x,r)$ is often called a granulometry, and its derivative is referred to as *pattern spectrum* [75, 114]. The granulometry is an increasing function (or decreasing in the case of the opening-sieve), with jumps at the scales where

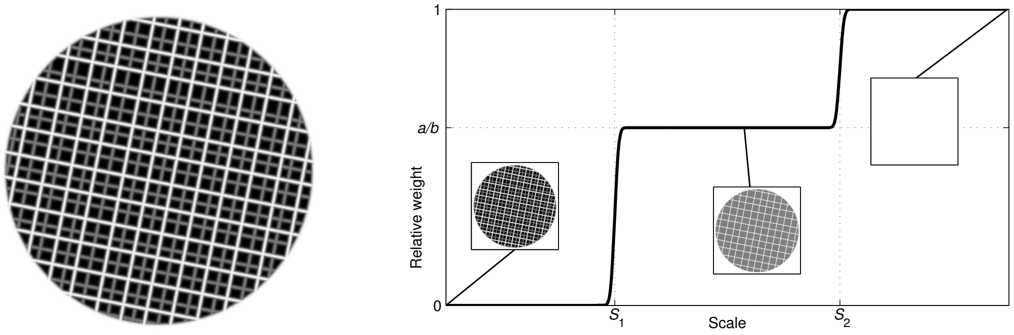


Figure 3.1: *Demonstration of the granulometry.* Left: *Synthetic test image with two structures at different scales superimposed. The image is formed by lines of gray-value a and b.* Right: *Abstracted granulometric curve that shows two jumps at the two scales present in the image.*

image features disappear (as in Figure 3.1). The pattern spectrum has peaks at these locations.

The difference between a pattern spectrum and a size distribution is the normalization, as we will see next.

3.2 The Size Distribution

Based on the closing scale-space, it is relatively easy to construct a size distribution of the dark objects. As mentioned in the introduction, the weight of each of the sets of rocks provides a point of the size distribution of these rocks. Similarly, the integral of each of the images in the closing scale-space can be used to construct a cumulative distribution. This distribution is rotation and translation invariant, because the closing is. By normalizing the cumulative distribution such that it ranges from 0 to 1, we make it independent of the image size and contrast, as well as the area fraction of the image covered by the objects. The cumulative distribution is thus defined as

$$H(r) = \frac{\int F(x,r)dx - \int F(x,0)dx}{\int F(x,\infty)dx - \int F(x,0)dx} \quad , \tag{3.2}$$

where $F(x,\infty)$ is the original image closed with an infinite structuring element, and is thus equal to the image filled with its maximal gray-value. A size distribution is obtained by taking the derivative of $H(r)$,

$$h(r) = \frac{d}{dr}H(r) \quad . \tag{3.3}$$

What Is Actually Being Measured

Note that, as the structuring element used is isotropic, the closing operation removes a dark object (or part thereof) if its smallest diameter is smaller than the structuring element. Therefore, whenever we mention the size of an object, we actually mean its smallest diameter. To use another diameter (e.g. the length) as a selection criterion, it is possible to use the RIA closing presented in Chapter 2. We use such a granulometry in an application in Section 7.3.

Using other types of closings, or other shapes for the structuring element, different granulometries are obtained that measure different features of the dark objects under examination (see Section 6.1). Using openings instead of closings, light objects are measured.

Difference Between Multi-Scale Structure and Two-Phase Structure

Some structures contain different scales. Think about a telephone cable, composed of many bundles, each of which is made out of hundreds of thin wires. The wires are part of two structures at different scales. The morphological scale-space as described in this section is capable of finding both scales. Take as an example the structure in Figure 3.1, which is formed by lines of gray-values a and b. A closing at scale S_1 fills the smaller dark regions with gray-value a. Another closing at scale S_2 fills the larger regions with gray-value b. This results in a single point belonging to two objects, its volume being split over two levels of the distribution: at level S_1 with weight a/b, and at level S_2 with weight $(b-a)/b$.

The method as described here to calculate a size distribution assumes a homogeneous image. If this is not the case, the distribution will obviously be different when estimated from different regions of the image. The structure in Figure 3.2 produces the same size distribution to that in Figure 3.1. To be able to distinguish these cases, some additional analysis would be required, either on the original image (e.g. a homogeneity analysis) or on the scale-space $F(x,r)$.

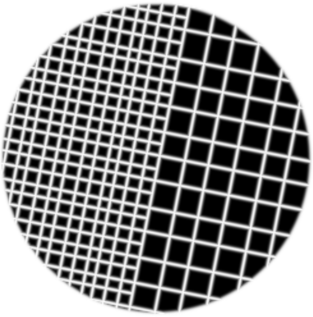


Figure 3.2: *Synthetic test image with inhomogeneous structure that produces the same size distribution as the image in Figure 3.1.*

Objects with Sloped Edges

It is obvious, when looking at the description of the sieve, that image features composed of gray-value ramps will be separated into many scales. For example, imagine a dark object of which the gray-value decreases towards its center (e.g. an inverted pyramid). In this case, what we have (according to the closing-sieve) is many superimposed shallow objects of decreasing size (much like the Russian matryoshka dolls). Thus, what a person objectively sees as a single object, is being turned into a multi-scale object, and spread out into many levels of the sieve.

We found that clipping the image's gray-values alleviated some of the symptoms related to this. In some specific cases an appropriate pre-processing step (e.g. high pass-filtering, line or edge detection) solves this problem. See Chapter 6 for more information on these topics.

3.3 Discrete Granulometries

A discrete sieve is a discrete operation (with discrete structuring elements), applied to a digitized image, and resulting in a discrete scale-space. This requires:

- Discretizing (by point-sampling) the input image: $f(x) \rightarrow \hat{f}[k]$, where $\hat{f} : \mathbb{Z}^n \rightarrow \mathbb{R}$, $k \in \mathbb{Z}^n$, $x = k\Delta$ and Δ the sample spacing.
- Discretizing the scales: $r \rightarrow i$. We sample the scale-axis at $r = s[i]$, where $i \in \mathbb{N}$ and $s : \mathbb{N} \rightarrow \mathbb{R}$ a scale-generating function. For logarithmic sampling, $s[i] = 2^{i/p}$, with p the number of scale samples per octave.

- Discretizing (also by point-sampling) the structuring elements: $D(x, s[i]) \rightarrow \hat{D}[k, i]$.

This results in a discrete scale-space computed by

$$\hat{F}[k,i] = \left(\phi_{\hat{D}[m,i]} \hat{f}\right)[k] \quad . \tag{3.4}$$

Discretizing the input image poses no challenge. As mentioned in the introduction of this thesis, $f(x)$ can be sampled without loss of information as long as it is band-limited. Sampling the structuring element, on the other hand, causes a large discretization error. Due to this and the lack of sampling invariance of the closing operation itself, the computed $\hat{F}[k,i]$ is not equal to the sampled version of $F(x,r)$ at $x = k\Delta$, $r = s[i]$. Any difference between $\hat{F}[k,i]$ and $F(k\Delta, s[i])$ is caused by various discretization errors. Also, because $F(x,r)$ is not band-limited, it is not possible to reconstruct it from $F(k\Delta, s[i])$. This is important especially along the scale-axis, where it might be interesting to find the location of peaks in $h(r)$.

We need to stress here that pattern spectra as in e.g. [75] are discrete implementations of the sieve with a series of structuring elements that do not necessarily increase uniformly in size. These are defined to satisfy the granulometric property, and the shape may vary to accomplish this. In this section, we define a discrete granulometry that approximates the continuous one, but does not satisfy the granulometric property. That is, the shape and size of the structuring elements must approximate those of the continuous granulometry as defined above. The more accurately this is accomplished, the better the granulometric property will be satisfied.

Sampling the Flat Structuring Element

Sampling a binary function causes severe quantization errors. This has two problematic consequences:

- The original function cannot be reconstructed given the samples. The problems occur in a strip along the boundary. The width of this strip is equal to the sample spacing Δ, the relative error is thus proportional to Δ.
- The size increments of a finely sampled scale-space become very erratic due to the irregular increase in size of the sampled disk as a function of the radius.

An obvious solution is not to use flat but gray-valued structuring elements, which can be sampled more accurately. A prominent example of a gray-valued structuring element is the parabola [32]. However, a parabolic closing does not produce the desired result, because a sieve that uses parabolic closings will split a single object over a whole range of scales: the object is only partially filled with one closing, larger closings will fill the object further. This makes the transformation of the granulometry into a size distribution difficult, and its results would be even more difficult to interpret. More likely is a band-limited disk, which has a smooth transition and can be sampled correctly, but is still quite flat. We will examine such a gray-value structuring element Section 3.5. Section 3.4 discusses how to implement a binary disk minimizing the discretization error. These two possible implementations are compared in Section 3.6.

Sampling the Scale-Axis

We still have not discussed how to sample the scale-axis. There is relatively little literature on this topic. In most articles, one-pixel increments are used as a default solution. We suggest to use logarithmic sampling, so as to keep the relative error constant. One might want to distinguish between 3-pixel objects and 4-pixel ones, but not between 100-pixel objects and 101-pixels ones. Additionally, large objects are sparse in an image, causing a linearly sampled size distribution to be inaccurate at large scales. Particle or pore-size distributions often are log-normal [6], meaning that, when logarithmically sampled, they resemble a normal distribution.

A sampled $F[k,i]$ also produces a sampled $H[i] = H(r)|_{r=s[i]}$. In this case, the derivative in Equation (3.3) can only be obtained by approximation,

$$h[i] = \frac{H[i+1]-H[i]}{s[i+1]-s[i]} \quad . \tag{3.5}$$

An upper bound to the number of scale samples that are useful is given by the discretization of the structuring element. If too many samples are taken, then the differences between these structuring elements are very small, and the discretization error has a large influence on the results. That is, the averaging effect introduced by the limited number of samples along the scale-axis hides some of the errors made in the spatial domain. This links the scale sampling to the sampling of the structuring element. This notion is illustrated in Section 3.6.

Even for band-limited images, the scale-space $F(x,r)$ is not band-limited along the scale-axis.[2] Therefore, $H(r)$ is also not band-limited. This makes it impossible to obtain all information on $H(r)$ using a pre-defined set of scales $s[i]$. Using adaptive sampling, however, it might be possible to find the location of large jumps in $H(r)$ (i.e. the peaks of

[2] Nor is it band-limited in any other direction.

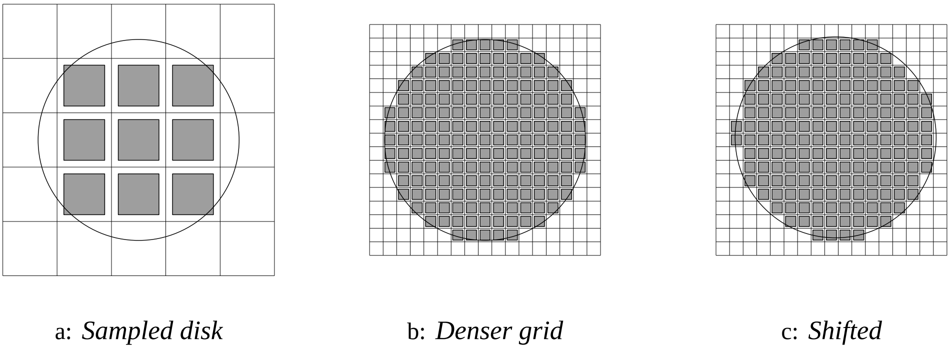

a: *Sampled disk* b: *Denser grid* c: *Shifted*

Figure 3.3: a: *Disk of radius* 1.85 *pixels, sampled (expected area* = 10.7521*, sampled area* = 9*).* b: *Same disk, sampled on a grid four times as dense (area* = 11.0625*). This is the structuring element one would use after subsampling the image four times.* c: *Same disk, centered at* $(0.19, 0.31)$ *(area* = 10.7500*).*

$h(r)$). Note that $H(r)$ can be sampled at random locations; it is not necessary to do this in a fixed order or with fixed steps. Such a procedure is similar to finding zero-crossings of a function numerically.

3.4 Sampling the Binary Structuring Element

Section 3.3 discussed problems encountered when discretizing the granulometry. One of the issues is how to discretize the structuring element, which, as discussed earlier, should be flat if one wishes to obtain a size distribution. This section proposes two changes to the discrete disk that greatly diminish the discretization error. Both these changes are evaluated in Section 3.6.

Increasing Accuracy with Interpolation

We stated in the previous section that the discretization error occurs in a strip along the boundary of the disk, whose width is equal to the pitch of the sampling grid. By decreasing this pitch the strip becomes thinner and the error diminishes (see Figure 3.3b).

The scaling property of the closing, as in Equation (1.29),

$$S_{\alpha^{-1}} \phi_{S_\alpha(D)} S_\alpha(f) = \phi_D(f) \tag{3.6}$$

teaches us that we can up-scale the input image and the structuring element, and down-scale the result. Replacing the left-hand side by a discrete closing, the equality is only true for $\alpha \rightarrow \infty$. Increasing α (i.e. up-sampling the image), the discrete closing becomes a better approximation to the continuous closing. Optionally, one could down-sample[3] the input image for the larger structuring elements, so that the discretization error is approximately equal for all levels of the scale-space.[4]

Increasing Accuracy with a Shift

Due to the symmetry of a sampled disk on a square grid, all discretization effects are enhanced: when increasing the size of a discrete disk, a multiple of four pixels are added to it. Placing the center of the disk away from the origin, this symmetry is broken (Figure 3.3c), allowing the disk to be sampled more accurately: $D(x,r) = \{x \mid \|x-\delta\| \leq r\}$. The optimal shift δ can be determined experimentally by computing the mean square relative error of the area for disks of many different sizes, for each δ. This results in the graph of Figure 3.4. There is a clear minimum at $\delta = (0.19, 0.31)$, and, due to symmetry, another one at $\delta = (0.31, 0.19)$. Counting the samples that fall inside a shape is an unbiased estimator for the area [141]. Therefore, minimizing the mean square error of the estimated area is a good way of determining the optimal shift δ. Figure 3.5 shows the relative error made when centering the disk at this location in comparison to centering it at the origin. Remember that the continuous closing is invariant to translation of the structuring element. For higher-dimensional structures, the same experiment can be performed to determine the optimal location of the origin. For the three-dimensional sphere the optimal location is $\delta = (0.16, 0.24, 0.34)$.

The one-dimensional isotropic structuring element makes this concept easier to explain. If a segment is centered at $\delta = 0$, two pixels are added (one at each end) at the same time when increasing its size (i.e. only lengths with an odd number of pixels are possible). Centering it at $\delta = 0.5$ the same is true, but now it is always even in size. Optimal asymmetry is obtained by centering it at $\delta = 0.25$. In this case, one pixel at a time is added to the structuring element. That is, more distinct lengths are possible. This makes it possible to sample the granulometry more accurately.

[3] Down-sampling before applying a closing should be done such that local or regional maxima are preserved. This can be accomplished by dilating the image with a square structuring element of size equal to the down-sampling factor, and sub-sampling the result.

[4] This also greatly diminishes the computational cost of the operation at large scales.

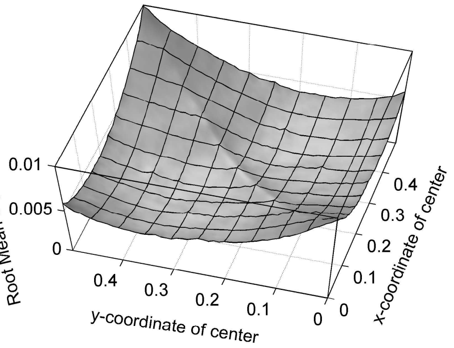


Figure 3.4: *Mean square relative error made when discretizing Euclidean disks (with random radii), for different positions of the disk's center with respect to the sampling grid. There is a minimum at coordinates* $(0.19, 0.31)$*, but the exact position is not very important, because quite a large region around this minimum produces small errors.*

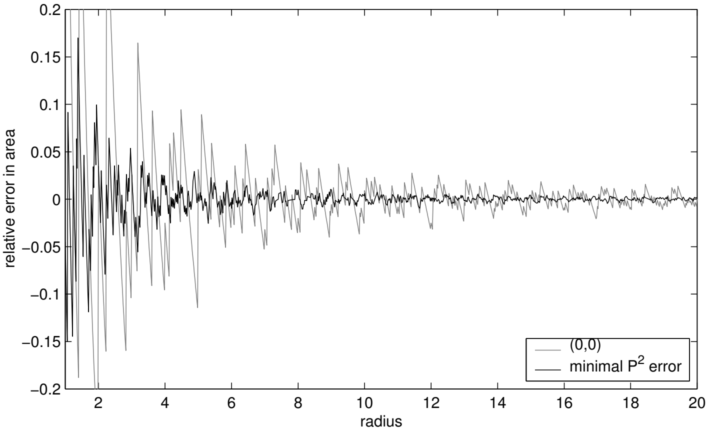


Figure 3.5: *Relative error in area for a disk centered at* $(0, 0)$*, and a disk centered at* $(0.19, 0.31)$*, the location of the minimum in Figure 3.4.*

3.5 Using Gray-Value Structuring Elements

Instead of using a binary structuring element, it is possible to use a gray-value structuring element. A disk with a certain minimal diameter can be sampled correctly if its edge is smooth. We use an error function to represent the transition from foreground to background; this is equivalent of convolving the binary shape with a Gaussian kernel. Both the error function and its derivative, the Gaussian function, are approximately band-limited, and can be sampled with very little loss of information. The sample spacing should be smaller or equal to σ [66, 139] for this to be the case. The image to be used as a structuring element is therefore defined as

$$D(x,r,\sigma) = A \cdot \frac{1}{2}\left\{-1 - \operatorname{erf}\left(\frac{|x|-r}{\sqrt{2}\sigma}\right)\right\} \quad , \tag{3.7}$$

where σ is larger or equal to 1, r is the radius of the disk, and x is the coordinate vector (see Figure 3.6). Note that the gray-value of the disk is 0, and the background has a value of $-A$. A is the scaling of the image, and depends on the gray-value range in the image to be processed.

It is not directly clear, however, how to scale this image $D(x,r,\sigma)$. It is obvious that the height A of the disk must be larger than the range of gray-values in the image. If it is not, the edge of the image used as structuring element will influence the morphological operation, which is not desirable (see the subsection on gray-value structuring elements on page 30*ff*). But this height will also influence the shape of the disk. Even though the disk is approximately band-limited for any A, its slopes are not invariant to gray-value scaling. Since morphological operations can be written as an interaction between slopes [32], it follows that this scaling definitely has an influence on the result of the operation. By relating the value of A to the range of gray-values in the image, the operation is invariant to gray-value scaling of the image, but not invariant to e.g. impulse noise (which increases the gray-value range), or gray-value scaling of individual objects in the image.

Figure 3.7 contains examples of a dilation with a gray-value structuring element scaled with different quantities. It shows what happens when A is smaller than the gray-value range of the input image, and what happens when A is chosen too large. We will be using a value of 1.0233 times the gray-value range of the input. This value is chosen to make the cut-off point of the structuring element image[5] to be $|x| \leq r + 2\sigma$.

[5] This provides the size of the image needed to generate the structuring element. By limiting this size, the operation can be made more efficient.

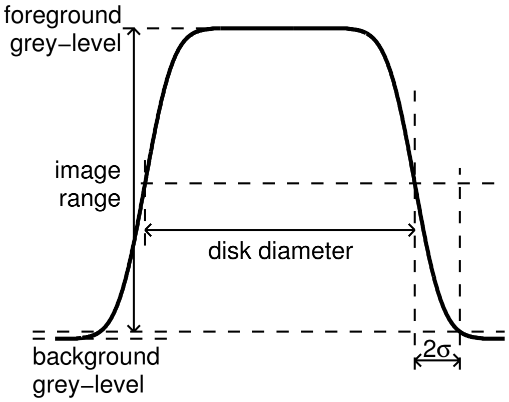


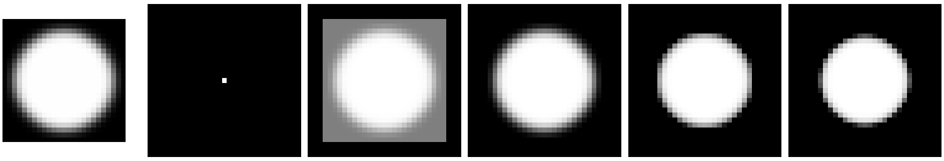
Figure 3.6: *Construction of the band-limited disk with a Gaussian profile. The profile is given by the error function as a function of the distance to the center of the disk. The difficulty lies in the normalization (with respect to the image range) and defining the width of the disk.*

3.6 Method Evaluation

Rotation Invariance

To illustrate the ideas on the sampling of the sieve presented above, the closing-sieve is applied to 13 rotated versions of a synthetic image (Figure 3.1, left). It is a superposition of two line patterns, the finer one forming squares half the size, and with a gray-value 50% lower, than the coarser one. The position of the lines that compose the image have been distorted by white noise (standard deviation of 1 pixel).

Figure 3.8 shows size distributions for this image, as calculated using the closing-sieve without shifting the structuring element. The various distributions result from choosing different subsampling factors (1, 2, 4 or 8 times, with interpolation) and scale-sampling densities (1, 2 or 4 samples per octave). The best precision is obtained when the image is subsampled eight times. However, the discretization error is accentuated by a scale-sampling rate that is too high. That is, the finer the scale sampling, the more influence the discretization errors have on the distribution. The reason is that the difference in the area of the disks between subsequent scales is smaller for a finer scale sampling; this difference should always be larger than the discretization error. This gives a relation between the accuracy of the sampling of the structuring element, and the number of scale samples that can be computed.



a: SE b: *Input* c: $A = 128$ d: $A = 260.86$ e: $A = 512$ f: $A = 1024$

Figure 3.7: *Effect of the scaling of the band-limited disk. From left to right, the structuring element used, the input image (gray-value of 255 in the middle pixel), and the result of the dilation after scaling the structuring element by various quantities. Note how the scaling of the structuring element needs to be larger than the image range, or else its edge becomes part of the structuring element. Also note that the apparent size of the disk in the results becomes smaller as the scaling of the structuring element increases.*

We repeated this experiment using the shifted disk as structuring element. The optimal shift of $(0.19, 0.31)$ found through Figure 3.4 was used. The results are shown in Figure 3.9. It can clearly be seen that this greatly improves the precision of the method, especially at small scales. Using such a disk, an subsampling factor of 2 might be enough to obtain the precision at small scales otherwise only obtained with an subsampling factor of 8. The improvement is insignificant for the very large scales.

We repeated the above experiment again using band-limited disks as structuring elements, but without subsampling in the input image. Figure 3.10 compares the results with those of the binary structuring elements without subsampling. Band-limited disks yield a better precision than binary disks, but only for scales starting at about 4. Smaller disks are ill-defined (they cannot be sampled correctly, and therefore produce erroneous results). Even though a band-limited disk is sampling invariant, meaning it can be resampled on a different grid (e.g. a shifted grid), computing a granulometry with a shifted gray-value structuring element produces different results. This is because the morphological operation itself is not sampling invariant. The size distribution obtained by a shifted gray-value disk is more precise than that obtained by one centered on the sampling grid.

Precision

A better approximation to rotation invariance does not necessarily yield higher discriminating qualities. To determine if this is the case, we defined band-limited objects with a fixed length, a random orientation, and a random width taken from a log-normal dis-

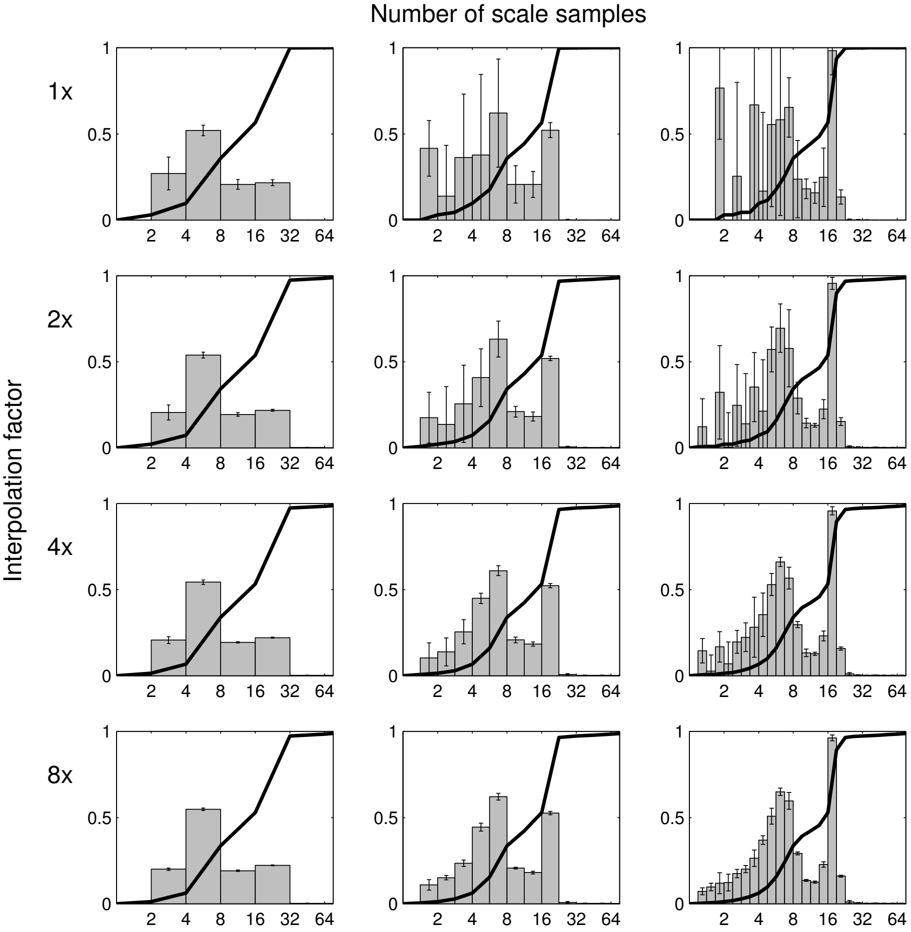


Figure 3.8: *Size distribution measured using the sieve with different subsampling factors (with interpolation) and scale sampling densities (*1, 2 *and* 4 *samples per octave), and averaged over results on* 13 *rotated versions of the test image in Figure 3.1. The structuring element used is a disk centered on the origin of the grid. The size of the detected dark objects is plotted on the x-axis, whereas the fraction of detected objects is plotted on the y-axis. The continuous line represents the average measured cumulative distribution. The error bars indicate the standard deviation.*

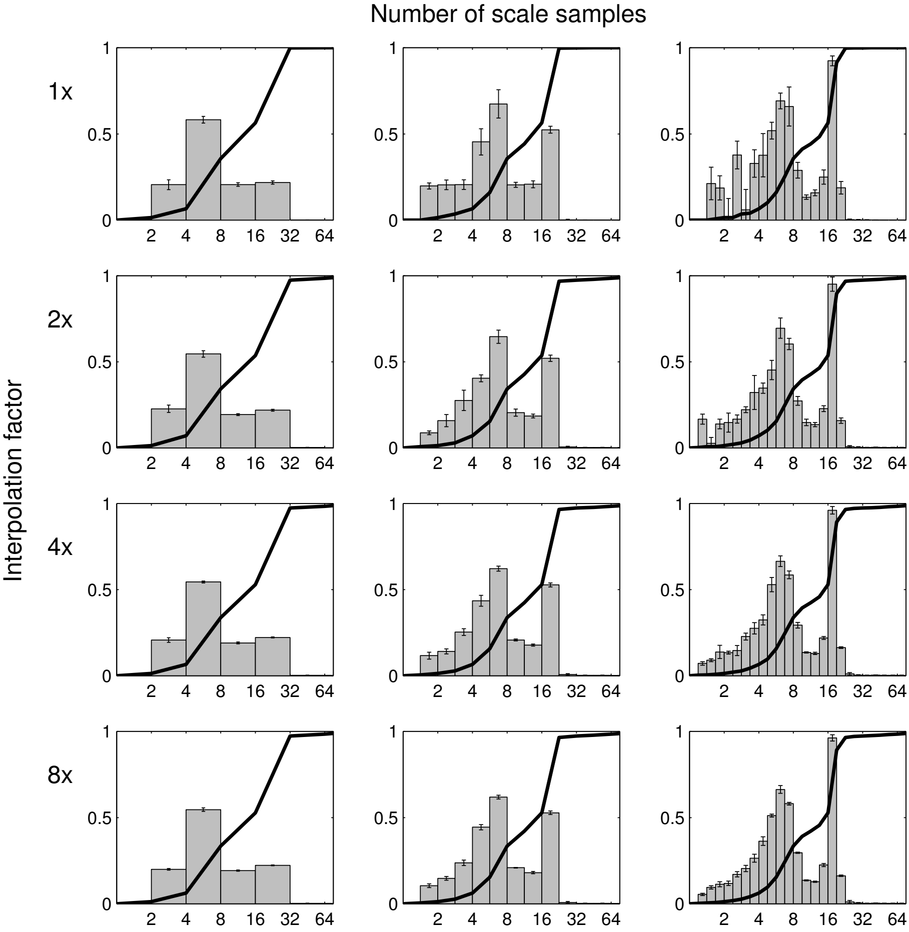


Figure 3.9: *Measured size distribution, as in Figure 3.8, using a structuring element optimally placed with respect to the sampling grid.*

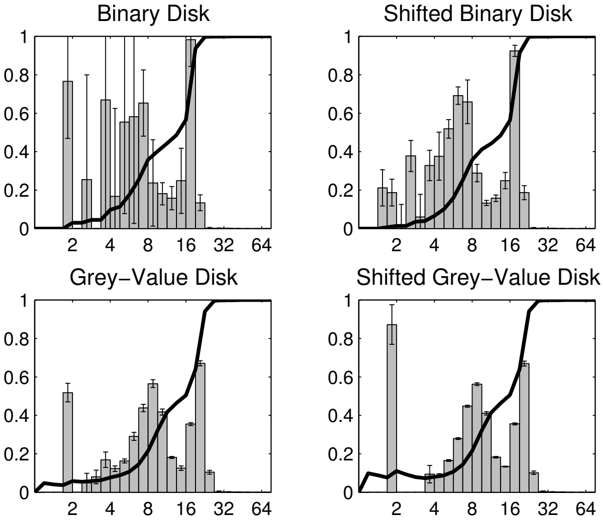


Figure 3.10: *Size distributions estimated using different implementations of the flat disk as structuring element. The top two graphs are identical to the top-right graph in Figures 3.8 and 3.9 respectively. The two bottom ones are obtained in the same way, but using band-limited disks as structuring elements. Notice how shifting the disk (bottom right) is also beneficial in the case of band-limited disks. The granulometries are not correct for scales up to about* 4 *pixels, because band-limited disks of those sizes cannot be sampled correctly.*

tribution. The granulometry measures this width. We created 1000 images of each of two classes, as in Figure 3.11, each containing 64 objects. The two classes are described by the distributions of the widths: $2^{N(\mu,\sigma^2)} = 2^{N(1.500,0.250)}$ and $2^{N(1.515,0.250)}$ (in pixels). The difference between the classes is thus a 1% shift in the mean of the logarithm of the width. We then estimated the size distribution of these objects is using a granulometry with four samples per octave, using

1. the non-modified closings, and
2. the closings modified by shifting the structuring element and subsampling (with interpolation) the input image four times (we did not use variable subsampling factors in this case).

On each estimate we fitted a volume-weighted log-normal distribution yielding an estimated log-mean value for each image.

We computed the estimated mean with a 95% confidence interval for each class based on 300 estimates randomly chosen from the set. If the confidence intervals do not overlap, the granulometry was able to distinguish the two classes based on 300 images of

Table 3.1: *Ability of the non-modified and modified granulometries to distinguish two size distributions with a very small difference in the means, based on the average over 300 estimates of each class. The modified granulometry distinguishes about 68%, whereas the non-modified one distinguishes about 61%.*

		non-modified		
		yes	no	totals
modified	yes	3022	371	3393
	no	5	1602	1607
	totals	3027	1973	5000

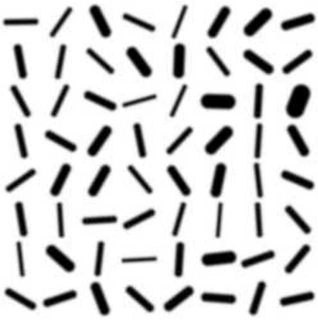


Figure 3.11: *One synthetic image from the sequence used to demonstrate the improved discriminating abilities of the granulometry with the proposed modifications.*

each class.[6] We determined for how many of 5000 such random subsets the improved granulometry and the standard granulometry were able to distinguish the two classes. The results are shown in Table 3.1. In 7.42% of the cases the improved granulometry was able to distinguish the classes where the standard method was not. In 0.10% of the cases the reverse was true. In total, the improved granulometry was able to distinguish the two classes in about 68% of the cases.

[6] 300 images seems to be the amount necessary to get a reasonable chance of distinguishing the two classes. With 200 images about 30% of the sets yield a positive result; with 500 images both methods achieve near 100% distinction rate.

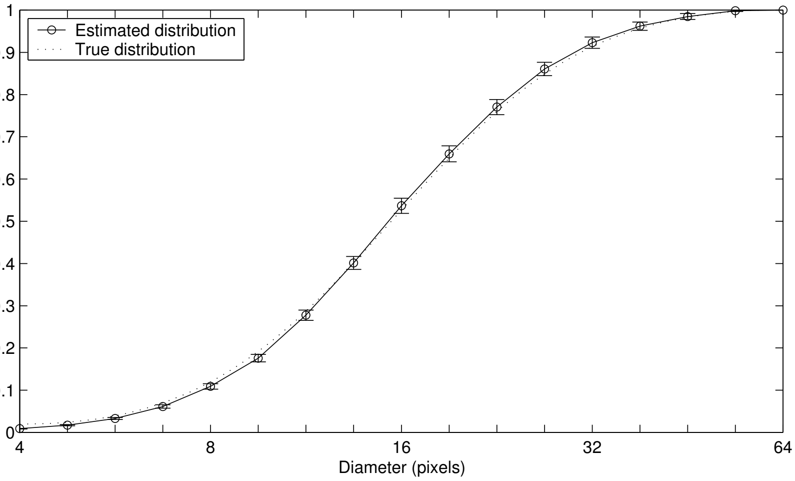


Figure 3.12: *Estimated and true size distributions for a sequence of test images similar to that in Figure 3.11. See text for more details.*

Accuracy

A similar experiment to the one above can be used to measure the bias of the estimated granulometries. We used a sequence of 200 images similar to the one in Figure 3.11. This time each image contains 16 objects whose widths are given by the distribution $2^{N(2.5,0.8)}$. We plotted the averaged estimated distribution, together with the 95% confidence interval and the theoretical distribution, in Figure 3.12. We used a shifted binary structuring element with subsampling. Note that the distributions are volume-weighted. Also, the distribution is nulled for diameters larger than 57 pixels, to make sure that the shapes do not overlap when generating the images. The plotted theoretical distribution was convolved with a small Gaussian kernel to match the smoothing in the band-limited objects (note that, because the objects are band-limited, their volume is spread out over a small set of scales; see page 61). The changes introduced by this in the distribution are noticeable only at the smaller scales because of the logarithmic scale axis. This smoothing is only an approximation to the actual change in the distribution caused by the band-limitness of the objects.

The two curves are in good agreement with each other except in a small region around 9 pixels diameter (and at 4 pixels as well, but the differences there are too small to be seen on the graph). The differences at these points are probably due to the approximation taken when computing the theoretical distribution (i.e. the smoothing we applied).

When a distinguished but elderly scientist states that something is possible, he is almost certainly right;
when he states that something is impossible, he is very probably wrong
(Arthur C. Clarke)

Chapter 4

Sampling-Free Morphology on One-Dimensional Images

As mentioned in the introduction, band-limited images, when sampled properly, do not loose any information in the sampling process. One of the consequences is that the original, continuous-domain function can be reconstructed from the samples. Another consequence is that linear filters[1] can be applied to the sampled image to obtain a result that is identical to that obtained when filtering the continuous-domain image. However, non-linear filters usually cannot be implemented in such a way that the results are comparable to continuous-domain filters. Morphological filters are no exception, as testifies Figure 4.1. We already saw some techniques in Sections 3.4 and 3.5 to better approximate the continuous-domain morphology. Other authors have suggested techniques involving curve evolution [25, 102]. In this chapter we go a step further and implement an algorithm that applies morphological operations (dilation, erosion and their combinations) to a continuous representation of a signal. The result of this operation is no longer an approximation, but an exact result. Alas, creating the continuous representation from samples does introduce errors. The main reason for this is that, because the whole signal is not available (the sampled image is finite in extent), it is not possible to use an ideal interpolator.

The *sampling-free morphology* introduced here is applied to a continuous representation of the signal, yielding again a signal in that same representation. It is possible to sample this signal, but this would introduce aliasing because it is not band-limited. For some operations, like the granulometry (Chapter 3), this is not a problem, because the result of the operation needs to be integrated to obtain a measurement result. That is, it is not necessary to obtain a sampled result. The continuous representation used in this chapter

[1] And some non-linear operations as well.

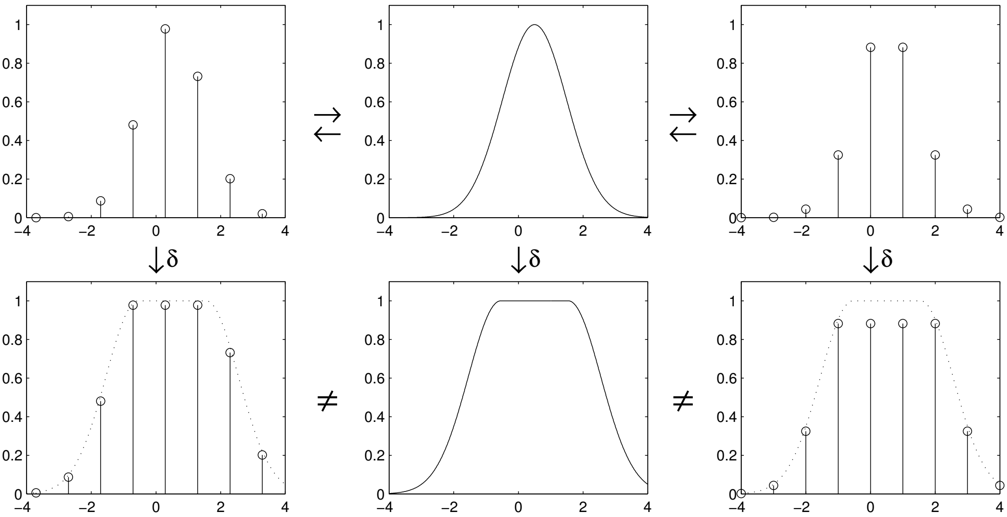


Figure 4.1: *The discrete dilation is not translation invariant, as this example shows. In the middle of the top row is a continuous signal. We sample this signal twice, using uniform sampling, but with a different offset of the sampling grid. We are still able to recreate the original signal from both these instances, but the results of the dilation are different. Neither result is the same as a sampled version of the result of the continuous dilation.*

allows for the computation of the integral of the signal. We will be putting this to use when evaluating the algorithm.

Section 4.1 describes the continuous representation, Section 4.2 explains the algorithm for the dilation, and Section 4.3 shows how to apply it to compute the erosion, closing and opening. The algorithm is described for one-dimensional signals only. In Section 4.5 we summarize our ideas about a possible multi-dimensional implementation, although it has not been implemented. Although possible, the implementation would be quite complex. In Section 4.4, the one-dimensional algorithm is applied to some test images to show the advantage of the algorithm over the classical discrete morphology.

4.1 Continuous Representation of a Signal

To reduce the sampling error of morphological operations, a continuous representation of the signal is required, a function $f : \mathbb{R} \to \mathbb{R}$ defined on an interval $[x_0, x_N]$. We must be able to

- represent band-limited signals accurately,
- represent signals with discontinuities in the first and higher derivatives, and
- obtain such a representation from a set of given samples.

We propose to use a piece-wise polynomial function, which is easy to work with. If we limit ourselves to third-order polynomials, zero-crossings, maxima and minima can be found analytically. Also, it is possible to construct a good approximation of a band-limited function with third-order polynomials [130].

Representing a 1D Signal as a Piece-Wise Polynomial

To represent a continuous one-dimensional function as a set of third-order polynomial segments, the following information is required:

- Starting point of each polynomial (x_i)
- Polynomial coefficients (a_i, b_i, c_i, d_i)
- Length of each polynomial (l_i)

Since the function we are representing is defined everywhere in the signal domain, the end point of a polynomial is equal to the starting point of the next one. Thus the length is redundant, and we only need to store the starting points of each polynomial and the end point of the last polynomial. The function is then written as a collection of segments $S_i(x)$

$$S_i(x) = a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3 \quad , \tag{4.1}$$

$i \in [0,1,2,...N-1]$, plus a right bound x_N.

Certain operations on such a representation are trivial. For example, shifting the function just requires incrementing or decrementing the starting points x_i, and negating the function is accomplished by negating all the polynomial coefficients. Other operations we apply to the polynomial function are sampling (evaluating the function at chosen locations) and integration. The integral over the function is the sum of the integral over each segment, determined by

$$\int_{x_0}^{x_N} f(x)dx = \sum_{i=0}^{N-1} \tfrac{1}{4}d_i x_{i+1}^4 + \tfrac{1}{3}c_i x_{i+1}^3 + \tfrac{1}{2}b_i x_{i+1}^2 + a_i x_{i+1} \quad . \tag{4.2}$$

Converting the Sequence of Samples into a Piece-Wise Polynomial

To create the piece-wise polynomial representation $f(x)$ from the given samples $f[n]$, we require an interpolation function that has certain characteristics:

– The resulting function must have as many continuous derivatives as possible (since the original band-limited signal is infinitely differentiable). We use third-order polynomials, thus we require that the second-order derivative be continuous.

– It must be a local representation. That is, the zone of influence of a single pixel must be limited, because only a limited number of samples is available.

– It must be capable of producing a polynomial representation.

An interpolator that satisfies these constraints is the cubic spline interpolator [20, 26, 131]. It produces polynomial segments in between each two sample points. Although its impulse response decays quite quickly, it requires a filter with an infinite impulse response (IIR) to determine the polynomial coefficients. This IIR filter can be implemented recursively [129]. Note that a spline of infinite order equals the ideal interpolator (the sinc function) [130]. Thus, a cubic spline is an approximation of the ideal interpolator.

When the input samples are noisy, it might be better to use a least squares spline [131]. In this case, the reconstructed function does not need to be equal to the samples at the sample locations, and thus can be smoother. Furthermore, by computing the piecewise polynomial in this way the number of pieces is reduced, which makes subsequent processing faster as well.

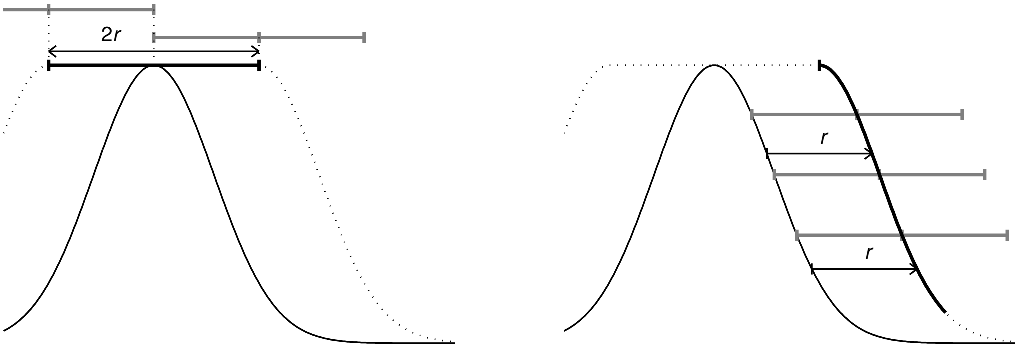
4.2 Sampling-Free Dilations

Examining the 1D dilation operation with a flat, compact structuring element B, one can readily see that the result is composed of plateaus (constant sections) as well as slopes with the exact same shape as can be found in the input signal (see Figure 4.2). Let us define the set B as

$$B = \{x | x \in [-r, r]\} \quad . \tag{4.3}$$

The plateaus are formed when, at a point x, the maximum value over the neighborhood B comes from a local maximum[2] (see Figure 4.2a). At points near x, the maximum over the neighborhood will also come from the same local maximum, and will therefore receive the same value. These plateaus will have a width of at most $2r$, centered on the local maximum.

[2] We do not distinguish regional and local maximum, using the latter term for both.



a: *Creating plateaus*

b: *Shifting slopes*

Figure 4.2: *Construction of the dilated function.* a: *Maxima create plateaus in the output.* b: *Slopes are shifted by a fixed distance, dictated by the size of the structuring element. In these graphs, the thin black line is the input signal, the dotted black line is the output signal, the thick black line shows how the output signal is constructed, and the thick gray lines give the size of the structuring element (*$2r$*).*

The sloped regions are produced when the maximum over B does not come from a local maximum. In this case, it must come from the border of the structuring element (see Figure 4.2b). At nearby points, the resulting value also comes from the same edge of the neighborhood. Therefore, a slope is created that is an exact copy of a slope from the input signal, shifted by r or $-r$. This is a result from the slope transform [19, 32, 76, 77].

Thus, for a one-dimensional signal, the output of the dilation with a flat, compact structuring element is the point-wise (or, in our case, the segment-wise) maximum of three functions:

– the input signal translated by r: $f(x-r)$,

– the input signal translated by $-r$: $f(x+r)$, and

– a signal composed of plateaus centered around each of the local maxima.

Note that approximately three comparisons for each output segment are required: one when creating the plateau function, and two when taking the maximum over the three segment functions. This is consistent with the van Herk algorithm, which requires three comparisons for each output pixel [46]. Creating the plateau function requires less than one comparison for each output segment because it typically contains far fewer segments than the input image.

The above analysis is valid for flat, compact, symmetric structuring elements. Any non-symmetric, flat, compact structuring element C, defined by

$$C = \{x | x \in [-r-d, r-d]\} = \{x | x+d \in [-r, r]\} \quad , \tag{4.4}$$

can be converted into a symmetric structuring element B by translating the input or the output signal:

$$\delta_C(f) = \delta_B T_d(f) = T_d \delta_B(f) \quad . \tag{4.5}$$

A non-compact structuring element can be constructed with the union of compact structuring elements:

$$\delta_{[\bigcup_i B_i]}(f) = \bigvee_i \delta_{B_i}(f) \quad . \tag{4.6}$$

Thus, the above analysis suffices for any flat structuring element.

Extending the algorithm for gray-value structuring elements is more complex, but might be accomplished through the slope transform [19, 32, 76, 77]. We have not pursued this.

Creating the Plateau Function

Creating the function composed of the plateaus requires that all local maxima be found. This is accomplished by examining the first and second order derivatives of each of the polynomials:

$$S_i'(x) = 0 \quad \wedge \quad S_i''(x) < 0 \quad \Leftrightarrow \quad x \text{ is a local maximum.} \tag{4.7}$$

Note that these derivatives can be obtained analytically from the spline representation. Finding the zero crossings of the first derivative is accomplished by solving a quadratic equation. Additionally, in the result of a previous morphological operation there can be maxima in the form of cusps and plateaus. These will be found only on knots (boundary points between polynomial segments), and are identified by comparing the derivatives of both polynomials at those points:

$$S_i'(x_{i+1}) \geq 0 \quad \wedge \quad S_{i+1}'(x_{i+1}) \leq 0 \quad \Leftrightarrow \quad x \text{ is a local maximum;} \tag{4.8}$$

if they are both equal to 0, x_{i+1} might be on a plateau that forms a local minimum. It is not necessary to specifically detect this situation, because it will not influence the result in any way.

Each maximum found generates a plateau of size $2r$. The plateau function is obtained by taking a portion of each of these segments such that, where they overlap, only the one

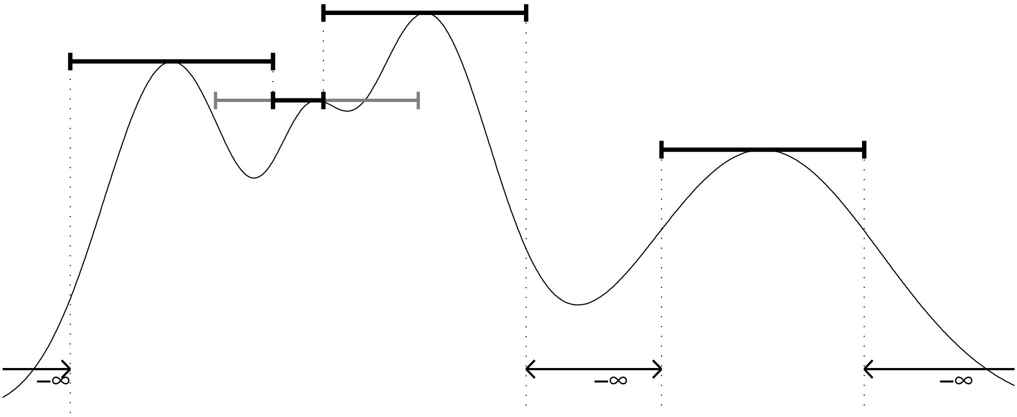


Figure 4.3: *Construction of the plateau function. At each local maximum a plateau (0^{th} order polynomial) with the size of the structuring element is set. In the case of overlapping plateaus, the one with the highest value is kept intact; the other one must be cropped. Empty regions are filled with segments of value $-\infty$ so that the function is defined everywhere in the signal domain.*

with the largest value is retained (see Figure 4.3). To do this, we sort all found maxima according to their value, largest first. We then add a 0^{th} order polynomial segment, ranging from $x-r$ to $x+r$, and with value $f(x)$, for each maximum at x. Each segment added should not overlap with any of the polynomials already present in the function, so it must be cropped to the available space. At the end of this process, eventual 'holes' must be filled with segments of value $-\infty$, so that the generated function is defined everywhere in the signal domain, and can be stored in the same manner as the input signal.

Computing the Maximum over the Segment Functions

The last step is to find the function that is the maximum of the three functions. This is a two-step process in which first two functions are compared, and then the result is compared to the third. To avoid complicated exceptions in the algorithm, we pad the three functions with zero-order polynomials so that they span the same interval (from x_0-r to x_N+r). The translated versions of the input signal are extended with the edge value (so as to keep them continuous). The function containing the plateaus is extended with $-\infty$.

This function comparison is very simple, but potentially generates quite a lot of segments. For each (portion of a) segment $S_i^1(x)$ in one function that spans the same region as another (portion of a) segment $S_i^2(x)$ in the other function, the intersection points $S_i^1(x) = S_i^2(x)$ must be found (this is a cubic equation, the solution can be found in Bronstein [26]). There are up to three intersection points, and thus up to four sub-segments. For each of these, the polynomial with the larger value is used to construct the output signal.

4.3 Sampling-Free Erosions, Closings and Openings

Since the erosion is the dual of the dilation, it can be implemented by inverting the signal, applying the dilation, and inverting the result again:

$$\varepsilon_B(f) = -\delta_B(-f) \quad . \tag{4.9}$$

As stated above, inverting the piecewise polynomial function is easily accomplished by negating all the polynomial coefficients.

The closing ϕ is created by applying an erosion to the result of the dilation,

$$\phi_B(f) = \varepsilon_{\breve{B}}[\delta_B(f)] \quad , \tag{4.10}$$

and the opening γ is constructed the other way around,

$$\gamma_B(f) = \delta_{\breve{B}}[\varepsilon_B(f)] \quad . \tag{4.11}$$

The algorithm as described above can be applied to its own result, so that implementing closings and openings becomes trivial.

4.4 Method Evaluation

A First Examination of the Algorithm

We extracted a line out of a band-limited image to apply our methods to. Figure 4.4 shows two portions of this line, along with the reconstructed continuous function, the result of a discrete dilation (i.e. one applied to the samples directly) and that of the sampling-free dilation proposed here. Figure 4.5 shows the results of the discrete and sampling-free closings on the same signal.

In these figures we can see that the sampling-free dilation reaches higher values than the discrete variant at some points, especially on plateaus. The value of this signal at these points is equal to the value of the true local maximum of the input signal (or rather of the cubic spline approximation). Likewise, the closing has higher values at the plateaus (the continuous version is equal only in exceptional cases). The differences would be larger if the structuring element did not have a length equivalent to an integer number of pixels.

Granulometry

We created a signal of which we know the function that represents the granulometric curve. To the samples of this signal we applied a granulometry (as in Chapter 3) with both the sampling-free and discrete closings, and compared the results to the theoretical granulometric curve.

The signal we used is a sine,

$$f(x) = sin\left(\frac{2\pi x}{T}\right) \quad , \tag{4.12}$$

with T the period. The sampling distance is 1, meaning that T must be larger than 2 for error-free sampling. The theoretical granulometric curve is described by

$$h(r) = \begin{cases} \frac{1}{\pi} sin\left(\frac{r\pi}{T}\right) - \frac{r}{T} cos\left(\frac{r\pi}{T}\right) & \text{for } r < T, \\ 1 & \text{for } r \geq T, \end{cases} \tag{4.13}$$

with r the size of the structuring element. We used two periods: $T_1 = \frac{200}{9}$ and $T_2 = \pi$. These values were chosen so that each period of the sine has a different offset with respect to the sampling grid. Both signals can be correctly sampled at a rate of 1. The first one can be interpolated very accurately using cubic splines, whereas the second will produce larger errors due to the inability of the spline to correctly reconstruct high-frequency signals (see Figure 4.6). The frequency characteristic of the cardinal cubic spline can be found in [131, Figure 2]. The spline interpolation on the second signal produces a result that is obviously not an exact reproduction of the input signal. Therefore, the result of the granulometry is inaccurate as well. However, it lies much closer to the theoretical curve than the result of the discrete granulometry. Another obvious drawback of the discrete granulometry is the discreteness of the structuring element, which can only be constructed with integer lengths. Because of this, the granulometry with the sampling-free closing could be sampled much more densely.

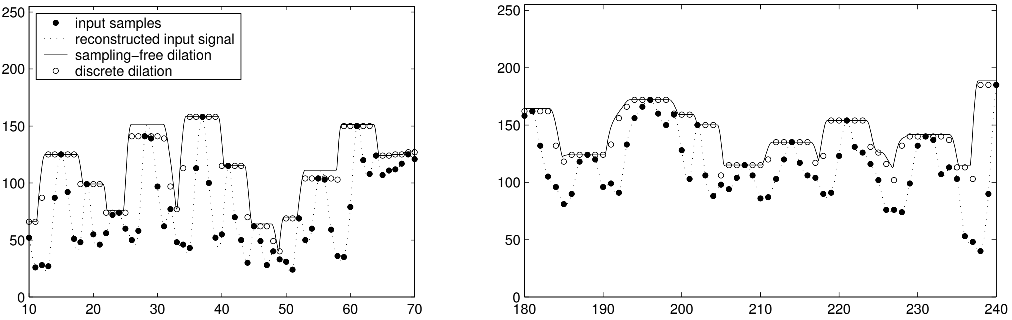


Figure 4.4: *Two interesting portions of a 1D signal, together with its sampling-free dilation. The open dots give the values of the discrete dilation for comparison. The structuring element has a length of* 5 *pixels.*

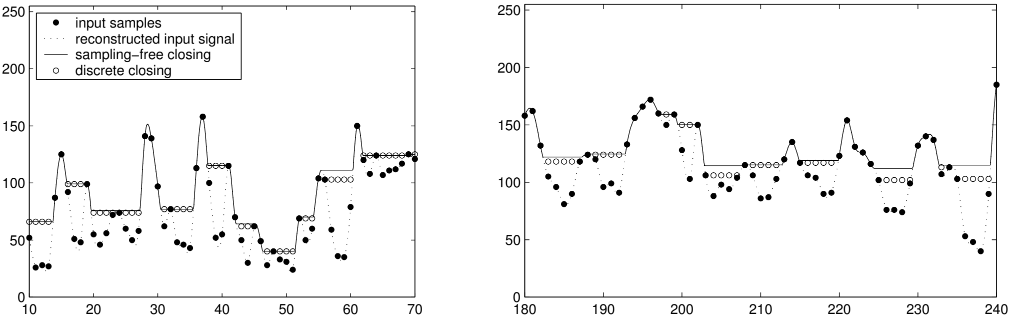


Figure 4.5: *Two interesting portions of a 1D signal, together with its sampling-free closing. The open dots give the values of the discrete closing for comparison. The structuring element has a length of* 5 *pixels.*

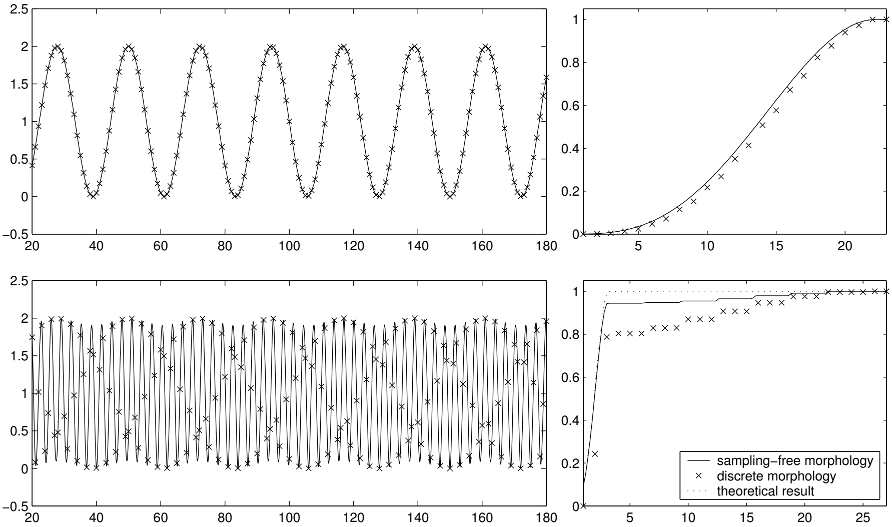


Figure 4.6: *Granulometry of a sine function sampled at different rates. On the left are the samples and the continuous function created with cubic splines. On the right is the result of the granulometry, computed with both discrete and continuous-domain morphology, compared to the theoretical granulometric function. For the top signal, the theoretical and sampling-free granulometries match exactly; therefore, the dotted line is hidden by the continuous one.*

We repeated the above experiments after adding noise to the input samples (see Figure 4.7). The results show more or less the same characteristics, except that the granulometric curves deviate a bit more from the theoretical (noiseless) values.

4.5 Extension to Multi-Dimensional Images

Morphological operations can be defined for images of any dimensionality. Therefore, we would like to extend our algorithm to multi-dimensional images as well. This is, however, not an easy task.

Obviously, extension to multi-dimensional images by processing each dimension separately will not work. In this case, maxima lying in between raster lines will be missed. This shows that it is necessary to create a patch representation of the image (using multi-dimensional cubic splines), and work on that.

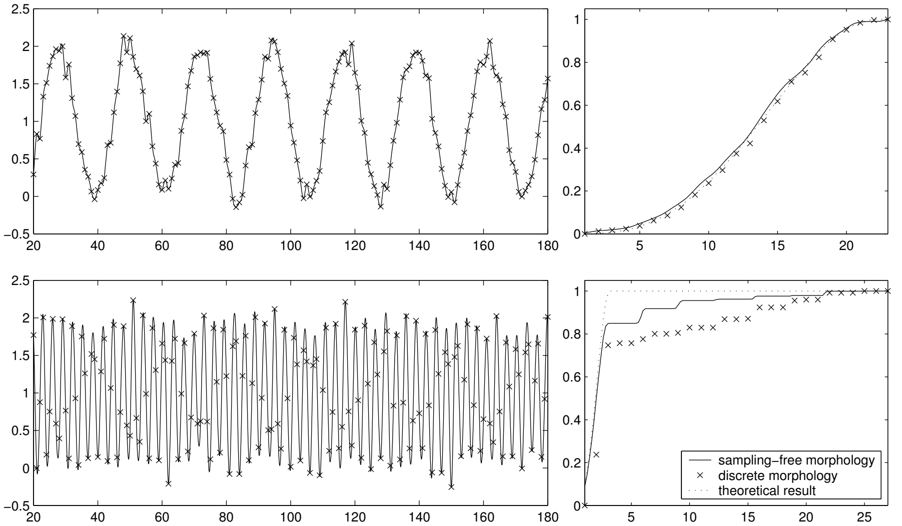


Figure 4.7: *Granulometry of a sine function sampled at different rates. Noise was added to the samples prior to the analysis. On the left are the samples and the continuous function created with cubic splines. On the right is the result of the granulometry, computed with both discrete and continuous-domain morphology, compared to the theoretical granulometric function of the noiseless signal.*

However, using multi-dimensional structuring elements with this representation also introduces a problem: we would need to create a translated version of the input image for each point along the boundary of this structuring element. Since there are an infinite number of these points, this is an impossible task. If we would simplify the structuring element by taking only a limited number of points along the contour, we would again miss some of the local maxima.

Thus, we should limit ourselves to one-dimensional structuring elements working on a patch representation of a multi-dimensional image. Now we can implement the same operations we suggested above: create two translated versions of the input image (one for each end of the structuring element), and an image consisting of plateaus centered around each of the local maxima in the image; then take the maximum over these three images. In this case, however, local maxima are all points for which there is a maximum in the direction of the structuring element. These points form lines (in a two dimensional image; in an n-D image this is a $(n-1)$-D plane). The plateaus we create are therefore patches with a zero-order polynomial in one direction, and some third-order polynomial in all the orthogonal directions. These polynomials are taken from the input patches along the local maxima line.

The only problem with this approach is that the patches produced by a dilation have very complex boundaries (given by third-order polynomials). This makes an implementation difficult, although the mathematical description remains relatively easy.

As seen in Section 1.1, using linear structuring elements it is possible to create more complex multi-dimensional structuring elements such as the rectangle, the hexagon, the octagon, etc. These shapes are increasingly better approximations of a disk. Thus, it would be possible to create an arbitrarily accurate approximation to the isotropic structuring element using multi-dimensional sampling-free morphology.

Experiment and theory often show remarkable agreement when performed in the same laboratory
(Daniel Bershader)

Chapter 5

Discrete Morphology with Line Structuring Elements

We have seen in Section 1.1 that many structuring elements can be decomposed into line segments. For example, the square, hexagon and octagon, which are increasingly accurate approximations of the disk, can be decomposed into two, three and four line segments, respectively. Thus, it is possible to create an arbitrarily accurate approximation of a disk by increasing the number of line segments used. The advantage of using lines instead of n-dimensional structuring elements is a reduction in the computational complexity. Furthermore, it is possible to implement a dilation or erosion by a line segment under an arbitrary orientation with only 3 comparisons per pixel, irrespective of the length of the line segment, using a recursive algorithm [46, 118].

Adams [2] showed how to create an optimal discrete disk using dilations with line segments. These disks are only approximations of the sampled Euclidean disk. The optimality is a trade-off between accuracy and efficiency. For multi-scale closings with these disks, however, absorption does not hold. Jones and Soille [54] improved on this by using periodic lines, so that the absorption property is satisfied. Nonetheless, these structuring elements sacrifice accuracy to gain implementation efficiency. The results developed in Section 3.4 are not valid for disks implemented in this way.

The morphology introduced in Chapter 2 also uses line segments. It is for these operators that we study the implementation of discrete line segments here. Examples of other applications of line structuring elements are roads in airborne images [27, 55], grid patterns on stamped metal sheets [128], and structure orientation estimation [119, 120]. We start by introducing Bresenham lines, the basic discrete lines. The most simple implementation of the dilation uses a Bresenham line segment as structuring element. For efficiency purposes, one might compute regional maxima over a Bresenham line across the image (using the recursive algorithm mentioned above). The drawback is that this operation is not even translation invariant in the discrete sense (i.e. invariant over integer pixel shifts). Jones and Soille [54] introduced periodic lines, which are studied in Section 5.2. Using periodic lines, it is possible to construct recursive dilations that are translation invariant in the discrete sense. After that we introduce operations obtained by interpolating the image to obtain regional maxima over line segments (Sections 5.3 and 5.4), and a gray-value structuring element that implements an approximately band-limited line segment (Section 5.5).

All of these approaches are compared in Section 5.6. They are then tested for rotation invariance and translation invariance. Section 5.7 addresses the problem we mentioned earlier in Chapter 2: we cannot compute an operation under all orientations (the orientation-axis of the RIA morphology must be sampled as well); how many operations do we need to compute?

The sampling-free morphology introduced in Chapter 4 should be able to implement morphological operations along arbitrary line segments without error. We leave this method out of the comparison since we do not have an implementation for multi-dimensional images.

5.1 Basic Discrete Lines: Bresenham Lines

Bresenham [24] published an algorithm to draw a line segment of any orientation on a plotter that could only draw horizontal, vertical and diagonal lines. The algorithm combines small portions of these lines to form a line segment of any orientation. In image processing, *Bresenham lines* are formed by steps in the eight cardinal directions of the grid.

To efficiently implement a dilation with a line segment of any orientation, the recursive algorithm proposed by van Herk [46] can be applied to a Bresenham line crossing the image [118], as in Figure 5.1 (lines can be tiled to cover the whole image). This results in, at each point, the maximum over some pixels (along the line) at each side of that point. The problem is that, for neighboring pixels, the configuration of this neighborhood is different. Take as an example a line that goes up one pixel for each two that it goes right. Such a line is drawn by making one step right and one diagonally up (see Figure 5.2). There are two ways of starting this line (one of the two steps must be taken first), so that each pixel along this line is embedded in one of two different neighborhoods. The dilation along this line will therefore be computed with two different structuring elements (both versions are an equally good approximation of the continuous line segment), the results of both alternated from pixel to pixel. When the image is translated horizontally

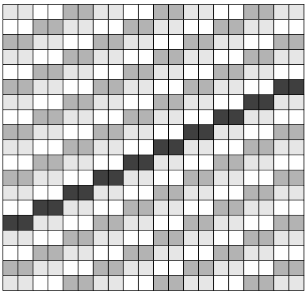


Figure 5.1: *A Bresenham line across the image can be tiled so that each pixel in the image belongs to a single line. Along these lines it is possible to compute the dilation (or any other operation).*

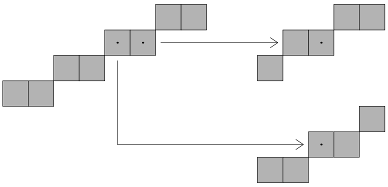


Figure 5.2: *The problem with a Bresenham line is that each pixel along the line is embedded in a differently shaped neighborhood. Each of these neighborhoods are equally good approximations of the continuous line segment.*

by one pixel, and translated back after the operation, a different result is produced than when the operation is computed without translation.

Only the horizontal, vertical and diagonal lines can be used to compute dilations that are translation invariant (in the discrete sense). For all other orientations, the shape of the structuring element changes from point to point in the image. This should not pose a significant problem for band-limited images. All shapes used are equally poor approximations of the continuous line segment. The error introduced because of this outweighs the problems caused by the shape-change due to the recursive implementation.

We implemented this method by skewing the image in such a way that all pixels belonging to the Bresenham line are aligned on a row (or column, depending on the orientation of the line) of the image (that is, each column is shifted by an integer amount of pixels). On this skewed image the operations can be applied along the rows, and the result must be skewed back.

Soille and Talbot [120] proposed to use the intersection of the closings (or the union of the openings) along all possible Bresenham lines of the desired orientation. In the example above, where there are two possible Bresenham lines representing the same continuous line; this would be the minimum of two closings. Using this method, discrete translation invariance is assured. But, in relation to the goal we set in the Introduction, this operation is not satisfactory. Most importantly, the operation is still not translation invariant in the continuous-domain sense. Also, the width of the line changes according to the orientation with respect to the sampling grid: the more possible Bresenham lines exist for the desired orientation, the thinner the structuring element will appear (this is because the operation is less restrictive, and the line segment is allowed to "wiggle" in between the image features). Less importantly for us, this method is not applicable for dilations or erosions (since the intersection of dilations is not a dilation and the union of dilations leads to a dilation with a thick line segment).

Another problem with the discrete line segment (whether implemented with a recursive algorithm or not) is that the length, defined by an integer number of pixels, depends on the orientation of the segment. For each orientation, there is a different set of lengths that are possible to construct. As the same time, the number of samples per unit length changes with the orientation.

5.2 Periodic Lines

Periodic lines were introduced by Jones and Soille [54] as a remedy to the lack of (discrete) translation invariance of the morphological operations along Bresenham lines. A periodic line is composed of only those points of the continuous line that fall exactly on a grid point, see Figure 5.3. These lines are thus formed of disconnected pixels, except for lines in one of the three cardinal orientations. When considering only these points, it is possible to use a recursive implementation along the periodic lines that is translation invariant in the discrete sense. However, because of the sparseness of the points along such a line, they are not useful except in constructing more complex structuring elements.[1] For example, by dilating a periodic line segment with a small connected segment, one creates a connected line segment, as in Figure 5.4. Thus, to implement a (discrete) translation-invariant dilation, one would compute a dilation with a periodic

[1] Using periodic lines instead of connected discrete lines in building a discrete disk, the absorption property is satisfied [54]. Discrete approximations to the Euclidean disk should thus only be constructed in this way to be used in a granulometry. Note that in Chapter 3 we ignored the absorption property of the discrete disk to create a more accurate granulometry, at the risk of having a (slightly) non-increasing granulometric function.

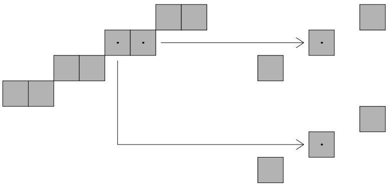


Figure 5.3: *The problem of the Bresenham line can be solved by using only a limited number of pixels on the line. This way, each neighborhood is the same, although it is no longer connected. This is a periodic line.*

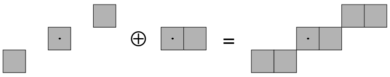


Figure 5.4: *By dilating a periodic line segment with a small structuring element, it is possible to join it up. This further limits the available lengths of the structuring element to multiples of the period.*

line segment, and on the result apply another dilation with a small connected line segment (which does not need to be implemented recursively because it is so small).

The drawbacks of this method are the small number of orientations for which it is useful (there are only few orientations that produce a short periodicity, for longer periodicities the line segment needed to connect the periodic line is longer as well), and the limited number of lengths that can be created (the length is a multiple of the periodicity, which depends on the orientation).

Because the result of this implementation is the same as that obtained by a direct (non-recursive) implementation using a Bresenham line segment as structuring element, we do not consider it separately in the comparison of Section 5.6.

5.3 Interpolated Lines by Skewing the Image

We mentioned above that operations along a Bresenham line can be implemented by skewing the image, applying the operation along a column (or row), and skewing the image back. In this section we consider image skews with interpolation (that is, the rows or columns of the image are not shifted by an integer number of pixels, but by sub-pixel quantities); see Figure 5.5.

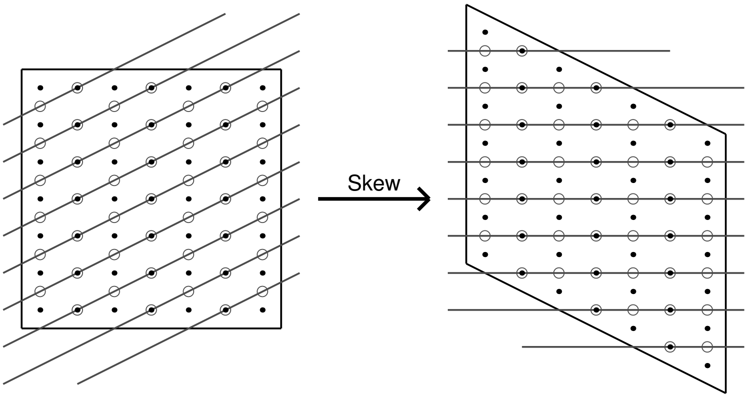


Figure 5.5: *After skewing the image, horizontal lines correspond to lines under a certain orientation with respect to the image data. Some of the original image samples (·) fall exactly on these lines, but most samples used (○) lie in between original grid points. The value at these points is obtained by interpolation.*

The interpolation method used is an important factor in the correctness of the output. The better the method is, the smaller the error will be. We used cubic convolution [56] to implement the skews. This method is a good compromise between accuracy, computational cost and window size.[2]

The lines obtained in this way are interpolated, but have the same number of samples as the Bresenham line of the same parameters. It is expected that these result in a somewhat better approximation to translation invariance. The major drawback is that the result needs to be skewed back. As stated before, morphological operations do not produce band-limited images, and therefore the results are not sampled properly. Interpolating the result of a morphological operation is questionable at best.

The reason we need to interpolate in the resulting image is that the result of the morphological operation is computed at the points along the continuous line laid across the image, and not at the grid points of the output image. There are few columns (as many as there are points in the periodic line representation for the selected orientation) with no sub-pixel shift. For these columns, interpolation of the output is not required, and the result is at its best.

[2] Remember that the image is not infinite in size, and therefore it is not possible to use the ideal interpolator. The window size is important because it determines the portion of the image affected by the border.

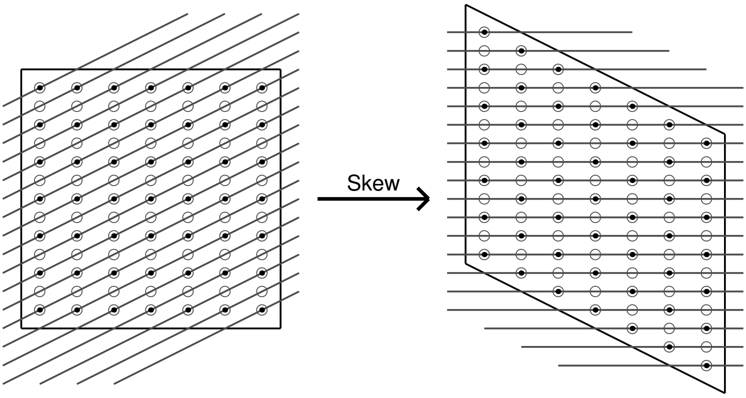


Figure 5.6: *At the expense of more computation, it is possible to directly compute each of the output columns, so that the inverse skew is not required. Not having to interpolate in the result of a morphological operation produces the most accurate result.*

To improve the result on the other columns, it might be interesting to sample the lines more densely before applying the morphological operation. This makes the inverse skew more accurate because the aliasing introduced by the operation will be less severe. In Section 3.4 we also used interpolation to increase the accuracy of morphological operations.

5.4 True Interpolated Lines

The interpolated lines presented above are at their best on only a few columns (or rows) of the image. It is, of course, also possible to accomplish the same accuracy for all output pixels. In this case, for each output pixel, samples along a line that goes exactly through it are computed by interpolation, as in Figure 5.6. On these computed samples the operation is performed.

To compute these lines somewhat efficiently, we resort again to the skew. By changing the offset of the image for the skew, it is possible to select which group of columns gets an integer shift. The result is only stored at these columns. The image must be skewed many times to compute the result for all the columns, but it is not necessary to skew back the output images. The number of skews that need to be computed is equal to the periodicity of the periodic line with the same orientation.

Again, as for all discrete line segments mentioned up to now, the number of samples used in the computation of the morphological operation depends not only on the length of the segment, but also on the orientation. Line segments along the grid are the densest, and diagonal segments have the least number of samples. Thus, for some orientations it is more probable to miss a local maximum (i.e. the maximum falls in between samples) than for others. This makes the approximation to continuous-domain translation invariance better for horizontal and vertical lines than for diagonal lines, and also has repercussions for the rotation invariance. Ideally, one would like to sample each of these lines equally densely. To do so, it would be necessary to add columns to the image when skewing. As mentioned above, this also enables the creation of sub-pixel segment lengths, in a similar fashion to the interpolation used in Section 3.4 to increase the accuracy of the isotropic closing.

Alternatively, rotating the image instead of skewing it also alleviates this problem. However, when rotating, only a limited set of samples falls exactly on output samples, and in the worst case this happens only for the sample in the origin of the rotation. This means that a larger number of operations is required to compute the result of the operation at all output pixels.

We have not corrected for the number of samples along the line segment in the comparison below.

5.5 Band-Limited Lines

A last option when implementing morphology with discrete line segments is to use gray-value structuring elements, which allows to construct band-limited lines. Such a segment is rotation and translation invariant, and does not have a limited set of available lengths. The drawback is that the line is thicker, but this should not be a problem for band-limited images, since it should contain only thick lines as well.

In Section 3.5, an approximately band-limited disk was generated with a Gaussian profile. It was stated there that a Gaussian function, as well as its integral, is band-limited in good approximation, and can be sampled at a rate of σ with only a very small error. An approximately band-limited line can be generated in a similar fashion to the disk, again using the error-function along the length of the line, and using the Gaussian function in the other dimensions.

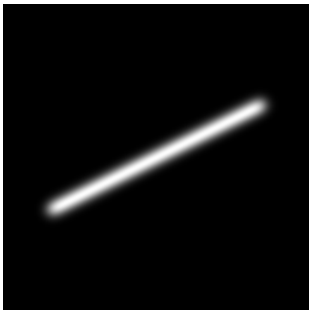


Figure 5.7: *An approximately band-limited line segment constructed with Equation (5.1).*

Let us define a two-dimensional image $L_{(\ell,\sigma)}$, to be used as a structuring element, by

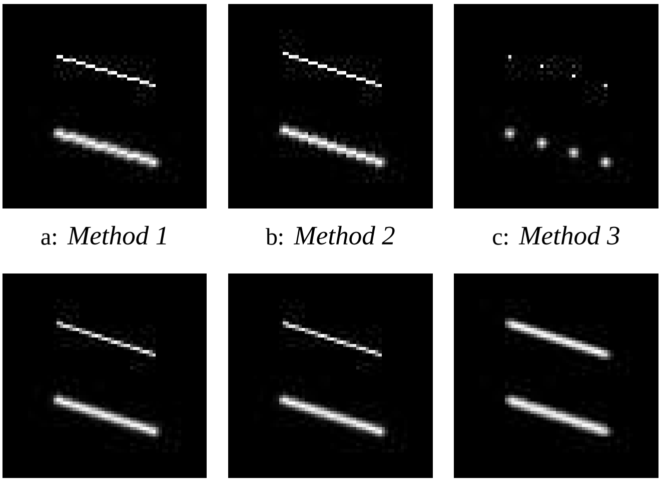
$$L_{(\ell,\sigma)}(x,y) = A \cdot \frac{1}{2}\left\{1 - \operatorname{erf}\left(\frac{\ell - 2\,|x|}{2\sigma}\right)\right\} \cdot \exp\left(\frac{-y^2}{2\sigma^2}\right) \quad , \qquad (5.1)$$

where ℓ is the length of the line segment, x is the coordinate-axis in the direction of the segment, and y is the coordinate-axis perpendicular to it. This is equivalent to an infinitely thin line convolved with a Gaussian. Again, setting σ to 1 is enough to obtain a correctly sampled structuring element. Figure 5.7 shows an example of such a band-limited line segment. Of course, generating line segments in higher-dimensional images is trivial: y needs to be substituted by a vector. Note that the gray-value of the segment is 0, and the background has a value of $-A$. Like in Section 3.5, it is required to choose an appropriate value for A, relative to the gray-value range in the image to be processed. The same conclusions apply here.

5.6 Comparison of Discrete Line Implementations

We have implemented the following versions of the dilation and the opening with a line segment structuring element:

- Method 1: with a Bresenham line segment as structuring element.
- Method 2: along Bresenham lines across the image (Section 5.1).
- Method 3: with periodic lines (Section 5.2).
- Method 4: along interpolated lines across the image (Section 5.3).
- Method 5: with true interpolated lines (Section 5.4).
- Method 6: with an approximately band-limited line segment (Section 5.5).



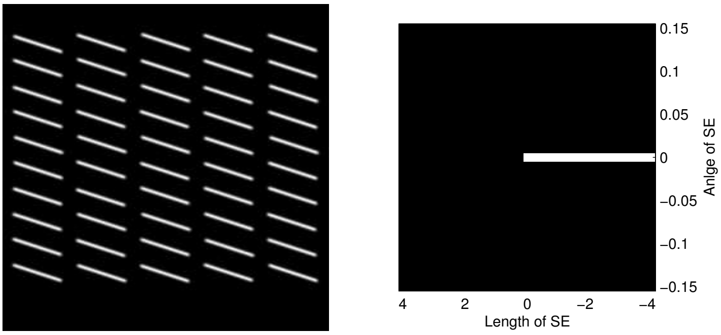
a: *Method 1* b: *Method 2* c: *Method 3*

d: *Method 4* e: *Method 5* f: *Method 6*

Figure 5.8: *Sample dilation with different implementations of the line segment structuring element. This gives an idea about the shape of the structuring element used. The input image contains a delta pulse and a Gaussian blob.*

Figure 5.8 shows the dilation with each of these methods applied to an image with a discrete delta pulse and a Gaussian blob. This figure gives an idea about the shape used in the operation. Methods 1 and 2 produce discrete line segments, whereas methods 4 and 5 produce line segments with gray-values that do not exist in the input image. As expected, using a periodic line produces a disjoint collection of points. Finally, method 6 produces the thickest, but also the smoothest, line segment.

To compare these different methods, an image was generated that contains many line segments of fixed length and orientation, but varying sub-pixel position (see Figure 5.9a). They were drawn using Equation (5.1). Openings were applied to this image, changing both the length and orientation of the structuring element, and using each of the implemented methods. The result of each operation is integrated (taking the sum of the pixel values), and plotted in a graph. It is expected that this results in a value of 1 for the openings in which the orientation of the structuring element matches that of the segments in the input image, and the length ℓ is smaller or equal to the length of these segments; the result should be 0 for any other parameter of the structuring element (as in Figure 5.9b). Since the image contains band-limited lines, it is expected that the result obtains a smooth transition from one state to the other. The more the result approximates the ideal situation, the better the specificity of the operator is.



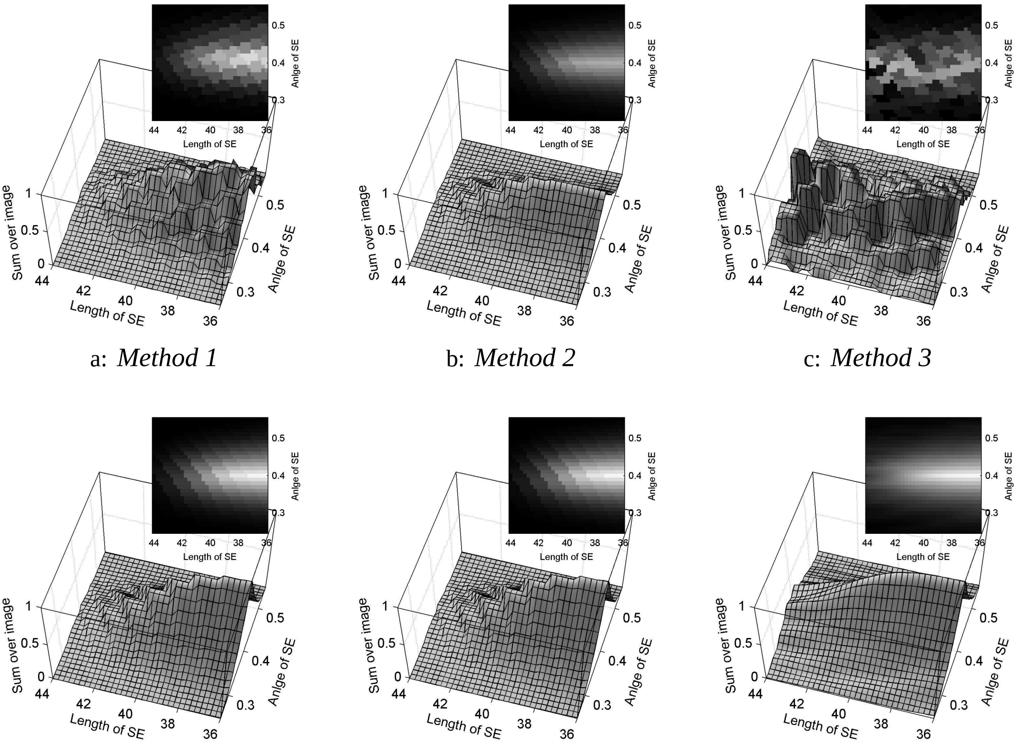
a: *Input image*

b: *Ideal response*

Figure 5.9: *Input image and ideal response for graphs shown in Figure 5.10. A value of* 1 *is expected for the openings in which the orientation of the structuring element matches that of the segments in the input image, and the length ℓ is smaller or equal to the length of these segments. A value of* 0 *is expected for any other parameter of the structuring element. Because the input image is band-limited, a smooth transition is actually obtained.*

The results are plotted in Figure 5.10. There are a couple of things that readily come to mind when comparing these graphs:

- All methods produce a similar result, with the exception of the periodic lines (method 3). This is due to the fact that the periodic line segment is disjoint, and therefore can “fit” inside two image features at once. For most of the orientations, the periodic line segment consists of only 2 points.
- The two discrete, non-interpolated implementations (methods 1 and 2), never reach values approximating 1. The interpolated and gray-value methods (methods 4, 5 and 6) reach higher values, closer to the ideal value of 1.
- The three methods that work along lines across the image (methods 2, 4 and 5) show a stair-like dependency on the length. This is because of the discretized lengths of these segments. Note that the actual length of the structuring element depends on the orientation. This dependency is less obvious in method 1 because there the orientation is discrete as well. This results in plateaus in which the structuring element does not change.
- There are very few differences between the two interpolated methods (methods 4 and 5).
- The result of the gray-value method (method 6) is very smooth, but shows some



a: *Method 1* b: *Method 2* c: *Method 3*

d: *Method 4* e: *Method 5* f: *Method 6*

Figure 5.10: *Comparison of different implementations of the opening with a line segment structuring element. See text for details. The input image has line segments of length 40 pixels, under an orientation of 0.4 rad.*

"ringing". This can be explained by the sampling of the structuring element and the image: morphological filtering uses the maximum or minimum value in a neighborhood, and it depends on whether a sample exactly hits such a maximum or minimum that it can be found or not. By modifying slightly the orientation of the line, a different set of samples will sit close to maxima or minima (i.e. the ridge of the line). Interpolation (as in Section 3.4) should diminish this ringing effect.

Taking these observations into account, it can be said that the interpolated methods and the gray-value method produce results more consistent with the expectations than the discrete methods. Also, for this test image, it does not appear to be necessary to use method 5, since it produces a result very similar to method 4. Method 4 is, of course,

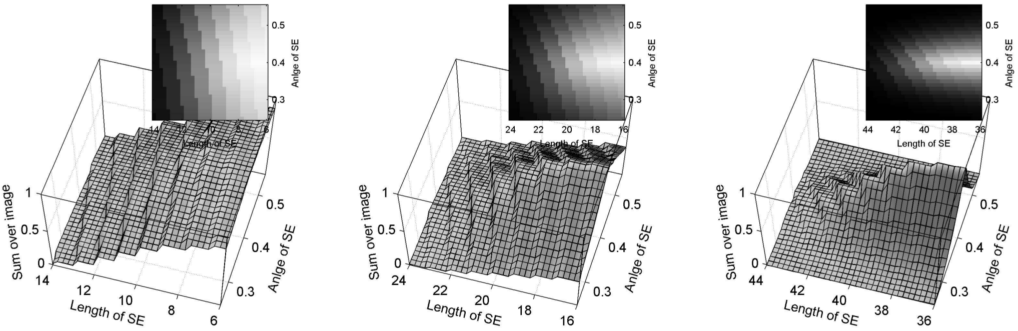


Figure 5.11: *Evaluation of method 4 (opening along an interpolated line). These graphs were obtained by changing the length of the line segments in the input image.* From left to right: *10, 20 and 40 pixels long.*

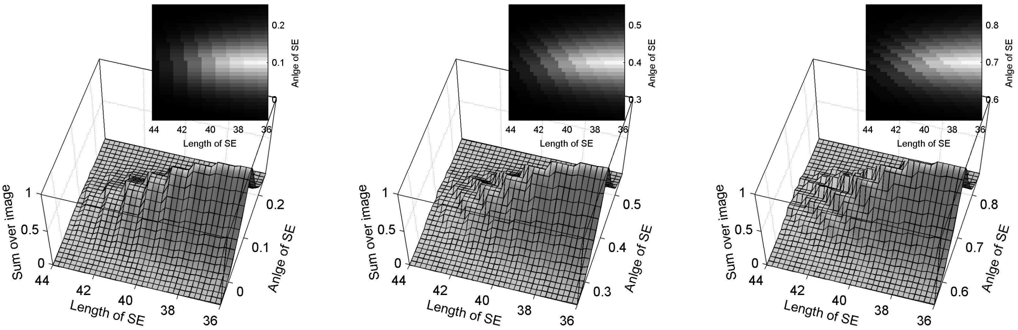


Figure 5.12: *Evaluation of method 4 (opening along an interpolated line). These graphs were obtained by changing the orientation of the line segments in the input image.* From left to right: *0.1, 0.4 and 0.7 rad.*

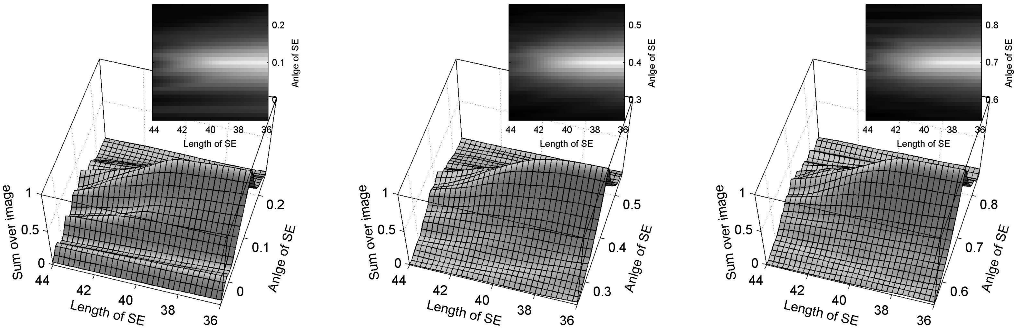


Figure 5.13: *Evaluation of method 6 (opening using a gray-value line segment). These graphs were obtained by changing the orientation of the line segments in the input image.* From left to right: *0.1, 0.4 and 0.7 rad.*

much simpler and computationally cheaper.

To further examine the interpolated method (method 4), the experiment was repeated changing the length of the line segments in the image. The results are shown in Figure 5.11. When decreasing the length, the angular selectivity decreases as well. This can be seen by the wider response in the angular direction. Also, because of the discrete segment lengths, it is impossible to distinguish lengths that differ by less than one pixel after skewing. That is, the minimal length difference that can be detected depends on the orientation. This can be seen in the graphs obtained by changing the orientation of the line segments in the image (Figure 5.12). The length of the steps in these graphs change with the selected orientations.

This angular dependency of the length does not occur with the gray-value morphology (see Figure 5.13). The only thing that changes in these graphs is the strength of the ringing effect. The smaller the angle, the larger this effect, because there will be larger sections of the ridge far away from any sample.

5.7 Angular Selectivity

In Chapter 2 we introduced an operation (RIA morphology) that applies a morphological filter with a line structuring element under all orientations, and then selects, for each output pixel, the orientation that produces the largest response. To implement this operation, we need to select a finite number of orientations at which to apply the filter. In this section we determine suitable values for this number.

Response along the Orientation Axis

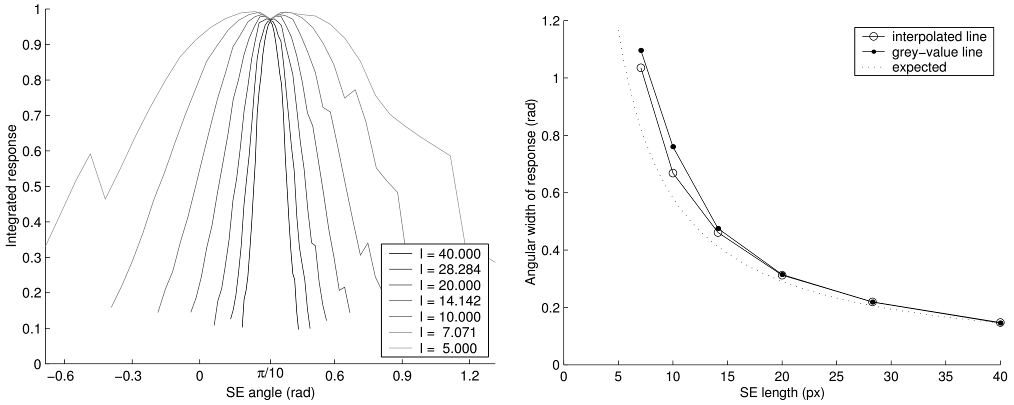
Section 2.3 describes the orientation-space that is generated by e.g. the RIA closing. Recalling Equation (2.37),

$$F(x,\varphi) = [\phi_{L_\varphi}(f)](x) \quad , \tag{5.2}$$

and Equation (2.38),

$$[\phi_L^{\triangleleft}(f)](x) = \bigwedge_{\varphi} F(x,\varphi) \quad , \tag{5.3}$$

we see that the RIA closing is formed by the minimum projection along the orientation axis of this orientation-space. If it were possible to interpolate along this axis, it would also be possible to define a maximum number of orientations that need to be sampled (in a similar fashion to the spatial sampling of band-limited signals according to the



a: *Angular response* b: *Angular width*

Figure 5.14: *The response along the orientation-axis of the morphological orientation-space.* a: *Integrated result of the closing with various line lengths ℓ, as a function of the orientation of the line.* b: FWHM *of the response in* a, *together with a line* FWHM$\propto \ell^{-1}$.

Nyquist criterion [88, 108]). Unfortunately, due to the non-linear nature of the closing, this is not possible. The shape of the response along this axis depends on the input image in a non-linear way, and will, in general, not be band-limited. Figure 5.14a shows this response, integrated over the spatial dimensions, for the interpolated line segment (method 4) applied to the test image used earlier (Figure 5.9a). As found in Section 5.6, the longer the line segment is, the narrower the response will be along orientation-axis. One would expect this width to be inversely proportional to the length. The full width at half the maximum (FWHM) is plotted in Figure 5.14b. This width is a measure for the angular width of the response. As can be seen in this figure, it does depend on the inverse of the segment length.[3] It does not give us, however, the number of orientations that should be used in the RIA closing.

To obtain this quantity, we assume that the distance along the orientation-axis between samples is given by q/ℓ, ℓ being the segment length and q being some constant. This is equivalent to taking $\lfloor \pi\ell/q \rfloor$ samples between 0 and π radian. By varying the value of q and computing a granulometry we find a suitable value for q. The granulometry was applied to an image similar to that in Figure 5.9a, but with the lines at arbitrary orientations. We used RIA openings with interpolated line segments. The results are shown in

[3] The disagreement for shorter line segments is due to the band-limit of the objects in the image.

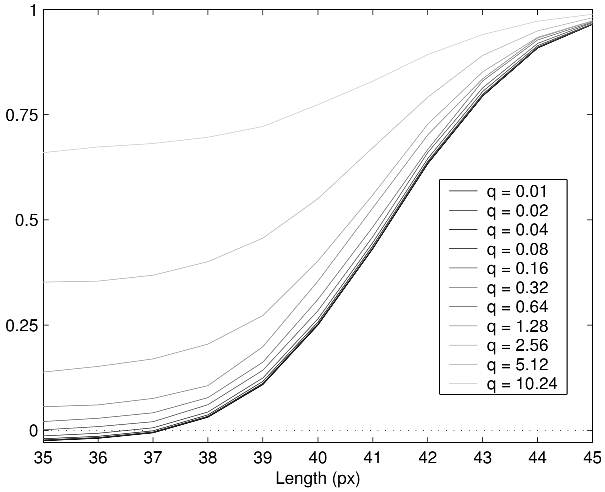
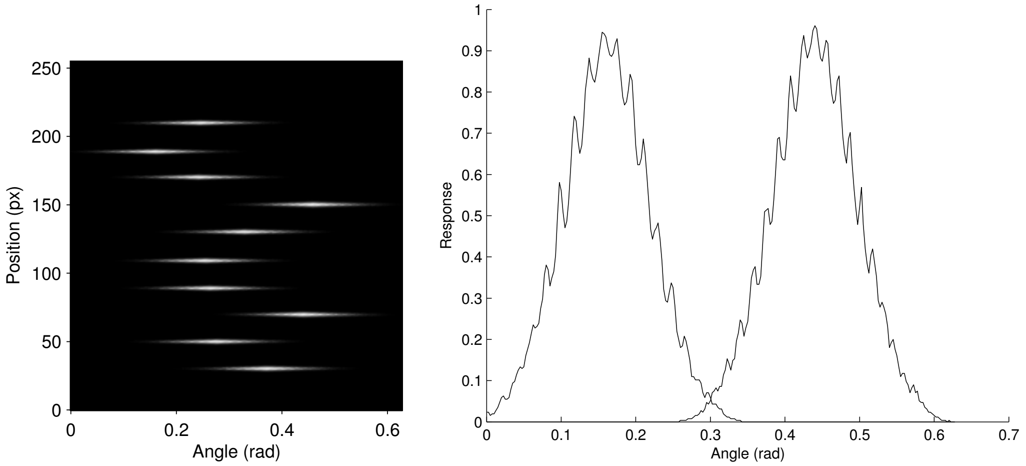


Figure 5.15: *Granulometries computed using* RIA *openings with different number of orientations sampled. The number of orientations is given by $\pi\ell/q$.*

Figure 5.15. Obviously, increasing q diminishes the accuracy of the granulometry. For values of $q = 0.16$ and lower, the graph is quite consistent. Therefore, setting q between 0.1 and 0.2 should produce good results. This implies 600 to 1200 orientations when computing a RIA closing with a line of 40 pixels in length. We have used $q = 1$ in most applications to decrease the computation time. This yields a small underestimation.

Figure 5.15 shows a granulometric line (for smaller q) that is negative at smaller ℓ. This is an error introduced by the interpolation, and does not occur with methods that do not interpolate, such as the gray-value structuring element.

Figure 5.14a shows a small dip at the maximum, that gets more pronounced for smaller lengths. The reason for this is related to the shape of the orientation-space $F(x,\varphi)$, and the integration over the spatial dimensions. As can be seen in Figure 5.16, the response along the φ-axis at a single pixel is far from smooth. This is because of the varying lengths of the line segment as a function of the orientation. By summing many such profiles, a smoother curve results. But apparently a dip at the maximum coincides for many of these profiles. Again, using gray-value structuring elements this behavior is not observed, due to the smoother orientation-space produced by them.



a: *x-φ plane of $F(x,\varphi)$*

b: *Two φ lines from $F(x,\varphi)$*

Figure 5.16: *The morphological orientation-space as used to compute the granulometries in Figure 5.15. As can be seen in* b*, the response along the orientation-axis is not smooth, even though the input image is.*

Opportunity is missed by most people because it is dressed in overalls and looks like work
(Thomas Edison)

It's true hard work never killed anybody, but I figure, why take the chance?
(Ronald Reagan)

Chapter 6

Assorted Topics Related to Granulometries

When using a granulometry in practice, many small difficulties arise that pose an obstacle to an all-purpose analysis package. Many of these difficulties can be overcome by adequate pre-processing, or by changing the closing (or opening) used in the granulometry. Section 6.1 is an overview of alternative granulometries, obtained by using the various algebraic closings available. Section 6.2 explores different pre-processing steps that we used in this thesis and other projects. These pre-processing operations are aimed at correcting for noise, uneven intensity across the objects, and objects that intersect the image border. All of these problems introduce additional errors in the estimated size distributions.

Section 6.3 introduces an algorithm that implements a flooding. It was developed as a pre-processing stage before a granulometry, but it can be used as a sieve in itself and as a segmentation algorithm.

6.1 Alternative Granulometries

Chapter 3 explores the sieve (and granulometry) with structural closings or openings. However, any one parametric family of algebraic closings or openings that satisfy the absorption property can be used to create a sieve. Section 1.2 enumerated different algebraic closings. We re-visit them here to explore how they would function in a granulometry. At the same time, each of them is compared to the isotropic closing to see what the advantages and disadvantages are.

Remember that, wherever we mention the closing, the opening can be substituted. Depending on whether the interesting features are dark or light, one should use the closing or the opening respectively.

Rank-Min Closings

The rank-min closing is similar to a structural closing, except that a certain number of pixels is ignored when computing the maximum over the structuring element. As said earlier, this is implemented by a rank filter (instead of a dilation) followed by an erosion. It is usually employed instead of a structural closing to reduce the effects of the noise in the input image.

The rank-min closing satisfies the absorption property only if the rank is fixed. However, to reduce the effect of noise, one would like to ignore a number of pixels that is relative to the size of the structuring element (i.e. a fixed percentage, not a fixed number). Such a closing does not satisfy the absorption property, and therefore cannot be used for a granulometry.

Instead of using rank-min closings, it is better to reduce noise in the input image by a pre-processing step (Section 6.2).

Rank-min closings as a granulometry might be useful only for structures with certain characteristics, e.g. where the interesting features are formed by agglomeration of smaller objects, which one does not need to see in the granulometry (as in Figure 6.1). A rank-min closing is able to adapt the structuring element to fit these agglomerated objects. Figure 6.2 shows an example of this. Note how, for increasing length of the structuring element, less of the object is considered “a straight line”, but also more of the background noise is removed. The increase of contrast obtained by using a rank-min closing instead of a structural closing is most noticeable with the 95th percentile. When using a lower percentile, the effective line length is also reduced, thereby increasing the amount of noise left in the image after the operation, but the contrast is not increased further.

Closings by Reconstruction

The difference between a structural closing and a closing by reconstruction is that, if the structuring element fits somewhere inside a feature, it is left intact by the reconstruction, whereas only the parts where the structuring element fits are kept by the structural closing. A sieve with such a closing has the advantage of correctly classifying all pixels belonging to an object, no matter how irregular its shape is. Where a structural closing would split an object into portions by size, the closing by reconstruction keeps all pixels together. Figures 6.3 and 6.4 illustrate this difference with an example.

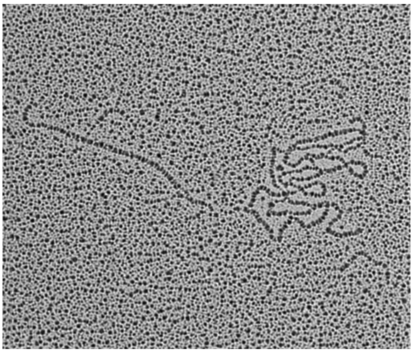


Figure 6.1: *Scanning electron microscope image of polysaccharide strands, imaged by spraying the strands onto mica, then evaporating Pt/C onto the mica (cooled to* $-185°C$*) at an angle of* $6°$ *[3]. The dark dots composing the image are caused by clusters of metal particles (this effect is called self-shadowing). The objects can be identified by a larger density of these clusters.*

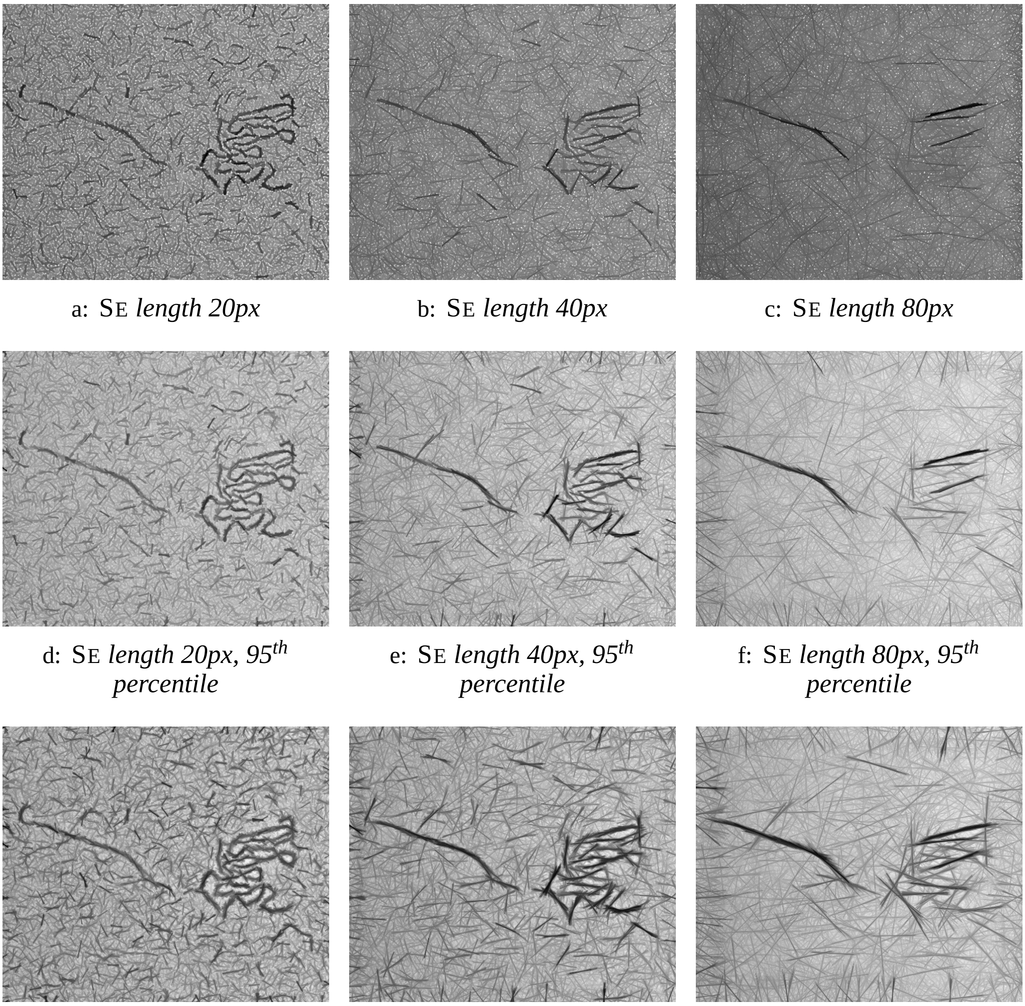
The drawback is that, when objects are touching, they are seen as a single object by the reconstruction, the smaller one being classified on the same scale as the larger one. Thus it depends on the application whether using a closing by reconstruction instead of a structural closing would be beneficial.

Attribute Closings

If a closing by reconstruction were beneficial, one might instead use an area closing. Area closings remove objects of a given area (number of pixels). This classifies objects on their area instead of on their smallest diameter. For irregular-shaped objects this might be a more interesting characteristic. Note that a closing by reconstruction is an attribute closing, and the disadvantage mentioned there is valid here as well: if objects are touching, it is no longer possible to separate them with this method.

Floodings

A flooding is a very different kind of closing. Since it is based on the watershed transform, it is constructed by a hierarchical segmentation of the image. Depending on the location within this hierarchy, more of fewer regions are closed. As this location within the hierarchy is not directly related to a size, it is not possible to construct a size distribution from floodings. But it is possible to obtain a granulometry that can be used as a



a: SE *length 20px*

b: SE *length 40px*

c: SE *length 80px*

d: SE *length 20px, 95th percentile*

e: SE *length 40px, 95th percentile*

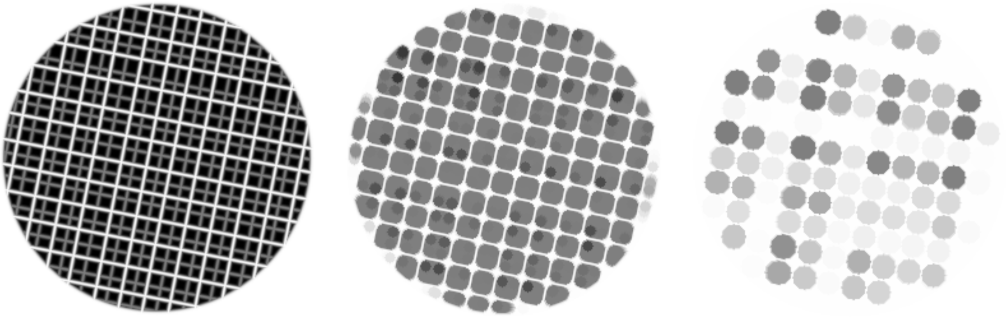
f: SE *length 80px, 95th percentile*

g: SE *length 20px, 80th percentile*

h: SE *length 40px, 80th percentile*

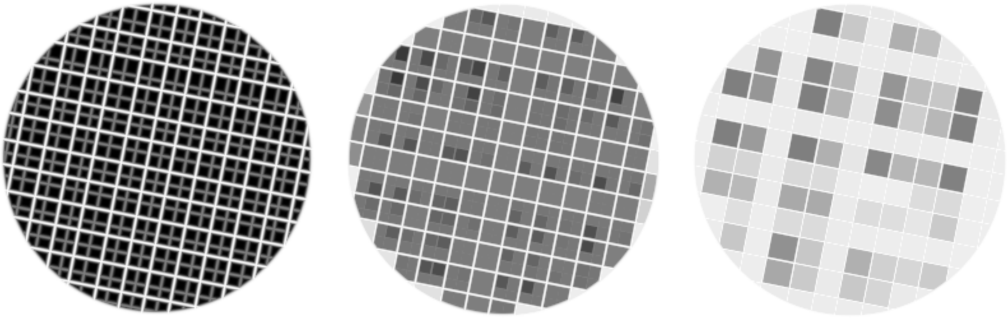
i: SE *length 80px, 80th percentile*

Figure 6.2: a-c: *Result of the* RIA *closing with line segments of 20, 40 and 80 pixels, applied to the image in Figure 6.1.* d-f: *Result of the* RIA *rank-min closings with line segments of 20, 40 and 80 pixels. The rank was chosen at 95% of the total structuring element (although for a granulometry the rank has to be fixed, not relative to the size of the structuring element).* g-i: *Idem, with the rank set to 80% of the total structuring element. Note how allowing a small part of the structuring element not fitting the shape can increase the contrast of the filtered image, although if too much slack is allowed, the structuring element can fit almost anywhere in the image.*



a: *Input* b: SE *diameter 8px* c: SE *diameter 18px*

Figure 6.3: *Isotropic structural closing applied to a test image.*



a: *Input* b: SE *diameter 8px* c: SE *diameter 18px*

Figure 6.4: *Isotropic closing by reconstruction applied to a test image. Compare to Figure 6.3*

characterization of the structure, without assigning any physical quantities or properties to it.

We implemented a flooding (described in Section 6.3) that, using a couple of parameters, produces an image from within the segmentation hierarchy. By increasing one of the parameters, a sieve is obtained, as can be seen in Figure 6.5. This type of granulometry describes the image in a rather different way compared to the size distributions obtained with other closings or openings, and might be useful in comparing similar structures.

6.2 Pre-processing

The imaging process itself may hamper the image in such a way that the results of a granulometry are useless. The three properties we found most destructive are noise, uneven gray-value across the objects (or pores, matrix, etc.) due to uneven illumination, texture or other object properties, and the image border due to a finite field of view. These are analyzed in the three subsections below, and adequate pre-processing is discussed. Obviously, if these aspects can be dealt with by changing the image acquisition process, one should do so.

Noise Reduction

As stated in the introduction, all images are noisy. If the signal-to-noise ratio (SNR) is good enough, the noise will just add a little to the noise generated by the sampling problems of the structuring elements, the quantization of gray-values, the scale sampling, etc. If, however, the SNR is too low, it will affect the granulometry in other ways as well. There are three major ways in which noise in the image affects the (closing) granulometry (a similar analysis is valid for the opening granulometry):

- Pixels where noise lowers the actual value add volume to the very small scales. This is just an offset for the curve.
- Pixel values that are raised above the level of the matrix (the inter-pore space) add volume to the very large scales. That is, the value of $\int F(x,\infty)dx$ in Equation (3.2) increases.
- Pixels inside a pore whose value is increased will divide this pore, adding volume to the smaller scales and removing volume from the larger scales. This effect of the noise is the most destructive.

Thus, noise adds to the smallest and largest measured scales, thereby stretching the granulometric curve to make it less steep, and also shifts volume from larger scales to smaller scales. The two thick, gray lines in Figure 6.6a show this.

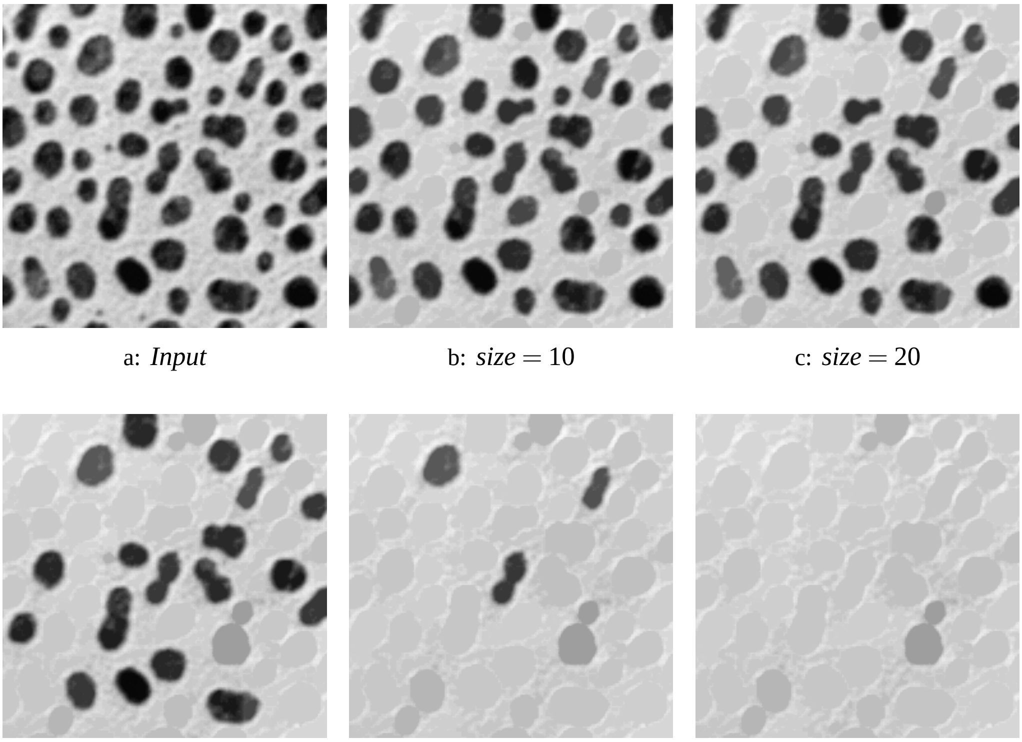
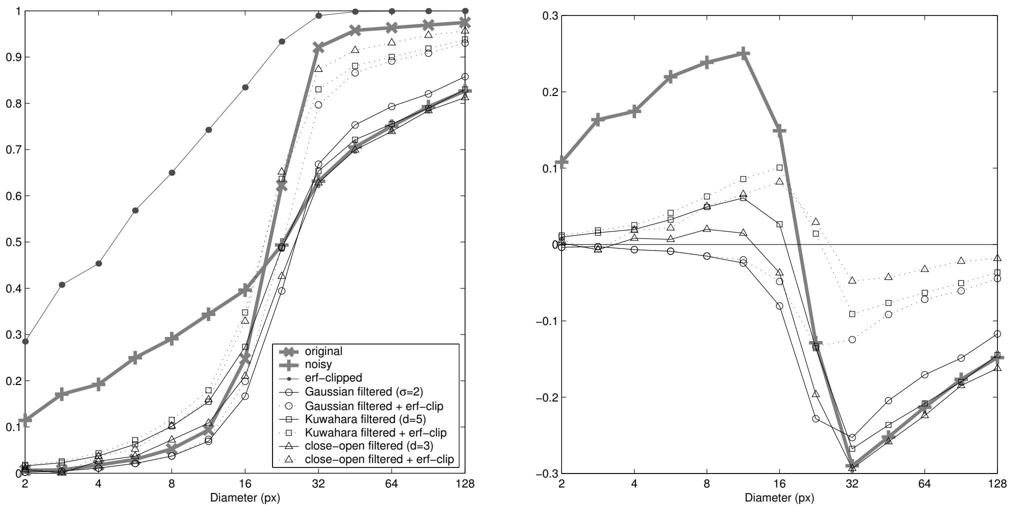


Figure 6.5: *Demonstration of the flooding as a sieve.* b-f: *results of the UpperEnvelope flooding with increasing merging parameter 'size'. This parameter can be interpreted in the following way: the flooding iteratively increases the gray-value of the darker regions until two of these regions touch. At this point, the size of both is compared to the 'size' parameter of the algorithm; if one is smaller than this value, the two regions are merged and the flooding of this new, larger region can continue. If not, the flooding of both regions is stopped. Thus, the larger the parameter is chosen, the more regions are merged and the larger the flooded regions are. See Section 6.3 for more information on this algorithm.*



a: *Size distributions*

b: *Differences with noise-free distribution*

Figure 6.6: *Size distributions obtained from a test image, with and without noise, and size distributions obtained after various noise-reduction filters were applied. The small-scale changes induced by noise can be corrected for easily with any noise-reduction filter. The large-scale changes, however, require a clipping. In this case we used the soft clipping as in Equation (6.1). The result of the median filter, not shown here, is very similar to that of the Kuwahara filter.*

Reducing noise is traditionally accomplished by filtering the image. Some very well known filters include the uniform and Gaussian filters, which are linear, and the median filter (see e.g. [49, 53, 127]) and the Kuwahara filter [64, 65, 85], which are non-linear. Both the median and Kuwahara are capable of increasing the SNR, without affecting the edges and other features too much. An excellent alternative are the open-close or close-open filters [104, 121]. These are obtained by combining the closing and the opening, and reduce the noise by removing small features that are either darker or lighter than their surroundings. Obviously the order in which they are applied is significant, but, assuming a band-limited image with normally-distributed, independent noise, and using a small structuring element, both filters will produce very similar results. A strong property for noise-reduction filters is *self-duality* (invariance to negation) [105]. Linear filters, as well as the median and Kuwahara filters mentioned above, are self-dual. The open-close and close-open, however, are not (they are each other's dual). By averaging the result of these two operators, self-duality is obtained [134].

All of these noise reduction filters smooth the flat regions of the image. The linear filters mentioned above will blur the edges, whereas the non-linear ones might not affect edges much, or even enhance them. But all of these filters will remove small objects. This is intentional, since noise forms small features. The consequence is that these filters effectively flatten (or null) the measured granulometric curve for very small scales. This is not important because possible interesting objects present at these scales are swamped by the noise anyway. To measure them it is necessary to increase the sampling density or reduce the noise level before sampling.

Figure 6.6 shows the effect of these noise-reduction filters. All are very effective at removing small objects (correcting for the volume added to the small scales), but the large-scale effects of noise (e.g. large dark objects created by small regions of increased matrix values at locations far apart) could only be removed by clipping the gray-values. This is discussed in the next subsection.

Normalizing the Gray-Value of Objects in an Image

If an object has an uneven gray-value across its extent, or if the various objects in an image have different intensities, the volume-weighing of the granulometry produces a large error in the estimated size distribution. Remember that the granulometry weighs the objects by summing the gray-values. Thus, the more even these gray-values are, the more accurate the granulometry is.

Also, when measuring pores, the intensity of the matrix in between these pores is of great importance. If the matrix in between two pores has a lower gray-value than the rest of the matrix (for example because it is very thin, and the point-spread function of the imaging device has blurred it), the volume of each of the two pores will be split over two levels: that of the pore itself, and that of the union of the two pores. That is, there is a shift of weight from smaller to larger scales. Another effect of non-uniform matrix gray-values, mentioned in the previous subsection, is that small regions of increased gray-values create very large, though shallow, pores. These add potentially large quantities of volume to the larger scales of the granulometry.

To overcome these problems, different stretching techniques are available. If the goal is to flatten both the background and the foreground, without changing the bandwidth of the image much (so that it is still correctly sampled and the localization of the edges is not impaired), the *soft clipping* (or *error-function clipping*) is an excellent technique [139]. Much like *hard clipping* (in which gray-values larger than the upper threshold t_{high} or smaller than the lower threshold t_{low} are set to the respective threshold values, and none of the other gray-values is changed), soft clipping removes gray-values

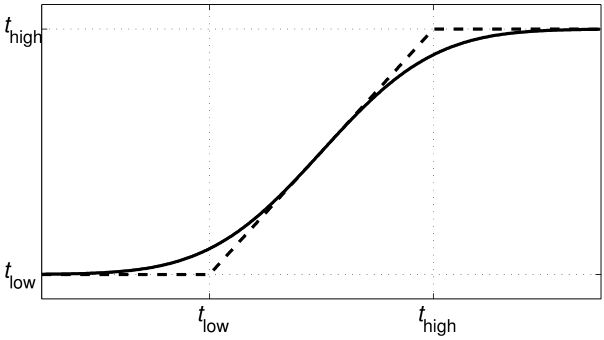


Figure 6.7: *The error function as a transfer function. The discontinuous line is the transfer function of the hard clip.*

that are outside the target range. However, this is accomplished by stretching all pixel values according to the error function (Figure 6.7). With adequate normalization this yields,

$$f_c(x) = \text{erf}\left(\frac{\sqrt{\pi}}{2}\,\frac{f(x)-t_{\text{med}}}{t_{\text{high}}-t_{\text{med}}}\right)(t_{\text{high}}-t_{\text{med}})+t_{\text{med}} \quad , \tag{6.1}$$

with $t_{\text{med}} = \frac{1}{2}(t_{\text{high}}+t_{\text{low}})$. The factor $\frac{\sqrt{\pi}}{2}$ is to make the slope of this transfer function 1 near t_{med}. The error function keeps the transitions from one phase to the other smooth, but removes the transitions within a single phase. See Figure 6.8 for a demonstration.

Another technique is illustrated in Section 6.3 and Figure 6.13.

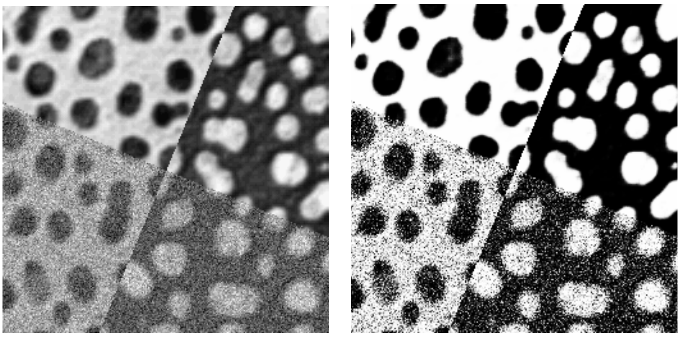
Correcting for Image Border Effects

Because an image is not infinite in extent (which coincidentally would make imaging quite difficult since the structures that are imaged are not infinite in extent either), it has borders. The borders of the image cause problems in two different ways:

– the result of a filter near the edge is undefined because the neighborhood used in the operation falls partially outside the image, and

– some objects in the image are incomplete.

To overcome the first difficulty, the image is usually extended in some predefined way (e.g. by making it periodic or mirroring the image data at the border). For morphological operations, which are selection operations (the result at a given point is the value of a neighboring point), we usually restrict the neighborhood to the image domain. That is, the result of the operation is always the value of one of the pixels inside the image. For a



a: *Input* b: *Soft clipping*

Figure 6.8: *Demonstration of soft clipping on a test image. Note how the edges of the objects remain smooth while both the objects and the background are flattened.*

closing this means that the possible shifts of the structuring element are restricted; pores intersected by the image border are regarded smaller than they really are.

This leads to the second problem. Cropped objects should not be regarded when computing a size distribution. However, if all cropped objects are discarded, the resulting size distribution will be biased because larger objects have a larger probability of intersecting the boundary (and therefore a larger probability of not being considered when estimating the size distribution). One of the findings in stereology (see also page 26) is an unbiased counting procedure [84]. We consider the image as a region in a larger plane. If this plane is divided into adjacent rectangular regions (of which the image is one), a tessellation is obtained. One should be able to assign each object in the scene to one of the regions. If this is done correctly, the number of objects assigned to each region is an unbiased estimator of the total number of objects (or the density). An exclusion line (which goes along the boundary between regions, separates the plane in two, and has a shape as in Figure 6.9) should be defined to assign objects to a particular region. All objects that fall (partially) inside the region should be counted, except the ones that touch the exclusion line. Note that this procedure is independent of the shapes and sizes of the objects.

6.3 The UpperEnvelope Algorithm

As mentioned in Section 1.2, a flooding is a form of algebraic closing. This section introduces an algorithm that implements such a flooding. Although it was devised as a

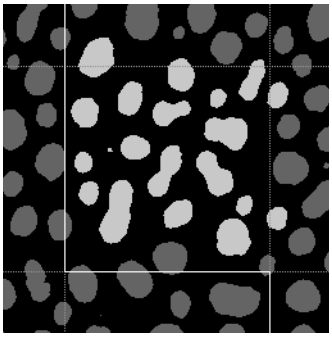


Figure 6.9: *For an unbiased counting of objects, a selection procedure like this one is required. Here, all objects that fall (partially) within the given cell but do not cross the white line (exclusion line) are selected. Note that if the same procedure is used on adjacent cells, none of the objects is counted twice or skipped.*

pre-processing step for the granulometry, it can also be used as a segmentation tool or as the closing operation in the granulometry itself. This section describes the algorithm. For applications see Sections 6.1 and 6.2.

This algorithm is based on the watershed transform. We therefore first explain a priority-queue algorithm that implements a watershed, as in [136, 138]. This type of algorithm sorts the pixels in the image according to their gray-value, and then processes each pixel only once. The special case where a group of pixels have the same gray-value (a plateau) should be treated carefully. These pixels need to be considered in the correct order, so that the watershed line lies half-way on the plateau. To accomplish this, we assume that the sorting algorithm will sort all pixels with the same gray-value according to their distance to pixels with lower gray-value. Because this is irrelevant to the performance of our UpperEnvelope algorithm, we do not discuss how to implement this. For a review of correct watershed algorithms see [99].

We have integrated a merging step into the algorithm. This is useful to reduce the number of local minima due to noise.

The Watershed

The watershed algorithm can be summarized by the following actions:

1. Each local minimum is used as a seed, and is assigned a label.
2. Labels are grown according to the image gray-values.

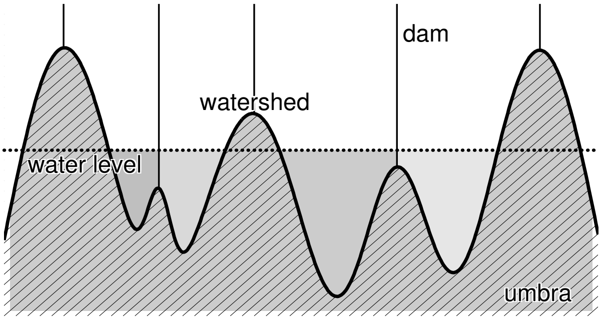


Figure 6.10: *As the water level rises, watersheds are built to keep the catchment basins separate.*

3. When two labels grow together, a decision is made whether to merge them or not.
4. If they are not merged, the line along which these labels meet is marked as a watershed.
5. Labels continue to grow in all directions without crossing any watersheds.
6. The watersheds define the segmentation.

This algorithm is comparable to flooding the 'landscape' formed by the gray-values in the image (see Figure 6.10). As the water level raises in each of the pools (local minima), they are kept separate by watersheds (dams). The merging decisions can be made using information such as the region size and depth. We allow a region that is both small and shallow to be merged to another region.

Pseudo-code that implements this algorithm can be found in Figure 6.11. As can be seen there, pixels are sorted according to their gray-value. This allows us to process the pixels from lower to higher gray-value with a single pass through the image, instead of one pass for every possible gray-level [136, 138]. As noted above, this version does not handle correctly the plateaus in the image. For these features some extra processing is required. Also, since each pixel is addressed only once, it is actually not necessary to mark the watershed pixels: all pixels in the label image that are still 0 at the end of the algorithm form the watersheds.

The UpperEnvelope

The UpperEnvelope algorithm proposed here is a slight modification on the watershed algorithm described above. The difference is that, once two labels meet, both are marked

```
Input: input image, optionally connectivity and parameters on merging

Create label image (initialize to zero)
Create table indices with indices into image, sorted according to value in input image;
    for plateaus, sort indices according to distance to lower gray-value pixels
Prepare table label_state containing information on each label for merging strategy:
    size, min (minimum value)

Assign label 1 to pixel indices[1]
Update label_state[1]
lastlabel ← 1

for each i in indices starting at $2^{nd}$ element:
    labels ← labels of processed neighbors of i
    case no neighbors:
        Increase lastlabel
        Assign label lastlabel to pixel i
        Initialize label_state[lastlabel]
    case 1 neighbor:
        Assign labels[1] to pixel i
        Update label_state[labels[1]]
    case more neighbors:
        if all labels can be merged:
            Rename each label in labels into labels[1]
            Assign labels[1] to pixel i
            Update label_state[labels[1]]
        else:
            (Pixel i is a watershed pixel)
        endif
endfor
```

Figure 6.11: *A watershed algorithm; see text for details.*

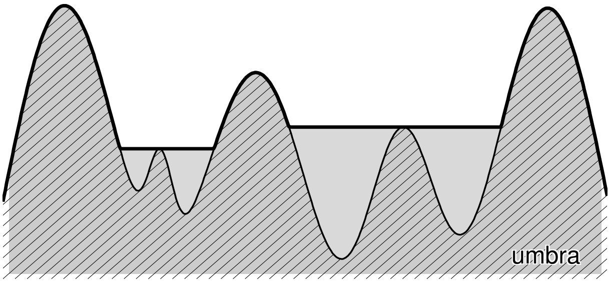


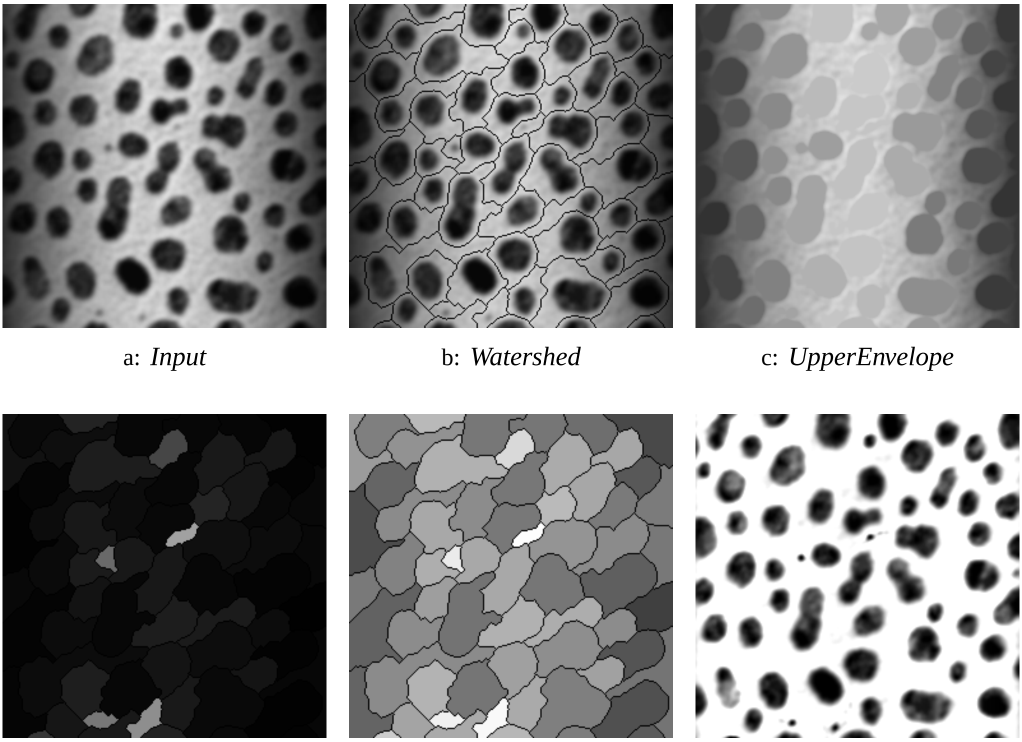
Figure 6.12: *The upperenvelope is an increasing transformation of the image based on the watershed transform.*

‘stopped’, and do not grow further. The flooding is obtained when these stopped regions are filled with their maximal gray-value. The space in between them is not modified (the input gray-value is retained). See Figure 6.12. To keep the watershed properties, however, it is necessary to keep the growth process going. The simplest solution is to record the gray-value of the pixel that caused a region to be ‘stopped’. At the end of the watershed algorithm, all pixels within the catchment basin with a gray-value lower or equal to this recorded value compose the ’stopped’ region.

As discussed on page 37, the UpperEnvelope algorithm implements a flooding. That is, g, the result of applying the algorithm to f, satisfies $g = f \vee \varepsilon(g)$ [83].[1] However, the algorithm itself is not a closing because it is not idempotent. It does satisfy the increasingness criterion.

We have applied the algorithms described in this section to a test image, and show the results in Figure 6.13. As can be seen there, it is possible to generate a ‘bottom’ image using the watershed results. This image can then be used to stretch the resulting regions so that each one has an equal depth. This way, each of the regions contributes equally to the granulometry. Figure 6.14 contains the pseudo-code for the UpperEnvelope algorithm.

[1] Although not obvious in the referred article, the erosion needs to be restricted for this to be a sensible definition; ideally, one should use an infinitesimal erosion.



a: *Input*

b: *Watershed*

c: *UpperEnvelope*

d: *‘Bottom’ image*

e: *‘Bottom’ image (log)*

f: *Normalized blobs*

Figure 6.13: *Demonstration of the algorithm described in Section 6.3. Note how the shading has little effect on the result of the watershed. Subfigure* e *is the same as* d*, but using a logarithmic mapping of the gray-values.* f: *Difference between UpperEnvelope and the input image, stretched using the ‘bottom’ image. Again, the shading has little effect on the result.*

```
Input: input image, optionally connectivity and parameters on merging

Create label image (initialize to zero)
Create table indices with indices into image, sorted according to value in input image
Prepare table label_state containing information on each label for merging strategy:
  size, max (maximum value), min (minimum value), stopped)

Assign label 1 to pixel indices[1]
Update label_state[1]
lastlabel ← 1
for each i in indices starting at 2nd element
  labels ← labels of processed neighbors of i
  case no neighbor:
    Increase lastlabel
    Assign label lastlabel to pixel i
    Initialize label_state[lastlabel]
  case 1 neighbor:
    Assign labels[1] to pixel i
    Update label_state[labels[1]]
      (max only updated if labels[1] is not stopped)
  case more neighbors:
    if all labels can be merged:
        (meaning labels[2..end] fit the requirement for merging)
      Rename each label in labels into labels[1]
      Assign labels[1] to pixel i
      Update label_state[labels[1]]
        (max only updated if labels[1] is not stopped)
    else:
      for each lab in labels that is not stopped:
        Mark lab as stopped
        Update label_state[lab]
      endfor
      (Pixel i is a watershed pixel)
    endif
endfor

Create output image as copy of input
Paint each output pixel with the max value for label if it would increase its value

The bottom image can be generated by painting each pixel with the min value for label
```

Figure 6.14: *The UpperEnvelope algorithm; see text for details.*

If the only tool you have is a hammer, you tend to see every problem as a nail
(Abraham H. Maslow)

Chapter 7
Applications

This chapter presents a few of the applications we worked on during the course of this project. All of these applications came from Unilever Research and Development Vlaardingen, and are reported here with their permission.

Section 7.1 presents the results obtained on a milk protein gel. The goal here is to find differences between untreated milk protein gel and milk gel treated with enzymes or enzymes and a substrate. When the treatments are performed with very small quantities, the differences are minute. Using isotropic granulometries with the improvements proposed in this thesis it is possible to detect these differences. This project was performed in collaboration with E.C.M. Bouwens, and submitted for publication in [74].

Section 7.2 examines a similar structure, also of a dairy-based product, but this time the evolution in time of the characteristic length is the interesting property. Mixtures with varying concentration of amylopectin are recorded in time while they are allowed to sediment. The increase in characteristic length, which depends on the concentration of amylopectin, is estimated using a granulometry at each time step, and the results are compared to those obtained with a classical method, Fourier analysis. This project was performed in collaboration with P.W. de Bont, and submitted for publication in [18].

Finally, Section 7.3 attempts to automate the determination of the quality of rice. The number of broken kernels determines the price of rice, but counting these is a very subjective process. In this project we aim at counting the fraction of broken kernels by scattering rice randomly on a flat-bed scanner. Classical image analysis methods (those that rely on segmentation of the image) would require the rice kernels to be carefully separated by hand, a tedious and time-consuming process. Using a granulometry with RIA openings to measure the length of the kernels, it does not matter if the kernels are touching one another because no segmentation needs to be done. This project was performed in collaboration with G. van Dalen, and published in [71].

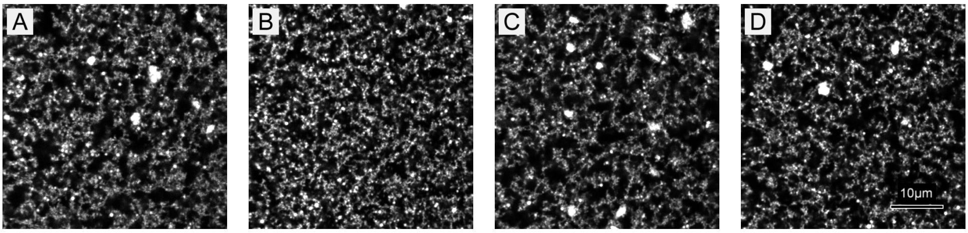


Figure 7.1: CLSM *slices of a milk protein gel after different treatments. The difference of the second sample (B) with respect to the other ones is obvious.*

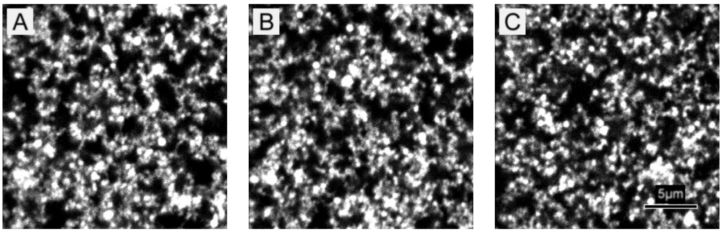


Figure 7.2: CLSM *slices of a milk protein gel after different treatments, as in Figure 7.1. In this case, the differences are not visible.*

7.1 Detecting Minute Differences in Structure

To study the effect of the addition of an enzyme and its substrates on the protein network of a milk gel, we added the substrate with and without the enzyme to the milk protein gel. The substrate alone could induce some side effects, which were expected to change the microstructure of the protein network (visible in the pore sizes of this network). The enzyme, however, is able to convert the substrate in such a way as to minimize these side effects. The addition of enzymes and substrates to milk protein would lead to enzymatic oxidation of this protein, possibly yielding a network with larger pore sizes. This is the effect of chemical oxidation versus enzymatic oxidation, and these experiments are part of a study to confirm these processes.

Because the effect of these additives to the milk protein gel should be visible in the pores of the protein network, we applied the granulometry as described in Chapter 3 to CLSM (Confocal Laser Scanning Microscope) fluorescence images of these products. Rhodamine was used to stain the proteins. Therefore, protein aggregates show up as regions with high pixel intensity. The dark regions contain water, dissolved protein and other components that do not contribute to the protein network. These we call the pores.

The first experiment is carried out on milk protein gel (UHT treated, low fat milk with 2% whey protein added) after four different treatments (see Figure 7.1):

- Class A is the original product.
- Class B is the same product with the substrate added.
- Classes C and D are treated with different amounts of the substrate and enzyme mixture.

All four products have similar, but not identical, properties. Only class B is visually distinguishable from the others. The images are 512 by 512 pixels.

The second experiment is carried out on similar products (made using skimmed milk powder to reduce the amount of whey protein). We have 32 uncorrelated images of each of the classes A, B and C, labelled in the same way as above; one of each is shown in Figure 7.2. They are 256 by 256 pixels. The differences are not discernible by eye.

The main difference between the products of the two experiments is the amounts of whey protein. These images confirm that the substrate heavily affects these proteins.

We need to note here that the obtained images are two-dimensional sections out of a three-dimensional structure. This implies that a size distribution obtained from such an image does not represent a size distribution of the three-dimensional pores, but rather that of random intersections of those pores with a plane (pore profiles). It could be possible to obtain a distribution for the three-dimensional pores from it using techniques from stereology (see [84, 124, 143] for an overview, as well as the text on page 26). However, as we are only interested in comparing the three products with one another, this is not required. If the pore size distributions of two three-dimensional structures are the same, then the distributions obtained from an intersection of these pores with a plane are the same as well. Therefore we are able to directly compare the distributions obtained from these two-dimensional images.

Although a three-dimensional analysis might provide better discriminating power, we were unable to do so because the milk protein network was too opaque to obtain a sufficiently representative three-dimensional image using CLSM; besides, the resolution along the optical axis is too low to image the pertinent structure.

Applying the Granulometry

All images were contrast-stretched, making 5% of the pixels black, and 5% white, using erf-clipping. This counteracts the effect of the clipping that occurred during data acquisition due to a too high dynamic range (see Section 6.2). A cumulative size distribution

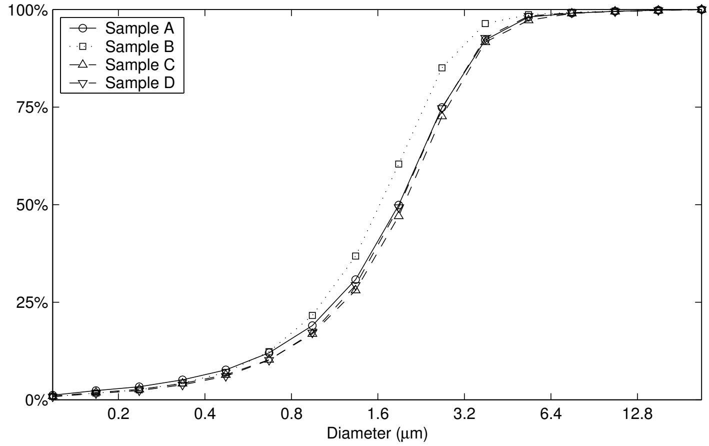


Figure 7.3: *Cumulative size distributions for the pore profiles of the samples in Figure 7.1. Note how samples C and D are more similar to A than to B.*

of the pore profiles, as presented in Chapter 3, with two samples per octave and binary disks as structuring elements, is then computed from each image.

The cumulative size distributions obtained in the first experiment are depicted in Figure 7.3. It shows that sample B is indeed distinguishable from A, C and D, which are very much alike. Sample B has smaller pore profiles, which can be seen by its cumulative distribution being above the others.

The second experiment produces the cumulative distributions in Figure 7.4, which shows the averages over each of the 32 distributions acquired for each class. The error bars show the 95% confidence interval for the mean of each point. This graph shows that class C has a distribution that is very similar to that of A; the pore profiles in class B are somewhat larger. The mean values at some scales are statistically different for class B compared to both A and C. Nothing can be said about differences between classes A and C.

7.2 Time Evolution of Characteristic Length

Polysaccharides are often used as thickening agents or stabilizers in dairy-based products. They improve the product texture as well as the physical stability. However, protein and polysaccharide mixtures are usually unstable; they tend to demix in a process

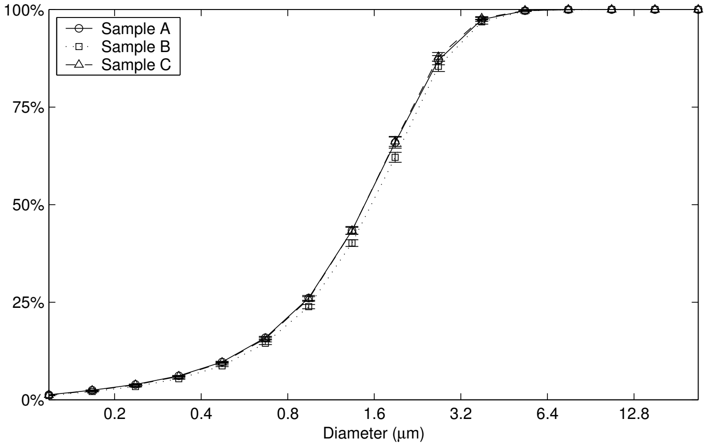


Figure 7.4: *Averaged size distributions for the pore profiles of the three samples in Figure 7.2, with the 95% confidence intervals for these means.*

referred to as phase separation. To study phase separation of casein colloids (the protein) and amylopectin (the polysaccharide), four samples were prepared, each containing 2.7% (weight) casein, and varying amylopectin quantities (1.0, 1.6, 1.9 and 4.8% weight). A series of images for each of these samples was recorded using CLSM. The casein was labelled with Rhodamine B (which did not influence the macroscopical phase separation), meaning that the bright parts in the images are the protein-rich phase and the dark parts compose the protein-poor (and hence polysaccharide-rich) phase. These four samples were placed in a glass container in which they could be stirred until the image recording began. The first image in each series was taken immediately after the stirring was stopped, after which one image was taken at 5 second intervals (except for the 4.8% mixture, for which the interval was 60s). Measurements were performed at room temperature at a depth of about 5 to 10μm from the bottom of the container.

Figure 7.5 shows the evolution of the four samples during the first 30 minutes. In this figure the phase-separation of the 1.6% and the 1.9% mixtures is evident. The 1.0% solution is stable, so it does not demix. The 4.8% sample, however, is not stable but does not show signs of phase separation in the first 30 minutes. In fact, it takes hours for the demixing to be clearly visible. This is attributed to the formation of a protein aggregate network [17], and is also predicted by coarsening theory for mixtures of gelling biopolymers [89].

The phase separation in this type of samples is typically measured by estimating the characteristic length of the structure. The characteristic length Λ is defined as the aver-

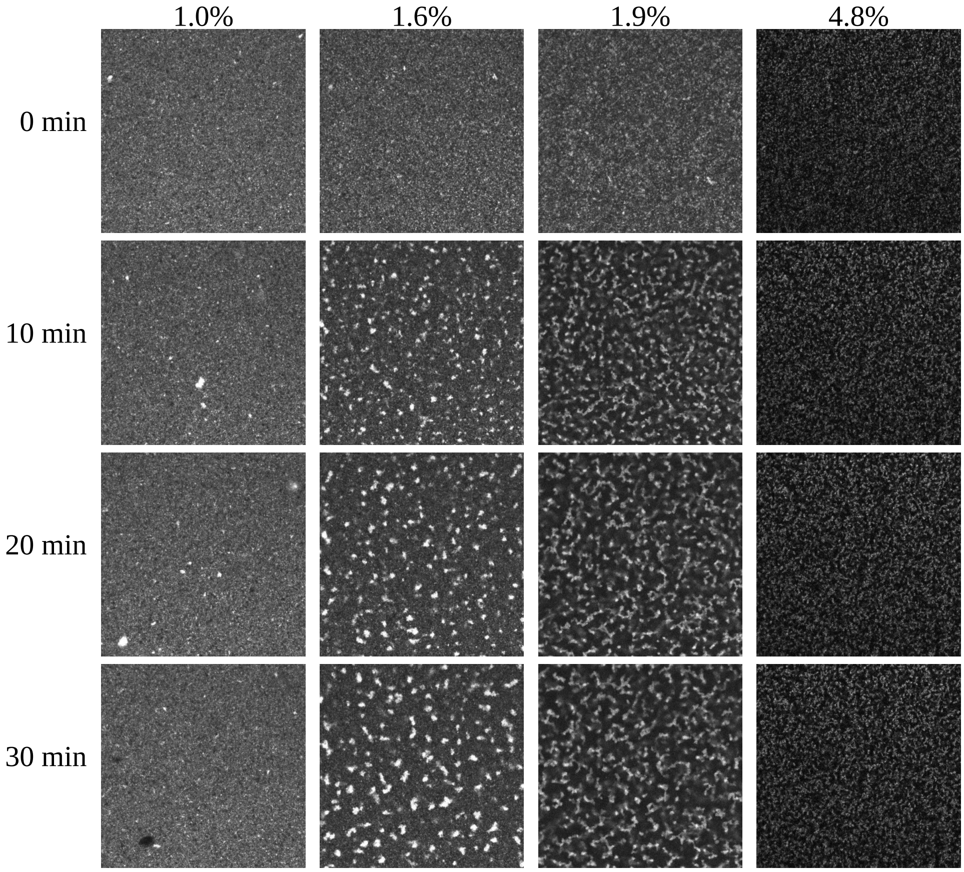


Figure 7.5: *Evolution of the four samples during the first 30 minutes after the stirring stopped. The 1.0% sample is stable and does not phase-separate; the 4.8% sample is not stable but very viscose, so the phase separation is not visible in the first half-hour. The other two samples demix rather quickly.*

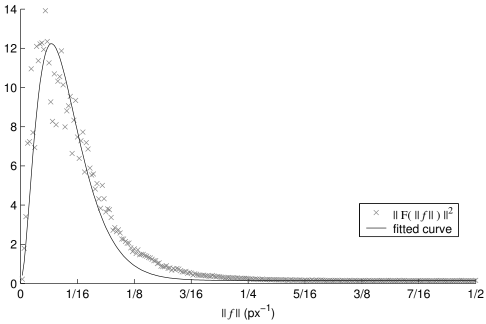


Figure 7.6: *Radially averaged frequency spectrum of one of the images in Figure 7.5, together with a fitted function. The fit is not perfect in this case, but the position of the peak matches the data relatively well. This position is related to the characteristic length.*

age distance between repeating patterns, and can be approximated by adding the average pore size and the average particle size. This length is expected to satisfy power-law behavior in time: $\Lambda(t) \propto t^{\alpha}$ [17, 18].

Like the previous application, we are using two-dimensional cross-sections of a three-dimensional structure. Unlike the previous application, here we are trying to derive measurements that relate to this three-dimensional structure. To do so we need to assume the mixture is isotropic, and realize that the results might be biased.

Results

To measure the characteristic length we applied the granulometry to both the dark and light phases, and used the median as a good estimate of the average size. The sum of the two medians gives the estimate of the characteristic length. We use the median instead of the average because it can be computed more easily from the measured size distribution. We used an error-function clip (see Section 6.2) tuned to the 99$^{\text{th}}$ percentile, to avoid a bias in the size distribution due to noise.

The size distribution is expected to be biased after all, because the sieve measures the minimal distance between structures, not the average distance. Because of the isotropy, this bias should be small.

We also employed a method often used when analyzing this type of structure: Fourier analysis [125, 135]. The power-spectrum of one of the images in Figure 7.5 yields a ring (after suitable high-pass filtering to remove shading and the peak at $f = 0$). Radial averaging (taking the average value as a function of $\|f\|$) results in data as in Figure 7.6. We fitted the function

$$p(\|f\|) = \left(b \, \frac{\|f\|}{a} \right)^2 \exp\left(-2 \frac{\|f\|}{a} \right) \tag{7.1}$$

to this data, yielding a peak location a related to the characteristic length by $\Lambda = \Delta / a$, Δ being the pixel pitch. The function $p(\|f\|)$ does not fit all data equally well, as can be seen in Figure 7.6. But the peak location seems to match pretty well for most images. The error-function clip required for the granulometry did not influence the results of this method.

Figures 7.7 and 7.8 contain the results for the granulometry and the Fourier method, respectively, and show a good agreement between the two. The power-law constants, although not identical, are quite similar. Even though both methods have their inaccuracies and biases, these appear not to be very large in this case.

In the 1.0% sample some very large, bright blobs appear at two time intervals. These are probably floating to the bottom of the vial, intersecting the imaging plane at some point in time. As can be seen in the graphic display of the results, the Fourier method is heavily influenced by these blobs, whereas the granulometry is not.

Finally, the 4.8% sample may have been imaged with a different gain setting of the CLSM, or maybe the larger percentage of amylopectin causes the dye to be less effective, but in Figure 7.5 we can see that for $t = 0$ the image of this sample is darker than for the other samples. This is the reason that, for the granulometry, the initial average object size is smaller in this sample than in the others. Because we do not expect a difference at this point, and the Fourier method did not detect such a difference, we conclude that this is a problem of the granulometry.

7.3 Counting Broken Rice Kernels

The fraction of broken rice kernels in a batch is used to determine its quality. The milling process employed to extract the kernels from their husk breaks a certain amount of them. Broken rice causes the consumer's perception of quality to decrease, along with the price. This makes it economically important to determine the fraction of broken kernels.

Because manual counting is both expensive and subjective (different people apparently produce different results), an automated system is required. A flatbed scanner is an ideal

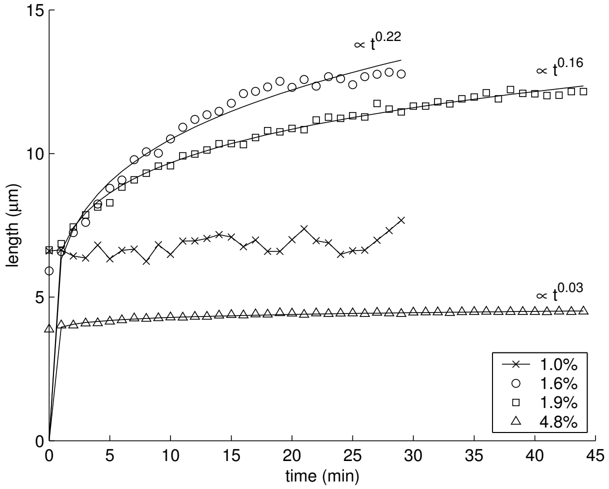


Figure 7.7: *Median pore size plus median object size as a function of time, as measured with the granulometry. The 4.8% sample has a lower value at $t = 0$ than the other samples. This is due to the apparently smaller objects in the images of this sample.*

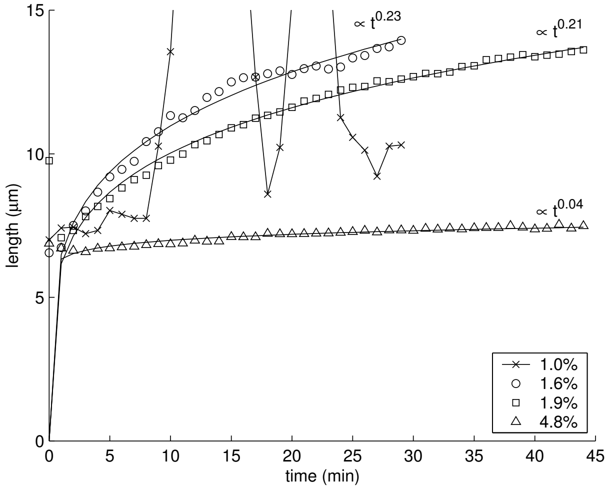


Figure 7.8: *Characteristic length as a function of time, as measured with the Fourier method. The 1.0% sample shows a larger oscillation here than with the granulometry. This is due to some very large, bright blobs appearing in the sequence at two time intervals. These are probably floating to the bottom of the vial, intersecting the imaging plane at some point in time.*

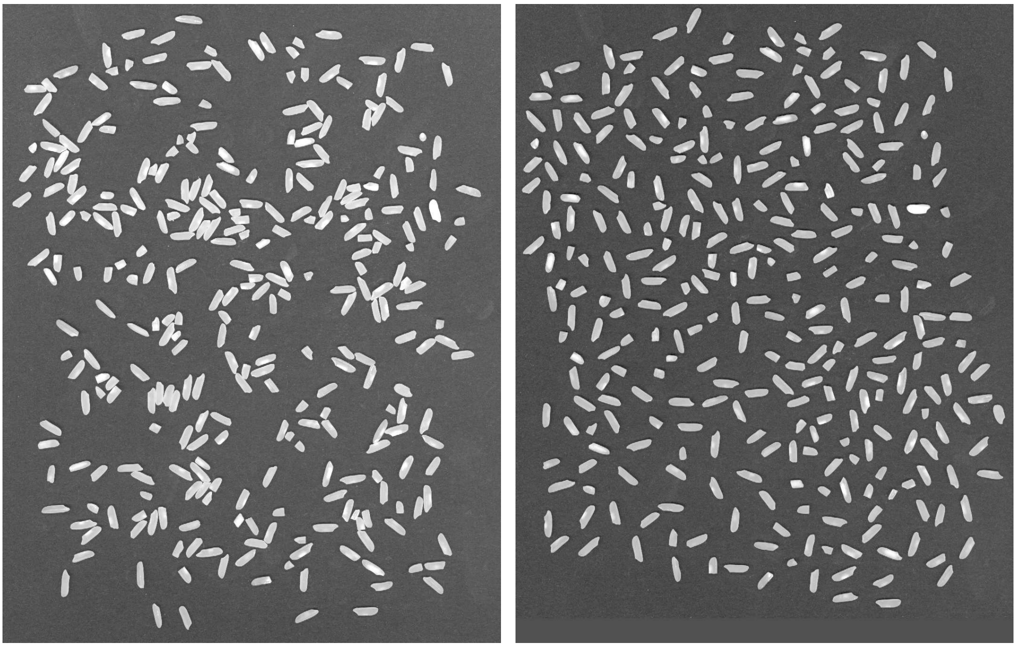


Figure 7.9: *Two images of rice kernels. The image on the left has been made after carefully separating all kernels to make segmentation easy.*

instrument to image rice, but it takes a lot of time to distribute the rice kernels on it in such a way that segmentation is possible. Therefore, we have applied a granulometry (with RIA openings), as a segmentation-free measurement technique. From the length distribution it is possible to derive the fraction of broken kernels. A disadvantage of using a scanner is that the background is far away from the scanning surface. This, as well as ambient light, cause the background of the image to be very noisy. This problem can be overcome with appropriate pre-processing.

Figure 7.9 shows two images of rice kernels obtained by placing the rice on a flatbed scanner. The image on the left has all kernels manually separated before acquisition, which takes about 15 minutes. The one on the right contains the same kernels randomly scattered on the scanning surface. As stated before, it is not trivial to correctly segment such an image. Thus, the classical measuring paradigm (threshold, label, measure the segmented objects) is not easily applied.

In total we have 6 images of the same sample, 20% of which consists of broken kernels:

- two images with only the broken kernels (one touching, one separated; Figure 7.10),

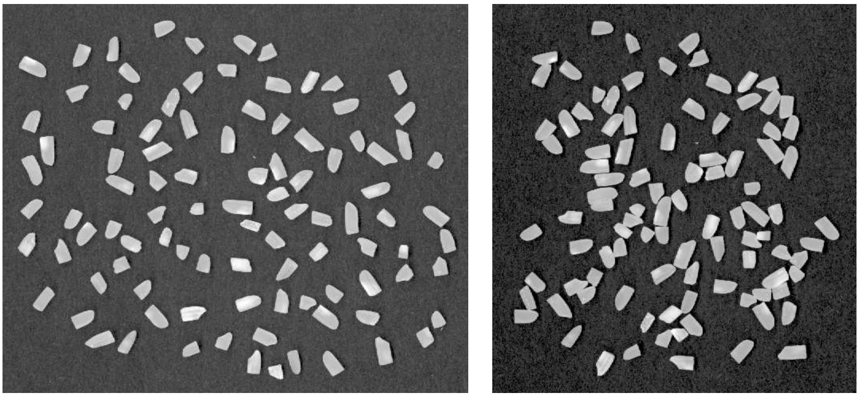


Figure 7.10: *The broken rice kernels used in the images from Figure 7.9.*

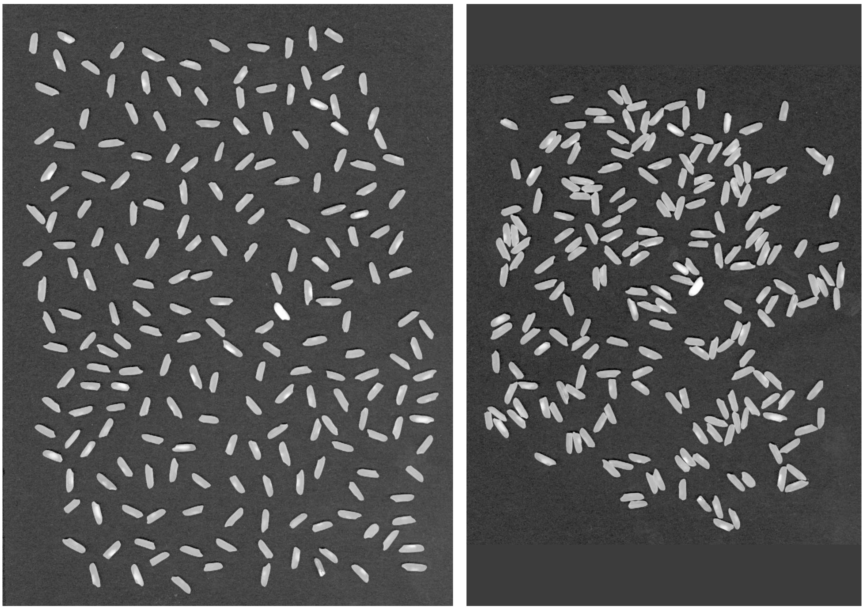
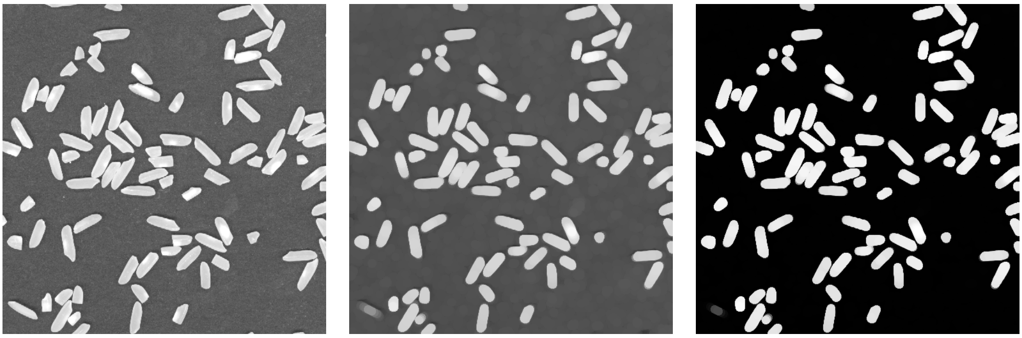


Figure 7.11: *The intact rice kernels used in the images from Figure 7.9.*



a: *Input* b: *Opening* c: *Soft-clipping*

Figure 7.12: *Preprocessing of the images. First, an opening removes thin elements* (b)*, which are not counted in the length distributions. Then, an error-function clip (soft-clipping) is applied* (c).

– two images with only the intact kernels (again, one touching, one separated; Figure 7.11), and
– two images with all kernels (again, one touching, one separated; Figure 7.9).

Applying the Granulometry

When studying the rightmost image in Figure 7.9, we notice that the places where the rice grains touch are quite wide, and do not show any decrease of gray-value. This means that the line segments of the RIA opening would be able to cross the boundary without penalty, causing an overestimation of the lengths. This overestimation can be avoided by increasing the width of the line segments used in the RIA opening. The goal is to avoid the structuring element from spanning the boundaries between the rice kernels. The best way to accomplish this is to create a structuring element that has approximately the shape of a rice kernel. Any other shape (such as elliptic or rectangular) would cause a bias towards smaller scales. To simplify the task of creating such a structuring element, we employ some pre-processing that simplifies the shapes in the image. The pre-processing is discussed below; refer to Figure 7.12.

Because we use thick line segments as structuring elements, all kernels and portions of kernels that are thinner than these lines will be put into the smallest scale of the granulometry. This adds a bias to the graph. To overcome this bias we remove these grains using an opening with a disk of diameter equal to the width of the line segments. Since very few rice kernels are too thin, removing them introduces only a very small

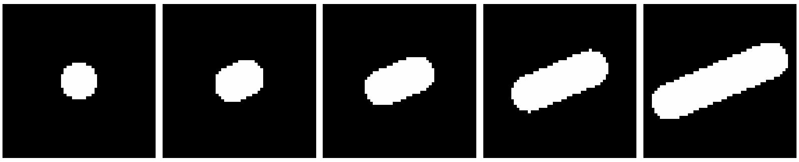

a: $l = w$ b: $l = \sqrt{2}w$ c: $l = 2w$ d: $l = 2\sqrt{2}w$ e: $l = 4w$

Figure 7.13: *Part of the sequence of structuring elements used.* a *is the structuring element used in the pre-processing step, so it defines the starting point of the sequence. From* b *to* e, *the lengths l increase exponentially with a factor* $\sqrt{2}$, *to obtain a logarithmic sampling of the scales; here it is expressed in terms of w, the width, which is 13 pixels in this case.*

imprecision in the measurements (assuming they do not have a larger probability of being broken). The thinner portions of the kernels that are also removed (which are usually the parts surrounding the germ) cause these kernels to be somewhat shorter. This yields an average underestimation of the lengths of four pixels (result obtained experimentally). This systematic error yields a shift to the left of the cumulative length distribution, but can be accounted for if necessary.

The second operation that is applied to the images is an error-function clip (see Section 6.2). Its need is two-fold: removing noise in the background, and equalizing the gray-value over the rice kernels. Some of these contain a chalky portion, caused by an unbalanced growing process (Figure 7.12a). This chalky portion is imaged whiter than the rest of the kernel, and would influence the length distribution by adding weight to the smaller scales.

After these pre-processing steps the rice kernels have a somewhat uniform shape with a half-disk at each end. Rotated versions of such a shape are easy to generate. We do not need to use interpolation as with the thin line segments in Chapter 5, because these thicker line segments yield a smaller discretization error. Figure 7.13 shows some of the structuring elements used. Using gray-value variants of these shapes (band-limited shapes like the disks in Section 3.5 and the line segments in Section 5.5), very similar results were obtained. The width of the lines was 13 pixels, and $\lceil \frac{1}{2}\pi l \rceil$ orientations were used for the RIA opening with length l. In the terminology used in Section 5.7, this is equivalent of using a value of q equal to 2. This is a significantly smaller value than that suggested earlier. However, because of the width of the lines, less orientations are required. Even so, an over-estimation of the cumulative size distribution at the smaller scales should be expected.

The Classical Measuring Paradigm

To compare our results with those obtained with an existing algorithm, we measured the length distribution using the Feret length measure (longest projection) on the thresholded and segmented image. This works well on the images where the kernels have been manually separated before acquisition, but produces poor results on the images with touching kernels (a similar thing has been done before in e.g. [112, 148]). The algorithm we used to determine the Feret length uses a chain-code representation of the object boundary, which can be easily rotated. The longest projection of the boundary is used as the object length. We used the estimated lengths to build a cumulative volume-weighted length distribution that could be compared with the granulometric curve.

Results

The length distributions of the rice kernels, measured from the available images using the proposed granulometry (using four samples per octave, which means that $s[i] = 2^{i/4}$), are plotted in Figure 7.14. The results for images with touching kernels are almost identical to the results for images with all kernels separated. For the classical method this is not the case (see Figure 7.15). Touching kernels in this case yield a very large over-estimation of the lengths.

Comparing these two figures also reveals that the granulometry method produces a result very close to that obtained by the classical method (on the images with the separated rice kernels). The largest difference is that the granulometry measures some volume at the small scales for the intact kernels. As discussed above, this is due to the very limited number of orientations used in the RIA openings. This bias, however, is easy to overcome by studying the whole curve instead of specifying a single threshold value (which would be quite nice to do since this would require only the computation of a single RIA opening instead of a whole granulometry).

Figure 7.16 studies the effect of the opening in the pre-processing stage. It shows the result of the classic measuring algorithm with and without this pre-processing operation. It turns out that the discriminating threshold moves from about 5.4mm to about 4.8mm. However, the discriminating ability is not affected at all.

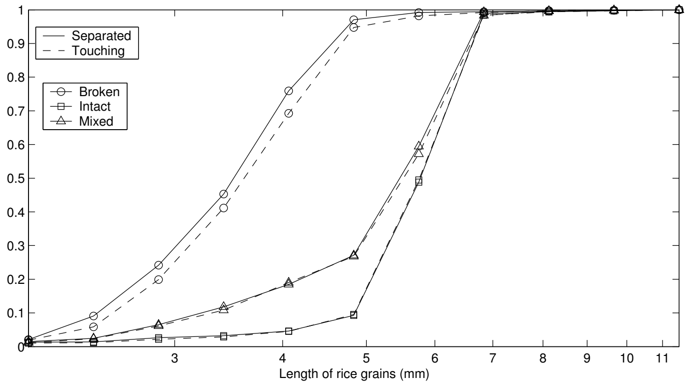


Figure 7.14: *Cumulative distribution measured for the various input images using a granulometry with* RIA *openings with thick line segments. This figure shows that it is possible to measure the fraction of broken kernels without having to manually separate each of them: the influence of the contact between rice kernels is very small.*

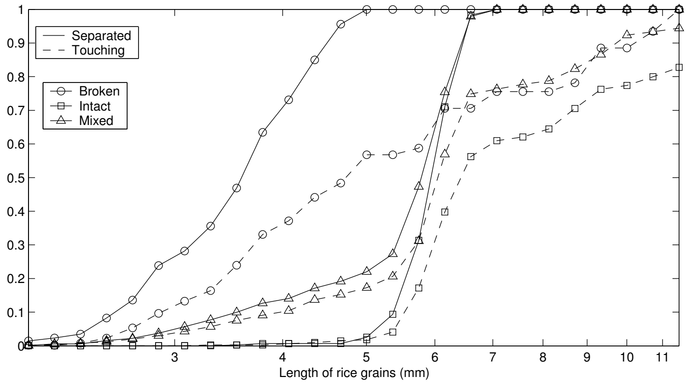


Figure 7.15: *Cumulative distribution measured for the various input images using the classical measuring paradigm. This figure shows that it is easy to measure the fraction of broken kernels in this way, given that they are all separated manually (continuous lines). The procedure breaks down for a random distribution of rice grains on the scanning surface (dashed lines).*

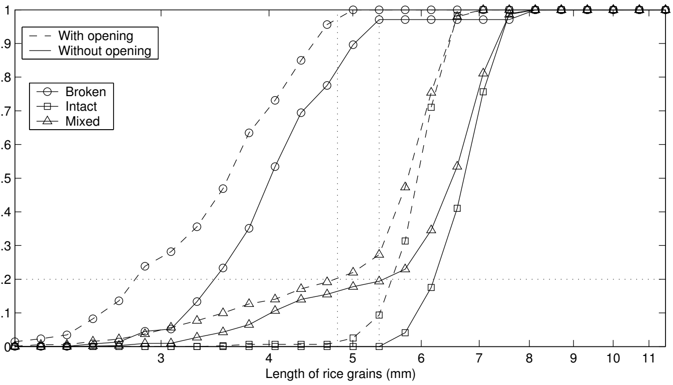


Figure 7.16: *Cumulative distribution measured for the various input images using the classical measuring paradigm, with and without the opening we employed before the granulometry. This graph shows the effect of this pre-processing step: the distribution is biased (shifted towards smaller lengths) meaning that the threshold to discriminate broken from intact kernels becomes approximately 4.8mm instead of 5.4mm, as should be used on the original images. (We can also see that on the image with the broken grains, in spite of the effort put into it, there are still two kernels touching a bit; these were separated by the opening in the pre-processing stage.)*

To steal ideas from one person is plagiarism; to steal from many is research
(Anonymous)

Chapter 8

The Radon Transform

The generalized Radon (or Hough) transform is a well-known tool for detecting parameterized shapes in an image. The Radon transform is a mapping between the image space and a parameter space. The coordinates of a point in the latter correspond to the parameters of a shape in the image. The amplitude at that point corresponds to the amount of evidence for that shape. In this paper we discuss three important aspects of the Radon transform. The first aspect is discretization. Using concepts from sampling theory we derive a set of sampling criteria for the generalized Radon transform. The second aspect is accuracy. For the specific case of the Radon transform for spheres, we examine how well the location of the maxima matches the true parameters. We derive a correction term to reduce the bias in the estimated radii. The third aspect concerns a projection-based algorithm to reduce memory requirements.[1]

8.1 Introduction

One of the first stages in image analysis is the extraction of primitives, such as lines, edges, curves or simple textures, from an image. In this paper we focus on curve detection, or more precisely, *shape* detection. In three- and higher-dimensional spaces, manifolds (N-dimensional), such as a spherical membrane, are as interesting as curves (one-dimensional). In general we are interested in a given *family of shapes*. Our assumption is that the members of this family can be described by a set of parameters. The task, then, is to find the parameters corresponding to the best fitting member of the family of shapes. The standard method for detecting parameterized shapes is based on a family of transformations, which includes the Radon [95] and Hough [47] transforms.

[1] This paper will be submitted to *Pattern Recognition* as “The Generalized Radon Transform: Sampling, Accuracy and Memory Considerations”, C.L. Luengo Hendriks, M. van Ginkel, P.W. Verbeek and L.J. van Vliet.

The organization of the paper is as follows: we continue the introduction by briefly discussing shape representation, followed by a short tutorial overview of the different transformations. The main point of this overview is to show that the different transformations are in fact different manifestations of a single unifying transform as has been described earlier, with some minor differences, in [30, 93, 113, 122]. To clarify the definitions and results obtained, we will use the detection of hollow hyper-spheres in D-dimensional space (circles in 2D) throughout the paper.

The transform we describe is continuous and should operate on continuous images, but to allow computer processing we must work with sampled images. Likewise, the transform should be discretized. This is one of the main points of the paper and is discussed in Section 8.2. To allow discretization we must, in most cases, replace the original transform by a regularized version. The consequences of the regularization are discussed briefly in Section 8.3. We study the particular case of a Radon transform for hollow spheres in more detail: the regularization in combination with a normalization for the surface area of the sphere leads to a bias in the estimated radius. This is the second contribution of the paper. The third contribution of the paper is a projection-based scheme to reduce the memory requirements of the transform-based approach and is described in Section 8.4. We verify our results by performing some experiments on the aforementioned hollow spheres.

Shape representation using generalized functions

Before proceeding, we introduce the following notation:

$\vec{x}$ The spatial coordinates

$I(\vec{x})$ The D-dimensional image containing the N-dimensional shapes

$\vec{p}$ The vector containing the parameters of the curve. Often a subset of the parameters specifies the location of the shape. It is, therefore, sometimes convenient to write $\vec{p} = \{\vec{q}, \vec{x}_o\}$, with $\vec{x}_o$ the location of the shape (the center of the sphere), and $\vec{q}$ the remainder of the parameters (the radius of the sphere).

$c(\vec{p})$ A member of a class of shapes described by the parameter vector $\vec{p}$.

$\vec{c}(\vec{s}; \vec{p})$ The coordinates of a point belonging to the shape $c(\vec{p})$. The coordinates $\vec{s}$ allow us to specify a specific point on the shape.

$\mathscr{C}(\vec{x};\vec{p})$ A set of constraint functions that together define the shape. The number of constraint functions depends on the dimensionality of the shape: $D-N$ constraints are necessary to describe a N-dimensional shape. For a point that lies on the shape, all the constraint functions evaluate to zero: $\mathscr{C}_i(\vec{x};\vec{p}) = 0$ for all i.

$C(\vec{p},\vec{x})$ A kernel, also called template, that represents the shape given by $\vec{p}$ as an image with spatial coordinates $\vec{x}$. We can model the image I as a sum of several of these templates.

Shapes can be described in different ways. The notation $\vec{c}(\vec{s};\vec{p})$ represents the shape. For a circle in 2D centered at $\vec{x}_o$ and with radius r this becomes

$$\vec{c}(\phi;\{r,\vec{x}_o\}) = \vec{x}_o + r\begin{pmatrix}\cos\phi\\ \sin\phi\end{pmatrix} \quad , \tag{8.1}$$

with ϕ a free coordinate letting us specify an arbitrary point on the circle. Alternatively, a shape can be defined through the specification of a constraint; this is known as the implicit representation. In the case of a circle:

$$\mathscr{C}(\vec{x};\{r,\vec{x}_o\}) = 0 \quad \text{with} \quad \mathscr{C}(\vec{x};\{r,\vec{x}_o\}) = \|\vec{x}-\vec{x}_o\| - r \quad . \tag{8.2}$$

Now recall that the shapes we are looking for are embedded in an image and not directly available as a set of points. This means that standard results from differential geometry, such as the expression for the curvature of a plane curve [123]

$$\kappa = \frac{\dot{x}\ddot{y}-\dot{y}\ddot{x}}{(\dot{x}^2+\dot{y}^2)^{\frac{3}{2}}} \tag{8.3}$$

are not directly applicable. In this example, the curvature of a curve embedded in an image in the form of an isophote can still be obtained through the well-known result for the isophote curvature [60, 63, 133].

In general, however, it may be beneficial to make the embedding of the shapes explicit. The basic ingredients for such a description are the constraint-based description and the Dirac delta function. The theoretical basis for this description can be found in Gel'fand et al. [36, Chapter III, Section 1], who give a very lucid account of this subject matter. It is not our intention to give a complete exposition of this material; we will merely touch upon the essentials.

Consider an N-dimensional shape in D-dimensional space. At any point on the shape we can define a local coordinate system. We will denote the local coordinate vector by

$\vec{u}$. The first N coordinates $u_{1\ldots N}$ span the subspace in which the shape lies and the $D-N$ remaining coordinates $u_{(N+1)\ldots D}$ span the subspace normal to the shape. In fact, these last coordinates act as constraint functions: we can set $\mathscr{C}_i = u_{i+N}$. We can describe the infinitesimal neighborhood I^n by

$$I^n(\vec{u}) = \delta(u_{(N+1)}, \ldots, u_D) \quad . \tag{8.4}$$

If we choose an orthogonal coordinate system, then this reduces to

$$I^n(\vec{u}) = \prod_{i=N+1}^{D} \delta(u_i) \quad . \tag{8.5}$$

Let us examine two simple examples in three-dimensional space. The x-y plane is described by the constraint $z=0$. Therefore,

$$I_{[x\text{-}y \text{ plane}]}(x,y,z) = \delta(z) \quad , \tag{8.6}$$

represents this plane. A line along the x-axis is described by the constraint $y=0$ and $z=0$:

$$I_{[\text{line along } x]}(x,y,z) = \delta(y)\delta(z) \quad . \tag{8.7}$$

Because of the simplicity of these two examples, we did not need a local coordinate system, but in general this *is* necessary. There is one last aspect we must consider. The constraints $z=0$ and $a(x,y,z)z=0$ with $a(x,y,z)>0$ represent the same shape, but the images $I(x,y,z)=\delta(z)$ and $I(x,y,z)=\delta(a(x,y,z)z)$ are not the same. The difference lies in how different points of the shape contribute to a volume integral over the image. This is immediately obvious when using the scaling property of the Dirac delta: $\delta(a(x,y,z)z) = (1/a(x,y,z))\delta(z)$. A priori, all the points of the shape should contribute equally. We can ensure this by scaling the u_i such that they correspond to the Euclidean distance (in image space $\vec{x}$) to the shape. If the constraint functions are chosen according to these principles, we write

$$I(\vec{x}) = \delta(\mathscr{C}(\vec{x};\vec{p})) \quad . \tag{8.8}$$

The Radon transform

The Radon transform is named after J. Radon who showed how to describe a function in terms of its (integral) projections [95]. The mapping from the function onto the projections is the Radon transform. The inverse Radon transform corresponds to the reconstruction of the function from the projections. The original formulation of the Radon

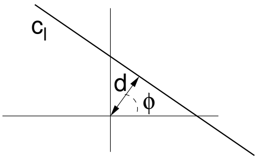


Figure 8.1: *The* normal *parameterization of a line. The parameters are the distance d from the line to the origin, through the normal of the line that intersects the origin, and the angle ϕ between that same normal and the x-axis, as indicated in the diagram.*

transform is:

$$\mathscr{R}\{I\}(d,\phi) = \int_{\mathbb{R}} I(d\cos\phi - s\sin\phi, d\sin\phi + s\cos\phi)ds \quad , \tag{8.9}$$

with the projection along the lines $c_l(d,\phi)$, and the parameterization as given in Figure 8.1. Within the realm of image analysis, the Radon transform is mostly known for its role in computed tomography. It is used to model the process of acquiring projections of the original object using x-rays. Given the projection data, the inverse Radon transform, in whatever form (most notably backprojection), can be applied to reconstruct the original object.

The Radon transform can also be used for shape detection. We reformulate the Radon transform:

$$\mathscr{R}\{I\}(d,\phi) = \int_{(x,y) \text{ on } c_l(d,\phi)} I(x,y)\,dx\,dy = \int_{\mathbb{R}^R} I(x,y)\delta(x\cos\phi + y\sin\phi - d)\,dx\,dy \quad . \tag{8.10}$$

It is now trivial to generalize the Radon transform to arbitrary shapes $c(\vec{p})$. We give three equivalent formulations, leaving it to the reader to decide which is clearest:

$$\mathscr{R}_{c(\vec{p})}\{I\}(\vec{p}) = \int_{\vec{x} \text{ on } c(\vec{p})} I(\vec{x})\,d\vec{x} = \int_{\mathbb{R}^N} I\big(\vec{c}(\vec{s};\vec{p})\big)\,||\frac{\partial\vec{c}}{\partial\vec{s}}||\,d\vec{s} = \int_{\mathbb{R}^D} I(\vec{x})\delta(\mathscr{C}(\vec{x};\vec{p}))\,d\vec{x} \quad . \tag{8.11}$$

For our study of the discretization of the transform, we choose the third formulation. The mathematical properties of this generalized form of the Radon transform have been extensively studied in [35].

Now imagine that there is a shape in the image with parameter set $\vec{a}$. When $\vec{p} \neq \vec{a}$, the Radon transform will evaluate to some finite number which is proportional to the number of intersections between the shapes $c(\vec{p})$ and $c(\vec{a})$, as illustrated in Figure 8.2. However, when $\vec{p} = \vec{a}$, the Radon transform yields a large response (a peak in the parameter space).

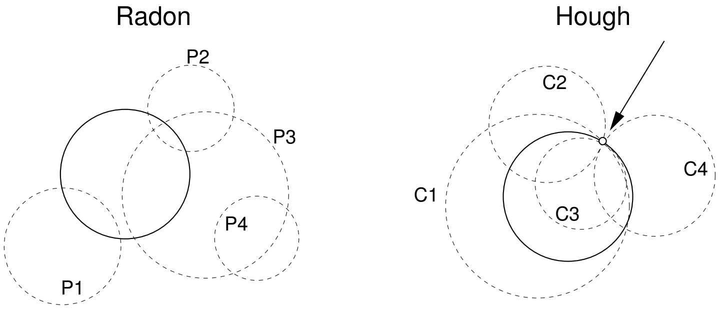


Figure 8.2: *The Radon and Hough transforms explained. Left: the Radon transform. Integrating the intensity values along each of the candidate curves P1–P4 yields small numbers. Only if a candidate curve happens to fully coincide with a curve in the image (the solid black circle), will the integral yield a large response. Right: the Hough transform. The indicated point can be accounted for by the presence of any of the indicated candidate curves C1–C4. If we consider all the points in the image in turn, we get many curves that account for individual points, but only candidate curves that correspond to a true curve (in this case the solid circle) will account for all of the points.*

This response is proportional to the N-dimensional hyper-volume of the shape. We can now interpret the Radon transform as follows: it provides a mapping from image space to a *parameter space* spanned by the parameters $\vec{p}$. The function created in this parameter space, $P(\vec{p})$, contains peaks for those $\vec{p}$ for which the corresponding shape $c(\vec{p})$ is present in the image. Shape detection is reduced to the simpler problem of peak detection.

The third formulation of the Radon transform in equation (8.11) demonstrates an important reason for using generalized functions. In this notation, we can recognize the form of a linear integral operator[2] $\mathscr{L}_K$ with kernel C:

$$(\mathscr{L}_C I)(\vec{p}) = \int_{\mathbb{R}^D} C(\vec{p},\vec{x}) I(\vec{x})\, d\vec{x} \quad . \tag{8.12}$$

Therefore, if we allow the kernel C to be a generalized function, then we can express the Radon transform in this format, which is particularly convenient to study its discretization. In case of a Radon transform, the kernel C is of the form: $C(\vec{p},\vec{x}) = \delta(\mathscr{C}(\vec{x};\vec{p}))$. In terms of shape detection, the role of the operator $\mathscr{L}_C$ is to compute the match (the inner product) between the image and a template C for a given parameter set $\vec{p}$. Here we see the connection between the Radon transform and template matching.

[2] This is known as a Fredholm operator.

Often, the parameters $\vec{p}$ consist of the position of the shape $\vec{x}_o$ and the actual shape parameters $\vec{q}$. In this case the kernel has a special (shift-invariant) structure:

$$C(\{\vec{q},\vec{x}_o\},\vec{x}) = C(\{\vec{q},\vec{x}_o+\vec{d}\},\vec{x}+\vec{d}) \quad \text{for any } \vec{d} \quad . \tag{8.13}$$

The operator $\mathscr{L}_C$ now reduces to a set of convolutions,

$$(\mathscr{L}_C I)(\vec{q},\vec{x}_o) = \big(K_C(\vec{q}) *_{\vec{x}} I\big)(\vec{x}_o) \quad \text{with} \quad K_C(\vec{q},\vec{x}) = C(\{\vec{q},\vec{x}\},\vec{0}) \quad . \tag{8.14}$$

This implies a large speed-up: using the convolution property of the Fourier transform, each convolution reduces to a multiplication in the Fourier domain. We may also note that the use of generalized functions as convolution kernels is widespread and necessary to describe the identity system, derivators and integrators.

Use of the Radon transform for shape detection dates back to 1965 [15]. The technique these authors describe is essentially a Radon transform. Rosenfeld [101] describes this technique (for straight lines) in Section 8.4.e. Neither [15] nor [101] identify this technique as the Radon transform.

The Hough transform

Rosenfeld [101] describes *two* techniques for curve detection: the first corresponds, as stated in the previous section, to the Radon transform. The second is a transform due to Hough [47], which through work by early adopters [33, 57, 81, 109] has become very popular.

The Hough transform was originally defined to detect straight lines in black and white images, and seemingly inherently discrete. As it is trivial to generalize the Hough transform to other shapes and gray-value images, we describe it in this extended form. We set up an N-dimensional accumulator array, each dimension corresponding to one of the parameters of the shape looked for. Each element of this array contains the number of votes for the presence of a shape with the parameters corresponding to that element. The votes themselves are obtained as follows. Consider each point in the input image in turn. Now we determine which shapes this point, with gray-value g, could be a member of; see Figure 8.2. We increment the vote for each of these shapes with g. Of course, if a shape with parameters $\vec{p}$ is present in the image, all of the pixels that are part of it will vote for it, yielding a large peak in the accumulator array. The Hough transform, like the Radon transform, is a mapping from *image space* to a *parameter space*.

The literature on the Hough transform has focussed on several aspects: error analysis [4, 59, 87, 94, 109, 110, 111, 132], reduction of the computational complexity (see

Section 8.4), extensions [5, 11, 37] (to e.g. other shapes or use of additional information, such as orientation), and choosing the appropriate parameterization [33, 145, 147]. An extensive survey of the Hough transform literature up to 1988 is given in [50].

It is possible to construct a continuous formulation for the Hough transform by noticing that the "voting" procedure accumulates the results in parameter space, which is equivalent to integration. So how do we arrive at a formulation in terms of an integration? The answer is simple: instead of considering to which points in parameter space a point in input space contributes, we may turn around and ask "which points in input space contribute (and how much) to any given point in parameter space?". This line of reasoning has been used by Stockman and Agrawala [122], and Sklansky [113] to link the Hough transform to template matching. Strangely enough, none of these authors have taken the next logical step: defining the continuous Hough transform through the continuous formulation of template matching, as given in equation (8.12). This final step was first made explicit by Princen et al. [93].

The starting point in the work by Princen et al. [93] is somewhat different from that given above and perhaps more in the spirit of the "Hough frame of mind". At the basis for their formulation are the constraint functions $\mathscr{C}$. For any given point $\vec{x}$ in the input space, the constraint(s) $\mathscr{C}(\vec{x};\vec{p})$ trace out a manifold in the parameter space spanned by the parameters $\vec{p}$. If we consider more points $\vec{x}$, we get more of these manifolds. If consider points $\vec{x}$ on a shape with parameters $\vec{p}_0$, then the corresponding manifolds will intersect each other at the point $\vec{p}_0$ in parameter space. Nevertheless, in the final step to obtain the kernel description (8.12), the authors return to the reasoning in [122] and carry it over to the continuous domain.

The Radon and Hough transform can therefore both be written in the form of (8.12) with kernel functions of the form $\delta(\mathscr{C}(\vec{x};\vec{p}))$. The apparent difference between the two is only superficial and basically amounts to a difference in the computational interpretation of (8.12):

Reading paradigm (Radon): For each $\vec{p}$, collect all the values of $I(\vec{x})$, apply the template weights $K(\vec{x};\vec{p})$, and sum everything.

Writing paradigm (Hough): Initialize the entire function $P(\vec{p})$ to zero. For each point $\vec{x}$ in the input image determine its contribution, weighted with $K(\vec{x};\vec{p})$, to each of the points in $P(\vec{p})$ and update $P(\vec{p})$.

The difference in interpretation can be beneficial: if the input data is sparse, the Hough paradigm offers an immediate reduction in computation time. Vice versa, if we are interested in only a few points in parameter space, the Radon paradigm is to be preferred.

The equivalence of the Radon and Hough transforms, as well as the fact that they can be seen as a form of template matching, has been known for a long time. It seems to have been tacitly noted by Griffith [40]: the technique he describes is basically a Radon transform and goes on to refer to Hough [47] as an earlier example of a projection-based (i.e. Radon) technique. Stockman and Agrawala [122] and Sklansky [113] pointed out the equivalence of the Hough transform and template matching. Deans [30] discussed the equivalence of the Radon and Hough transforms, starting from the Radon transform and showing that it has the same properties as the Hough transform.

Finally, it should be pointed out that Princen et al. [93] claim that the Radon transform and the Hough transform are only equivalent “under certain restrictive conditions”. This reasoning is flawed for two reasons. First, they compare the Radon transform to the original, discrete definition of the Hough transform, instead of their continuous definition of the Hough transform. Second, they do not seem to have realized that the Radon transform can be written in the form of a kernel operation as well. Indeed, the kernel as defined in [93] does not make use of generalized functions. The consequence is that their kernel has measure zero, meaning that the integral will evaluate to zero *unless the input data contains impulse functions*. As we discuss in the next section, we are interested in a more general model for the input data.

8.2 Sampling the Radon transform

As mentioned in the section about the Hough transform, several authors have investigated the discretization effects associated with it. Here, we will investigate the effects of sampling on the continuous formulation of the Radon/Hough transform.

It is important to discriminate between two models for the input data. The first is used in [58, 93]: the input space is continuous and contains points at arbitrary (sub-pixel) locations,

$$I(\vec{x}) = \sum_i \delta(\vec{x} - \vec{x}_i) \quad . \tag{8.15}$$

In this model, noise corresponds to perturbations in the position of the points. Given this model, the conditions under which $P(\vec{p})$ can be sampled were given by Kiryati and Bruckstein [58]. Basically, the manifolds that are stamped in parameter space must be band-limited, very much like the use of Parzen windows for creating smooth probability density functions [34].

In our work we consider a different model. In most image analysis problems we start out with a continuous gray-value image, which is subsequently sampled for computer processing. Assuming the image is band-limited and properly sampled according to the Nyquist criterion [21], the sampled image represents the continuous function perfectly: the original can be reconstructed from the sampled image. The sampled image still contains all the subtleties of the continuous original, including sub-pixel information. Curves embedded in the image are faithfully represented. Sethian [106] starts out with planar curves and subsequently embeds these as zero-crossings in a gray-value image for this reason. By choosing the operators to be applied to the gray-value image appropriately, the embedded planar curves can be manipulated (shrunk or expanded for instance).

A large class of images is band-limited by virtue of the image acquisition process. Images obtained using an optical system are band-limited. If, however, the image is not band-limited, such as input data of the form given in (8.15), the image must be low-pass filtered to make it band-limited, implying a certain loss of resolution.

We now wish to process the sampled image such that the discrete processing and the result thereof are also faithful to their original continuous counterparts. Under which conditions is it possible to replace a continuous operations by the chain of operations consisting of sampling, a discrete operation, and reconstruction? This principle is known as *sampling invariance* and has been investigated by Verbeek [133] and van Vliet [139]. They have considered convolutions and multiplicative combinations of convolutions.

Here we follow van Ginkel [37] and study the conditions under which equation (8.12) is sampling invariant. There are two aspects. Keeping $\vec{p}$ fixed, we will first consider under which conditions we may replace I and C by sampled (along $\vec{x}$) versions and the integral by a summation. If these conditions are satisfied, we may compute $P(\vec{p})$ for an arbitrarily chosen $\vec{p}$. We must then show that it is possible to sample the parameter space $P(\vec{p})$, so that we only need to evaluate $P(\vec{p})$ on a discrete set of points. For simplicity we restrict ourselves to a one-dimensional example: $\vec{p} \to p$ and $\vec{x} \to x$. The Fourier axes corresponding to p and x are denoted by $\tilde{p}$ and $\tilde{x}$ respectively. The sampling distance along x is Δx, the discrete coordinate corresponding to x is n, i.e. the sampled version of $I(x)$ is $I(n\Delta x)$. We first investigate under which conditions the following is true:

$$P(p) \;= \int_{\mathbb{R}} C(p,x)I(x)dx = \Delta x \sum_{n\in\mathbb{Z}} C(p,n\Delta x)I(n\Delta x) \quad . \tag{8.16}$$

We denote the band-limit (along x) of the product $C(p,x)I(x)$ by $\mathrm{b}_x\{CI\}$. With p fixed, the sampling criterion for the computation of this integral is a relaxed version of the Nyquist criterion [133],

$$s_x > \mathrm{b}_x\{CI\} \quad , \tag{8.17}$$

where s_x is the sampling frequency along the x-axis. Also the band-limit of CI can be expressed in that of C and I,

$$\mathrm{b}_x\{CI\} \leq \mathrm{b}_x\{C\} + \mathrm{b}_x\{I\} \quad . \tag{8.18}$$

It follows that both the kernel C and the image I must be band-limited to allow discretization. Proper sampling of the image I is a prerequisite for any image analysis and therefore introduces no new restrictions. This is in general not true for the kernel C, on which we must impose a bandwidth limitation. This clearly leads to a different Radon transform, but reflects a conscious choice with well-understood consequences. These will be discussed in Section 8.2. The alternative, sampling C without imposing a band-limit first, leads to aliasing effects.

We can compute $P(p)$ for an arbitrary value of p. If $P(p)$ is band-limited, it can be safely sampled, provided the correct (Nyquist) rate is used. We determine whether $P(p)$ is band-limited by computing its Fourier transform:

$$\mathscr{F}\{P\}(\tilde{p}) = \mathscr{F}\left\{\int_{\mathbb{R}} C(p,x)I(x)dx\right\}(\tilde{p}) \; = \int_{\mathbb{R}} \mathscr{F}_p\{C(p,x)\}(\tilde{p},x)I(x)dx \quad . \tag{8.19}$$

If C is band-limited along the p-axis with band-limit $\mathrm{b}_p\{C\}$, then the integral above evaluates to zero for $\tilde{p} > \mathrm{b}_p\{C\}$, which means that $P(p)$ is band-limited as well.

The discussion above also holds for the complete multi-dimensional operation: our argument holds for each spatial dimension x_i separately and for each parameter dimension p_j as well. The same ideas also extend trivially to other sampling schemes, such as the hexagonal grid.

Band-limiting the kernel C

The Gaussian filter is approximately band-limited with critical sample spacing σ [139] and corresponding band-limit $b = \frac{1}{2}\sigma^{-1}$. Its properties, in particular good simultaneous frequency and spatial localization [91], and not introducing new structure [62], make it a good choice for band-limiting $C(\vec{p},\vec{x})$. We obtain C_b, a band-limited version of C, as follows:

$$C_b(\vec{p},\vec{x}) = C(\vec{p},\vec{x}) * G(\vec{p},\vec{x};\Sigma) \quad . \tag{8.20}$$

The diagonal covariance matrix Σ reflects that we impose band-limitation along each dimension separately.

By its nature the function $C(\vec{p},\vec{x}) = \delta(\mathscr{C}(\vec{x};\vec{p}))$ is in general very sparse: for any given

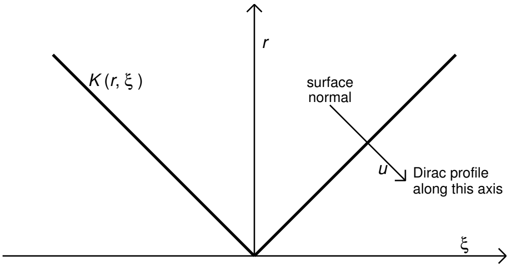


Figure 8.3: *The function $K(r,\vec{x})$ is a cone. For correct normalization, it is required that this function behave like the Dirac delta function along the normal to the surface.*

$\vec{p}$, the points $\vec{x}$ which belong to the shape span some curve or manifold in $C(\vec{p},\vec{x})$. The Radon transform for hyper-spheres provides a convenient example to investigate the structure of $C(\vec{p},\vec{x})$ and the effects of band-limitation. The parameter vector $\vec{p}$ consists of the center $\vec{x}_o$ of the D-dimensional sphere and its radius r: $\vec{p} = (x_1, \cdots, x_D, r)$. The kernel C becomes

$$C(\{r,\vec{x}_o\},\vec{x}) = K(r,\vec{x}_o - \vec{x}) \quad \text{with} \quad K(r,\vec{\xi}) = \delta\big(\frac{1}{2}\sqrt{2}(\|\vec{\xi}\| - r)\big) \quad . \tag{8.21}$$

The function K represents a cone. Each point on the surface of this cone should have equal weight as discussed in Section 8.1. In essence, we want the integral over the truncated function K, $\int_0^R \int_{\mathbb{R}^D} K(r,\vec{\xi})d\vec{\xi}dr$, to equal the surface area of the truncated cone with a base of radius R. The cone has a single coordinate u normal to the cone, see Figure 8.3. This coordinate is chosen such that it corresponds to the Euclidean distance to the surface: $u = \frac{1}{2}\sqrt{2}(\|\vec{\xi}\| - r)$. In Section 8.3 we discuss some issues that lead to a different choice for the normalization.

What is the effect on K of the Gaussian smoothing applied to C? Let us first consider the effect of the smoothing applied along the $\vec{p}$-axes. All parameters share the same units and it is therefore logical to use the same σ_K along each dimension. The effect on a local surface patch, if it can be considered planar locally ($\sigma_K \ll r$), is that the Dirac profile is substituted by a Gaussian profile

$$\begin{aligned} K_b(r,\vec{x}_o - \vec{x};\sigma_K) &= K(r,\vec{x}_o - \vec{x}) *_{\vec{x}_o,r} G(\vec{x}_o,r;\sigma_K) \approx \\ & G\big(\frac{1}{2}\sqrt{2}(\|\vec{x}_o - \vec{x}\| - r);\sigma_K\big) = \sqrt{2}\, G(\|\vec{x}_o - \vec{x}\| - r;\sqrt{2}\,\sigma_K) \quad . \end{aligned} \tag{8.22}$$

Because of the linear dependence between r and $\vec{x}_o$, the actual smoothing obtained along the grid axis is a factor $\sqrt{2}$ too large, as can be seen in Equation (8.22). This means that we can reduce the size of the Gaussian by that factor:

$$K_b(r,\vec{x}_o-\vec{x};\sigma_K)=\sqrt{2}\,G(\|\vec{x}_o-\vec{x}\|-r;\sigma_K) \quad . \tag{8.23}$$

The last step is to apply a Gaussian along the $\vec{x}$-axes. Because the structure of K along axes $\vec{x}_o$ and $\vec{x}$ is not independent, the Gaussian smoothing along $\vec{x}_o$ implies a Gaussian smoothing along $\vec{x}$. This makes it unnecessary to apply the Gaussian smoothing along the $\vec{x}$-axes, unless the required smoothing σ_s along the $\vec{x}$-axes is larger than that required along the $\vec{p}$-axes. A consequence is that the regularization in the parameter space only, as proposed in [58], is in this case sufficient to avoid discretization errors.

The consequences of the imposed band-limit are as follows: as long as the Gaussian is small with respect to the curvature of C, the effects of the Gaussian are negligible. In fact, it is possible to interpolate in $P(\vec{p})$ and obtain sub-pixel accuracy. High-curvature structures of C correspond either to highly curved shapes (such as the point of our cone, where r is small) or to shapes which vary rapidly as a function of the parameters. In neither case is it reasonable to expect good results anyway. The band-limitation does lead to a bias in the estimated radius. This effect and how it can be compensated for using a normalization term is the topic of Section 8.3.

A perfect discretization?

Under some circumstances it is possible to avoid discretization errors altogether. We first consider the convolution case [90, Section 8.4]. Let $l(x)$ be an ideal low-pass filter with a cut-off frequency that corresponds to the band-limit of the image $I(x)$, and $f(x)$ the filter to be sampled. Applying $l(x)$ to $I(x)$ has no effect: $I(x)*l(x)\equiv I(x)$. This leads to:

$$I(x)*f(x)=[I(x)*l(x)]*f(x)=I(x)*[l(x)*f(x)] \quad , \tag{8.24}$$

both $I(x)$ and the term between the brackets are band-limited and can be sampled.

This principle is also applicable to equation (8.12), but only along the $\vec{x}$ dimensions. Only in special cases, such that of the hyper-spheres, is this sufficient. In the general case it remains necessary to impose a band-limit on the parameter axes.

Ideal low-pass filters have some undesirable properties; ringing artifacts being the most important. It is also possible to achieve the same effect with Gaussian filters, but at a certain cost. If we oversample the image $I(x)$ by a factor three, and choose the size σ of the Gaussian equal to the new sampling distance, we have, to a very good approximation, $I(x)*G(x)\equiv I(x)$.

These ideas are interesting, but in practice the influence of the regularization is minor, so there is no need to resort to oversampling.

8.3 Accuracy of the parameter space

Due to the definition of the function $K_b(r,\vec{x};\sigma_K)$ in (8.22), large hyper-spheres will produce a higher value (higher confidence) than small ones in the parameter space. The consequence is two-fold:

- a few disconnected sections in the input will be selected as a large hyper-sphere with a higher confidence than a smaller but complete hyper-sphere, and
- the radii of hyper-spheres with thick walls will be over-estimated.

To avoid this, the spheres that compose $K_b(r,\vec{x};\sigma_K)$ should be normalized. That is, the integral of their gray-values should remain constant for any r. This results in

$$K_b(r,\vec{x};\sigma_K) = \frac{1}{S_D(r)}\, G(\|\vec{x}\| - r;\sigma_K) \quad , \tag{8.25}$$

where $S_D(r) = Qr^{-(D-1)}$ is the surface area of a D-dimensional hyper-sphere of radius r, and Q is a constant that depends on D.

This normalization causes a distortion of the shape of the cone, which in turn leads to an underestimation of the radius. We showed in [69][3] that the error in the position of a maximum in $P(\vec{p})$ along the r-axis, assuming $r \gg \sigma_K$, is dominated by the position of the maximum along the r-axis of $K_b(r,\vec{x};\sigma_K)$. To correctly estimate the radius of hyper-spheres, this maximum should lie at $\|\vec{x}\|$, but is shifted to $\|\vec{x}\| + \varepsilon(\|\vec{x}\|)$ by the normalization (see Figure 8.4). We find the position of the sifted maximum by equating the derivative of Equation (8.25) to zero,

$$\frac{\partial K_b}{\partial r} = Q\left[\frac{-(D-1)}{r^D} + \frac{\|\vec{x}\| - r}{\sigma_K^2 r^{D-1}}\right] G(\|\vec{x}\| - r;\sigma_K) = 0 \quad , \tag{8.26}$$

$$-(D-1)\,\sigma_K^2 - (\|\vec{x}\| - r)\,r = 0 \quad , \tag{8.27}$$

and solving for $\varepsilon = r - \|\vec{x}\|$, which yields

$$\varepsilon(\|\vec{x}\|) \approx -\frac{(D-1)\,\sigma_K^2}{\|\vec{x}\|} - \frac{(D-1)^2\,\sigma_K^4}{\|\vec{x}\|^3} + O\left(\|\vec{x}\|^{-5}\right) \quad . \tag{8.28}$$

[3] Printed in Appendix A.

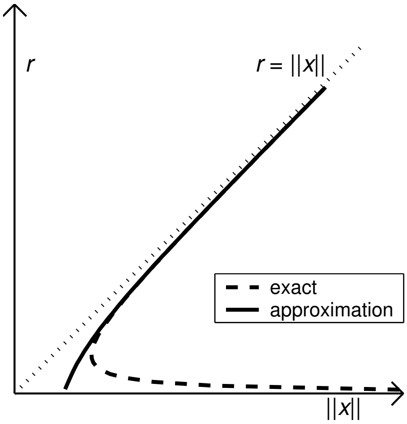


Figure 8.4: *The normalized function $K_b(r, \vec{x}, \sigma_K)$ from Equation (8.25) is distorted according to Equation (8.27). The correction we use is an approximation valid for larger r, and given by Equation (8.28).*

We will be using only the first term of this equation. As shown in [69], when taking into account the shapes in the image a new term of $O\left(\|\vec{x}\|^{-3}\right)$ should be added. However, it is not possible to correct for it since no assumptions about the image can be made.

Thus, the measured radius, R_m, is given by $R_m = \|\vec{x}\| + \varepsilon(\|\vec{x}\|)$. We substitute $\|\vec{x}\|$ by R_c, which we then can change such that R_m becomes equal to $\|\vec{x}\|$. R_c is the radius we use for the creation of the convolution kernel $K_b(r, \vec{x}; \sigma_K)$ at r.

$$\|\vec{x}\| = R_c - \varepsilon(R_c) \quad \Longrightarrow \quad R_c = \frac{1}{2}\|\vec{x}\| + \sqrt{\frac{1}{4}\|\vec{x}\|^2 - (D-1)\sigma_K^2} \quad . \tag{8.29}$$

$$K_b(r, \vec{x}; \sigma_K) = \frac{1}{S_D(r)}\, G(\|\vec{x}\| - R_c(r); \sigma_K) \quad . \tag{8.30}$$

8.4 Reducing memory requirements

The parameter space for the Radon transform typically has more dimensions than the input image. This implies that these parameter spaces might not fit into the available computer memory. This constraint has traditionally prevented wide-spread use of these transforms for 3D images.

Many authors have tackled this problem in a variety of manners. Most notably, Ballard and Sabbah [12] propose to partition the parameter space into two or more spaces with independent parameters, which can be computed sequentially. For example, it is possible to first locate all the object centers, and in a second stage determine the sizes. For (hollow) spheres, this requires the use of gradient information to locate centers [38], and is only practical for very few objects due to the cost of the second stage. Atherton and Kerbyson[9] encode the radius as the phase in the complex parameter space, thereby simplifying the this approach. Hsu and Huang [48] also use a dimensionality reduction to detect 3D ellipsoids (with 6 parameters, the axes are supposed to lie on the grid). They split the 6D parameter space into two 4D parameter spaces, which have to be combined to find the objects.

Another method often employed involves splitting the parameter space into overlapping regions, from which the maxima are extracted. This does not involve a reduction of dimensionality, but incurs a penalty in computational cost because of the overlap. In the case of a sphere, it is natural to split the parameter space along the r-axis, since a slice $P(r_i,\vec{x}_o)$ is computed by a single convolution. We will call this method the Sliding Window method (SW).

We propose a different approach to reduce the memory requirements. Spheres can be detected very efficiently by storing only the maximum projection along the r-axis of P, together with the location of these values on the r-axis (if one is prepared to ignore concentric spheres). That is, we keep

$$S\left(\vec{x}_o\right) = \max_r \left\{P\left(r,\vec{x}_o\right)\right\} \tag{8.31}$$

and

$$R\left(\vec{x}_o\right) = \arg\max_r \left\{P\left(r,\vec{x}_o\right)\right\} \quad . \tag{8.32}$$

The local maxima in $S(\vec{x}_o)$ indicate the location of the center of the spheres, and $R(\vec{x}_o)$ gives the corresponding radii. Both of these can be computed by a small modification of the Radon algorithm. Instead of storing all the $P(r_i,\vec{x}_o)$ slices, we propose to take the point-wise maximum of each slice with the previously computed intermediate result. This does not add any computational cost to the algorithm, since finding the local maxima needs to be done anyway. This maximum projection even simplifies this task. We call this method the Maximum Projection method (MP), and should be both faster and much less memory-hungry than the SW method.

The resulting parameter space $S(\vec{x}_o)$ is not band-limited. But, if the spheres are clearly identifiable and well separated, it turns out to have nicely-shaped peaks (i.e. the neighborhoods of the local maxima are band-limited or nearly so). Thus, it is still possible to obtain the center of the sphere with sub-pixel accuracy. However, the r-axis at each position has been discretized to sampling locations. The accuracy to which r can be estimated depends on the number of samples taken, not the band-limit of $K_b(r, \vec{x}_o; \sigma_K)$ along the r-axis.

It is possible to implement such a Radon transform for other shapes as well, in which case the maximum projection can be taken over more than one dimension. That is, only the spatial dimensions need to be kept, all other dimensions can be collapsed into one maximum image and one or more maximum position images, of which there are as many as the number of parameter dimensions that are collapsed.

8.5 Results

Evaluation

To demonstrate the claims made in the previous sections, we computed the Radon transform of 25 synthetically generated, 3D test images, 128^3 pixels in size, each containing 20 spheres of different radii (between 6 and 18 pixels) at random, sub-pixel locations. Some of the spheres were touching, but none were overlapping. These spheres had a Gaussian profile (with $\sigma_i = 1$), thereby approximating band-limitness. We computed the Radon transform with the two methods explained above (SW and MP), using the normalized kernel of Equation (8.25) and the corrected kernel of Equation (8.30) (setting $\sigma_K = 1$). The SW method uses a window of 7 slices in the radius direction, from which 2 slices overlap other regions. It required five times as much memory, and took about twice as much time to finish, as compared to the MP method. This is because the algorithm we used to find the local maxima is relatively expensive compared to the convolutions themselves.

We evaluated both methods by computing the differences between the true parameters of the spheres and the estimated ones. Table 8.1 compares the errors in the location for the different methods. We found that both SW and MP found the location of the spheres with the same precision and accuracy (for any given sphere the parameters found by the two methods are almost identical; differences are in the order of 2%). The bias is very small, not significant in relation to the standard deviation. Figure 8.5 shows the errors in the estimated radii for the SW method with and without correcting the kernel for bias. The MP method found the rounded values of the radii found by the SW method.

To examine the influence of noise, we added Gaussian noise to the images used above and repeated the experiments; the results are also shown in Table 8.1 and Figure 8.5.

Table 8.1: *Error made when estimating the position of spheres in synthetic 3D images. The numbers shown are standard deviations of the error (in pixels) in the x, y and z-directions for the various methods (estimated from 500 spheres of radius between 6 and 18 pixels).*

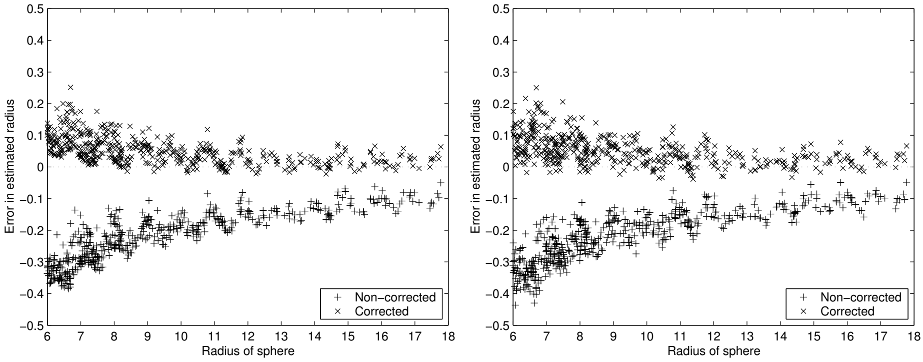
		$std(\delta x)$	$std(\delta y)$	$std(\delta z)$
noiseless image	non-corrected SW	0.055509	0.057290	0.059277
	corrected SW	0.055204	0.056107	0.060896
	corrected MP	0.054276	0.055121	0.059372
noisy image	non-corrected SW	0.067684	0.066870	0.070712
	corrected SW	0.067816	0.065254	0.071266
	corrected MP	0.067446	0.064059	0.070625

The input images have a SNR of 2 (with SNR= $\frac{\max I(\vec{x})-\min I(\vec{x})}{\sigma_N}$, $I(\vec{x})$ the uncorrupted image and σ_N the standard deviation of the noise probability density distribution). The standard deviation in the errors do not increase much for this noise level. This shows that the projection method is a good approximation with or without noise, and shows that the Radon transform itself is robust with respect to noise.

The graphs in Figure 8.6 are computed without the use of interpolation (by densely sampling along the r-axis), and show the attainable accuracy in the estimation of the radius with and without normalization and/or correction. They also show the effect of the width σ_K of the probes and the width σ_i of the spheres in the image. The non-normalized method is heavily influenced by σ_i, but not at all by σ_K. Normalizing the kernel makes the transform almost independent of σ_i, but it becomes dependent on σ_K. The bias correction proposed in Equation (8.30) removes this dependency, but the assumption made ($r \gg \sigma$) breaks down for smaller radii and larger σ.

Ballotini

As a demonstration application, we used a rather poor-quality X-ray micro-CT image of ballotini (small, hollow glass beads, see Figures 8.9a and 8.10a). Some of the glass walls give a very wide response in the imager (probably caused by refraction or reflection). In one such region many small spheres can be fitted. To avoid this, we replaced the kernel



a: SNR$=\infty$ b: SNR$=2$

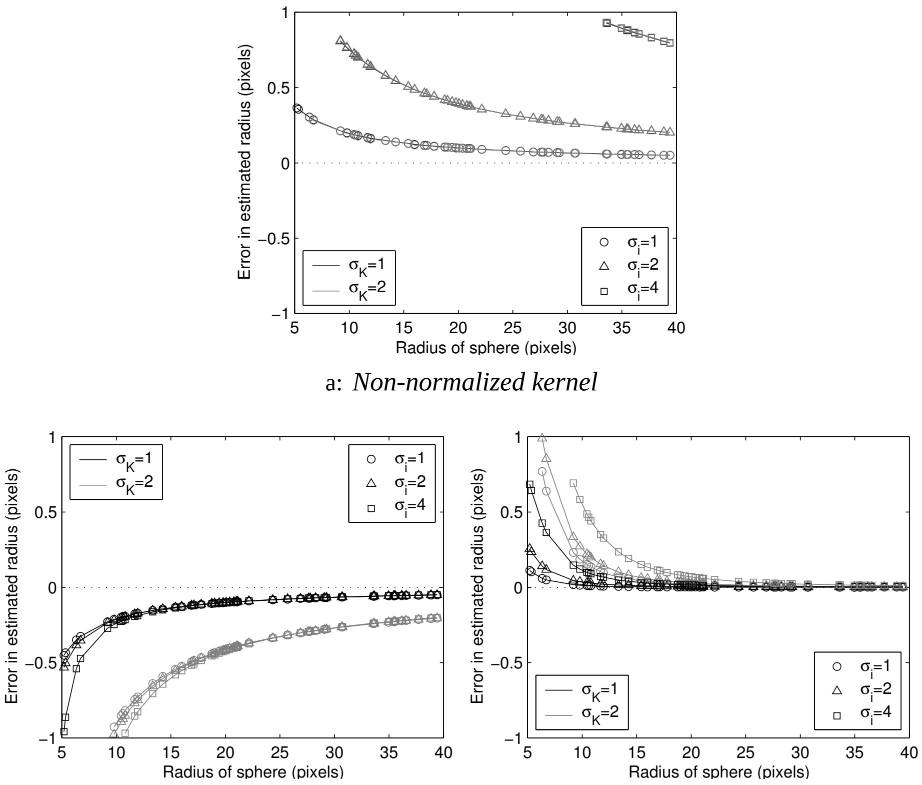
Figure 8.5: *Error made when estimating the radius of spheres in synthetic 3D images. The bias of the corrected method is due to the approximations used when computing Equation (8.28). The oscillatory nature of the lines is due to the interpolation along the r-axis: the error is smaller for values near sample points.*

K_b by a kernel K'_b that penalizes for high gray-values inside the sphere:

$$K'_b(r,\vec{x}_o;\sigma_K) = K_b(r,\vec{x}_o;\sigma_K) - K_b(r-d,\vec{x}_o;\sigma_K) \quad , \tag{8.33}$$

with K_b the corrected kernel as given in Equation (8.30), and d the difference in radius. By requiring that the inner part of the sphere be empty, the discriminating abilities of the transform (for these images) are greatly enhanced (compare image b to e in Figures 8.9 and 8.10). The computational complexity remains the same. We set $d = 4\sigma_K$, such that the sphere $K_b(r,\vec{x}_o;\sigma_K)$ is not affected too much (see Figure 8.7), since that would cause a large underestimation of the radius. In the synthetic test images of the evaluation we performed earlier, this setting leads to a slightly larger bias: an underestimation of about 0.1 pixel for large spheres, see Figure 8.8. This bias might be corrected for in the same manner as before.

To find the spheres in the parameter space $S(\vec{x}_o)$, a threshold is used to decide which local maxima are important enough to represent a sphere in the input images. More complex decision rules could be used, but are outside the scope of this paper. Figures 8.9 and 8.10 show the results for two different slices of the 3D image.



a: *Non-normalized kernel*

b: *Normalized kernel*

c: *Corrected kernel*

Figure 8.6: *Minimal error in estimation of the radius of spheres in synthetic 3D images, obtained without interpolation. These graphs show the effectiveness of the normalization and correction of the kernel $K_b(r, \vec{x}; \sigma_K)$, as well as the effect of σ_K and the σ_i used to create the input images.*

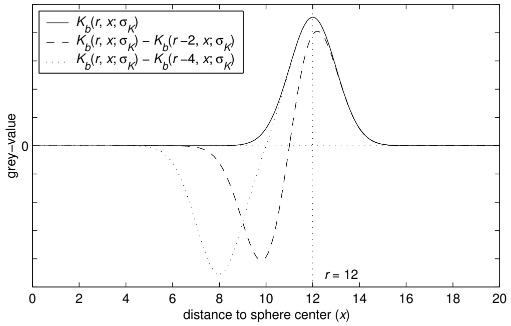


Figure 8.7: *Effect of the negative inner sphere: if it is too close to the outer sphere, the profiles will overlap too much, causing an outwards shift of the outer sphere. A distance of* 4σ *introduces only a small error. (In this graph,* $\sigma = 1$.*)*

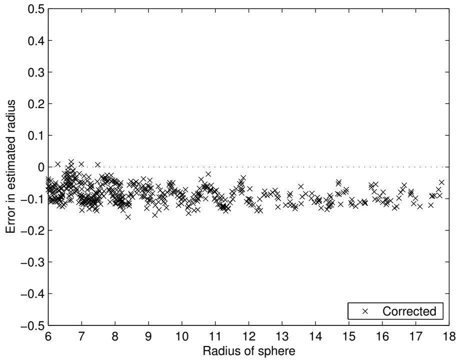
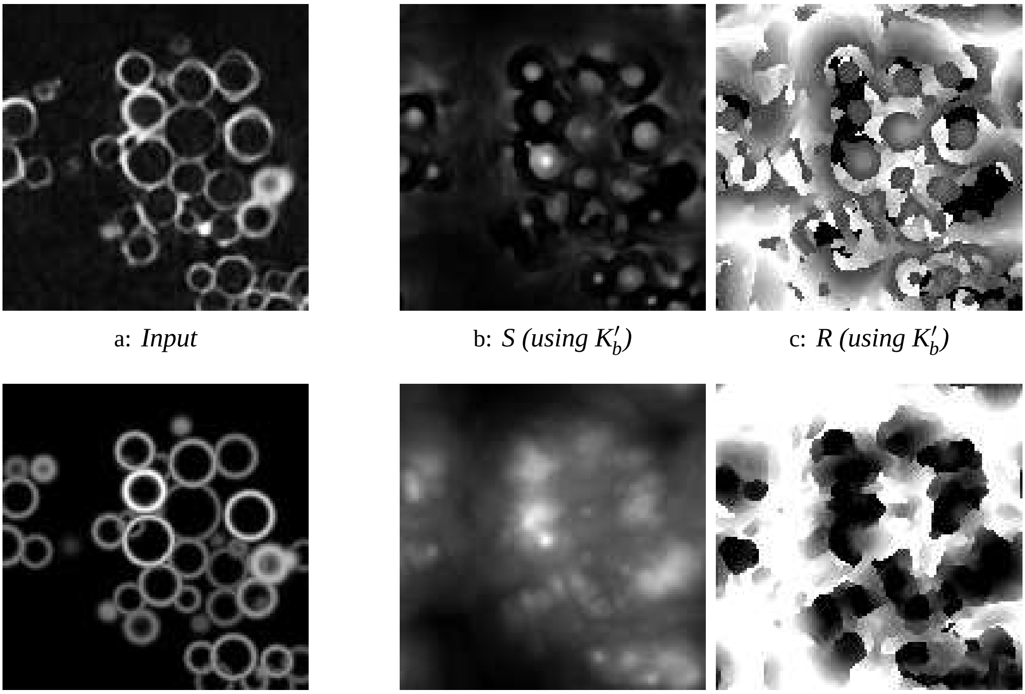


Figure 8.8: *Error made when estimating the radius of spheres in synthetic 3D images, using a negative inner sphere in the convolution kernel. Compare to the results for the corrected kernel in the left graph of Figure 8.5, which is computed on the same images. The underestimation here is about* 0.1 *pixel for large spheres.*



a: *Input*

b: *S (using K_b')*

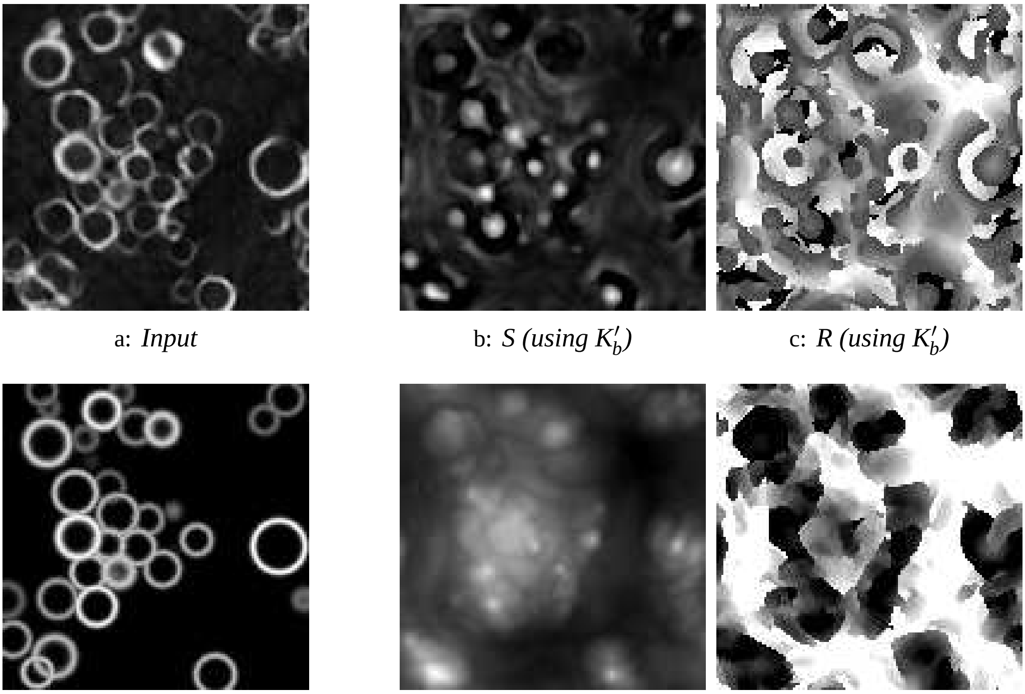
c: *R (using K_b')*

d: *Output*

e: *S (using K_b)*

f: *R (using K_b)*

Figure 8.9: *One slice of the 3D ballotini image and the corresponding slices of the results of the Radon transform. a: Slice of the input image. b: $S(\vec{x}_o)$ and c: $R(\vec{x}_o)$ obtained using the modified $K_b'(r,\vec{x}_o;\sigma_K)$ from (8.33). d: Image generated with the found parameters. e: $S(\vec{x}_o)$ and f: $R(\vec{x}_o)$ obtained using the regular $K_b(r,\vec{x}_o;\sigma_K)$ from (8.30).*



a: *Input*

b: *S (using K_b')*

c: *R (using K_b')*

d: *Output*

e: *S (using K_b)*

f: *R (using K_b)*

Figure 8.10: *One slice of the 3D ballotini image and the corresponding slices of the results of the Radon transform. a: Slice of the input image. b: $S(\vec{x}_o)$ and c: $R(\vec{x}_o)$ obtained using the modified $K_b'(r,\vec{x}_o;\sigma_K)$ from (8.33). d: Image generated with the found parameters. e: $S(\vec{x}_o)$ and f: $R(\vec{x}_o)$ obtained using the regular $K_b(r,\vec{x}_o;\sigma_K)$ from (8.30).*

A conclusion is the place where you got tired of thinking
(Arthur Block)

Chapter 9
Conclusions

Throughout this thesis we have seen how the granulometry can be applied to accurately estimate the (volume-weighted) size distribution of objects in an image. Such a size distribution can be used, for example, to characterize the structure being imaged. Several aspects are examined (Section 9.1), including the creation of the structuring elements used, the selection of the morphological operation, the preparation of the image (in the form of pre-processing), and the conversion of the granulometric curve to a cumulative size distribution. When examining these aspects, we focus on the continuous image being represented by the samples in the discrete image. The reason for this is that we are interested in measuring a property of real-world objects, which have been imaged and sampled for computer processing. Applying discrete morphological operations on such an image yields measurements related in part to the coincidental sampling positions with respect to the objects. That is, when applying morphology to measure object properties, different results are obtained each time the same objects are imaged. One important aspect of this thesis is substantially reducing these discretization errors. The results on this aspect are summarized in Section 9.2.

Section 9.3 presents the conclusions from Chapter 8 about the Radon transform. This transform detects parameterized shapes in an image, and can therefore also be used to construct a size distribution. The most important difference between the Radon transform and the granulometry is that the former is linear, whereas the latter is strongly non-linear. Both methods do not require any form of segmentation, although they can benefit from pre-processing.

9.1 The Granulometry

A granulometry is the projection of a morphological scale-space on the scale axis. The morphological scale-space is built with a sieve: an operation that is extensive (or anti-extensive), increasing and absorbing. The structural opening and the closing always satisfy the first two of these properties, but not always the third. The structuring element used must be chosen carefully. In the continuous domain, structuring elements such as the disk do satisfy the absorption property.

The closing sieve is defined by

$$F(x,s) = [\phi_{B(x,s)}(f)](x) \quad , \tag{9.1}$$

with s the scale-axis of the scale-space. The granulometry is the projection of F onto this axis.

Once the granulometry has been computed, a volume-weighted, cumulative size distribution follows by normalization:

$$H(r) = \frac{\int F(x,r)dx - \int F(x,0)dx}{\int F(x,\infty)dx - \int F(x,0)dx} \quad . \tag{9.2}$$

We propose logarithmic sampling of the scale-axis, so that the relative error of the discrete operation remains constant across all scales, and improper statistics are avoided when estimating the size distribution (usually larger objects are fewer in number than small ones).

Selecting a Suitable Morphological Operation

As stated above, the structural opening or closing with a disk as structuring element can be used in the granulometry. A disk (or a sphere in three dimensions) is an isotropic structuring element and as such has the additional, desirable property of rotation invariance. The granulometry that uses it measures the width of objects.

But there are other morphological operations that satisfy the sieving properties. Among these are:

- RIA opening and closing, to measure object length. These operations complement the isotropic structuring element that measures the smallest object diameter. The RIA operations can measure all other diameters.
- Rank-max opening and rank-min closing, which ignore a certain amount of pixels. Both isotropic and RIA morphology can be applied in this way to measure, for example, objects formed by agglomeration of smaller particles.
- Opening and closing by reconstruction, only to be used on objects easily segmented. Touching objects will be considered a single object by these filters.

- Attribute openings and closings, also only to be used on objects easily segmented. Instead of measuring the largest or smallest diameter, other attributes such as area can be measured.
- Floodings, that would produce a granulometry not related to a size distribution, but potentially useful in characterizing a structure.

The Pre-Processing

Another important aspect of analyzing an image is preparing it in such a way that the chosen method performs best. This is often called pre-processing, and involves noise removal, contrast enhancement, artifact removal, etc.

One of the most important effects noise has on the granulometry is flattening it. It adds volume to both the small and large scales, thereby making the curve less steep. We have seen that a combination of low-pass filtering (either by linear filters such as the Gaussian filter or by non-linear filters such as the median, Kuwahara or open-close filters) and soft-clipping provide the best pre-processing of an image for accurate size distribution estimation.

Since the granulometry computes a volume-weighted distribution based on the gray-value volume of the objects in the image, it is important to give all objects a uniform gray-value over their extent, and to give all objects in the image the same gray-value. If this does not happen, some objects will have a larger influence on the estimated distribution. We propose to use soft-clipping to equalize the gray-values of both the foreground and the background.

Finally, it is important to know that, if the acquisition can be improved, it is worth doing so. Any pre-processing needed to make the images fit for analysis also yields unwanted side-effects. These should be avoided, if possible. For example, as seen in Section 6.2, none of the noise reduction filters are perfect. They disturb the image in some way, and are unable to remove all the effects of the noise.

9.2 Sampling and Morphology

As stated before, we are interested in measuring a property of real-world objects, which have been imaged and sampled for computer processing. Therefore, it is important that the operations used for measuring are independent of the sampling. A major part of this thesis deals with the actual algorithms for the opening and closing, which lack the sampling-invariance property. By applying some algorithmic changes (as described below) to these filters, we were able to dramatically reduce the discretization errors, thereby improving the discrete approximation to the continuous filters. We examined two structuring element shapes: disks (or balls in 3D) and line segments.

Interpolation in Morphological Operations

Increasing the density of the sampling grid directly translates to a reduced discretization error. Assuming band-limited and correctly sampled input images, interpolation introduces no error (except close to the image border). However, because the structuring element itself also is sampled more densely, the relative error of the operation decreases. For the granulometry, where only the projection on the scale-axis is required, no sub-sampling needs to be done after the operation. If sub-sampling were required, the improvement would be less significant.

For the line segment as structuring element, interpolation can be used in another manner: by obtaining the exact function values along the line instead of using the nearest samples. We have shown how such a strategy diminishes the errors due to sampling in the opening with a line segment.

Shifting the Disk

We also found that by slightly shifting the disk with respect to the sampling grid, the area of the binary structuring element as a function of the scale parameter becomes smoother and better resembles its ideal continuous counterpart (this can, of course, not be done for the erosion and dilation, since these are not invariant to translations of the structuring element). This effect can be explained by the loss of symmetry of the discretized structuring element. As a result, the granulometry can be sampled more densely. The optimal shift found was $(0.19, 0.31)$ in 2D and $(0.16, 0.24, 0.34)$ in 3D. Additionally, this modification reduces the errors due to the lack of rotation invariance. This improvement is equivalent to a 4-fold subsampling for smaller scales, but less noticeable at the larger scales because of the chosen logarithmic sampling.

Gray-Value Structuring Elements

Both the disk and the line segment can also be implemented using gray-value structuring elements by creating approximately band-limited shapes. We propose an error-function to simulate the edge of a band-limited object. This is equivalent to convolving the binary shape with a Gaussian kernel. One of the more difficult aspects of using such a structuring element is the scaling of its gray-values, which must be related to the gray-value range of the image. We use a factor of 1.0233 for this relation, making the gray-value range of the structuring element just a bit larger than that of the image. This factor is chosen to make the cut-off point of the structuring element to be 2σ pixels from the edge of the shape.

The gray-value structuring elements performed better than the original binary ones. For the disk, the performance in the rotation-invariance test was equivalent to subsampling 4 to 8 times, depending on the disk size. For the line segment, the differences are smaller since the proposed binary version also uses interpolation. However, the increase in computational cost might reduce the usefulness of the gray-value structuring elements. This is especially true for the line segment, which in its binary form can be computed with a one-dimensional operation, but as a gray-value structuring element has the same dimensionality as the image.

Sampling-Free Morphology

The only way of completely avoiding discretization errors in mathematical morphology is using an alternative image representation. For one-dimensional images we have proposed to use a piece-wise polynomial representation, based on spline interpolation. Because this is a continuous representation, discretization effects are no longer relevant. The structuring element is limited to flat shapes (defined by a domain). With a dedicated algorithm, dilations and erosions can be computed on this representation, yielding a new image in the same representation.

A granulometry with these operations was shown to have a smaller bias than the discrete operation. This bias is introduced by the (non-ideal) cubic spline interpolation, not the subsequent morphological operations. Additionally, the shape of the structuring element is not limited to an integer number of pixels.

Extending the algorithm to higher-dimensional images is very complicated, but should be possible. Extending the algorithm to gray-value images might also be possible by using the slope transform.

Sampling the Orientation Axis

For use in RIA morphology, we also require a finite number of orientations under which to compute the morphological operations. We found, as was expected, that this number is directly proportional to the line length. If this number is expressed as $\pi\ell/q$, a simple experiment showed that setting q between 0.1 and 0.2 will yield good results. Nonetheless, we often used a value of 1; this leads to a small underestimation of the measured length.

9.3 The Radon Transform

We have given the conditions under which the Radon transform can be computed free of discretization errors. Assuming that the input image is correctly sampled, these conditions are met by imposing a band-limit on the operator function. This has no consequences for sufficiently smooth shapes. The parameter space that results is band-limited, which allows sampling and interpolation, and thus sub-pixel accuracy in the estimated parameters.

To avoid a larger weight being assigned to larger shapes, the operator function should be normalized. We studied the effect of this normalization in the case of the Radon transform for spheres, and propose a way of correcting for the bias this introduces in the estimated radius.

The Radon transform reduces to a convolution for position-type parameters, yielding a large speed-up. We propose a memory-efficient algorithm, computing (through convolution) a single r-slice of $P(r,\vec{x}_o)$ at a time. We keep track of the maximum projection and the argument-maximum projection along the r-axis as we compute the slices. We argue that this approach can be used for other shapes as well.

We have applied this modified Radon transform to a 3D image of glass hollow beads. To compute its parameter space we have employed a convolution kernel that contains not only a sphere, but also a second, smaller, concentric sphere with negative gray-values. The resulting parameter space has a much higher discriminating ability than that which would result from the same transform with a single sphere.

For every complex problem, there is a solution that is simple, neat, and wrong.
(Henry L. Mencken)

Appendix A
Underestimation of the Radius in the Radon Transform for Circles and Spheres

In this technical report we compute the underestimation of the radius in the Radon transform for circles and spheres. Our implementation of the Radon transform uses spheres with a Gaussian profile, and normalizes the gray-value of each of the spheres so that a very large sphere matching only a couple of segments will not get a higher confidence (value of the peak in the parameter space) than a very small sphere completely matched in the image. This normalization causes an underestimation of the radius.[1]

A.1 Introduction

The Radon transform for an N-dimensional hyper-sphere is defined as

$$P(\vec{x},r) = \int C_b(\vec{p},r) I(\vec{x}-\vec{p}) d\vec{p} \quad , \tag{A.1}$$

with

$$\begin{aligned} C_b(\vec{x},r) &= \frac{1}{S_N(r)} \delta\big(\frac{1}{2}\sqrt{2}(\|\vec{x}\| - r)\big) * G(\vec{x};\sigma_p) \\ &= \frac{1}{S_N(r)} G(\|\vec{x}\| - r; \sigma_p) \end{aligned} \tag{A.2}$$

[1] This report is published as “Underestimation of the Radius in the Radon Transform for Circles and Spheres”, C.L. Luengo Hendriks, M. van Ginkel and L.J. van Vliet, PH Report number PH-2003-02, Pattern Recognition Group, Delft University of Technology, The Netherlands.

the convolution kernel that defines the shape of the sphere, and $I(\vec{x})$ the input image. In these equations, $S_N(r)$ is the surface area of the sphere of radius r. $G(\vec{x};\sigma)$ denotes the Gaussian function:

$$G(\vec{x};\sigma) = \frac{1}{\left(\sigma\sqrt{2\pi}\right)^N}\, e^{-\frac{1}{2}\left(\frac{\|\vec{x}\|}{\sigma}\right)^2} \quad . \tag{A.3}$$

Let us assume that the input image has a single sphere with a Gaussian profile centered at $\vec{x}=0$, having a radius of R and a Gaussian parameter σ_i:

$$I(\vec{x}) = G(\|\vec{x}\| - R;\sigma_i) \quad . \tag{A.4}$$

$P(0,r)$, has a shape given by the integral of the product of two Gaussian curves (that of $C_b(\vec{x},r)$ and that of $I(\vec{x})$),

$$\begin{aligned} P(0,r) &= \int C_b(\vec{p},r)\, I(0-\vec{p})\, d\vec{p} \\ &= \int \frac{1}{S_N(r)}\, G(\|\vec{p}\| - r;\sigma_p)\, G(\|\vec{p}\| - R;\sigma_i)\, d\vec{p} \\ &= S_N(1)\int_0^\infty \frac{1}{S_N(r)}\, G(\rho - r;\sigma_p)\, G(\rho - R;\sigma_i)\, d\rho \quad . \end{aligned}$$

The location of the maximum along r is then used as an estimate of the radius of the sphere in the input image.

Using the substitutions $\sigma_s^2 = \sigma_p^2 + \sigma_i^2$, $\frac{1}{\sigma_e^2} = \frac{1}{\sigma_p^2} + \frac{1}{\sigma_i^2}$ and $s = \sigma_e^2\left(\frac{r}{\sigma_p^2} + \frac{R}{\sigma_i^2}\right)$, The product of the Gaussians can be re-written as

$$\begin{aligned} & G(\rho - r;\sigma_p)\, G(\rho - R;\sigma_i) \\ &= \frac{1}{\sqrt{2\pi}\sigma_p} \exp\left(-\frac{1}{2}\frac{(\rho-r)^2}{\sigma_p^2}\right) \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2}\frac{(\rho-R)^2}{\sigma_i^2}\right) \\ &= \frac{1}{2\pi\sigma_p\sigma_i} \exp\left(-\frac{1}{2}\left[\frac{\rho^2 - 2\rho r + r^2}{\sigma_p^2} + \frac{\rho^2 - 2\rho R + R^2}{\sigma_i^2}\right]\right) \\ &= \frac{1}{2\pi\sigma_p\sigma_i} \exp\left(-\frac{1}{2\sigma_e^2}\left[\rho^2 - 2\rho\sigma_e^2\left(\frac{r}{\sigma_p^2} + \frac{R}{\sigma_i^2}\right)\right] - \frac{1}{2}\left(\frac{r^2}{\sigma_p^2} + \frac{R^2}{\sigma_i^2}\right)\right) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2\pi\sigma_p\sigma_i} \exp\left(-\frac{1}{2\sigma_e^2} \left[\rho - \sigma_e^2 \left(\frac{r}{\sigma_p^2} + \frac{R}{\sigma_i^2} \right) \right]^2 - \frac{(r-R)^2}{2\sigma_s^2} \right) \\
&= \frac{\sigma_e\sigma_s}{\sigma_p\sigma_i}\, G(\rho - s; \sigma_e)\, G(r - R; \sigma_s) \\
&= G(\rho - s; \sigma_e)\, G(r - R; \sigma_s) \quad .
\end{aligned}
$$

Using this, one obtains

$$
\begin{aligned}
P(0,r) &= \frac{S_N(1)}{S_N(r)}\, G(r-R;\sigma_s) \int_0^\infty \rho^{N-1} G(\rho - s; \sigma_e)\, d\rho \\
&= \frac{1}{r^{N-1}}\, G(r-R;\sigma_s) \int_{-s}^\infty (x+s)^{N-1} G(x;\sigma_e)\, dx \\
&= \frac{1}{r^{N-1}}\, G(r-R;\sigma_s) \int_{-s}^\infty \sum_{k=0}^{N-1} \binom{N-1}{k} x^k s^{N-k-1} G(x;\sigma_e)\, dx \\
&= \frac{1}{r^{N-1}}\, G(r-R;\sigma_s) \sum_{k=0}^{N-1} \binom{N-1}{k} s^{N-k-1} \int_{-s}^\infty x^k G(x;\sigma_e)\, dx \quad .
\end{aligned}
\tag{A.5}
$$

The integral can be split

$$
\begin{aligned}
\int_{-s}^\infty x^k G(x;\sigma_e)\, dx &= \int_0^\infty x^k G(x;\sigma_e)\, dx + \int_0^{-s} x^k G(x;\sigma_e)\, dx \\
&= 2^{\frac{k-1}{2}} \sigma_e^{k+1} \left[\Gamma\left(\frac{k+1}{2}\right) + \gamma\left(\frac{k+1}{2}, \frac{s^2}{2\sigma_e^2}\right) \right] \quad ,
\end{aligned}
$$

but this does not lead us anywhere. The second term, the lower incomplete Gamma function, depends on r (through s) but cannot be solved for r analytically. Therefore, we cannot determine the location of the maximum along r without choosing a dimensionality.

A.2 The 2D case

$$
P(0,r) = \frac{1}{r}\, G(r-R;\sigma_s) \int_{-s}^\infty (x+s) G(x;\sigma_e)\, dx \quad . \tag{A.6}
$$

The solution to the integral is given by

$$\begin{aligned} & s\int_{-s}^{\infty} G(x;\sigma_e)\,dx + \int_{-s}^{\infty} xG(x;\sigma_e)\,dx \\ &= s\left[\int_{0}^{\infty} G(x;\sigma_e)\,dx - \int_{0}^{-s} G(x;\sigma_e)\,dx\right] + \int_{-s}^{\infty} xG(x;\sigma_e)\,dx \\ &= s\left[1-\mathrm{erf}(\frac{-s}{\sqrt{2}\sigma_e})\right] + \sigma_e^2\,G(-s;\sigma_e) \quad . \end{aligned}$$

Assuming $R > 3\sigma_i$ or so (if this is not true, the input image will not have recognizable circles), and knowing that the position of the peak is close to its expected location, $r \approx R$, it can be assumed that $R > 3\sigma_e$. This means that the error function takes a value of approximately -1, and the Gaussian of 0.

$$P(0,r) \approx \frac{2s}{r}\,G(r-R;\sigma_s) = 2\sigma_e^2\left(\frac{1}{\sigma_p^2}+\frac{R}{\sigma_i^2 r}\right) G(r-R;\sigma_s) \quad .$$

To find the position of the maximum, we equate the derivative to zero.

$$\frac{dP}{dr} \approx 2\sigma_e^2\left[\frac{-R}{\sigma_i^2 r^2} + \left(\frac{1}{\sigma_p^2}+\frac{R}{\sigma_i^2 r}\right)\frac{R-r}{\sigma_s^2}\right] G(r-R;\sigma_s) = 0 \quad .$$

Solving for $x = r-R$ yields

$$\begin{aligned} & 0 = \frac{-R}{\sigma_i^2 r^2} + \frac{R-r}{\sigma_p^2\sigma_s^2} + \frac{R^2-Rr}{\sigma_i^2\sigma_s^2 r} \\ \Longrightarrow\quad & 0 = \sigma_s^2\sigma_p^2 R - \sigma_i^2 R r^2 + \sigma_i^2 r^3 - \sigma_p^2 R^2 r + \sigma_p^2 R r^2 \\ & \;\; = \sigma_i^2 x^3 + (2\sigma_i^2+\sigma_p^2)Rx^2 + \sigma_s^2 R^2 x + \sigma_s^2\sigma_p^2 R \quad , \end{aligned}$$

a third-degree equation. Again assuming $R > \sigma_i$ and x is close to 0,

$$0 \approx (2\sigma_i^2+\sigma_p^2)Rx^2 + \sigma_s^2 R^2 x + \sigma_s^2\sigma_p^2 R$$

has a simpler solution. We select the root closest to 0.

$$x \approx \frac{-\sigma_s^2 R}{2(2\sigma_i^2+\sigma_p^2)} + \sqrt{\frac{\sigma_s^4 R^2}{4(2\sigma_i^2+\sigma_p^2)^2} - \frac{\sigma_s^2\sigma_p^2}{(2\sigma_i^2+\sigma_p^2)}} \quad . \tag{A.7}$$

A fifth-order approximation around $1/R = 0$ of this is given by

$$\begin{aligned} x &\approx -\frac{\sigma_p^2}{R} - \frac{\sigma_p^4}{R^3}\frac{\sigma_p^2+2\sigma_i^2}{\sigma_s^2} + O\left(R^{-5}\right) \\ &= -\frac{\sigma_p^2}{R} - \frac{\sigma_p^2}{R^3}\left(\sigma_p^2+\sigma_e^2\right) + O\left(R^{-5}\right) \quad . \end{aligned} \tag{A.8}$$

Note that both σ_s and σ_e depend on the input image, but the first term (which is good for an approximation of the third order), depends only on σ_p, defined by the algorithm.

A.3 The 3D case

$$P(0,r) = \frac{1}{r^2}\, G(r-R;\sigma_s) \int_{-s}^{\infty} (x+s)^2 G(x;\sigma_e)\, dx \quad . \tag{A.9}$$

The solution to the integral is given by

$$\begin{aligned} &s^2 \int_{-s}^{\infty} G(x;\sigma_e)\, dx + 2s \int_{-s}^{\infty} xG(x;\sigma_e)\, dx + \int_{-s}^{\infty} x^2 G(x;\sigma_e)\, dx \\ &\quad \{\text{partial integration of last term}\} \\ &= s^2 \int_{-s}^{\infty} G(x;\sigma_e)\, dx + 2s \int_{-s}^{\infty} xG(x;\sigma_e)\, dx - \sigma_e^2 xG(x;\sigma_e)\Big|_{-s}^{\infty} + \sigma_e^2 \int_{-s}^{\infty} G(x;\sigma_e)\, dx \\ &= (s^2+\sigma_e^2)\left[\int_0^{\infty} G(x;\sigma_e)\, dx - \int_0^{-s} G(x;\sigma_e)\, dx\right] \\ &\qquad\qquad + 2s \int_{-s}^{\infty} xG(x;\sigma_e)\, dx - \sigma_e^2 xG(x;\sigma_e)\Big|_{-s}^{\infty} \\ &= (s^2+\sigma_e^2)\left[1 - \operatorname{erf}(\frac{-s}{\sqrt{2}\sigma_e})\right] + \sigma_e^2 s\, G(-s;\sigma_e) \quad . \end{aligned}$$

Again the error function takes a value of approximately -1, and the Gaussian of 0.

$$\begin{aligned} P(0,r) &\approx \frac{2}{r^2}\left(s^2+\sigma_e^2\right)\, G(r-R;\sigma_s) \\ &= \frac{2}{r^2}\left(\sigma_e^4\left(\frac{r}{\sigma_p^2}+\frac{R}{\sigma_i^2}\right)^2 + \sigma_e^2\right) G(r-R;\sigma_s) \\ &= \frac{2}{\sigma_s^4}\frac{\left(\sigma_i^2 r + \sigma_p^2 R\right)^2 + \sigma_i^2\sigma_p^2\sigma_s^2}{r^2}\, G(r-R;\sigma_s) \quad . \end{aligned}$$

To find the position of the maximum, we equate the derivative to zero.

$$\begin{aligned}\frac{dP}{dr} \approx \frac{2}{\sigma_s^4} \Bigg[& \frac{2\left(\sigma_i^2 r + \sigma_p^2 R\right)\sigma_i^2}{r^2} \\ & + \frac{\left(\sigma_i^2 r + \sigma_p^2 R\right)^2 + \sigma_i^2 \sigma_p^2 \sigma_s^2}{r^2} \left(\frac{R-r}{\sigma_s^2} - \frac{2}{r}\right)\Bigg] G(r-R;\sigma_s) = 0 \quad .\end{aligned}$$

Solving for $x = r - R$ yields

$$\begin{aligned} & 0 = \frac{2\left(\sigma_i^2 r + \sigma_p^2 R\right)\sigma_i^2}{r^2} + \frac{\left(\sigma_i^2 r + \sigma_p^2 R\right)^2 + \sigma_i^2 \sigma_p^2 \sigma_s^2}{r^2} \left(\frac{R-r}{\sigma_s^2} - \frac{2}{r}\right) \\ \implies \quad & 0 = 2\left(\sigma_i^2 r + \sigma_p^2 R\right)\sigma_i^2 \sigma_s^2 r + \left(\left(\sigma_i^2 r + \sigma_p^2 R\right)^2 + \sigma_i^2 \sigma_p^2 \sigma_s^2\right)\left(Rr - r^2 - 2\sigma_s^2\right) \\ & \quad = \sigma_i^4 x^4 + (\sigma_i^2 + 2\sigma_s^2)\sigma_i^2 R x^3 + \left((2\sigma_i^2 + \sigma_s^2)R^2 + \sigma_p^2 \sigma_i^2\right)\sigma_s^2 x^2 \\ & \qquad + \left(\sigma_s^2 R^2 + 3\sigma_p^2 \sigma_i^2\right)\sigma_s^2 R x + 2(R^2 + \sigma_i^2)\sigma_p^2 \sigma_s^4 \quad .\end{aligned}$$

a fourth-degree equation. Again assuming $R > \sigma_i$ and x is close to 0,

$$0 \approx \left((2\sigma_i^2 + \sigma_s^2)R^2 + \sigma_p^2 \sigma_i^2\right)x^2 + \left(\sigma_s^2 R^2 + 3\sigma_p^2 \sigma_i^2\right)Rx + 2(R^2 + \sigma_i^2)\sigma_p^2 \sigma_s^2$$

has a simpler solution. We select the root closest to 0.

$$\begin{aligned} x \approx & \frac{-\left(\sigma_s^2 R^2 + 3\sigma_p^2 \sigma_i^2\right)R}{2(2\sigma_i^2 + \sigma_s^2)R^2 + 2\sigma_p^2 \sigma_i^2} \\ & + \sqrt{\frac{\left(\sigma_s^2 R^2 + 3\sigma_p^2 \sigma_i^2\right)^2 R^2}{\left(2(2\sigma_i^2 + \sigma_s^2)R^2 + 2\sigma_p^2 \sigma_i^2\right)^2} - \frac{2(R^2 + \sigma_i^2)\sigma_p^2 \sigma_s^2}{(2\sigma_i^2 + \sigma_s^2)R^2 + \sigma_p^2 \sigma_i^2}} \quad .\end{aligned}$$

A fifth-order approximation around $1/R = 0$ of this is given by

$$\begin{aligned} x & \approx -\frac{2\sigma_p^2}{R} - \frac{2\sigma_p^2}{R^3} \frac{2\sigma_p^4 + 4\sigma_p^2 \sigma_i^2 + \sigma_i^4}{\sigma_s^2} + O\left(R^{-5}\right) \\ & = -\frac{2\sigma_p^2}{R} - \frac{2\sigma_p^2}{R^3}\left(\sigma_p^2 + \sigma_s^2 + \sigma_e^2\right) + O\left(R^{-5}\right) \quad . \end{aligned} \tag{A.10}$$

As in the 2D case, the first term (which is good for an approximation of the third order), depends only on σ_p, defined by the algorithm.

A.4 Kernel Normalization

To show that the most important part of the under-estimation of the radius is caused by the normalization, we compute the position of the maximum along the r-axis of $C_b(\vec{x},r)$. We expect this maximum to be close to $\|\vec{x}\| = R$. We recall the definition of $C_b(\vec{x},r)$,

$$C_b(\vec{x},r) = \frac{1}{S_N(r)} G(\|\vec{x}\| - r; \sigma_p) = \frac{K}{r^{N-1}} G(R-r;\sigma_p) \quad ,$$

with K some constant that depends on the dimensionality N. The derivative along r is given by

$$\frac{dC_b}{dr} = K \left[\frac{-(N-1)}{r^N} + \frac{R-r}{\sigma_p^2\, r^{N-1}} \right] G(R-r;\sigma_p) \quad .$$

Equating it to zero and solving for $x = r - R$ yields

$$\begin{aligned} 0 = -\frac{N-1}{r^N} + \frac{R-r}{\sigma_p^2\, r^{N-1}} &= -(N-1)\sigma_p^2 + (R-r)r \\ &= -(N-1)\sigma_p^2 - x(x+R) \quad , \end{aligned}$$

$$x \approx -\frac{(N-1)\sigma_p^2}{R} - \frac{(N-1)^2\sigma_p^4}{R^3} + O\left(R^{-5}\right) \quad . \tag{A.11}$$

This explains the first term of equations (A.8) and (A.10), as well as a portion of the second term. The rest of those equations is due to the asymmetry of the peak resulting from the convolution. This asymmetry is caused by the curvature of the two interacting shapes.

To correct for this bias, we need to draw the kernel with an alternative radius R'. This should be selected such that the maximum in the r-direction lies exactly at r. For larger R this is:

$$\begin{aligned} & r = R' + \frac{(N-1)\sigma_p^2}{R'} \\ \Longrightarrow \quad & 0 = R'^2 - rR' + (N-1)\sigma_p^2 \\ \Longrightarrow \quad & R' = \frac{1}{2}r + \sqrt{\frac{1}{4}r^2 - (N-1)\sigma_p^2} \quad . \end{aligned} \tag{A.12}$$

If you want anything done well, do it yourself;
this is why most people laugh at their own jokes
(Bob Edwards)

Acknowledgement

This thesis would have been much less interesting[2] without the contribution of many people. Most of all, my Ph.D. advisor and mentor, Lucas van Vliet, always full of ideas, always ready to help. He not only taught me how to do research, he also taught me how to write it all down. Needless to say, the people at Unilever Vlaardingen contributed much, especially Han Blonk, Geert van Kempen and Gerard van Dalen. I want to thank them and everybody else there for the very interesting projects, the huge amount of data, and the warmth with which they have received and helped me. I also want to thank Scott Singleton and Dave Rowlands at Unilever Colworth House for a very productive and interesting time in Bedford. My profound appreciation to Ted Young for being a great source of inspiration. I might never have found my way to this group if he had not been such a marvellous teacher.

Of course the colleagues with which I spent such a large part of my days had much influence on this thesis. I have conducted very fruitful discussions with Frank Faas, Michael van Ginkel, Pavel Paclík, Dick de Ridder, Bernd Rieger and David Tax.[3] Many of the ideas flowing from these discussions ended up in my research. I just hope they profited as well. I very much appreciate all the senior staff members, Bob Duin, Pieter Jonker, Piet Verbeek, Lucas van Vliet, Albert Vossepoel and Ted Young for being willing and able to impart their thorough knowledge. Ronald Lichteringen, Wouter Smaal, Klara Schaafsma and Wim van Oel, whom we call “support staff”, have done much more than just support. I am also grateful to everybody in PHAIO for the all beers, barbecues, xblast and FPS sessions, chess and “Colonisten” matches, etc., that are so important for the creative process.

[2] *Even* less interesting according to Frank...

[3] I hope I am not forgetting anybody here! During these five years there were many Ph.D. students in this group. I have exchanged ideas with all of them.

I would probably not have ended up where I am now without some of the people that have been in my life: As undergraduate students, Sander Nievaart, Haipo Chan and I studied together. Sander was the one bringing discipline and making me work. My uncle Peter Dahmen, a TU Delft alumnus, helped me in my choice to study physics at Delft. Francisco Guijarro, my high-school math teacher, awoke my interest in mathematics with his infectious enthusiasm for the subject. Thanks to all of them.

Finally I would like to thank my parents. They gave me all the opportunities to do what I wanted to do, and bought me lots of Lego Technic. I am very grateful to them.

I don't want to achieve immortality through my work,
I want to achieve immortality through not dying
(Woody Allen)

Curriculum Vitae

Cris Luengo was born in Leiden, the Netherlands, August 7, 1974. Shortly after his 8th birthday, he moved with his parents to Alfaz del Pi, in Alicante, Spain, and a couple of years later to neighboring Altea, where his parents still live. In 1992 he graduated from the Colegio Cooperativa Almedia, in Callosa d'En Sarrià, and decided to study Applied Physics in Delft. In November 1998 he received his M.Sc. degree after finishing a research project entitled "Improved Resolution in Infrared Imaging Using randomly Shifted Images", at the Pattern Recognition Group, under guidance of Dr. Lucas J. van Vliet.

He stayed in that group to work on his Ph.D. research project, under supervision of this M.Sc. advisor. The project was financed by the Ministry of Economic Affairs under the IOP program, and started officially on January 1st, 1999. Unilever Research Laboratories in Vlaardingen was the industrial partner in this project, and did not stop providing him with problems and data to work on. Han Blonk, Geert van Kempen and Gerard van Dalen at Unilever were involved in this project.

During the four years of his contract, Cris did quite a bit of research, not all of which is presented in this thesis, as well as some other things:

- He worked on the DIP*lib* project, wrote an interface to MATLAB, and created DIP*image*, a MATLAB toolbox. DIP*image* is now used by many research groups and educational institutions throughout the world (see http://www.ph.tn.tudelft.nl/DIPlib).
- From 2001 through 2003 he worked for the Computer Service Labs (CSP), which provides self-study courses to M.Sc. students. One of his tasks was to create a hands-on image processing course (tn3531-p).
- He participated in setting up and teaching the laboratory parts of courses like CBP (in 1999, 2000, 2001 and 2003, image processing for industrial applications) and EMBO (in 2002, a microscopy and image analysis course for European Ph.D. students in the field of micro-biology).

All of these things took so much time, that his contract was extended by a full year.

For 10 weeks in the summer of 2001 (August, September and October), he worked at Unilever Research Colworth House (Sharnbrook, Bedfordshire, UK), under the supervision of Scott Singleton, and together with Dave Rowlands, on analysis of three-dimensional images obtained by micro-CT. He did research on the analysis of tea leaves, ice crystals, ballotini, and what not, and on improving the images obtained by the (house-built) micro-CT machine. This period resulted in an internal report and (eventually) two papers.

He will be joining the Life Sciences Division at the Lawrence Berkeley National Laboratory (Berkeley, California) in February 2004.

Journal Papers

Improving Morphological Sieves to Detect Minute Differences in Structures, C.L. Luengo Hendriks, L.J. van Vliet, G.M.P. van Kempen and E.C.M. Bouwens, *submitted to* IEEE *Transactions on Image Processing*.

Time Evolution of Phase Separating Milk Protein and Amylopectin Mixtures, P.W. de Bont, G.M.P. van Kempen, C.L. Luengo Hendriks and R. Vreeker, *submitted to Food Hydrocolloids*.

The Generalized Radon Transform: Sampling, Accuracy and Memory Considerations, C.L. Luengo Hendriks, M. van Ginkel, P.W. Verbeek and L.J. van Vliet, *in preparation*.

Discrete Morphology with Line Structuring Elements, C.L. Luengo Hendriks and L.J. van Vliet, *in preparation*.

Filtering Ring Artifacts in X-ray Computed Tomography, C.L. Luengo Hendriks and L.J. van Vliet, *in preparation*.

Conference Proceedings

The Generalized Radon Transform: Sampling and Memory Considerations, C.L. Luengo Hendriks, M. van Ginkel, P.W. Verbeek and L.J. van Vliet, in *Computer Analysis of Images and Patterns* (*Proceedings of the Tenth International Conference,* CAIP *2003, Groningen,* NL; N. Petkov and M.A. Westenberg, eds.), vol. 2756 of LNCS, pp. 681–688, Springer-Verlag, Berlin, 2003.

Discrete Morphology with Line Structuring Elements, C.L. Luengo Hendriks and L.J. van Vliet, in *Computer Analysis of Images and Patterns* (*Proceedings of the Tenth International Conference,* CAIP *2003, Groningen,* NL; N. Petkov and M.A. Westenberg, eds.), vol. 2756 of LNCS, pp. 722–729, Springer-Verlag, Berlin, 2003.

Basic Morphological Operations, Band-Limited Images and Sampling, C.L. Luengo Hendriks and L.J. van Vliet, in *Scale Space Methods in Computer Vision* (*Proceedings of the Fourth International Conference, Scale-Space 2003, Isle of Skye,* UK; L.D. Griffin and M. Lillholm, eds.), vol. 2695 of LNCS, pp. 313–324, Springer-Verlag, Berlin, 2003.

Segmentation-Free Estimation of Length Distributions Using Sieves and RIA *Morphology*, C.L. Luengo Hendriks and L.J. van Vliet, in *Proceedings of the Third International Conference on Scale-Space, Vancouver, Canada* (Kerckhove, M., ed.), vol. 2106 of LNCS, pp. 389–406, Springer-Verlag, Berlin, 2001.

A Rotation-Invariant Morphology for Shape Analysis of Anisotropic Objects and Structures, C.L. Luengo Hendriks and L.J. van Vliet, in *Proceedings of the Fourth International Workshop on Visual Form, Capri, Italy* (C. Arcelli, L.P. Cordella and G. Sanniti di Baja, eds.), vol. 2059 of LNCS, pp. 378–387, Springer-Verlag, Berlin, 2001.

Morphological Scale-Space to Differentiate Microstructures of Food Products, C.L. Luengo Hendriks and L.J. van Vliet, in *Proceedings of the 6th Annual Conference of the Advanced School for Computing and Imaging, Lommel, Belgium* (L.J. van Vliet, J.W.J. Heijnsdijk, T. Kielmann and P.M.W. Knijnenburg, eds.), pp. 289–293, 2000.

Improving Resolution to Reduce Aliasing in an Undersampled Image Sequence, C.L. Luengo Hendriks and L.J. van Vliet, in *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications* (M.M. Blouke, G.M. Williams, N. Sampat and T. Yeh, eds.), vol. 3965 of *Proceedings of* SPIE, pp. 214–222, San Jose, California, 2000.

Improving Spatial Resolution in Exchange of Temporal Resolution in Aliased Image Sequences, L.J. van Vliet and C.L. Luengo Hendriks, in *Proceedings of the 11th Scandinavian Conference on Image Analysis, Kangerlussuaq, Greenland* (B.K. Ersboll and P. Johansen, eds.), vol. 1, pp. 493–499, Pattern Recognition Society of Denmark, Lyngby, 1999.

Technical Reports

Underestimation of the Radius in the Radon Transform for Circles and Spheres, C.L. Luengo Hendriks, M. van Ginkel and L.J. van Vliet, Technical Report PH-2003-02, Pattern Recognition Group, Faculty of Applied Sciences, Delft University of Technology, 2003.

3D Structure Quantification of Tea Leaf and Ice Cream by Image Analysis and X-Ray Micro-Computed Tomography, C.L. Luengo Hendriks, D. Rowlands and S. Singleton, Technical Report, Unilever Research Colworth, Bedford, United Kingdom, 2001.

The Use of Morphological Sieves to Differentiate Microstructures of Food Products, C.L. Luengo Hendriks, G.M.P. van Kempen, L.J. van Vliet, S. Singleton and J.C.G Blonk Technical Report, Unilever Research Vlaardingen, Vlaardingen, The Netherlands, 2000.

Improved Resolution in Infrared Imaging Using Randomly Shifted Images, C.L. Luengo Hendriks, L.J. van Vliet and K. Schutte, Technical Report FEL-98-S315, TNO Physics and Electronics Laboratory, The Hague, The Netherlands, 1998.

References

[1] Acton, S.T., "Fast Algorithms for Area Morphology," *Digital Signal Processing*, vol. 11, pp. 187–203, 2001. Referred to on pg. 37

[2] Adams, R., "Radial Decomposition of Discs and Spheres," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 5, pp. 325–332, 1993. Referred to on pg. 89

[3] den Adel, R. and Blonk, J.C.G., *Vertical Pt/C Replication; a Visualisation Technique for Macromolecules*, Tech. Rep. VD 01 0304, Unilever Research Vlaardingen, Vlaardingen, 2001. Referred to on pg. 109

[4] Aguado, A.S., Montiel, E., and Nixon, M.S., "Bias Error Analysis of the Generalised Hough Transform," *Journal of Mathematical Imaging and Vision*, vol. 12, no. 1, pp. 25–42, 2000. Referred to on pg. 147

[5] Aguado, A.S., Montiel, M.E., and Nixon, M.S., "Ellipse Detection via Gradient Direction in the Hough Transform," in *Proceedings of the Fifth IEE International Conference on Image Processing and its Applications*, pp. 375–378, Edinburgh, 1995. Referred to on pg. 148

[6] Aitchison, J. and Brown, J.A.C., *The Lognormal Distribution*, Cambridge University Press, London, 1957. Referred to on pg. 63

[7] Alvarez, L. and Morel, J.M., "Morphological Approach to Multiscale Analysis: From Principles to Equations," in *Geometry-Driven Diffusion in Computer Vision* (ter Haar Romeny, B.M., ed.), pp. 229–254, Kluwer Academic Publishers, Dordrecht, 1994. Referred to on pg. 57

[8] Asano, A., Ohkubo, T., Muneyasu, M., and Hinamoto, T., "Primitive and Point Configuration Texture Model and Primitive Estimation Using Mathematical Morphology," in *SCIA 2003, 13th Scandinavian Conference on Image Analysis* (Bigun,

J. and Gustavsson, T., eds.), vol. 2749 of *Lecture Notes on Computer Science*, pp. 178–185, Springer-Verlag, Berlin, 2003. Referred to on pg. 57

[9] Atherton, T.J. and Kerbyson, D.J., "Using Phase to Represent Radius in the Coherent Circle Hough Transform," IEE Colloquiem on Hough Transforms, 1993. Referred to on pg. 156

[10] Bakker, P., *Image Structure Analysis for Seismic Interpretation*, Ph.D. thesis, Pattern Recognition Group, Faculty of Applied Sciences, Delft University of Technology, Delft, 2002. Referred to on pg. 41

[11] Ballard, D.H., "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981. Referred to on pg. 148

[12] Ballard, D.H. and Sabbah, D., "Viewer Independent Shape Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 6, pp. 653–660, 1983. Referred to on pg. 156

[13] Bangham, J.A., Campbell, T.G., and Aldridge, R.V., "Multiscale Median and Morphological Filters for 2D Pattern Recognition," *Signal Processing*, vol. 38, no. 3, pp. 387–415, 1994. Referred to on pg. 57

[14] Bangham, J.A., Chardaire, P., Pye, C.J., and Ling, P.D., "Multiscale Nonlinear Decomposition: The Sieve Decomposition Theorem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 529–539, 1996. Referred to on pg. 57

[15] Bazin, M.J. and Benoit, J.W., "Off-line Global Approach to Pattern Recognition for Bubble Chamber Pictures," *IEEE Transactions on Nuclear Science*, vol. 12, pp. 291–295, 1965. Referred to on pg. 147

[16] Beucher, S., "Watersheds of Functions and Picture Segmentation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1928–1931, 1982. Referred to on pg. 38

[17] de Bont, P.W., van Kempen, G.M.P., and Vreeker, R., "Phase Separation in Milk Protein and Amylopectin Systems," *Food Hydrocolloids*, vol. 16, pp. 127–138, 2001. Referred to on pg. 129, 131

[18] de Bont, P.W., Luengo Hendriks, C.L., van Kempen, G.M.P., and Vreeker, R., "Time Evolution of Phase Separating Milk Protein and Amylopectin Mixtures," *submitted to Food Hydrocolloids*. Referred to on pg. 23, 125, 131

[19] van den Boomgaard, R. and Smeulders, A., “The Morphological Structure of Images: The Differential Equations of Morphological Scale-Space,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 16, no. 11, pp. 1101–1113, 1994. Referred to on pg. 79, 80

[20] de Boor, C., *A Practical Guide to Splines,* Springer-Verlag, New York, 2001. Referred to on pg. 78

[21] Bracewell, R.N., *The Fourier Transform and its Applications,* 3rd ed., McGraw-Hill, New York, 2000. Referred to on pg. 150

[22] Breen, E.J. and Jones, R., “An Attribute-Based Approach to Mathematical Morphology,” in *Mathematical Morphology and its Applications to Image and Signal Processing* (Maragos, P., Schafer, R.W., and Akmal Butt, M., eds.), pp. 41–48, Kluwer Academic Publishers, Boston, 1996. Referred to on pg. 37

[23] Breen, E.J. and Jones, R., “Attribute Openings, Thinnings, and Granulometries,” *Computer Vision and Image Understanding,* vol. 64, no. 3, pp. 377–389, 1996. Referred to on pg. 37

[24] Bresenham, J.E., “Algorithm for Computer Control of a Digital Plotter,” *IBM Systems Journal,* vol. 4, no. 1, pp. 25–30, 1965. Referred to on pg. 90

[25] Brockett, R.W. and Maragos, P., “Evolution Equations for Continuous-Scale Morphological Filtering,” *IEEE Transactions on Signal Processing,* vol. 42, no. 12, pp. 3377–3386, 1994. Referred to on pg. 75

[26] Bronstein, I.N., Semendjaev, K.A., Musiol, G., and Mühlig, H., *Taschenbuch der Mathematik,* 4th ed., Verlag Harri Deutsch, Thun, Frankfurt am Main, 1999. Referred to on pg. 78, 82

[27] Chanussot, J. and Lambert, P., “An Application of Mathematical Morphology to Road Network Extractions on SAR Images,” in *Mathematical Morphology and its Applications to Image and Signal Processing* (Heijmans, J.A.M. and Roerdink, J.B.T.M., eds.), pp. 399–406, Kluwer Academic Publishers, Dordrecht, 1998. Referred to on pg. 89

[28] Chen, M.H. and Yan, P.F., “A Multiscaling Approach Based on Morphological Filtering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 694–700, 1989. Referred to on pg. 57

[29] Cheng, F. and Venetsanopoulos, A.N., “An Adaptive Morphological Filter for Image Processing,” *IEEE Transactions on Image Processing,* vol. 1, no. 4, pp. 533–539, 1992. Referred to on pg. 37

[30] Deans, S.R., "Hough Transform from the Radon Transform," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 2, pp. 185–188, 1981. Referred to on pg. 142, 149

[31] Digabel, H. and Lantuéjoul, C., "Iterative Algorithms," in *Quantitative Analysis of Microstructures in Materials Sciences, Biology and Medicine* (Chermant, J.L., ed.), pp. 85–99, Dr. Rieder-Verlag, Stuttgart, 1978. Referred to on pg. 38

[32] Dorst, L. and van den Boomgaard, R., "Morphological Signal Processing and the Slope Transform," *Signal Processing*, vol. 38, no. 1, pp. 79–98, 1994. Referred to on pg. 62, 67, 79, 80

[33] Duda, R.O. and Hart, P.E., "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972. Referred to on pg. 147, 148

[34] Duda, R.O., Hart, P.E., and Stork, D.G., *Pattern Classification*, 2nd ed., Wiley, New York, 2001. Referred to on pg. 149

[35] Gel'fand, I.M., Graev, M.I., and Vilenkin, N.Y., *Generalized Functions. Volume 5, Integral Geometry and Representation Theory*, Academic Press, New York, 1966. Referred to on pg. 145

[36] Gel'fand, I.M. and Shilov, G.E., *Generalized Functions. Volume 1, Properties and Operations*, Academic Press, New York, 1964. Referred to on pg. 143

[37] van Ginkel, M., *Image Analysis using Orientation Space based on Steerable Filters*, Ph.D. thesis, Pattern Recognition Group, Faculty of Applied Sciences, Delft University of Technology, Delft, 2002. Referred to on pg. 56, 148, 150

[38] van der Glas, M., Vos, F.M., and Vossepoel, A.M., "Assessment of Center of Rotation of the Glenohumeral Joint by a Modified Hough Transform," in *Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention* (Niessen, W.J. and Viergever, M.A., eds.), vol. 2208 of *Lecture Notes on Computer Science*, pp. 1207–1209, Springer-Verlag, Berlin, 2001. Referred to on pg. 156

[39] Goetcherian, V., "From Binary to Grey Tone Image Processing Using Fuzzy Logic Concepts," *Pattern Recognition*, vol. 12, pp. 7–15, 1980. Referred to on pg. 38

[40] Griffith, A.K., "Edge Detection in Simple Scenes Using A Priori Information," IEEE *Transactions on Computers*, vol. 22, no. 4, pp. 371–381, 1973. Referred to on pg. 149

[41] ter Haar Romeny, B.M. (ed.), *Geometry-Driven Diffusion in Computer Vision*, Kluwer Academic Publishers, Dordrecht, 1994. Referred to on pg. 48

[42] Hadwiger, H., *Vorlesungen über Inhalt, Oberfläche und Isoperimetrie*, Springer-Verlag, Berlin, 1957. Referred to on pg. 26

[43] Haralick, R.M., Sternberg, S.R., and Zhuang, X., “Image Analysis Using Mathematical Morphology,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, 1987. Referred to on pg. 25

[44] Heijmans, H.J.A.M., *Morphological Image Operators*, vol. 25 of *Advances in Electronics and Electron Physics*, Academic Press, San Diego, California, 1994. Referred to on pg. 25

[45] Heijmans, H.J.A.M. and Ronse, C., “The Algebraic Basis of Mathematical Morphology I. Dilations and Erosions,” *Computer Vision, Graphics, and Image Processing*, vol. 50, pp. 245–295, 1990. Referred to on pg. 25

[46] van Herk, M., “A Fast Algorithm for Local Minimum and Maximum Filters on Rectangular and Octagonal Kernels,” *Pattern Recognition Letters*, vol. 13, pp. 517–521, 1992. Referred to on pg. 32, 79, 89, 90

[47] Hough, P.V.C., “Method and Means for Recognizing Complex Patterns,” US patent nr. 3069654, 1962. Referred to on pg. 141, 147, 149

[48] Hsu, C.C. and Huang, J.S., “Partitioned Hough Transform for Ellipsoid Detection,” *Pattern Recognition*, vol. 23, pp. 275–282, 1990. Referred to on pg. 156

[49] Huang, T.S., Yang, G.J., and Tang, G.Y., “A Fast Two-Dimensional Median Filtering Algorithm,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979. Referred to on pg. 114

[50] Illingworth, J. and Kittler, J., “A Survey of the Hough Transform,” *Computer Vision, Graphics and Image Processing*, vol. 44, no. 1, pp. 87–116, 1988. Referred to on pg. 148

[51] Jackway, P.T., “Properties of Multiscale Morphological Smoothing by Poweroids,” *Pattern Recognition Letters*, vol. 15, pp. 135–140, 1994. Referred to on pg. 31

[52] Jackway, P.T. and Deriche, M., “Scale-Space Properties of the Multiscale Morphological Dilation-Erosion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 38–51, 1996. Referred to on pg. 57

[53] Jähne, B., *Digital Image Processing*, Springer-Verlag, Berlin, 2002. Referred to on pg. 114

[54] Jones, R. and Soille, P., "Periodic Lines: Definition, Cascades, and Application to Granulometries," *Pattern Recognition Letters*, vol. 17, no. 10, pp. 1057–1063, 1996. Referred to on pg. 89, 90, 92

[55] Katartzis, A., Pizurica, V., and Sahli, H., "Applications of Mathematical Morphology and Markov Random Field Theory to the Automatic Extraction of Linear Features in Airborne Images," in *Mathematical Morphology and its Applications to Image and Signal Processing* (Goutsias, J., Vincent, L., and Bloomberg, D.S., eds.), pp. 405–414, Kluwer Academic Publishers, Boston, 2000. Referred to on pg. 89

[56] Keys, R.G., "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981. Referred to on pg. 94

[57] Kimme, C., Ballard, D., and Sklansky, J., "Finding Circles by an Array of Accumulators," *Communications of the ACM*, vol. 18, no. 2, pp. 120–122, 1975. Referred to on pg. 147

[58] Kiryati, N. and Bruckstein, A.M., "Antialiasing the Hough transform," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 213–222, 1991. Referred to on pg. 149, 153

[59] Kiryati, N. and Bruckstein, A.M., "What's in a Set of Points?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 4, pp. 496–500, 1992. Referred to on pg. 147

[60] Kitchen, L. and Rosenfeld, A., "Gray-Level Corner Detection," *Pattern Recognition Letters*, vol. 1, no. 2, pp. 95–102, 1982. Referred to on pg. 143

[61] Klein, J.C. and Serra, J., "The Texture Analyzer," *Journal of Microscopy*, vol. 95, no. 2, pp. 349–356, 1972. Referred to on pg. 25

[62] Koenderink, J.J., "The Structure of Images," *Biological Cybernetics*, vol. 50, pp. 363–370, 1984. Referred to on pg. 57, 151

[63] Koenderink, J.J. and Richards, W., "Two-Dimensional Curvature Operators," *Journal of the Optical Society of America A*, vol. 5, no. 7, pp. 1136–1141, 1988. Referred to on pg. 143

[64] Kuwahara, M., Hachimura, K., Eiho, S., and Kinoshita, M., “Processing of RI-Angiocardiographic Images,” in *Digital Processing of Biomedical Images* (Preston, K. and Onoe, M., eds.), pp. 187–203, Plenum Press, New York, 1976. Referred to on pg. 114

[65] Kuwahara, M., Hachimura, K., and Kinoshita, M., “Image Enhancement and Left Ventricular Contour Extraction Techniques Applied to Radioisotope Angiocardiograms,” *Automedica*, vol. 3, pp. 107–119, 1980. Referred to on pg. 114

[66] Luengo Hendriks, C.L., *Improved Resolution in Infrared Imaging Using Randomly Shifted Images*, Master’s thesis, Pattern Recognition Group, Faculty of Applied Sciences, Delft University of Technology, Delft, 1998. Referred to on pg. 67

[67] Luengo Hendriks, C.L., van Ginkel, M., Verbeek, P.W., and van Vliet, L.J., “The Generalized Radon Transform: Sampling, Accuracy and Memory Considerations,” *in preparation*. Referred to on pg. 23

[68] Luengo Hendriks, C.L., van Ginkel, M., Verbeek, P.W., and van Vliet, L.J., “The Generalized Radon Transform: Sampling and Memory Considerations,” in *Computer Analysis of Images and Patterns, Proceedings of the Tenth International Conference, CAIP 2003* (Petkov, N. and Westenberg, M.A., eds.), vol. 2756 of *Lecture Notes on Computer Science*, pp. 681–688, Springer-Verlag, Berlin, 2003. Referred to on pg. 23

[69] Luengo Hendriks, C.L., van Ginkel, M., and van Vliet, L.J., *Underestimation of the Radius in the Radon Transform for Circles and Spheres*, Tech. Rep. PH-2003-02, Pattern Recognition Group, Delft University of Technology, 2003. Referred to on pg. 23, 154, 155

[70] Luengo Hendriks, C.L. and van Vliet, L.J., “A Rotation-Invariant Morphology for Shape Analysis of Anisotropic Objects and Structures,” in *Proceedings of the Fourth International Workshop on Visual Form* (Arcelli, C., Cordella, L.P., and Sanniti di Baja, G., eds.), vol. 2059 of *Lecture Notes on Computer Science*, pp. 378–387, Springer-Verlag, Berlin, 2001. Referred to on pg. 22

[71] Luengo Hendriks, C.L. and van Vliet, L.J., “Segmentation-Free Estimation of Length Distributions Using Sieves and RIA Morphology,” in *Proceedings of the Third International Conference on Scale-Space* (Kerckhove, M., ed.), vol. 2106 of *Lecture Notes on Computer Science*, pp. 389–406, Springer-Verlag, Berlin, 2001. Referred to on pg. 23, 125

[72] Luengo Hendriks, C.L. and van Vliet, L.J., "Basic Morphological Operations, Band-Limited Images and Sampling," in *Scale Space Methods in Computer Vision, Proceedings of the Fourth International Conference, Scale-Space 2003* (Griffin, L.D. and Lillholm, M., eds.), vol. 2695 of *Lecture Notes on Computer Science,* pp. 313–324, Springer-Verlag, Berlin, 2003. Referred to on pg. 22

[73] Luengo Hendriks, C.L. and van Vliet, L.J., "Discrete Morphology with Line Structuring Elements," in *Computer Analysis of Images and Patterns, Proceedings of the Tenth International Conference, CAIP 2003* (Petkov, N. and Westenberg, M.A., eds.), vol. 2756 of *Lecture Notes on Computer Science,* pp. 722–729, Springer-Verlag, Berlin, 2003. Referred to on pg. 22

[74] Luengo Hendriks, C.L., van Vliet, L.J., van Kempen, G.M.P., and Bouwens, E.C.M., "Improving Morphological Sieves to Detect Minute Differences in Structures," *submitted to IEEE Transactions on Image Processing*. Referred to on pg. 22, 23, 125

[75] Maragos, P., "Pattern Spectrum and Multiscale Shape Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 701–716, 1989. Referred to on pg. 57, 58, 62

[76] Maragos, P., "Morphological Systems: Slope Transforms and Max-Min Difference and Differential Equations," *Signal Processing*, vol. 38, no. 1, pp. 57–77, 1994. Referred to on pg. 79, 80

[77] Maragos, P., "Slope Transforms: Theory and Application to Nonlinear Signal Processing," *IEEE Transactions on Signal Processing*, vol. 43, no. 4, pp. 864–877, 1995. Referred to on pg. 79, 80

[78] Matheron, G., *Elements pour une Théorie des Milieux Poreux*, Masson, Paris, 1967. Referred to on pg. 25

[79] Matheron, G., *Random Sets and Integral Geometry,* Wiley, New York, 1975. Referred to on pg. 22, 35, 42, 51, 54, 57

[80] Matheron, G. and Serra, J., "The Birth of Mathematical Morphology," in *Mathematical Morphology, Proceedings of the VI International Symposium ISMM, Sydney, April 2002* (Talbot, H. and Beare, R., eds.), pp. 1–16, CSIRO Publishing, Collingwood, Australia, 2002. Referred to on pg. 25

[81] Merlin, P.M. and Farber, D.J., "A Parallel Mechanism for Detecting Curves in Pictures," *IEEE Transactions on Computers,* vol. 24, pp. 96–98, 1975. Referred to on pg. 147

[82] Meyer, F., “The Levelings,” in *Mathematical Morphology and its Applications to Image and Signal Processing* (Heijmans, J.A.M. and Roerdink, J.B.T.M., eds.), pp. 199–206, Kluwer Academic Publishers, Dordrecht, 1998. Referred to on pg. 37

[83] Meyer, F., “Flooding and Segmentation,” in *Mathematical Morphology and its Applications to Image and Signal Processing* (Goutsias, J., Vincent, L., and Bloomberg, D.S., eds.), pp. 189–198, Kluwer Academic Publishers, Boston, 2000. Referred to on pg. 37, 121

[84] Mouton, P.R., *Principles and Practices of Unbiased Stereology: an Introduction for Bioscientists*, Johns Hopkins University Press, Baltimore, 2002. Referred to on pg. 26, 117, 127

[85] Nagao, M. and Matsuyama, T., “Edge Preserving Smoothing,” *Computer Graphics and Image Processing*, vol. 9, pp. 394–407, 1979. Referred to on pg. 114

[86] Nakagawa, Y. and Rosenfeld, A., “A Note on the Use of Local Min and Max Operations in Digital Picture Processing,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, no. 8, p. 3, 1978. Referred to on pg. 26

[87] Niblack, W. and Petkovic, D., “On Improving the Accuracy of the Hough Transform: Theory, Simulations, and Experiments,” in *Proceedings of the IEEE Computer Society Conference CVPR*, pp. 574–579, 1988. Referred to on pg. 147

[88] Nyquist, H., “Certain Topics in Telegraph Transmission Theory,” *Transactions of the AIEE*, pp. 617–644, 1928. [reprinted in: Proceedings of the IEEE, vol. 90, pp. 280–305, 2002]. Referred to on pg. 18, 103

[89] Onuki, A. and Puri, S., “Spinodal Decomposition in Gels,” *Physical Review E*, vol. 59, no. 2, pp. 1331–1334, 1999. Referred to on pg. 129

[90] Oppenheim, A.V., Willsky, A.S., and Young, I.T., *Signals and Systems*, Prentice-Hall International, London, 1983. Referred to on pg. 153

[91] Papoulis, A., *Signal Analysis*, McGraw-Hill, New York, 1977. Referred to on pg. 151

[92] Park, K.R. and Lee, C.N., “Scale-Space Using Mathematical Morphology,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 11, pp. 1121–1126, 1996. Referred to on pg. 57

[93] Princen, J., Illingworth, J., and Kittler, J., “A Formal Definition of the Hough Transform: Properties and Relationships,” *Journal of Mathematical Imaging and Vision*, vol. 1, pp. 153–168, 1992. Referred to on pg. 142, 148, 149

[94] Princen, J., Illingworth, J., and Kittler, J., “Hypothesis Testing: A Framework for Analyzing and Optimizing Hough Transform Performance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 4, pp. 329–341, 1994. Referred to on pg. 147

[95] Radon, J., “Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten,” *Berichte Sächsische Akademie der Wissenschaften*, vol. 69, pp. 262–277, 1917. Referred to on pg. 141, 144

[96] Roerdink, J.B.T.M., *On the Construction of Translation and Rotation Invariant Morphological Operators*, Tech. Rep. AM-R9025, Centre for Mathematics and Computer Science, Amsterdam, 1990. [to appear in *Mathematical Morphology: Theory and Hardware* (Haralick, R.M., ed.), Oxford University Press]. Referred to on pg. 42

[97] Roerdink, J.B.T.M., “Mathematical Morphology with Noncommutative Symmetry Groups,” in *Mathematical Morphology in Image Processing* (Dougherty, E.R., ed.), pp. 205–254, Marcel Dekker, New York, 1993. Referred to on pg. 42

[98] Roerdink, J.B.T.M., “Group Morphology,” *Pattern Recognition*, vol. 33, no. 6, pp. 877–895, 2000. Referred to on pg. 42

[99] Roerdink, J.B.T.M. and Meijster, A., “The Watershed Transform: Definitions, Algorithms and Parallelization Strategies,” *Fundamenta Informaticae*, vol. 41, no. 1-2, pp. 187–228, 2001. Referred to on pg. 118

[100] Ronse, C., “Erosion of Narrow Image Features by Combination of Local Low Rank and Max Filters,” in *Proceedings of the Second IEE International Conference on Image Processing and its Applications*, pp. 77–81, London, 1986. Referred to on pg. 37

[101] Rosenfeld, A., *Picture Processing by Computer*, Academic Press, New York, 1969. Referred to on pg. 147

[102] Sapiro, G., Kimmel, R., Kimia, B.B., and Bruckstein, A.M., “Implementing Continuous-Scale Morphology via Curve Evolution,” *Pattern Recognition*, vol. 26, no. 9, pp. 1363–1372, 1993. Referred to on pg. 75

[103] Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982. Referred to on pg. 25, 48

[104] Serra, J., "Alternating Sequential Filters," in *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances* (Serra, J., ed.), pp. 203–216, Academic Press, London, 1988. Referred to on pg. 114

[105] Serra, J., "The Centre and Self-Dual Filtering," in *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances* (Serra, J., ed.), pp. 159–180, Academic Press, London, 1988. Referred to on pg. 114

[106] Sethian, J.A., "Curvature and the Evolution of Fronts," *Communication of Mathematical Physics*, vol. 101, no. 4, 1985. Referred to on pg. 48, 150

[107] Sethian, J.A., *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, 1996. Referred to on pg. 48

[108] Shannon, C., "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949. [reprinted in: Proceedings of the IEEE, vol. 86, pp. 447–457, 1998]. Referred to on pg. 18, 103

[109] Shapiro, S.D., "Transformations for the Computer Detection of Curves in Noisy Pictures," *Computer Graphics and Image Processing*, vol. 4, pp. 328–338, 1975. Referred to on pg. 147

[110] Shapiro, S.D., "Feature Space Transforms for Curve Detection," *Pattern Recognition*, vol. 10, pp. 129–143, 1978. Referred to on pg. 147

[111] Shapiro, S.D., "Properties of Transforms for the Detection of Curves in Noisy Pictures," *Computer Graphics and Image Processing*, vol. 8, pp. 219–236, 1978. Referred to on pg. 147

[112] Shatadal, P., Jayas, D.S., Hehn, J.L., and Bulley, N.R., "Seed Classification Using Machine Vision," *Canadian Agriculture Engineering*, vol. 37, no. 3, pp. 163–167, 1995. Referred to on pg. 138

[113] Sklansky, J., "On the Hough Technique for Curve Detection," *IEEE Transactions on Computers*, vol. 27, no. 10, pp. 923–926, 1978. Referred to on pg. 142, 148, 149

[114] Soille, P., *Morphological Image Analysis*, Springer-Verlag, Berlin, 1999. Referred to on pg. 25, 26, 28, 57, 58

[115] Soille, P., “Morphological Operators with Discrete Line Segments,” in *DGCI 2000, 9th Discrete Geometry for Computer Imagery* (Borgefors, G., Nyström, I., and Sanniti di Baja, G., eds.), vol. 1953 of *Lecture Notes on Computer Science*, pp. 78–98, Springer-Verlag, Berlin, 2000. Referred to on pg. 48

[116] Soille, P., “Morphological Texture Analysis: an Introduction,” *Lecture Notes in Physics*, vol. 600, p. 215:237, 2002. Referred to on pg. 57

[117] Soille, P., “Short Survey on Morphological Operators Based on Rank Filters,” *Pattern Recognition*, vol. 35, pp. 527–535, 2002. Referred to on pg. 25, 37

[118] Soille, P., Breen, E.J., and Jones, R., “Recursive Implementation of Erosions and Dilations Along Discrete Lines at Arbitrary Angles,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 562–567, 1996. Referred to on pg. 32, 89, 90

[119] Soille, P. and Talbot, H., “Image Structure Orientation using Mathematical Morphology,” in *Proceedings of the 14th IAPR International Conference on Pattern Recognition* (Jain, A.K., Venkatesh, S., and Lovell, B.C., eds.), vol. 1, pp. 1467–1469, IEEE Computer Society Press, Los Alamitos, 1998. Referred to on pg. 56, 89

[120] Soille, P. and Talbot, H., “Directional Morphological Filtering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1313–1329, 2001. Referred to on pg. 56, 89, 92

[121] Sternberg, S.R., “Grayscale Morphology,” *Computer Vision, Graphics and Image Processing*, vol. 35, pp. 333–355, 1986. Referred to on pg. 26, 114

[122] Stockman, G.C. and Agrawala, A.K., “Equivalence of Hough Curve Detection to Template Matching,” *Communications of the ACM*, vol. 20, no. 11, pp. 820–822, 1977. Referred to on pg. 142, 148, 149

[123] Stoker, J.J., *Differential Geometry*, John Wiley and Sons, Chichester, 1969. Referred to on pg. 143

[124] Stoyan, D., Kendall, W.S., and Mecke, J., *Stochastic Geometry*, 2nd ed., Wiley, Chichester, 1995. Referred to on pg. 127

[125] Tromp, R.H., Rennie, A.R., and Jones, R.A.L., “Kinetics of the Simultaneous Phase Separation and Gelation in Solutions of Dextran and Gelatin,” *Macromolecules*, vol. 28, no. 12, pp. 4129–4138, 1995. Referred to on pg. 131

[126] Tscheschel, A., Stoyan, D., and Hilfer, R., “Erosion-Dilation Analysis for Experimental and Synthetic Microstructures of Sedimentary Rock,” *Physica A*, vol. 284, pp. 46–58, 2000. Referred to on pg. 57

[127] Tukey, J.W., *Exploratory Data Analysis*, Addison-Wesley, Reading, Massachusetts, 1977. Referred to on pg. 114

[128] Tuzikov, A., Soille, P., Jeulin, D., Bruneel, H., and Vermeulen, M., “Extraction of Grid Patterns on Stamped Metal Sheets Using Mathematical Morphology,” in *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, vol. 1, pp. 425–428, The Hague, 1992. Referred to on pg. 89

[129] Unser, M., Aldroubi, A., and Eden, M., “Fast B-Spline Transforms for Continuous Image Representation and Interpolation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 277–285, 1991. Referred to on pg. 78

[130] Unser, M., Aldroubi, A., and Eden, M., “Polynomial Spline Signal Approximations: Filter Design and Asymptotic Equivalence with Shannon’s Sampling Theorem,” *IEEE Transactions on Information Theory*, vol. 38, no. 1, pp. 95–103, 1992. Referred to on pg. 77, 78

[131] Unser, M., Aldroubi, A., and Eden, M., “B-Spline Signal Processing: Part I - Theory,” *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–833, 1993. Referred to on pg. 78, 83

[132] van Veen, T.M. and Groen, F.C.A., “Discretization Error in the Hough transform,” *Pattern Recognition*, vol. 14, pp. 137–145, 1981. Referred to on pg. 147

[133] Verbeek, P.W., “A Class of Sampling-error Free Measures in Oversampled Band-Limited Images,” *Pattern Recognition Letters*, vol. 3, pp. 287–292, 1985. Referred to on pg. 143, 150

[134] Verbeek, P.W., Vrooman, H.A., and van Vliet, L.J., “Low-level Image Processing by Max-Min Filters,” *Signal Processing*, vol. 15, no. 3, pp. 249–258, 1988. Referred to on pg. 114

[135] Verhaegh, N.A.M., van Duijneveldt, J.S., Dhont, J.K.G., and Lekkerkerker, H.N.W., “Fluid-Fluid Phase Separation in Colloid-Polymer Mixtures Studied with Small Angle Light Scattering and Light Microscopy,” *Physica A*, vol. 230, pp. 409–436, 1996. Referred to on pg. 131

[136] Verwer, B.J.H., van Vliet, L.J., and Verbeek, P.W., "Binary and Grey-Value Skeletons: Metrics and Algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 5, pp. 1287–1308, 1993. Referred to on pg. 118, 119

[137] Vincent, L., "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, 1993. Referred to on pg. 38

[138] Vincent, L. and Soille, P., "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991. Referred to on pg. 118, 119

[139] van Vliet, L.J., *Grey-Scale Measurements in Multi-Dimensional Digitized Images*, Ph.D. thesis, Pattern Recognition Group, Faculty of Applied Physics, Delft University of Technology, Delft, 1993. Referred to on pg. 20, 67, 115, 150, 151

[140] van Vliet, L.J., Boddeke, F.R., Sudar, D., and Young, I.T., "Image Detectors for Digital Image Microscopy," in *Digital Image Analysis of Microbes; Imaging, Morphometry, Fluorometry and Motility Techniques and Applications* (Wilkinson, M.H.F. and Schut, F., eds.), pp. 37–64, John Wiley and Sons, Chichester, 1998. Referred to on pg. 18

[141] van Vliet, L.J. and Verbeek, P.W., "Edge Localization by MoG Filters: Multiple-of-Gaussians," *Pattern Recognition Letters*, vol. 15, no. 5, pp. 485–496, 1994. Referred to on pg. 65

[142] Vogt, R.C., "Morphological Operator Distributions Based on Monotonicity and the Problem Posed by Digital Disk-Shaped Structuring Elements," in *Digital and Optical Shape Representation and Pattern Recognition, SPIE Proceedings*, vol. 938, pp. 1019–1022, Orlando, Florida, 1988. Referred to on pg. 58

[143] Weibel, E.R., *Stereological Methods, Volume 1*, Academic Press, London, 1979. Referred to on pg. 127

[144] Weickert, J., *Anisotropic Diffusion in Image Processing*, Teubner, Stuttgart, 1998. Referred to on pg. 48

[145] Westin, C.F. and Knutsson, H., "The Möbius Strip Parameterization for Line Extraction," in *Computer Vision, Second European Conference on Computer Vision*, vol. 588 of *Lecture Notes on Computer Science*, pp. 33–38, Springer-Verlag, Berlin, 1992. Referred to on pg. 148

[146] Witkin, A.P., “Scale-Space Filtering,” in *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, vol. 2, pp. 1019–1022, Karlsruhe, Germany, 1983. Referred to on pg. 57

[147] Yuen, S.Y. and Ma, C.H., “An Investigation of the Nature of Parameterization for the Hough Transform,” *Pattern Recognition*, vol. 30, no. 6, pp. 1009–1040, 1997. Referred to on pg. 148

[148] Zayas, I., Converse, H., and Steele, J., “Discrimination of Whole from Broken Corn Kernels with Image Analysis,” *Transactions of the American Society of Agricultural Engineers*, vol. 33, no. 5, pp. 1642–1646, 1990. Referred to on pg. 138

Index of Terms

This list contains the page numbers where each term is defined. The term is emphasized in the text to make it easier to find. Terms from Chapter 8 (about the Radon transform) are not included.