Image Analysis Methods Based on Hierarchies of Graphs and Multi-Scale Mathematical Morphology

Peter F.M. Nacken

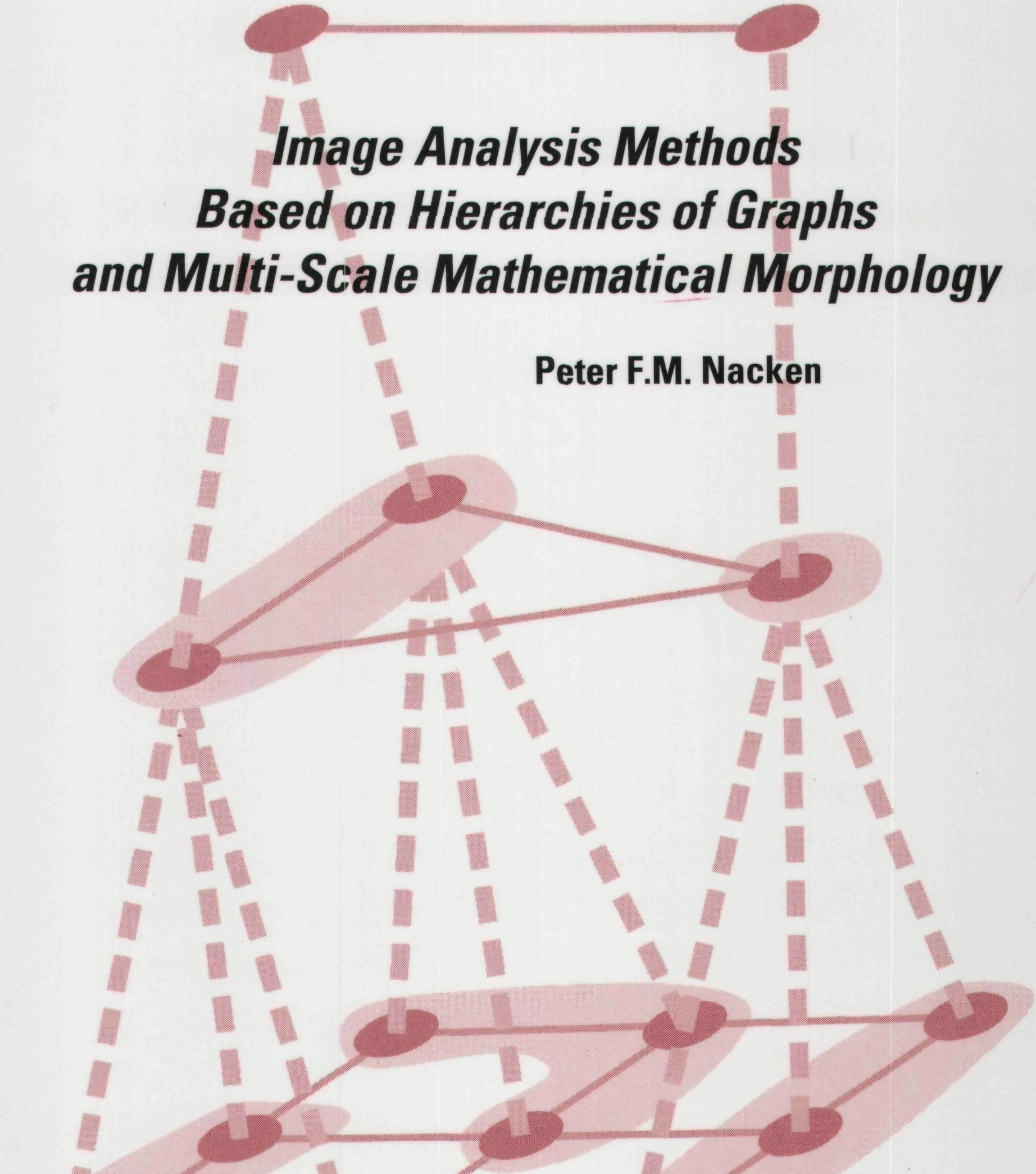


Image Analysis Methods Based on Hierarchies of Graphs and Multi-Scale Mathematical Morphology

Faculteit der Wiskunde en Informatica

Promotor :	Prof.dr.ir. F.C.A. Groen
Copromotores :	Dr.ir. H.J.A.M. Heijmans Dr. A. Toet

'Het schijnt dat u af en toe fraaie ontdekkingen doet.'
'Ik heb een enkele keer wel eens geluk gehad,' zei Inni,
'maar op uw terrein ben ik blind en doof.
U mag me rustig uitlachen.'

Cees Nooteboom, Rituelen

Voorwoord

Voor U ligt het resultaat van vier jaar onderzoek op het gebied van de beeldverwerking, uitgevoerd in een samenwerkingsverband van TNO Technische Menskunde, het Centrum voor Wiskunde en Informatica en de Universiteit van Amsterdam. Dit onderzoek is gefinancierd door de Stichting Informatica Onderzoek Nederland, met financiele steun van NWO.

Het bereikte resultaat was ondenkbaar geweest zonder de steun van de personen, die ik op deze plek wil bedanken.

Veel dank ben ik verschuldigd aan het mij begeleidend drietal, promotor Frans Groen en copromotores Henk Heijmans en Lex Toet. Ieder van hen heeft met zijn specifieke belangstelling en inbreng een unieke bijdrage geleverd aan dit proefschrift: Lex met zijn creatieve ideeën over hierarchische methodes, die het altijd beter doen dan je op het eerste gezicht verwacht, Henk met zijn nadruk op wiskundige juistheid en exacte formuleringen en Frans met de onophoudelijke vraag, niet alleen te laten zien *wat* je methode doet, maar ook door en door te begrijpen, *waarom* hij dat doet. Wat Frans, Henk en Lex echter gemeen hebben, is hun oprechte persoonlijke belangstelling. De diversiteit van hun benaderingen heeft het me niet altijd gemakkelijk gemaakt, maar is uiteindelijk zeer vruchtbaar geweest.

Graag wil ik ook Adri Steenbeek bedanken. De door hem gemaakte C++-classes hebben als basis gediend bij het programmeren van de in dit proefschrift beschreven graaf-methodes. Hoewel het niet direct zichtbaar is, is zijn werk op vele plaatsen in dit boekje aanwezig.

Ik bedank de medewerkers van TNO in Soesterberg en van het CWI voor de prettige sfeer, die onontbeerlijk is voor het schrijven van een proefschrift.

Während sechs Monaten hatte ich das Vergnügen, zu Gast zu sein bei der Fachschaft für Mustererkennung und Bildverarbeitung von Herrn Prof. W. Kropatsch der Technischen Universität Wien. Ich bedanke mich bei ihm und sein "PRIP-team", besonders bei Dieter Willersinn, für ihre Gastfreundschaft. In Wien hatte Ich die Gelegenheit, mit Leuten mit einen ganz eigenen und inspirierenden Blick auf hierarchische Methoden, zusammenzuarbeiten.

Van groot belang voor het ontstaan van dit proefschrift zijn ook de contacten met vele andere onderzoekers geweest, die met hun suggesties en eigen werk een belangrijke bron van inspiratie zijn geweest. Ik wil hier met name noemen: Adrian Baddeley, Rein van den Boomgaard, Annick Montanvert, Carol Orange en Luc Vincent.

Contents

1

Introduction

1.1. Image Analysis

Computer analysis of pictorial data is an important field in computer science, mathematics and electrical engineering. The human visual system shows incredible performance in handling such data. It can process huge amounts of information quickly and with great flexibility, and does this apparently without any effort.

The problem of programming a computer to make it 'see' has appeared to be very difficult. While even very simple computers surpass humans in numerical tasks and modern chess playing programs can beat all but the very best human players, even very primitive animals outdo any computer vision system in the task of seeing.

Yet there are image analysis problems where the machine performs better than human beings. Such problems are concerned with images displaying objects and situations known in advance. Typical examples are counting objects, measuring objects and quality checks on artifacts. For such problems, image analysis systems have been designed which perform their task quickly, accurately and, sometimes just as important, without getting tired.

The central issue in image analysis is making image contents explicit. Which aspect of the information present in an image is relevant, depends on the application at hand. Therefore, image contents can be described at various levels, ranging from the raw image data to a symbolic description using domain dependent symbols.

An image represents a series of measurements, performed on some set of physical objects called the scene. These measurements are arranged on a regular grid. Various quantities can be measured, corresponding to different imaging methods. Most commonly, the intensity of reflected light is measured, possibly in various bands of the electro-magnetic spectrum. There are also techniques which measure the reflection of sound waves (ultrasound medical imaging and seismology), the density of hydrogen atoms in a sample (MRI) or the transmission of X-rays, to mention a few.

The spatial arrangement of measurements corresponds to the spatial arrangement of the scene. This correspondence can be direct, for example if flat surfaces are recorded by a camera, or indirect in the case of MRI or ultrasound imaging. If a 3-D scene is recorded by a camera, the relation between the geometry of the scene and the geometry of the image is described by the perspective transform. In this situation, problems such as occlusion may occur.

In a raw image, all information is present implicitly, but not in a useful form. Image analysis systems must detect structure in such an—often very large—set of measurements. Image analysis always involves the identification of specific subparts of the image. These subparts can correspond to specific objects in the scene or to specific structures in the image. When the relevant subparts of the image have been detected, they can be analyzed in order to extract useful information on the image contents.

There is a large number of techniques for manipulating image subparts, corresponding to the large amount of image analysis applications. If image analysis is performed for object recognition, for example in a robot manipulation environment, the selected image subparts are compared with models in a database, and a list of recognized objects and their locations is produced. If image analysis is used for measuring properties of materials, numerical measurements are performed on the identified image subparts. Here, one can think of counting and measuring the holes and grains in a soil sample. The results of image analysis can also be output in a visual form, for example by overlaying the input image with colored overlays of specific image subparts. Such output is important when image analysis is performed in an interactive environment [81].

Although these examples show that specific problems require specific solutions, a large amount of generally applicable image analysis techniques has been developed over the years. Yet, the construction of a general purpose vision system, comparable in its abilities to the human visual system, is far from feasible at the moment. Neither is there a single, all-embracing theory of image analysis and computer vision.

The subjects discussed in this thesis are centered around a particular type of image representations, namely hierarchical ones. The purpose of the research presented here is to gain an insight in the possibilities of such representations and to develop a number of methods using them. The choice for a particular representation, rather than some specific problem, implies that the techniques described in this thesis are general purpose ones. At some instances, however, domain dependent knowledge is introduced in our methods, when they are applied in a specific situation.

1.2. Hierarchical Methods

An important class of image analysis methods is formed by hierarchical techniques. This field started in the mid seventies, when Tanimoto and Pavlidis presented the grey level pyramid [100] and Horowitz and Pavlidis presented the split-and-merge segmentation method [45]. Rosenfeld [88] wrote: "Pyramids, in general, are data structures that provide successively condensed representations of the information in the input image. What is condensed may be simply image intensity, so that the successive levels of the pyramid are reduced-resolution versions of the input image; but it may also be descriptive information about features in the image, so that successive levels represent increasingly coarse approximations to these features." When Rosenfeld wrote this in 1983, the phrase *pyramid* referred to a regular pyramid: an ordered stack of images defined on a regular grid whose dimensions were reduced by a factor of 2 between successive levels. Later, other data structures providing successively condensed image representations have been developed, such as the irregular pyramid [57, 76]. There are also image analysis methods which have the characteristics mentioned by Rosenfeld, without the use of an explicit representation of successively condensed image descriptions.

An important characteristic of hierarchical methods is the fact that global structures become local in higher levels of a hierarchy, in the sense that each descriptive element on a high description level integrates information over a larger part of the image plane. This integration can be performed by averaging image values over increasingly larger regions or by some more advanced method of approximating image contents. It is also possible, that image primitives which have significance at a larger scale are the only ones which survive in the higher description levels.

There are several motivations for using hierarchical techniques. First of all, the scale at which interesting structure is present in an image is in general not known in advance. For example, the size at which an object in a 3-D scene is represented in an image depends on the distance to the camera, which may vary. Therefore, it is attractive to perform analysis of image contents at several scales simultaneously. Many images contain interesting structures at various scales. Usually, there is not a single 'best' description of an image. It depends on the application at hand and on the availability of domain specific knowledge, which parts of a description are relevant. Therefore, it can be attractive to have a hierarchical image description, which can be inspected by other parts of the image analysis system.

A second motivation is efficiency of computation. In many applications, most of the operations can be performed initially on a coarse representation of an image. The information obtained by processing a coarse description is then used to restrict the more costly processing of the fine representation of the image to specific parts of the image plane. A divide-and-conquer approach can result in faster algorithms, even if computations are not performed on parallel hardware. This is for example the case for the computation of special types of convolutions, such as those occurring in the computation of the Gaussian and Laplacian pyramids [17]. The construction of specialized hardware is most attractive when the image is represented by regular data structures [107]. Mapping irregular data structures to parallel hardware is also feasible, but can lead to a large communication overhead [111].

A third motivation for the use of hierarchical methods is the fact that such methods allow uniform treatment of the image at different scales, thus bridging the gap between the most elementary descriptive elements such as pixels, and richer descriptive elements such as regions in an image. This is expressed by the fact that the operations performed for the construction of a hierarchy are scale invariant or that the same operations are performed in each level of a hierarchy.

1.3. Scale Space and Image Pyramids

Hierarchical image descriptions can be classified by the type of condensation represented by the hierarchy, i.e. the sense in which information is reduced in higher levels of the hierarchy. In scale space and image pyramids, the higher levels of the hierarchy contain information integrated over a larger part of the image plane or they represent larger parts of the image plane. The regions over which information is integrated in these representations, is fixed in advance.

In such descriptions, each level in the hierarchy is itself an image. These images are created by considering the input image at successively coarser resolutions or scales. The motivation for this approach is the observation that important structures in an image are visible at low resolutions, whereas details are only visible at high resolution.

A desirable characteristic of this type of methods is *recursive computability*: a coarse or low resolution version of an image can be computed from any image at a less coarse resolution, without having to consider the initial image. This means that images at progressively coarser resolutions contain less and less information.

In the continuous case, the levels of the hierarchy can be created by a convolution of the original image with Gaussian kernels of increasing size. Koenderink [53] has shown that the simplest convolution kernel which does not create any new grey values in the image, is the Gaussian kernel. Thus, an image $f\colon \mathbb{R}^2 \to \mathbb{R}$ is described in *scale space* by the function

$$F(x, y; s) = (f * G_s)(x, y), \tag{1.1}$$

where x and y are spatial coordinates and s is the scale parameter; G_s is a Gaussian function of width s. In scale-space, both the image plane and the scale parameter are continuous. The relation between the scale parameter s and the region over which information is integrated is determined by the width of the Gaussian convolution kernels.

Marr [63] was particularly interested in the evolution of the zero crossings of the Laplacian* in scale space. He conjectured that these zero crossings contain important information on the structure of images. Hummel and Moniot [46] have shown that Marr's conjecture on the significance of zero crossings of the Laplacian was correct: under some weak conditions, the original image can be reconstructed from the zero crossings of the Laplacians of the successively blurred images, combined with the values of the gradients of the Laplacians at those zero crossings.

Lindeberg [60] has proposed a version of scale space for images defined on a discrete grid, but with a continuous scale parameter. He imposed conditions similar to those of Koenderink, and derived a continuous family of discrete convolution kernels.

In the discrete case, a hierarchical representation can also be created by repeatedly filtering and sub-sampling an image. In this case, an image on a grid of size $2^k \times 2^k$ is represented on a family of grids of size $2^{k-i} \times 2^{k-i}$ ($i = 0, \ldots, k$). Each pixel in one of these grids corresponds with four pixels in the grid beneath it. Such a structure is called a *pyramid*. Although the pyramid in which the image dimensions are reduced by a factor 2 in each step is used most commonly, other possibilities exist. An overview is given by Kropatsch [56].

If each pixel contains the average grey value of the four corresponding pixels in the next lower level, the result is the pyramid which was originally proposed by Tanimoto and Pavlidis [100]. This pyramid is sometimes called a *quad-tree*.

A different type of pyramidal image description was introduced by Burt and Adelson [17]. Their pyramid also consists of a stack of square images, whose dimensions are reduced by a factor 2 in each step. In stead of storing the average value of four pixels in a level in a cell on the next level, they compute a weighted average over a larger number of pixels. The weights are chosen such that the images in higher levels of the pyramid approximate the convolution of the original image with a Gaussian. Therefore, this pyramid is called the *Gaussian pyramid*.

The *Laplacian pyramid* is computed from the Gaussian pyramid by subtracting subsequent levels, after a suitable interpolation of the higher level. The resulting levels can be interpreted as band-passed filtered versions of the original

* The Laplacian is the differential operator $\partial^2/\partial x^2 + \partial^2/\partial y^2$.

image. This allows for the interpretation of the levels of the Laplacian pyramid as the description of the structure which is present in an image *at a given scale.*

The successive levels of a pyramid need not be computed through a linear filter. Another possibility is the use of morphological filters, in combination with a suitable sampling scheme [42, 102]. Subtracting adjacent levels in the morphological pyramid also produces images which can be interpreted as the structure present at a given scale.

Pyramidal structures can also be used for the representation of curves. In this case, each cell in a level of a pyramid represents a square region in the base level of the pyramid, which corresponds to the image plane. The cells do not store the exact shape of the curve part within this square region, but only a condensed representation. This corresponds to the averaging process in the grey scale pyramid. In the chain pyramid due to Meer [70], each curve part represented in a cell is linked to curve parts in adjacent cells, yielding a doubly linked list in each level of the pyramid. In the most simple form of the chain pyramid, the cells carry no extra information and all the information is represented in the structure of the doubly linked list. If each cell carries the centroid of the curve segments it represents as an attribute, a smoothed version of the curve can be obtained by connecting these centroids. In the curve pyramid by Kropatsch [55], the information carried by each cell in the pyramid consists of the sides of the cell through which the curve enters and leaves the cell. Kropatsch used a $2 \times 2/2$ pyramid, in which the length of a curve description does not increase with the description level.

In the examples of size based hierarchies discussed so far, problems can occur if a structure in an image has more than one characteristic size. Consider for example an elongated object in a grey scale image. If this image is blurred, the object will disappear when the width of the convolution kernel is of the same order of magnitude as the *width* of the object. Yet, the *length* of the object seems to be just as good a measure for the object size.

1.4. Primitive-Based Hierarchies

In the previous section, we discussed hierarchical methods in which the levels of the hierarchy have a regular spatial structure. In this section, we will discuss some hierarchical methods in which the levels of the hierarchy are not fixed, but constructed adaptively to the image contents.

The descriptive elements used in these methods are *primitives,* which represent some aspect of the spatial structure in an image. As the primitives in the higher levels of the representation are not ordered on a regular grid, a graph formalism is natural for the representation of their arrangement: the primitives in each level are the vertices of a graph, while their interrelations are described by edges. Building a hierarchical representation using the graph formalism yields a *hierarchical graph.* An important application area for graph representations is *image representation.*

1.4.1. Primitives and Attributes

A *primitive* is defined as a condensed representation of the spatial structure present in a localized part of the image. The motivation for the choice of the phrases '*condensed* representation', 'representation of *spatial* structure' and 'structure in a *localized* part of the image' is as follows.

A condensed representation is used in stead of the raw image data because we want to retain only relevant information. When primitives are used to build a hierarchical image description, the use of condensed information enables the description of image structure at progressively coarser scales or at progressively higher abstraction levels.

For many image analysis applications, the precise values of the grey value at a given location is far less important than the geometric arrangement of the grey value pattern. In fact, in many situations no calibration of the grey value scale is performed, such that attributing a very precise meaning to a given grey value is not possible. Therefore, in this thesis emphasis is laid on the geometric or spatial structures in images.

The choice for localized primitives enables the construction of a hierarchy through aggregation of primitives which are spatially close to each other. Primitives can for example correspond to a point in the image, a line segment or a region.

A primitive can possess *attributes*, which provide a specification of some of its properties. The attributes of a region can be its area and average grey value or some other grey value statistics. A line segment or curve can carry its length, center of mass or the response of an edge detection filter along its outline, as an attribute.

In the methods to be described, primitives are handled through their attributes. Therefore, the attributes should represent all the properties which are essential for the application at hand. The extent to which a primitive is described by its attributes can vary. A region usually does not carry its shape as an attribute, so it cannot be reconstructed from its attributes. On the other hand, a line segment which carries its end points as it attributes is determined completely by them.

The purpose of the description of an image by a set of primitives is to make relevant information explicit and accessible and to discard irrelevant information. Therefore, reconstruction of the original raw image data from the describing primitive set is not always possible. What should be possible is the reconstruction of the image contents up to a given level of abstraction, depending on the application at hand. Perhaps the simplest illustration of this concept is found in optical character recognition. An optical character recognition system converts printed or hand-written text into an ASCII representation. All information on the visual appearance of the document is lost, but the information of the text is retained. It is possible to synthesize an image from an ASCII representation by printing it in a fixed font, resulting in a standard visual appearance.

This thesis presents a number of primitive-based hierarchical image descriptions. The development of such a description requires the choice of a primitive type, a method for extracting primitives from the image, and a way of aggregating primitives in order to generate more condensed descriptions. In this thesis, the construction of a hierarchy and the analysis of image structure go hand in hand. In the case of, for example, curve approximation (section 4.3), pairs of line segments are replaced by single, approximating line segments. This process reflects both the construction of a hierarchy and the piecewise linear approximation of a curve.

In this thesis, all levels of the hierarchy contain primitives of the same type. The aggregation of a group of primitives in a lower level produces a primitive of the same type in a higher level of the hierarchy, providing a more condensed representation of image structure. This approach is different from the one taken in structural pattern recognition [30], where primitives occur only in the lowest level, and hierarchical (treelike) structures composed of these primitives occur in the higher levels.

The primitive types to be used must enable the construction of a hierarchy. This means that they must have a meaning at all levels of abstraction and that they must permit aggregation into more condensed primitives of the same type. Suitable primitives are regions, line segments and curves. Using one of these primitive types, an input image can act as the lowest level of the hierarchy. If regions are used as primitives, pixels can be seen as (degenerate) regions. If the primitives are curves, the boundaries between pixels can be treated as (degenerate) curves.

In such an approach, corners are not suitable primitives. They can be present in every level of the hierarchy, but they cannot be aggregated, because a corner in a curve cannot be thought of as being composed of a number of smaller corners. It is of course possible to extract corners from a hierarchical description by curve parts, as was shown by Fermüller and Kropatsch [28].

If a grey scale image is considered as the height map of a mountain area, various primitives such as peaks, ridges and valleys can be detected. Sometimes, the development of such primitives is traced as the image is blurred [59], but it is also possible to detect these primitives in the original image and then manipulate with the primitives themselves [32]. This type of analysis can be performed on the original grey scale image, but is often applied to a gradient image. In mathematical morphology, a wide variety of segmentation algorithms, all based on the watersheds of a gradient image, has been developed [72]. The detection of ridges can also be performed on distance transformed images, yielding the skeleton [5]. Similar techniques have been used for multi-resolution terrain modeling in order to obtain elevation databases for flight simulators, e.g. Scarlatos and Pavlidis [91]. Topographic primitives such as peaks and ridges have the advantage that their definition does not require the selection of a 'magic number', such as a grey value threshold. On the other hand, they may not be robust in the sense that a small change in an image can cause a large change in its topographic structure. When topographic primitives are used, much information on the distance between various primitives is lost. Especially the instability of the skeleton is notorious: a small bump or indentation in the object border may cause the creation of long branches in the skeleton.

1.4.2. Hierarchies of Graphs

Every level of a hierarchical description consists of a number of primitives. Relations between these primitives can be defined implicitly by their arrangement on the image plane but they can also be represented in a more explicit form. An appropriate formalism for the description of interrelations between primitives is based on *graphs* [36]. Primitives act as the vertices of a graph and their interrelations are represented by edges. The edges of the graph can carry attributes, providing a specification of the relation between primitives. A graph will be denoted as $G = (V, E)$, where V is the set of vertices and E is the set of edges.

A classical example of a graph representation of a set of primitives and their relations is the region adjacency graph, which is illustrated in figure 1.1. Such a graph represents a partition of the image plane in a number of regions. A region is a connected subset of the image plane. Each region is represented by a vertex; vertices representing adjacent regions in the image plane are connected by an edge. The attributes of each region can be its area and average grey value; the attribute of an edge can be the length of the boundary between the corresponding pair of adjacent regions. The main application area for region based representations is in image segmentation, but they have also been used for object recognition [27].

When a hierarchical image representation is built from a number of primitives, embedded in a graph, the resulting structure is a hierarchy of graphs, as shown in figure 1.2. Each level in such a representation is a graph, in which primitives are used as vertices and the edges represent interrelations between them. Each primitive in a higher level in the graph comprises a number of

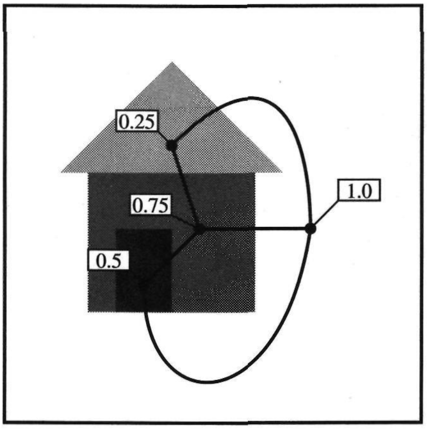


Figure 1.1: An image and its region adjacency graph, in which each vertex carries the grey value of its region as an attribute.

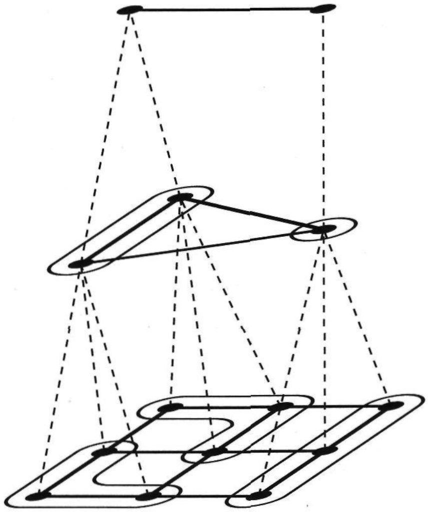


Figure 1.2: An example of a hierarchy of graphs, consisting of three levels. Edges in each level are indicated by solid lines and parent–child relations by dashed lines. The groups of children of each parent are encircled; these groups are represented by their parent on the next level.

primitives in the level below. The primitives in the lower level of an image description which are represented by a single primitive in the higher level, are called the *children* of that primitive; the higher level primitive is called the *parent*. The parent-child relations impose a tree-like structure on the vertices of the hierarchy of graphs. Formally, a hierarchy of graphs is defined as follows.

definition 1.1 *A hierarchy of graphs is a sequence* $(G_0, \ldots, G_n)$ *of graphs and a sequence* $(\pi_0, \ldots, \pi_{n-1})$ *of mappings* $\pi_i : V_i \rightarrow V_{i+1}$ *such that:*

(1) for $i = 0, \ldots, n-1$, $\pi_i(V_i) = V_{i+1}$ *;*

(2) for each $i = 0, \ldots, n-1$ *and each* $x \in V_{i+1}$, $\pi_i^{-1}(x)$ *is a connected subset of* G_i *;*

(3) for each $i = 0, \ldots, n-1$ *and* $x, y \in V_{i+1}$, $(x, y) \in E_{i+1}$ *if and only if there are* $x' \in \pi_i^{-1}(x)$ *and* $y' \in \pi_i^{-1}(y)$ *such that* $(x', y') \in E_i$.

For $x \in V_i$, the vertex $\pi_i(x) \in V_{i+1}$ is the parent of x; the vertices in $\pi_{i-1}^{-1}(x)$ are the children of x.

It is possible to discern bottom-up or data-driven image analysis methods and top-down or model driven image analysis methods. Bottom-up methods process image data in a uniform manner and apply general methods in order to aggregate data into primitives of gradually higher levels of representation. Top-down methods use application-specific knowledge, often in the form of object models. Such a model is then searched for in the image data. The image data is not processed uniformly, but goal-directed, in order to find support for or against the presence of a particular structure in the image.

This thesis is mainly concerned with bottom-up methods in which the same type of primitive is used in every level of the hierarchy. As such methods do not employ domain specific knowledge, they have a large potential application area. The choice for a single primitive type for all levels of the hierarchy allows for a uniform treatment of structures at different scales.

The applicability of bottom-up methods is restricted by the fact that not all types of knowledge or expectations on image contents can be used in a bottom-up context and because "errors" made during the construction of the hierarchy can not be repaired later. In this thesis, such limitations will show; in chapter 6, it will be indicated how a hierarchical representation, built in a bottom-up fashion, can be processed top-down in order to extract relevant image information and to remove artifacts created in the bottom-up pass.

1.4.3. Image Segmentation

Image segmentation is a poorly defined problem. There is general agreement [35, 45, 112] on the fact that a segmentation is a partition of the image plane such that each part satisfies some homogeneity criterion and such that the union of two adjacent parts does not. This definition, however, does not determine a unique segmentation for a given homogeneity criterion. Moreover, segmentation methods are often judged on subjective criteria, such as the extent to which they produce the same result a human would when segmenting an image.

Many approaches to image segmentation have been proposed (see Haralick and Shapiro [34] or Pal and Pal [82] for a review). Some techniques, such as those based on thresholding or edge detection, do not use regions during the segmentation process, but produce them only as a result. Other techniques do manipulate regions during the segmentation process. In such approaches, a set of regions is operated on by merging and/or splitting procedures, until a satisfactory segmentation is obtained. Although the goal of image segmentation is usually a single partition of the image plane, and not a hierarchical description, the hierarchical approach is very useful.

As it is not possible to search the space of all possible image partitions, an efficient search can be made by repeatedly splitting and merging regions. The region merging scheme of Brice and Fennema [15] can be seen as a hierarchical method *avant la lettre*. In this method, each connected component of the image grid which has a uniform grey value, is used as an initial region. Then regions are merged, until maximal regions are formed which still satisfy a homogeneity criterion. The application of a uniform merging criterion for regions varying between the small seed regions up to the final regions is typical for hierarchical methods.

The split-and-merge segmentation scheme of Horowitz and Pavlidis [45] starts with an initial segmentation of the image. Then, regions which do not satisfy a homogeneity criterion are split recursively, while groups of regions are merged if the resulting region satisfies the homogeneity criterion. After the split-and-merge procedure, merging of the remaining regions must be performed in order to remove artificial boundaries.

Another region based segmentation technique is segmentation by relinking, first described by Burt, Hong and Rosenfeld [18]. It manipulates a regular pyramid, in which each cell represents some region in the image plane. This region is not fixed, but depends on a set of *parent-child links* which assign to each cell in the pyramid (a child) one cell (its parent) in the next level. The parent-child links define a tree structure in which the pixels in the input image are the leaves. Each cell defines a subtree which has this cell as a root; and the cell is considered to represent the region in the image which consists of the leaves of this subtree. These parent-child links are updated iteratively, until the regions represented by the cells of some high level in the pyramid represent a suitable segmentation.

1.5. Morphological Approach to Image Analysis

The second class of image analysis methods discussed in this thesis is multiscale mathematical morphology. Mathematical morphology [95, 66] is a theory for image analysis based on set theoretic notions such as inclusion, union and intersection. Morphological operations require the selection of a probe called a *structuring element.* An image is analyzed by inspecting its interaction with the structuring element as this is moved around the image plane. Both the choice of the structuring element and the operation being performed with this structuring element determine the effect of a morphological operator. The structuring element acts as a kind of elementary building block. Image analysis using mathematical morphology usually consists of two steps: a transformation step, in which one or more morphological operations are performed on an image, extracting some geometric structure from the image, and a measurement step, in which certain geometrical characteristics of the transformed image (such as its area) are measured.

Initially, mathematical morphology was constructed for binary images, but it can also be applied to grey scale images and recently, it has been extended to the abstract framework of complete lattices [40].

Among the translated and scaled versions of a convex structuring element which are included in a binary image, there are maximal ones. An image can be considered as the union of maximal structuring elements contained in the image. These maximal structuring elements can be represented by their location and size. Decomposing an image as a union of maximal structural elements is a way of making global information local: a maximal structuring element covers a part of the image plane, while it can be represented by its center and size.

In order to provide an axiomatic treatment of size, Matheron [66] defined a class of morphological filters called *granulometries*. Such filters successively remove all image subparts smaller than some minimal size. As will be shown in chapter 9, granulometries are strongly connected with the representation of an image as the union of a collection of maximal spheres. Filtering an image by computing the union of its maximal structuring elements which exceed a given minimal size is an example of a granulometry.

For morphology on binary images, the structuring element is often chosen to be a sphere, but other choices are possible. For grey scale morphology, so-called *flat* structuring functions, which are constant on their support can be used, but other structuring functions such as the parabola have specific advantages [10].

1.6. Overview of this Thesis

In this thesis, several hierarchical image analysis methods using primitives are presented. This section gives an overview.

The chapters 2 through 6 present a number of methods for line detection and image segmentation, based on the recursive aggregation of small primitives into larger ones. The methods vary in the amount of structural information present in the image representation they use, and in the way groups of primitives which are to be aggregated, are computed.

Chapter 2 describes the detection of linear structures in an image by clustering short line segments. These line segments act as the primitives, which carry their end points as their attributes. Using a stochastic model of image formation, a metric on the set of all primitives is derived; this measures the probability that two primitives originate from the same structure in the scene. The distances between primitives defined by this metric define the relations between the primitives; no explicit representation in the form of a graph is used. The spatial arrangement of primitives in the image plane is used indirectly, as the metric takes the distance of primitives in the image plane into account.

Chapters 3 through 5 describe a number of image segmentation methods. The segmentation methods to be presented in this thesis all use hierarchies of region adjacency graphs as the image representation. They differ in the way the groups of regions to be aggregated into a single region are selected and in the type of information used in the process.

In chapter 3, groups of regions are formed by selecting a number of characteristic regions and assigning other regions to them. Each group consisting of a characteristic region and the regions assigned to it, is aggregated into a single region. The structure of the region adjacency graph is used here in two ways. Firstly, it is used to enforce the connectedness of all regions. Secondly, a cost function on paths in the graph is used for the computation of the assignments of regions to characteristic ones. A hierarchy is built by repeated application of the grouping procedure.

Chapter 4 presents a method for building a hierarchical description of an image by repeated aggregation of adjacent primitive pairs in a graph. The method can be used for arbitrary primitive types and is demonstrated both with line segments (for polygonal curve approximation) and with regions (for image segmentation).

The edges in the graph play an important role here: vertices can only be merged into a single primitive if they are connected by an edge. A homogeneity model for primitives is used to label each edge with a merge score which describes how well the merge of the two primitives attached to that edge would match the model. Thus, the measure used in this chapter evaluates the 'quality' of a condensed primitive *after* the aggregation, rather than the similarity of primitives *before* their aggregation.

The edge labels are used to guide the selection of pairs of primitives to be merged. Merges in the graph are independent and can therefore be performed in parallel. The corresponding action on a graph is called a *parallel graph contraction* [89].

In chapter 5, a hierarchical image segmentation method based on relinking is presented. The classical relinking method of Burt *et al.* [18] is adapted in such a way that connected regions are obtained. This requires the application of the hierarchy-of-graphs formalism. Relinking can be used in a completed hierarchy as well as during the construction of the hierarchy, each time a new level is created.

Then, the edges of the region adjacency graph are used to represent, in an explicit form, the boundaries between regions in the image. This is used in order to combine region and boundary information in the segmentation process.

In this chapter, edges are used in two ways. Firstly, they are used for the enforcement of connectivity of regions during the relinking procedure. Secondly, they are used for the representation of boundaries between regions in the image.

Chapter 6 presents a number of top-down operations which can be performed on the hierarchies that were described in chapter 5. Such operations are required in order to extract information from the hierarchical description, especially if the bottom-up construction phase has led to artifacts. If a model for image contents is available, this can be used to guide a top-down procedure. As an example, we present two methods for the detection of cells in muscle tissue. The first method performs a top-down search for combinations of regions which are convex and fall within a given size range. The second method performs a top-down refinement of region contours by using dynamic programming for detecting optimal paths in different levels of the hierarchy. This yields a coarse-to-fine approximation of object contours.

Chapters 7 through 9 discuss a number of basic and advanced operators in mathematical morphology, using chamfer discs as structuring elements. The notion of size is defined through the choice of a parameterized set of structuring elements. By performing morphological operations with such a set of structuring elements, a size-dependent analysis of image structure can be performed.

It will be argued in chapter 7 that the set of spheres defined by a so called chamfer metric is an attractive set of structuring elements, because they are good approximations of Euclidean discs on the discrete grid. This chapter describes a number of properties of chamfer metrics and is meant as a preparation for the following chapters.

The medial axis is closely related to the description of an image as a union of maximal structuring elements. Chapter 8 describes the characterization and computation of the medial axis defined by chamfer metrics.

The representation of an image as a union of maximal structuring elements leads to various questions concerning the distributions of the locations and the radii of these structuring elements. Size distributions can be seen as operations which filter an image by retaining only the image subparts which are larger than some minimal size.

The opening transform is a useful tool for the computation of a size distribution. This transform, which maps a binary image onto a grey scale image, will be discussed in chapter 9. There, we will also point out the relation with the pattern spectrum [61].

The last chapter of this thesis is chapter 10, in which the conclusions of the research described in this thesis are presented. Some open problems, loose ends and possible directions for further research are mentioned.

2

Clustering of Line Segments Using a Metric

2.1. Introduction

This chapter describes the extraction of image structure by progressive aggregation of primitives. As an example, the analysis of linear structures is discussed in detail. The primitives used are described by parameters in some space U, which is a manifold.

A metric is constructed, which measures how well two line segments can be merged into a single line segment. This metric is used in a simple clustering algorithm in order to extract linear structures from an image containing many small line segments.

The detection of linear structure in images is an important task in computer vision. In order to detect a large linear structure in an image, information from many locations must be combined. Many authors, like Nevatia and Babu [79] or Burns, Hanson and Riseman [16] have described the detection of lines in grey scale images. For each location in the image, they calculate the magnitude and the orientation of the gradient. A large value of the magnitude of the gradient at some location in the image is an indication for the presence of a line in the direction perpendicular to the direction of the gradient at that location. Information from different locations in the image is combined to find a large number of locations in the image where support for the presence of a line can be found.

In stead of looking at grey scale images, it is also possible to look at images which contain a number of short line segments. These line segments can be the output of an edge detection and line segment fitting algorithm, or they can be derived from a line drawing. A line segment is not considered as a group of collinear pixels in the image, but as a separate geometric structure, defined for example by its end points or by its center, length and orientation. This semi-symbolic representation of a line segment is the primitive with which the image representation presented in this chapter is built.

In many cases, short line segments lie along long straight lines. As these longer lines are the relevant structures in an image, information from different line segments must be combined in order to find the large linear structures.

One way of combining information in this case is the Hough transform [25]. In the Hough transform, all possible lines in the image plane are considered. For each line, the image is inspected in order to detect how strongly the image data support the presence of this line. As all possible lines must be considered, memory requirements for this technique are high. The performance, both in computational efficiency and in quality of the resulting image description, can be improved using a hierarchical technique [86].

Other techniques for detecting linear structures in images are based on grouping [9, 90]. In grouping techniques, linear structures are detected by replacing a number—often two—of short line segments by a single longer one, and repeating this until the large linear structures in the image are found. Such techniques usually consist of two independent parts. First it has to be determined which pairs of line segments will be (or may be) replaced by a single longer one. This "groupability" of two line segments depends for example on their collinearity and on the distance between their end points. Then, the actual replacement must be carried out. The detection of groups of line segments which are to be replaced is often based on a number of thresholds: the difference in orientation must not exceed a given value; the distance between the end points of two segments must not exceed a given value (which possibly depends on the length of the line segments), etc. Such criteria produce binary relations between line segments.

Scher *et al.* [92] describe the grouping of line segments on a single known carrier line into larger line segments. They introduce an evaluation function which measures how well the group of line segments being evaluated can be replaced by a single line segment. This evaluation function has a high value for groups of line segments with small gaps between them and for large groups of line segments. The evaluation function is used in various grouping schemes. The clustering method of Scher *et al.* can not be used to detect linear structures in two dimensional images, because the line segments must lie on a single, known carrier line. In the two dimensional case, such a carrier line is not known a priori and line segments which form a linear structure in an image do not lie exactly on a single carrier line, due to noise.

The purpose of this chapter is the construction of a metric on the set of all line segments which measures how likely it is that two segments are part of the same larger linear structure in the image. This depends on various aspects of the geometrical relation between two line segments, such as the distance between their end points and the collinearity of the segments. The metric expresses the geometrical relation in a single number.

There exist measures for different aspects of the geometrical relation between two line segments, but they do not describe all aspects. Collinearity, for example, can be measured as the average distance of the four end points of two segments, to a line fitted to those four end points. This is a good measure for collinearity but it does not depend on the distance between the segments. The distance between end points, on the other hand, is a good measure for the nearness of two segments, but it does not depend on their collinearity.

The appropriate type of measure to determine whether two line segments belong to a single linear structure is a metric. This is the case because of the transitivity of the collinearity relation. If two line segments A and B belong to a single linear structure and B and a third line segment C belong to a single linear structure as well, then A and C belong to a single linear structure. This is expressed by the triangle inequality, the essential characteristic of a metric. Transitivity does not only hold for the collinearity of line segments, but also for other relations such as the cocircularity of arc segments. There are, however, also geometrical relations for which transitivity does not hold, for example for line segments being sides of the same square.

In this chapter, a metric on the set of line segments will be constructed in a number of steps. These are subsequently the choice of a parameter space U, describing the set of all line segments in the plane, the association of each line segment in the plane with a function—which will be called the neighborhood function—on this space U, and the definition of a metric on the set of all square-integrable functions on U. The construction is based on well known mathematical techniques. It can be seen that a suitable choice of the neighborhood functions will indeed produce a metric which measures the groupability of line segments.

The metric described in this chapter can be used in a variety of clustering algorithms [48]. A numerical description of the relation between line segments provides a larger variety of choice of clustering algorithms than a binary description. In this chapter, a very simple grouping technique is used to verify the usefulness of the metric.

The rest of this chapter is organized as follows: In section 2.2, some general aspects of metric spaces are discussed and some examples are given. Section 2.3 describes how neighborhood functions can be used to construct metrics on the set of all primitives of a given type. The construction is illustrated by two simple examples. The metric for lines segments is discussed in section 2.4. In section 2.5, the usefulness of the metric is verified by using it in a simple line segment grouping scheme. Section 2.6 presents the conclusions of this chapter.

2.2. Metrics

Before the construction of metrics on sets of primitives is discussed, some general remarks on metric spaces are made. $\mathbb{R}_{\geq 0}$ is defined as the set of non-negative reals.

Definition 2.1 *A metric space is a pair (X, d) where X is some set and d is a function $X \times X \to \mathrm{IR}_{\geq 0}$ such that for all x, y and z in X:*
(1) $d(x,y) = 0 \Leftrightarrow x = y$.
(2) $d(x,y) = d(y,x)$.
(3) $d(x,y) + d(y,z) \geq d(x,z)$.

The third axiom in the definition of a metric is called the triangle inequality. It is this inequality that is not satisfied in many of the measures proposed in literature.

Let Y be a subset of X, where (X, d_X) is a metric space. Then Y can be provided with a metric in a straightforward way: for each pair $(y_1, y_2) \in Y \times Y$, the distance is defined as $d_Y(y_1, y_2) = d_X(y_1, y_2)$. This metric on Y is called the *induced* metric.

If X is a vector space provided with a norm $\|\cdot\|$ [23], then X can be provided with a metric. This metric is defined by $d(x,y) = \|x - y\|$, for all $x, y \in X$.

The set of all primitives of a given type, such as line segments, can be parameterized by some parameter space U. The dimension of this parameter space is equal to the number of degrees of freedom which each primitive has. Some parameters may have values in IR, but some parameters, for example those describing orientation, can have values only in a more limited range, like $[0, \pi)$. Therefore the parameter space U can have a structure which is more complicated than the structure of IR^n. In many cases, the parameter space corresponding with a set of primitives will be the Cartesian product of the spaces in which the different coordinates can assume values. Such spaces can be IR for position coordinates, $\mathrm{IR}_{>0}$ (the set of positive reals) for lengths and radii, or the unit circle for direction coordinates.

The construction described in this chapter requires a metric for measuring the distance between two functions defined on U. Actually, this metric will not compare any two functions described on U, but just square integrable ones, i.e. functions for which $\int_U f^2(t)\mathrm{d}t$ exists and has a finite value. This set of functions is called $\mathcal{L}^2(U)$. (Actually, instead of using functions, one should use equivalence classes of functions, defined by the equivalence relation $f \sim g \Leftrightarrow \int_U (f(t) - g(t))^2 \mathrm{d}t = 0$. In this chapter only continuous functions are used, so this has no practical consequences. For details, the reader is referred to Dieudonné [23])

$\mathcal{L}^2(U)$ is a vector space. An inner product on this vector space is defined by

$$\langle f|g\rangle = \int_U f(t)g(t)\mathrm{d}t. \tag{2.1}$$

This inner product induces a norm by $\|f\| = \sqrt{\langle f|f\rangle}$. The corresponding metric is given by $d(f,g) = \|f - g\|$, that is

$$d(f,g) = \left[\int_U (f(t) - g(t))^2 \mathrm{d}t\right]^{1/2}. \tag{2.2}$$

The subset $\mathcal{S}(U) = \{f \in \mathcal{L}^2(U) \mid \|f\| = 1\}$ is called the unit sphere in $\mathcal{L}^2(U)$. From now on $\mathcal{S}(U)$ will be provided with the induced metric. Because all functions in $\mathcal{S}(U)$ have norm 1, we have for all $f, g \in \mathcal{S}(U)$:

$$d^2(f,g) = 2 - 2\int_U f(t)g(t)\mathrm{d}t. \tag{2.3}$$

Note that $d(f,g)$ decreases as $\int_U f(t)g(t)\mathrm{d}t$ increases.

The construction of a metric for comparing primitives such as line segments is performed in a number of steps. Each line segment u is identified with a function f_u in $\mathcal{S}(U)$. As $\mathcal{S}(U)$ is a subset of $\mathcal{L}^2(U)$, the set of all line segments is identified with a subset of $\mathcal{L}^2(U)$. Then $\mathcal{L}^2(U)$ is provided with the metric as described above and the set of all line segments is provided with the induced metric.

2.3. Metrics on Sets of Primitives

Let us now return to our original problem. Let X be the set of all primitives of a given type, for example line segments. Assume that the primitive set can be parameterized by points in a parameter space U. Line segments can, for example, be parameterized by coordinates $(x, y, \theta, l) \in \mathbb{R}^2 \times [0, \pi) \times \mathbb{R}_{>0}$. The point (x, y) is the center of the line segment, the coordinate θ indicates that the direction of the line segment is $(\cos\theta, \sin\theta)$ and l is the length of the line segment.

If U is embedded in some $\mathbb{R}^n$, the simplest conceivable metric on U is the one induced by the Euclidean metric on $\mathbb{R}^n$. This metric, however, does not take into account the geometric properties we want to measure. For the example of line segments, the Euclidean distance between two line segments $(x_1, y_1, \theta_1, l_1)$ and $(x_2, y_2, \theta_2, l_2)$ is $[(x_1 - x_2)^2 + (y_1 - y_2)^2 + (\theta_1 - \theta_2)^2 + (l_1 - l_2)^2]^{1/2}$. The distance between two line segments with the same direction and length depends only on the Euclidean distance between the two center points. Thus, the induced metric is not sensitive to collinearity of line segments, a property which we want to measure. In general, the metric induced by an embedding of the parameter space U in some Euclidean space $\mathbb{R}^n$ does not work well, because it does not take into account the geometrical interpretation of the parameters for the primitives they describe.

If primitives are parameterized by coordinate vectors $(x_1, \ldots, x_n)$, their nearness can also be defined using thresholds. If a set of thresholds θ_i is defined, two primitives x and y are said to be related if $|x_i - y_i| < \theta_i$ for all i. In relation to this approach, a metric d can be defined as $d(x, y) = \max_i |x_i - y_i|/\theta_i$. Then two primitives are related if their distance is less than 1. This metric suffers from the same problems as the metric induced by the Euclidean metric, since the geometric interpretation of the coordinates is not taken into account. It would, for example, be nice to allow a larger deviation of the angles of short line segments, because the measurement of the direction of line segments is more uncertain for short line segments than for long ones.

We introduce a metric which overcomes the objections raised above. It takes into account the geometrical interpretation of the different parameters and their interactions. This is done by letting the metric depend on the model of image formation. This metric is constructed by embedding the parameter space U in the function space $\mathcal{S}(U)$ in a very specific way. Then the metric on $\mathcal{S}(U)$ induces a metric on U. The basic idea is to associate with each point $u \in U$ a so called neighborhood function. This neighborhood function must satisfy $f_u \in \mathcal{S}(U)$ and $f_u(t) \geq 0$ for all $t \in U$. Neighborhood functions are chosen in such a way that different points in U have different neighborhood functions.

As a simple example, consider the case where the primitive set is the set $\mathrm{I\!R}$ of real numbers. Let the neighborhood function of u be the indicator function of an interval of length 1 centered at u: $f_u(t) = 1_{[u-1/2, u+1/2]}(t)$. The primitive set has now been identified with a subset $\mathcal{F}(U)$ of $\mathcal{S}(U)$. $\mathcal{F}(U)$ is provided with the induced metric. In order to let d have a maximal value of 1, it is divided by $\sqrt{2}$. Equation (2.3) then gives:

$$d(u, v) = \left[1 - \int_U f_u(t) f_v(t) \mathrm{d}t\right]^{1/2}. \tag{2.4}$$

For the example mentioned above, it can easily be seen that

$$\int_{\mathrm{I\!R}} f_u(t) f_v(t) \mathrm{d}t = \max(0, 1 - |u - v|). \tag{2.5}$$

The metric thus defined on $\mathrm{I\!R}$ is

$$d(u, v) = \sqrt{1 - \max(0, 1 - |u - v|)} = \min(1, \sqrt{|u - v|}). \tag{2.6}$$

This metric is illustrated in figure 2.1a. Both $f_x(t)$ and $d(x, y)$ are translation invariant since $f_x(t) = f_0(x - t)$ and $d(x, y) = d(x - y, 0)$. The upper curve shows $f_0(t)$ and the lower curve shows $d(x, 0)$.

The neighborhood function has a nonzero value only on an interval of length 1. It can be seen that $d(x, y)$ has the maximal value 1 when x and y are so far apart that the intervals around x and y do not overlap. The larger the overlap of these intervals is, the smaller the value of the metric. The metric therefore measures the disjointness of the intervals, which can be interpreted as influence zones of the points x and y.

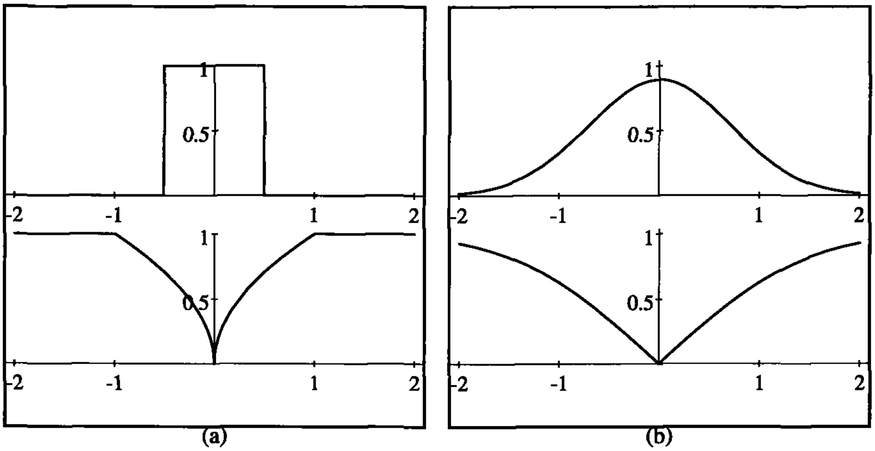


Figure 2.1: Two examples of metrics on IR. The upper curves show a neighborhood function $f_0(t)$ and the lower curves show $d(x, 0)$ for the metric defined by this neighborhood function.

Another metric on IR can be obtained by taking the Gaussian neighborhood function

$$f_u(t) = \sqrt[4]{2/\pi}\, e^{-(t-u)^2}. \tag{2.7}$$

The metric defined by this function is

$$d(u, v) = \sqrt{1 - e^{-(u-v)^2/2}}. \tag{2.8}$$

This metric is illustrated in figure 2.1b. There is again translation invariance, as $f_x(t)$ and $d(x, y)$ depend only on $x - t$ and $x - y$, respectively. The upper and lower curves show $f_0(t)$ and $d(x, 0)$. Note that the use of a Gaussian neighborhood function has the practical advantages that, firstly, the product of two Gaussian is again a Gaussian and, secondly, that the integral of a Gaussian over IR can be expressed in a closed form.

In this example, the neighborhood functions have a nonzero value everywhere. The metric now expresses the extent to which the neighborhood functions have different values at different locations of the parameter space IR.

Let us now return to the case of segments as primitives. The neighborhood function must be chosen in such a way that the metric it defines expresses the groupability of primitives. Therefore, the choice of a neighborhood function depends on how a structure in a scene gives rise to primitives in an observed image. Suppose that an image is created by measurement of some real structures in the scene. The measurement process is not perfect, so deformations occur. Let us choose the neighborhood function $f_u(t)$ to be the probability that a measured primitive u stems from measurement on a true structure t in the scene:

$$f_u(t) = P(T_{\text{true}} = t \mid T_{\text{meas}} = u). \tag{2.9}$$

This neighborhood function contains information on both the distribution of linear structures in the scene and on the characteristics of the measurement process. Using Bayes rule, we can write

$$P(T_{\text{true}} = t \mid T_{\text{meas}} = u) = \frac{P(T_{\text{meas}} = u \mid T_{\text{true}} = t)}{P(T_{\text{meas}} = u)} \times P(T_{\text{true}} = t). \tag{2.10}$$

The second term in the right hand side of (2.10) describes the probability distribution of structures in the scene, i.e. a priori world knowledge. No a priori knowledge is assumed, so this distribution is taken uniform. This yields an improper distribution (i.e. it integrates to ∞ in stead of 1), but as various factors will cancel in our calculations, this causes no problems. The denominator on the left hand factor describes the probability distribution of the primitives obtained by measurement. This distribution depends on the distributions of structures in the scene and on the image formation model. The numerator of the left hand factor describes the probability distribution of the primitives obtained by measurement, given the presence of a structures in the scene. Hence, this distribution represents the image formation model.

If it is assumed that different primitives in the image are generated by measurements of independent realizations of the distribution of structures in the scene, $f_u(t)f_v(t)$ is a measure for the probability that both u and v are generated by a measurement of the same structure t. Integrating this product over t gives a measure for the probability that both u and v are produced by measurements of the same structure in the scene. This integral can be regarded as a measure of nearness of the two primitives.

Thus, the choice of the neighborhood function depends on the underlying model for image formation. It is also possible to put a priori demands on the behavior of the metric (as we will see when the length of gaps between collinear line segments is discussed). These demands impose conditions on the neighborhood function and therefore on the image formation models which can be chosen.

In many cases, the relation between two primitives is invariant under the operation of some group G. In the case of line segments, for example, collinearity and nearness only depend on the relative positions of the primitives; they are invariant under translations and rotations of the whole primitive configuration. It is natural to choose neighborhood functions which are transformed into each other under the action of the group.

If G is transitive, only one neighborhood function needs to be defined; the others can be constructed from this function. This reduces the number of degrees of freedom in the choice of neighborhood functions. However, in many applications the number of symmetries will be so small that the symmetry group is not transitive. In the case of line segments, for example, there will be invariance under translation and rotation, but no scale invariance. Therefore, independent neighborhood function must be defined for each l_0. The neighborhood functions belonging to different x_0, y_0 and θ_0 are then fixed by these choices through the action of the group of Euclidean motions.

Note that the choice of a probability density function for $f_u(t)$ implies $\int_U f_u(t)\mathrm{d}t = 1$. The construction however required that $\int_U {f_u}^2(t)\mathrm{d}t = 1$ for all u. If we restrict to ***bounded*** probability density functions, the integral $\int_U f_u(t)\mathrm{d}t$ will be finite. If the symmetry group of the problem is transitive, the ratio between the integral of f_u and the integral of its square is fixed, as can be derived from the fact that all neighborhood functions can be transformed into each other by a simple coordinate transformation. If the value of $\int_U f_u^2(t)\mathrm{d}t$ has a value not equal to 1, $f_u(t)$ should be multiplied with some normalization factor.

If the symmetry group is not transitive, it may still be possible to choose the neighborhood functions in such a way that $\int_U f_u(t)\mathrm{d}t$ does not depend on u. In the example of line segments discussed in the next sections, the neighborhood functions will be chosen in such a way that they satisfy this condition.

Note that it is not necessary for the construction of the metric that the integral of f_u is constant. Yet it *is* required if a probabilistic interpretation is given to the metric.

It is also possible to take $f_u(t) = \sqrt{P(T_{\mathrm{true}} = t \mid T_{\mathrm{meas}} = u)}$, implying $f_u^2(t) = P(T_{\mathrm{true}} = t \mid T_{\mathrm{meas}} = u)$. The distance between u and v defined in this chapter is then equal to the so-called Hellinger distance [19] between the probability distributions $f_u^2(t)$ and $f_v^2(t)$.

2.4. The Metric on Line Segments

In the previous sections, the construction of a metric on a set of primitives has been presented. In this section, this procedure will be used to obtain a metric on the set of all line segments. The most important step in our construction is the choice of a model of image formation, leading to the choice of an appropriate neighborhood function.

If linear structures are computed by edge detection, followed by line fitting, distortions can occur. Lines can be interrupted or displaced from their original location and orientation. Additional processing is then necessary for the extraction of the complete linear structure in the scene.

Now let $u_0 = (x_0, y_0, \theta_0, l_0)$ be the coordinate representation of a line segment in the parameterization discussed previously. In order to define a metric, the neighborhood function $f_{u_0}(x, y, \theta, l)$ must be defined, which indicates how likely it is that a given measured segment u_0 stems from a measurement on a true linear structure with coordinates (x, y, θ, l).

Note that there is a desirable property of the metric which imposes an extra condition on the form of the neighborhood function. If two line segments lie on the same support line while they are separated by a gap which is large compared with their own lengths, the distance between them should be large, even if they could be caused by a single, very long, linear structure in the scene. If such a long structure is indeed present, the gap will be filled with other line segments, making a detection of the linear structure in the scene possible. In other words, it is assumed that distortions in the measurement process do not create large gaps in linear structures, but only short ones.

In order to implement this criterion, a neighborhood function must be chosen which does not extend very far in the direction of the line segment. Then, the length of a measured line segment gives no information on the length of the linear structure to which it belongs.

From equation (2.10) we have:

$$\begin{aligned} f_u(t) &= P(T_{\text{true}} = t \mid T_{\text{meas}} = u) \\ &= \frac{P(T_{\text{meas}} = u \mid T_{\text{true}} = t)}{P(T_{\text{meas}} = u)} \times P(T_{\text{true}} = t). \end{aligned} \tag{2.11}$$

The prior distribution of position and orientation of the linear structures in the scene is assumed to be uniform. Hence the second factor in the right hand side depends only on l_t. The local measurement property discussed above implies that the first factor does not depend on the length l_t of the linear structure in the scene. Hence we can take $f_u(t)$ of the form

$$f_u(t) = \psi_u(x_t, y_t, \theta_t)\phi(l_t). \tag{2.12}$$

There is no need to specify ϕ since

$$\int_U (f_{u_0} - f_{v_0})^2 \mathrm{d}\mu(t) = \int_{\mathbb{R}>0} \phi(l)^2 \mathrm{d}l \int (\psi_{u_0} - \psi_{v_0})^2 \mathrm{d}x\mathrm{d}y\mathrm{d}\theta \tag{2.13}$$

and $\int \phi(l)^2 dl$ is a constant, which can be disregarded when the values of the metric are compared.

We will now describe our model of image formation, leading to the choice of a neighborhood function. We will assume that the position and the orientation of a measured line segment are normally distributed around the true values for the linear structure in the scene. The neighborhood function of a point u_0 will have its maximum in the point u_0 itself and will decay towards 0 when moving away from u_0. Because position and orientation are normally distributed, the neighborhood function can be constructed using Gaussian functions, which depend on a single parameter (their width).

- The direction which is measured for the line segments can be corrupted. Therefore, a line segment should be "near" to other line segments at the same position but having a somewhat different direction. This is expressed by the width of the neighborhood function in the θ-direction (in U). The direction measured for a long line segment is a better indication for the direction of the underlying linear structure than the direction measured for a short segment. Therefore the width in the θ-direction of the neighborhood function of a short segment is larger than the width in the neighborhood function of a long segment.
- If a linear structure is broken into pieces, several short segments can result. The midpoints of these segments are displaced from the center of the original structure, in the direction of the linear structure. Therefore the neighborhood function of a line segment must have a relatively large width (proportional to the length of the measured line segment) in the direction in which the linear structure lies.
- A line segment can also be displaced from the linear structure in the direction orthogonal to the structure. The displacement in this direction will not be too large and does not depend on the length of the line segment.

The neighborhood function is now chosen as follows: For fixed θ, the function $\psi_{u_0}(x, y, \theta)$ is a two-dimensional Gaussian-like function centered at (x_0, y_0) with its long axis oriented in the θ-direction of the x-y–plane. The width σ_l in the θ-direction. i.e. the length direction of the line segment, is proportional to the length l_0 of the line segment. The width σ_w is fixed to some small value. The amplitude of the Gaussian-like function varies with direction: it is large for θ close to the direction θ_0, while it is small for values θ further away from the line direction θ_0.

Thus, the neighborhood function is described by:

$$f_{u_0}(x, y, \theta, l) = G_{\sigma_a(l_0)}(\theta - \theta_0) G_{\sigma_l(l_0)} \sigma_w (R_{\theta_0}^{-1}(x - x_0, y - y_0)) \phi(l) \quad (2.14)$$

Here $G_\sigma(x)$ is a Gaussian-like function of width σ:

$$G_\sigma(x) = \frac{1}{\sqrt[4]{\pi\sigma^2}} e^{-(x/\sigma)^2}. \tag{2.15}$$

G is a two-dimensional Gaussian-like function:

$$G_{\sigma_x\sigma_y}(x,y) = \frac{1}{\sqrt{\pi\sigma_x\sigma_y}} e^{-(x/\sigma_x)^2-(y/\sigma_y)^2}. \tag{2.16}$$

R_θ is a rotation over θ. As mentioned before, the factor ϕ cancels out in later calculations.

The symmetry group underlying this situation is the group of Euclidean motions of the image plane. There is no scale invariance, since the uncertainty in the orientation of a segment depends on its length. Note that, once the choice for Gaussian neighborhoods is made, there are only three degrees of freedom left: $\sigma_l(l_0)$, σ_w and $\sigma_a(l_0)$. Note that the values of these parameters depend only on the length l_0 of the line segment. The value of $\int_U f_{u_0}(t)\mathrm{d}t$ is

$$(2\pi)^{3/4}(\sigma_l(l_0)\sigma_w\sigma_a(l_0))^{1/2} \int_0^\infty \phi(x)\mathrm{d}x. \tag{2.17}$$

This value does not depend on u_0 if the product $\sigma_l(l_0)\sigma_w\sigma_a(l_0)$ does not depend on l_0. In the next section, the values of $\sigma_l(l_0)$, σ_w and $\sigma_a(l_0)$ will be chosen such that this condition is satisfied.

2.5. Experimental Results

To verify the usefulness of the metric defined in the previous section, a hierarchical clustering method based on this metric is examined. The clustering method used is a standard hierarchical clustering method [48]. It acts on a configuration of a number of primitives. The algorithm proceeds step by step. In each step, the two nearest primitives are replaced by a single one, such that a hierarchy of configurations with smaller and smaller numbers of primitives is created.

Let $\{s_1,\ldots,s_n\}$ be the set of all primitives in the original configuration. Let $L^{(k)} = \{s_1^{(k)},\ldots,s_{n-k}^{(k)}\}$ be the configuration in the k-th iteration step. $L^{(0)}$ is equal to $\{s_1,\ldots,s_n\}$. Successive levels are built by replacing the primitive pair having minimal distance by a single primitive:

$$L^{(k+1)} = \left(L^{(k)} - \{s_p^{(k)}, s_q^{(k)}\}\right) \cup \{c(s_p^{(k)}, s_q^{(k)})\}. \tag{2.18}$$

Here $s_p^{(k)}$ and $s_q^{(k)}$ are chosen such that $d(s_p^{(k)}, s_q^{(k)}) = \min_{i \neq j} d(s_i^{(k)}, s_j^{(k)})$ and $c(s_p^{(k)}, s_q^{(k)})$ is the *join* of $s_p^{(k)}$ and $s_q^{(k)}$, a line segment which replaces the original two segments.

The join of two line segments is defined as follows. First, the line is constructed which minimizes the mean squared distance of the four end points to the line. The join of the two segments is a segment of this line. The end points of the join are calculated by looking at the orthogonal projections of the two segments on the line. The join is the smallest segment of the line which contains both projections.

The clustering method is applied to a number of line segment configurations, derived from natural images. A Sobel gradient filter is applied to the original image. The edges in the image are detected by a peak filter. The edges detected by the peak filter are one pixel thick. Line segments are then fitted to the curves by connecting the first and the last point. If a curve is longer than a given number of pixels (in this case 6), the curve is cut into shorter pieces before fitting the line segments. In this way, line segments are fitted with reasonable accuracy to curves which contain bends or angles.

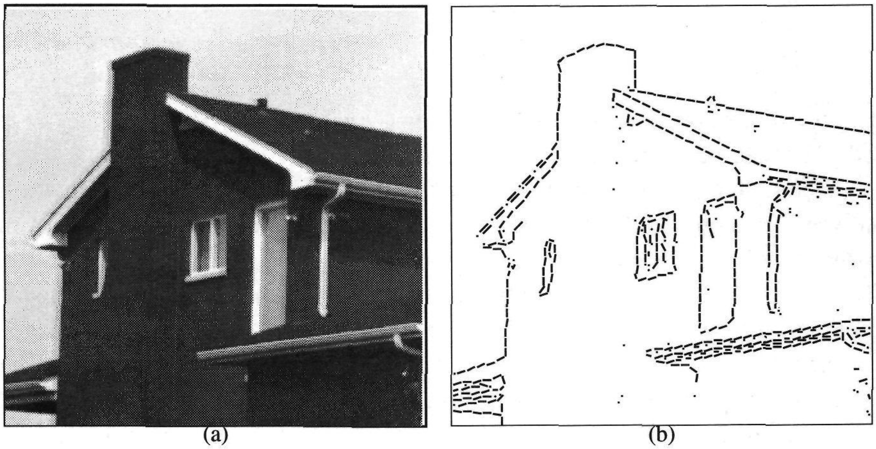


Figure 2.2: The house image and the extracted segments

We still have to choose the parameters $\sigma_l(l_0)$, σ_w and $\sigma_a(l_0)$. The following approach is taken to estimate a good set of values. Suppose that the end point of each line segment is displaced in the direction orthogonal to the underlying linear structure in such a way that the distance from the end point to the underlying line is uniformly distributed in the interval $[-D, D]$, for some constant D. It is assumed that the shifts in the position of the two end points are independent. The dislocation of the end points generates a sideways deviation of the center of the line segment and a distortion of the orientation of the line segment. In a first order approximation, the standard deviation of the orientation is $\sqrt{2/3}D/l_0$ and the standard deviation of the sideways displacement is $\sqrt{1/6}D$, where l_0 is the length of the line segment.

If these values are used for $\sigma_a(l_0)$ and σ_w, then $\sigma_l(l_0)$ must be linear in l_0 in order to guarantee that the product of the parameters is a constant. This implies that larger gaps can be bridged between long line segments, while only short gaps can be crossed between short line segments. Therefore, $\sigma_l(l_0)$ must be of the form Cl_0, for some constant C.

A cut-off $l_{\max}$ for the length l_0 of the line segment is used: line segments having a length larger than $l_{\max}$ have the same neighborhood function as line segments of length $l_{\max}$. This has two effects. Firstly, a minimal value $\sqrt{2/3}D/l_{\max}$ is introduced for the uncertainty in edge direction. Secondly, the spread in the length direction of the line segment is bounded by $Cl_{\max}$, thus avoiding that large gaps between line segments can be bridged.

The parameters for the metric are:

$$\begin{aligned}\sigma_l(l_0) &= C\min(l_0, l_{\max})\\ \sigma_w &= \sqrt{1/6}D\\ \sigma_a(l_0) &= \sqrt{2/3}D/\min(l_0, l_{\max})\end{aligned}$$

For most results presented here, the values $C = 1$, $l_{\max} = 40$ and $D = 2$ were used. The size of the images is 256×256.

Line segment clustering was performed for the house image 2.2. The initial configuration contained 509 line segments. Figure 2.3 shows the stages of the clustering process in which 250, 200, 150 and 120 segments are left. When 250 line segments are left, the linear structures begin to show quite clearly. When 150 segments are left, the detection of the linear structures in the image is more or less completed. At this stage, there are stray line segments left. When the clustering process is continued, unnatural groupings occur, which do not correspond to linear structures in the image. This can be seen in the picture where there are 120 line segments left.

It is therefore best to stop the clustering before undesirable clusterings occur. The distance at which such clusterings start to occur, depends on image contents. Therefore, it is not possible to prescribe a generally applicable distance threshold at which the process should be halted. Practical examples indicate that undesirable clusterings often occur if one of the two segments is a short noise segment which is not the result of previous clusterings, but has survived for a long time on its own without participating in the process.

In order to evaluate the noise sensitivity of the clustering algorithm, the algorithm is run on a number of synthetic images with a varying number of ad-

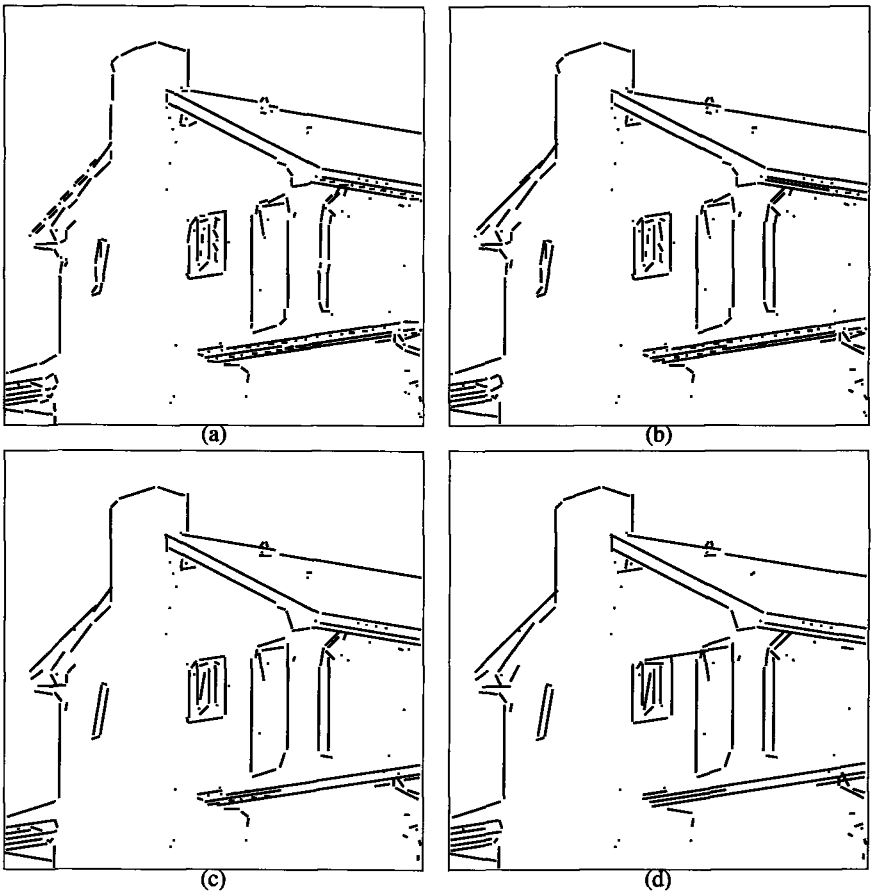


Figure 2.3: Stages in the clustering process with 250, 200, 150 and 120 segments left.

ditional noise segments. The signal consists of a number of short line segments forming a long horizontal line in the image. The positions and orientations of each of these line segments are distorted. The lengths of the line segments and of the gaps between them are distributed normally with an average of 6 and a standard deviation of 2. The noise consists of a varying number of line segments with uniformly distributed position and orientation.

The clustering process is continued until no line segment pairs with a distance smaller than 0.9998 remain. The results are shown in figure 2.4. In the result pictures, line segments which have not participated in the clustering process are not shown.

In all cases, the linear structure is detected. As the number of noise segment grows, the number of accidentally aligned noise segment grows as

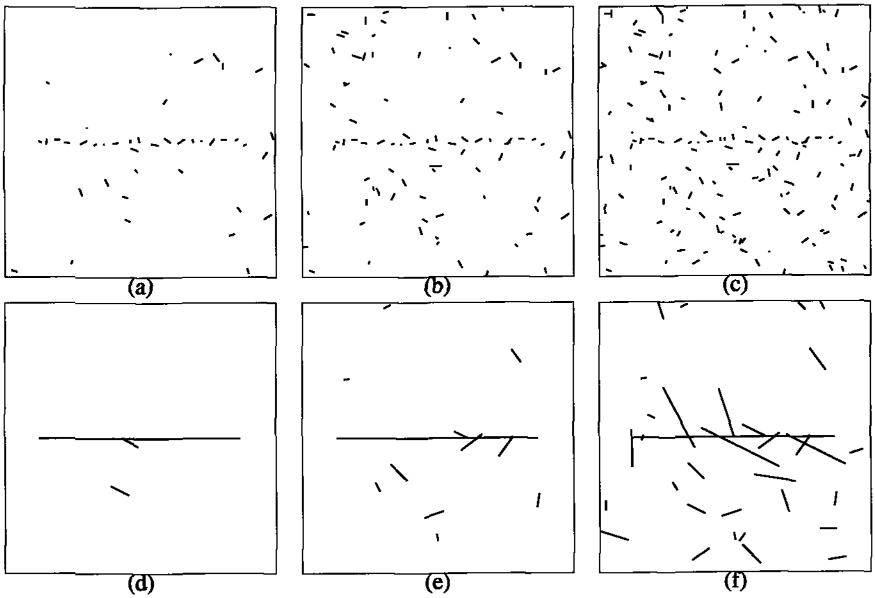


Figure 2.4: Results for synthetic images with noise. From left to right: 25, 75 and 150 noise segments.

well. Therefore, a number of linear structures is detected in the noise. The number of these accidental linear structure increases as the number of noise segments increases. In a more sophisticated approach, the image formation model could take into account that noise segments can be aligned accidentally and the neighborhood functions could be adapted accordingly.

The noise sensitivity has also been tested with the house image. The house image is contaminated with 100, 300 or 500 uniformly distributed noise segments and the clustering is performed until the number of remaining line segments is 170 larger than the number of added noise segments. The results are shown in figure 2.5. Line segments which have not participated in the clustering process are not shown. In all cases, the most important linear structures are detected, but there is also a number of linear structures detected in the background noise.

In order to examine the range of images for which the method gives good results, the clustering is applied to two other images, taken from the Cranfield benchmark set, which contain curved structures. The results are shown in figure 2.6. The pictures show, from left to right, the original image, the initial line segment configuration and the line configuration which is reached after a number of clustering operations, just before unwanted clusterings start to occur. The moment at which the clustering stops is selected such that the resulting

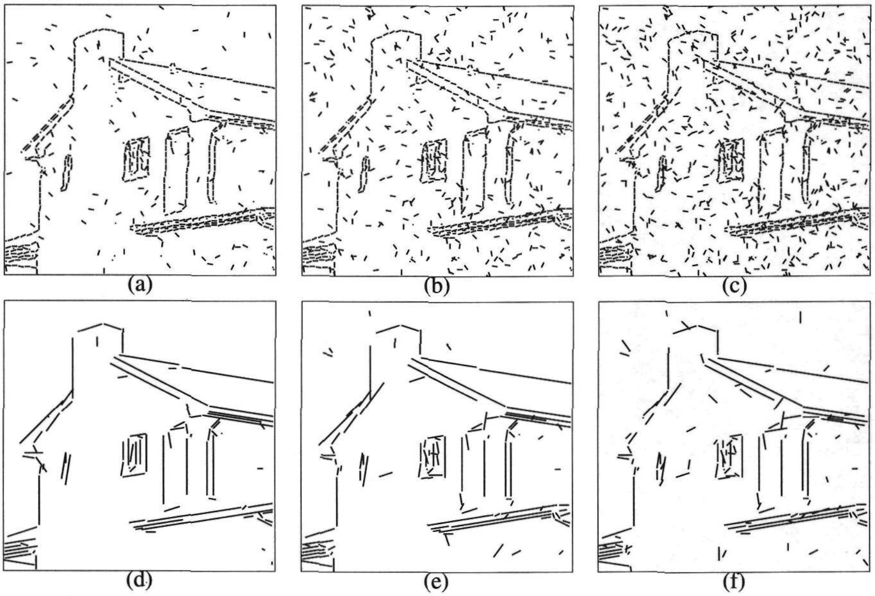


Figure 2.5: Results for house image with additional noise. From left to right: 100, 300 and 500 noise segments.

configuration is visually pleasing.

The linear structures in all images are detected quite well. If curves are present, a number of short segments is fitted along these curves.

Note that there is a number of long lines which are broken up in a staircase-like pattern, for example the diagonal line in the lower left hand corner of the image in the second row. This effect is caused by the discretization, in our implementation, of the line orientations: a line with the correct orientation can not be represented. This problem can be solved by increasing the orientation resolution.

In order to test the sensitivity of the clustering process to parameter changes, the clustering process is applied to the house image with various parameter settings.

The results are shown in figure 2.7. In all cases, the clustering process proceeded until there were 150 segments left. The parameters C, l_{max} and D are varied in the left, middle and right column, respectively. For each of the parameters C, l_{max} and D, a smaller and a larger value are tried. The results obtained with the smaller values (the normal values divided by 2) are presented in the top row and those with the larger parameter (the normal value multiplied by two) in the bottom row.

For all parameter settings, the most important linear structures are detected.

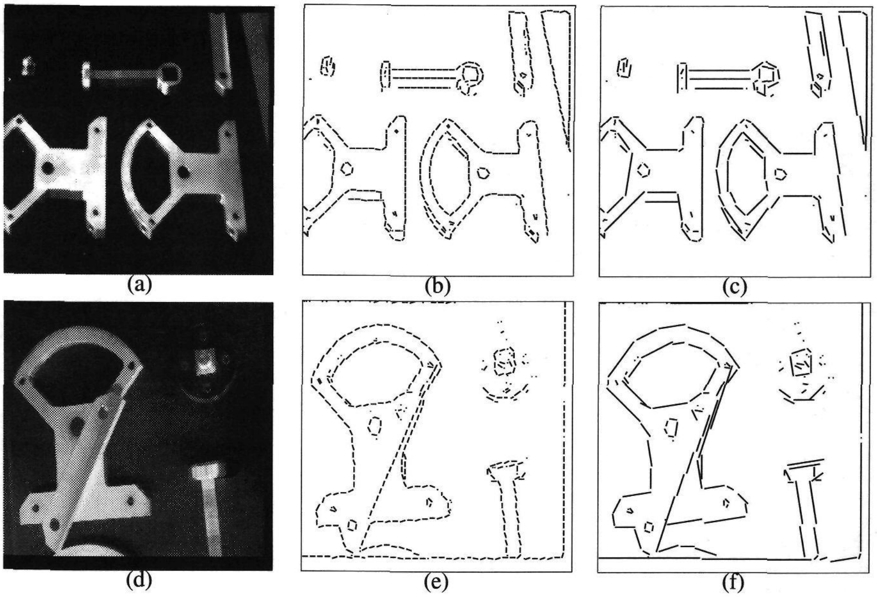


Figure 2.6: Result for different images.

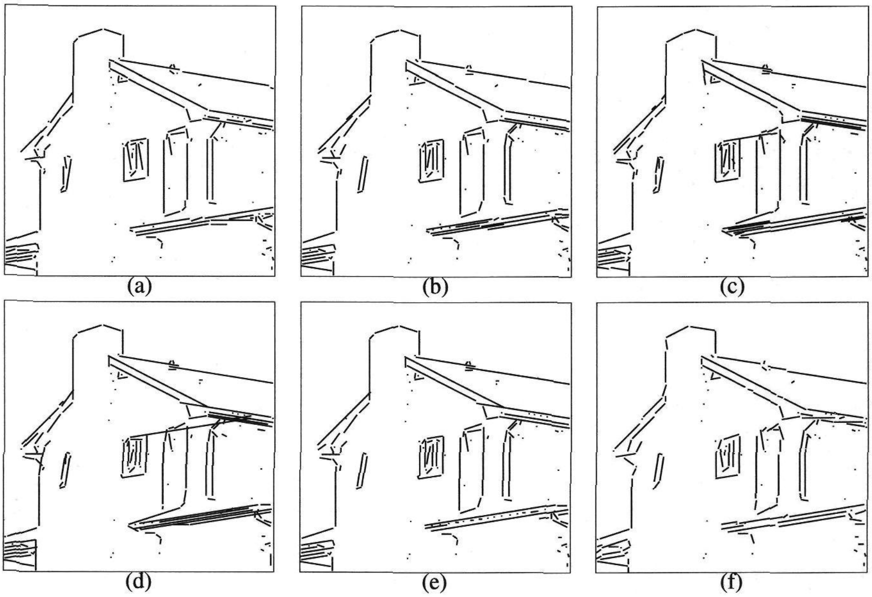


Figure 2.7: Results for different parameter settings. From left to right, C, $l_{\max}$ and D are varied. On the top row, the parameter is smaller, on the bottom row, it is larger.

In the left column, it can be seen that a large value of C encourages the bridging of large gaps (e.g. between the door and the window) and the formation of long segments (e.g. in the gutters). In the middle column, the effect of the cut-off length can be seen. A large cut-off length encourages the formation of large line segments, as can be seen for example in the roof. The right column shows the effect of the parameter D. A large value of D encourages merging long parallel line segments, as can been seen in the gutters. As such merges are discouraged by a small value of D, other merges, such as the one between the top of the window and the top of the roof, occur.

In a previous publication [77], the results with a different choice for $\sigma_l(l_0)$, σ_a and $\sigma_a(l_0)$ have been presented. Those results are comparable to the ones resented here. This also shows the robustness of our metric.

2.6. Conclusions

In this chapter, a metric for comparing line segments has been presented. This metric measures how well two line segments could be replaced by a single longer one. The metric has been constructed using a new technique, based on neighborhood functions. This technique is applicable to other situations as well. The behavior of the metric depends on the choice of the image formation model, which is reflected in the choice of the neighborhood functions. In the chapter, such a model and the corresponding neighborhood function have been presented. The number of free parameters left which must be fixed to "tune" the metric, is limited.

The metric has been tested by using it in a very simple line segment clustering process. This process detected large linear structures in images containing a number of short line segments, produced by edge detection and line fitting. The clustering process can detect large linear structures in images quite well, especially with regard to the simplicity of the clustering algorithm. This shows that the metric is indeed a good measure for the groupability of line segments. It has been demonstrated that our method is robust and shows graceful degradation, both in the presence of noise and under variation of the parameters.

3

Connected Clusterings in a Graph

3.1. Introduction

An important task in image processing and pattern recognition is finding clusters of 'similar' primitives in a large set [48]. In image processing, this typically amounts to image segmentation. An image is a collection of measurements, arranged in some spatial order. Therefore, two sets play a role when performing clustering on image data. The first set is the one that defines the spatial arrangement of the measurements. For raw images, this set is typically a continuous space $\mathbb{R}^n$ or a regular grid $\mathbb{Z}^n$. This set is called the *spatial support*. The other set is the set of possible measurement values. It usually contains a number of allowed grey values, represented by integers, but it can also be higher dimensional, like in the case of multi-spectral images. This set is called the *range* of the image.

In this chapter, clustering of grey values in an image will be discussed in the context of the *region adjacency graph*. In such graphs, each vertex represents a connected subset of the image plane. This subset can be as small as a single pixel. Two vertices are connected by an edge if the subsets they represent are adjacent. Repeated merging of regions in the image leads to the construction of a hierarchy of region adjacency graphs (cf. section 1.4).

Each level in the hierarchy of graphs acts as the spatial support of an image. The range of these graph images is a set of grey values, represented by integers. The image grid of the input image is considered as a 4-, 6- or 8-connected graph. This allows for a uniform treatment of all levels.

Suppose that a number of characteristic vertices has been selected for some graph image. A clustering can then be devised by assigning other vertices to one of those characteristic vertices. Each characteristic vertex then represents the cluster of all vertices which have been assigned to it. The assignment of vertices will be performed such that each vertex is assigned to a characteristic vertex that is 'close' to it. The selection of characteristic vertices will be discussed later on in this chapter.

The fact that we have to deal with two sets—spatial support and image range—can lead to difficulties. Should a vertex be assigned to a characteristic vertex to which it is near in spatial support, or to a characteristic vertex that has a similar grey value, or both?

If we consider only spatial support and ignore the range and assign each point to the nearest characteristic point, we obtain the Voronoï diagram of the set of characteristic points. If we consider only the range and ignore the spatial support, assignment of each primitive to the nearest characteristic primitive boils down to multilevel thresholding. Such a clustering procedure will result, in general, in groups of vertices which do not constitute connected subsets of the spatial support.

In this chapter, we will discuss clustering methods which take into account both the spatial support and the image range and which have the property that the vertices assigned to a given characteristic vertex constitute a connected subset of the spatial support.

In section 3.2, we will consider a number of criteria for clustering primitives which correspond to *increasing path distances*. It will be shown that, for such criteria, it is possible to construct a clustering in which each vertex is assigned to a characteristic vertex in an optimal way. The same algorithm can be used to compute a non-optimal connected clustering for other types of distances.

In section 3.3, some examples of hierarchies built based on various clustering criteria will be presented. Section 3.4 presents the conclusions of this chapter.

3.2. Optimal Clusterings for Increasing Path Distances

This section discusses clustering algorithms based on increasing path distances. First, we define this type of distances. Then, it is shown that there is an algorithm that computes clusters which are both optimal with respect to this distance and connected in image space.

Let $G = (V, E)$ be a graph with vertex set V and edge set E. This graph will be the measurements space of an image, i.e. there is a value $g(v)$ in the range R associated with each vertex v of the graph.

Definition 3.1 *Let $G = (V, E)$ be a graph and let $v, w \in V$. A path from v to w is a sequence $v_0 = v, v_1, \ldots, v_n = w$ of vertices such that $(v_i, 'v_{i+1}) \in E$ for all i.*

Various costs can be associated with a path $v_0, \ldots, v_n$. The value of such a cost function is often a nonnegative integer, but, as we will see later, other totally ordered sets N can be used. The smallest element in such a set will be denoted by 0.

Definition 3.2 *A function C assigning a cost in N to each path $v_0, \ldots, v_n$ is called an increasing path cost if there is a function $\phi : N \times N \to N$ and a function $d : R \times R \to N$ such that*

$$C(v_0, \ldots, v_n) = \phi(C(v_0, \ldots, n_{n-1}), d(g(v_{n-1}), g(v_n))) \qquad (3.1)$$

and $C(v_0) = 0$, where ϕ satisfies the following conditions:

1. $\phi(0, x) = x$.
2. $\phi(x, y) \leq \phi(x', y)$ *if* $x \leq x'$.
3. $\phi(x, y) \geq x$.

This definition implies that the cost of a path can be computed iteratively.

Definition 3.3 *Let C be a minimal path cost. Then an increasing path distance D is defined by*

$$D(v, w) =$$
$$\min\{C(v_0, \ldots, v_n) \mid v = v_0, v_1, \ldots, v_n = w \textit{ is a path from } v \textit{ to } w\}$$

Thus, the increasing path distance between two vertices is simply the minimal path cost over all paths between these two vertices. The following lemma shows that such a path can be chosen in an optimal way.

Lemma 3.4 *Let v, w be two vertices in a graph. Then there is a path $v_0 = v, v_1, \ldots, v_n = w$ such that*

$$D(v, v_i) = C(v_0, v_1, \ldots, v_i) \qquad (3.2)$$

for every i. Such a path is called an optimal path from v to w.

PROOF. This lemma can be proved with induction in $D(v, w)$. The lemma is true for $D(v, w) = 0$, since for every path $v_0 = v, v_1, \ldots, v_n = w$ with cost 0, $D(v, v_i) = 0$ holds for all i. (use definition 3.7, property 3.)

Now suppose that the lemma holds for all vertices u with $D(v, u) < D(v, w)$. Consider the path $v_0 = v, v_1, \ldots, v_n = w$ with cost $D(v, w)$ and let k be the last index for which $C(v_0, \ldots, v_k) > D(v_0, v_k)$. This implies $D(v_0, \ldots, v_k) < D(v, w)$. Replace the first part $v_0, \ldots, v_k$ of the path by an optimal path $v = v'_0, v'_1, \ldots, v'_{k'} = v_k$ from v to v_k. Then $v'_0, \ldots, v'_{k'}, v_{k+1}, \ldots, v_n$ is an optimal path. In order to show this, it is sufficient to show that

$$\phi(D(v, v_k), d(g(v_k), g(v_{k+1}))) = D(v, v_{k+1}).$$

Using property 2 in definition 3.2, it can be seen that the left hand side is smaller than or equal to

$$\phi(C(v_0,\ldots,v_k),d(g(v_k),g(v_{k+1}))) = C(v_0,\ldots,v_k,v_{k+1}) = D(v,v_{k+1}),$$

but as $D(v,v_{k+1})$ is smallest possible path cost for a path from v to v_{k+1}, equality must hold. ∎

We present some examples of increasing path costs. If $d(g_1,g_2) = 1$ for all g_1 and g_2 and $\phi(a,b) = a+b$, then $D(v,w)$ is the length of the shortest path from v to w, that is

$$D(v,w) = \min\{n \mid v_0,\ldots,v_n \text{ is a path from } v \text{ to } w\}. \tag{3.3}$$

If $d(g_1,g_2) = |g_1 - g_2|$ and $\phi(a,b) = a+b$, then $D(v,w)$ is the length of the shortest path from v to w , where each edge (v_1,v_2) is weighted by the difference between $g(v_1)$ and $g(v_2)$, that is

$$D(v,w) = \min\{\sum |g(v_{i+1}) - g(v_i)| \mid v_1,\ldots,v_n \text{ is a path from } v \text{ to } w\}. \tag{3.4}$$

It is not necessary to take $\phi(a,b) = a+b$. If we take, for example, $\phi(a,b) = \max(a,b)$ and $d(g_1,g_2) = |g_1 - g_2|$, then $D(v,w)$ is the minimum over all paths from v to w of the largest grey value step between adjacent vertices in such a path, that is

$$D(v,w) = \min\{\max_i |g(v_{i+1}) - g(v_i)| \mid v_1,\ldots,v_n \text{ path from } v \text{ to } w\}. \tag{3.5}$$

It is even possible to allow D to have non-numerical values. Suppose that we use only integer grey values. Consider the set of polynomials in X, and define a total ordering on these polynomials by

$$p(X) \leq q(X) \Leftrightarrow \lim_{X\to\infty} q(X) - p(X) = +\infty. \tag{3.6}$$

It is then possible to define an increasing path distance by choosing $\phi(a,b) = a+b$ and $d(g_1,g_2) = X^{|g_1-g_2|}$. This distance function corresponds with a path cost function which defines the cost of a path to be $\sum_i n_i X^i$, where n_i is the number of times a grey value step of height i occurs in the path. Thus, the distance is defined by

$$D(v,w) = \min\{\sum_i X^{|g(v_{i+1})-g(v_i)|} \mid v_0,\ldots,v_n \text{ is a path from } v \text{ to } w\}. \tag{3.7}$$

Suppose a subset $V^* \subset V$ of characteristic points of a graph image is given. We consider the problem of assigning each vertex in V to that vertex in V^* to which it is closest. Closeness will be measured by an increasing path distance D. We describe an algorithm which computes an assignment with the following properties:

1. If a vertex v is assigned to a vertex $v^* \in V^*$, then the equality $D(v^*, v) = \min_{w \in V^*} d(w, v)$ will hold.
2. The set of vertices assigned to each characteristic vertex $v^* \in V^*$ is connected.

In other words, we provide an algorithm which generates clusters which are both optimal with respect to the distance used, and connected in image space.

In the algorithm, three variables $\alpha(v)$, $\delta(v)$ and $\delta'(v)$ are associated with each vertex v. The variable $\alpha(v)$ represents for each vertex v the characteristic vertex to which it is assigned. The variable $\delta(v)$ represents the distance to this vertex and has value ∞ as long as the correct characteristic vertex has not been found. The variable $\delta'(v)$ represents the smallest cost of a possible path to a characteristic vertex through vertices v' for which $\delta(v')$ has already been set.

Algorithm 3.5 *Computation of an optimal, connected clustering in a graph image for an increasing path distance.*
Let $G = (V, E)$ be a graph with vertex values $g(v)$. Let V^* be a set of characteristic points. Let three variables $\delta(v)$, $\delta'(v)$ and $\alpha(v)$ be associated with each vertex $v \in V$.

1. *Put $\alpha(v) := v$ and $\delta(v) := 0$ for each $v \in V^*$ and $\delta(v) := \infty$ for all other vertices.*
2. *Put all vertices in $N(V^*) \setminus V^*$ in a list;*
   put $\delta'(v) := \min_{x \in N(v)} \phi(\delta(x), d(g(x), g(v)))$ for all vertices v in the list.
3. *Select the vertex v with the smallest value of $\delta'(v)$. Put $\delta(v) := \delta'(v)$ and put $\alpha(v) := \alpha(w)$, where w is a neighbor of v which satisfies $\delta(v) = \phi(\delta(w), d(g(v), g(w)))$.*
4. *Remove v from the list; put all neighbors w of v for which $\delta'(v) = \infty$ in the list and put $\delta'(w) := \min_{x \in N(w)} \phi(\delta(x), d(g(x), g(w)))$.*
5. *repeat steps 3 and 4 until the list is empty*

Theorem 3.6 *Consider the values $\alpha(x)$ computed by algorithm 3.5. Then, for each $v^* \in V^*$, the set $\{v \in V \mid \alpha(x) = v^*\}$ is connected.*

PROOF. It can be seen from the algorithm that each time an assignment $\alpha(v) := w$ is made, $w \in V^*$ holds.

After step 1, we have for each $v^* \in V^*$ the equality $\{v \in V \mid \alpha(v) = v^*\} = \{v^*\}$, so $\{v \in V \mid \alpha(v) = v^*\}$ is a connected subgraph.

Further assignments to α are made in step 3. When the assignment $\alpha(v) := v^*$ is made, there is a neighbor w of v such that $\alpha(w) = v^*$. Therefore, connectedness of $\{v \in V \mid \alpha(v) = v^*\}$ still holds after the assignment. Repeating this argument for each assignment to α shows that, after termination of the algorithm, $\{v \in V \mid v \in V^*\}$ is connected for each $v^* \in V^*$. ∎

Theorem 3.7 *Consider the values $\alpha(v)$ and $\delta(v)$ computed by algorithm 3.5. Then for each $v \in V$:*

$$\delta(v) = \min_{v^* \in V^*} D(v^*, v) = D(\alpha(v), v). \tag{3.8}$$

PROOF. Write $D_{\min}(v)$ for $\min_{v^* \in V^*} D(v^*, v)$. We will show that the attributes $\delta(v)$ of all vertices v will be assigned the value $D_{\min}(v)$ in increasing order of $D_{\min}(v)$.

In the first step, the assignment $\delta(v^*) := 0$ is performed for each $v^* \in V^*$.

Now suppose that the assignment $\delta(v) = D_{\min}(v)$ has been performed for all vertices v with $D_{\min}(v) < d$, for some d. It will be shown that in the subsequent series of assignments, $\delta(v)$ will be put to d for all v with $D_{\min}(v) = d$, before any δ receives a value larger than d. Let v be the next vertex for which an assignment to δ is made and let $v^* = v_0, v_1, \ldots, v_n = v$ be an optimal path (lemma 3.7) from $v^* \in V^*$ to v. There is a k in $\{0, \ldots, n\}$ such that $D_{\min}(v_i) = d$ for $i \in \{k, \ldots, n\}$ and $D_{\min}(v_i) < d$ for $i \in \{0, \ldots, k-1\}$. This implies $\delta(v_i) = D_{\min}(v_i)$ for $i \in \{0, \ldots, k-1\}$. If $d = 0$, then $k = 0$, but as $v_0 \in V^*$, it is sure that the equality $\delta(v_0) = 0$ already holds.

We will assume that some value larger than d will be assigned to $\delta(v)$ and deduce a contradiction. If some value larger than d is assigned to $v = v_n$ in step 3, then $\delta'(v) > d$ when this assignment is made. This implies that $\delta(v_{n-1}) > d$ when the assignment to $\delta'(v)$ was made in step 4, for otherwise, $\delta'(v)$ would have been put to $\phi(\delta(v_{n-1}), d(g(v_{n-1}), g(v_n))) = d$, where that last equality holds because of the optimal path property. By repeating this argument, it can be seen that some value larger than d has been assigned to $\delta(k)$. (Note that this claim holds both in the case $d > 0, k > n$ and in the case $d = 0, k = 0$.) But the assignment $\delta'(v_k) := d$ was made after the assignment $\delta(v_{k-1}) := D_{\min}(v_{k-1})$, so the assignment $\delta(v_k) := d$ would have been made earlier. Therefore the value d will be assigned to $\delta(v)$.

Note that, in step 3, $\alpha(v)$ is given a value such that $\delta(v) = D(v^*, v)$ is satisfied. ∎

Algorithm 3.5 computes connected optimal clusterings for increasing path distances. For other types of distances, such clusterings may not exist. Yet algorithm 3.5 can be used to compute sub-optimal connected clusterings for such distances.

3.3. Application to Hierarchical Clusterings

Algorithm 3.5 computes clusterings in graph images in which some set of characteristic points has been marked. The result of this computation can be represented as a graph in which each vertex represents all the vertices in a single cluster of the input graph. This section describes some results when algorithm 3.5 is applied repeatedly in order to compute a hierarchy of region adjacency graphs.

Application of the clustering algorithm requires the definition of a distance measure between vertices, which expresses similarity and nearness of vertices. This determines, which vertices will be clustered in a single region. Two distance measures will be used. The first one is the grey value weighted path length as defined by equation (3.4). This distance is an increasing path distance, so the clustering computed by algorithm 3.5 is an optimal one.

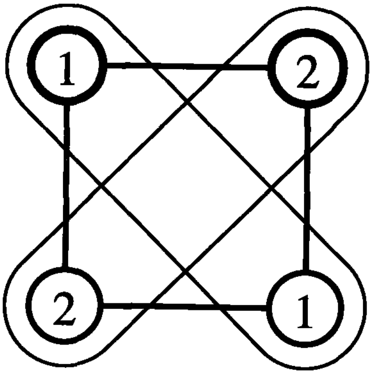


Figure 3.1: A situation where no optimal connected clustering exists. If the vertices at the top are the characteristic ones and the grey value difference is used, the optimal clustering consists of two non-connected groups, as indicated.

The second distance is the grey value difference defined by

$$C(v_0, \ldots, v_n) = |g(v_0) - g(v_n)|. \qquad (3.9)$$

This cost depends only on the grey values of the end points of a path, not on the vertices between them. This distance is not an increasing path metric, so an optimal connected clustering will not exist in general. In figure 3.1, it can be seen that the optimal clustering defined by such a metric can be disconnected. On the other hand, this distance has the advantage that it induces a multilevel threshold if an optimal clustering is determined without regarding connectivity.

Therefore, clusterings computed with this measure represent a compromise between thresholding and the preference for connected clusters.

In each level of the hierarchy, characteristic vertices are chosen by a random process in such a way that for each level, 25% of the vertices are selected as characteristic points. If some grey level occurs often in some part of the region, it is likely to be put into a larger cluster. In the construction of the next level, larger regions must therefore have a larger chance of being selected as characteristic regions.

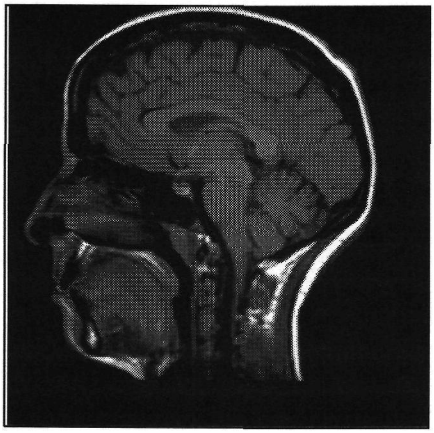


Figure 3.2: The original 256 × 256 image on which clustering is performed.

Therefore, characteristic regions are selected by the following process. A series of random points is generated with a uniform distribution over the image plane. A region is selected if it contains at least one point from the random set. Thus, regions with a large area have a larger chance of being selected.

Figure 3.3 shows the results obtained for the original image 3.2. The original image is 256 × 256 pixels large. Figure 3.3 shows the second, fourth and sixth levels, which contain 4096, 256 and 16 regions, respectively. The top row was constructed using the grey value difference criterion; the bottom row was constructed using the weighted path length criterion.

In Gaussian scale space, images are often described as a configuration of light and dark blobs and the evolution of these blobs is tracked when the scale parameter increases [103]. In the context of this section, light and dark blobs correspond to vertices which are lighter or darker, respectively, than all of their neighbors.

Therefore, we consider the type of hierarchy that can be constructed when all local minima and maxima are selected as characteristic points. In this case, the top level of the hierarchy does not consist of a single vertex, but of a graph

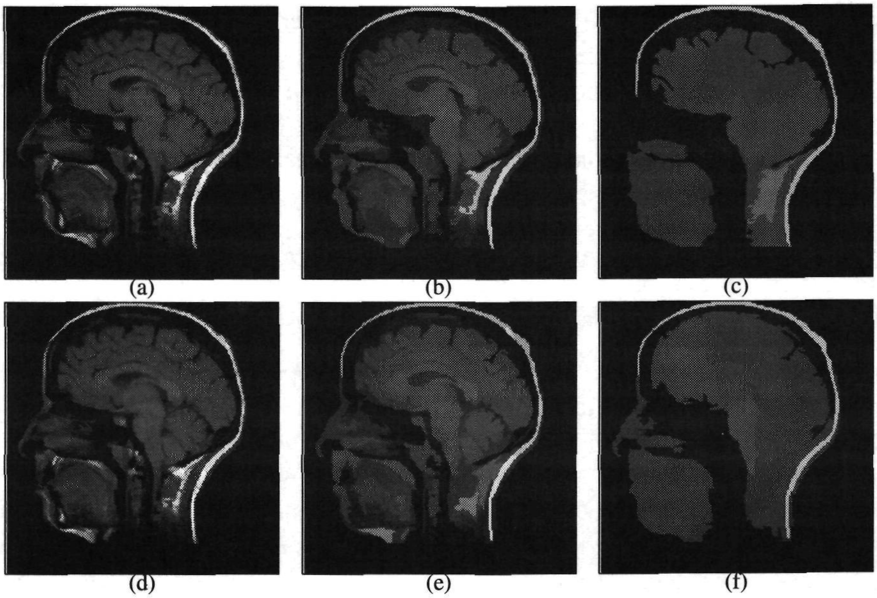


Figure 3.3: Clustering results. The top row was constructed with the grey value difference criterion, the bottom row with the weighted path length. From left to right, levels with 4096, 256 and 16 regions are shown.

in which each vertex is either a minimum or a maximum, for then each vertex is selected as a characteristic primitive. For this situation, some interesting properties can be derived. A *cycle* in a graph is a path in which the begin and end points coincide, and in which all other vertices are distinct. The number of distinct vertices in a cycle is called its length.

Lemma 3.8 *Let G be graph at the top level of a hierarchy created by selecting local extrema as characteristic points. Then each cycle in G has even length.*

PROOF. Upon stabilization, every vertex in the graph is either a minimum or a maximum. No two minima can be neighbors in the graph, nor can two maxima. Therefore, the vertices in a cycle form an alternating sequence. As a cycle is closed, this is possible only if it contains an even number of vertices. ∎

If the original graph is 4-connected or 6-connected, each level is a region adjacency graph is a planar graph. In a planar graph, *faces* are regions of the plane which are surrounded by edges and vertices which form a cycle, in such a way that this region contains no other vertices. The number of edges (or vertices) which surround a given face is called the *degree* of that face.

Kropatsch and Montanvert [57] have shown that the degree of the faces in a hierarchy of graphs does not increase with level, i.e. for each face in a higher level of the hierarchy, a face in a lower level of the hierarchy can be found which has at least the same degree. They state their result only for a particular class of hierarchies, but their proof is is applicable without change to the structure at hand.

If the original image is considered as a 6-connected graph, this has some interesting consequences.

Theorem 3.9 *Let G be the top level of a hierarchy which was constructed from a 6-connected image using extrema as characteristic points. Then G is a tree.*

PROOF. Suppose G is not a tree. As it is a planar graph, it must contain a closed face. As the degree of the faces in a 6-connected grid is 3, the degree of such a face in G can be at most 3. Each face of degree d defines a cycle of length d and according to lemma 3.8, d must be even. As $d \leq 3$, d must be equal to 2. Yet faces of degree 2 do not occur in a region adjacency graph (although it may be possible that multiple edges between vertices are required in order to retain the bounded face degree property [110]). This leads to a contradiction, so G must be a tree. ∎

Hierarchies based on extrema can be very high, because some configurations can contain many extrema, such that the number of vertices is reduced only slightly between successive levels. This is for example the case when a homogeneous region contains lighter and darker noise pixels. Figure 3.4 show three levels of a hierarchy constructed by selecting extrema as characteristic vertices. The 4th level (with 1136 vertices, figure 3.4a), the 16th level (with 681 vertices, figure 3.4b) and the 481st, top level (with 31 vertices, figure 3.4b) are shown. The lower levels of the hierarchy form a poor representation of the image, but major structures are lost in the top level. This is caused by the fact that the description as a nested sequence of light and dark blobs is not appropriate for this image: the brain, which is a major structure in the image, is connected to both lighter and darker structures, and therefore it disappears.

3.4. Conclusions

In this chapter, we have presented a clustering method which operates on graph images and generates connected clusters. We have shown that, for a particular class of clustering criteria, the clustering computed by this algorithm is optimal. The algorithm can be used for other types of distances as well, in which case it computes non-optimal clusterings.

This sub-optimal clustering has been demonstrated with the grey level distance, which is closely related to segmentation by thresholding. Hence, this

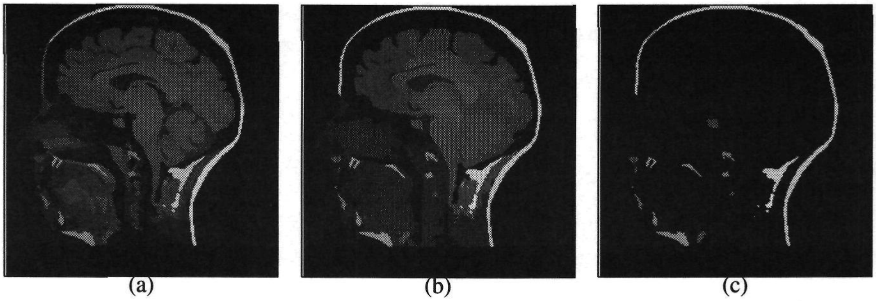


Figure 3.4: Three levels in a hierarchy defined by intensity extrema. The levels contain 1136 (a), 681 (b) and 31 (c) vertices.

method provides a compromise between segmentation by thresholding, and the desire to generate connected clusters.

Repeated application of clustering generates a hierarchical image description. If grey value extrema are selected as characteristic vertices, the top level of the hierarchy will contain only local extrema. If the base level graph is 6-connected, the graph structure in the top level contains no loops, i.e. it is a tree.

4

Model-Based Bottom-Up Grouping of Image Primitives

4.1. Introduction

Both grey level image segmentation and polygonal curve approximation can be considered as grouping problems. For grey-level image segmentation, pixels must be grouped in such a way that (1) the regions which they represent satisfy some homogeneity condition and (2) adjacent regions have distinct properties. For polygonal curve approximation, pixels must be grouped such that (1) the curve segments they represent can be approximated by a line segment and (2) adjacent groups (segments) have different orientations. This chapter presents a graph-based bottom-up grouping scheme in which both problems can be treated in a uniform manner.

Every segmentation algorithm must (implicitly or explicitly) adopt a *model* for homogeneous image regions. The most simple model for grey level image regions assumes a constant grey level value within each region. In some cases (e.g. for the segmentation of textured images) more intricate models may be required. Segmentation also requires an *error measure* for determining the deviation of an image region from the model. For the piecewise constant grey level model one often uses the root-mean-square value of the residues.

Many techniques for image segmentation have been proposed. They can be divided into two groups: those computing the homogeneous primitive groups (regions or curve parts) and those computing the boundaries between such groups. Region-oriented techniques compute connected groups of pixels which satisfy the region model. Edge-detection methods detect points in the image which satisfy a discontinuity model. It is not possible to evaluate all groupings. Groups of pixels which satisfy the model are typically found by splitting and/or merging groups repeatedly until the result fits the model within a given error.

Among bottom-up grouping schemes, region *merging* and region *growing* can be discerned. Region merging methods [15] first consider each pixel as an individual region. Two regions are replaced by their union if the latter satisfies the model. Merging continues until no union of adjacent regions satisfies the model. Region growing methods [112] first select a special set of pixels called *seeds*. Regions are grown by aggregating pixels to the seeds. This growth process continues until the image plane is covered by regions which satisfy the model. Each region in the final segmentation contains exactly one seed. The selection of appropriate seeds is a difficult problem, which requires procedures that are adapted to a particular class of images (e.g. [72]).

Bottom-up techniques use only local information, i.e. information from a restricted area. The regions over which information is collected increase progressively as the grouping process evolves. When global information becomes available, a clustering performed in the early stages may prove incorrect. Relinking methods [18] can be invoked to revise incorrect clusterings.

This chapter presents a merging scheme in which primitives are merged pairwise and in parallel. Primitives act as the vertices of a graph. The edges in the graph correspond to pairs of primitives which might be merged. In contrast to stochastic pyramid schemes [68, 76], in which an arbitrary number of adjacent regions can be replaced by their union, the bottom-up grouping scheme performs *pairwise region merging*.

The rest of this chapter is organized as follows. In the next section, grouping is presented in a hierarchical graph context and our method is described. In Section 4.3, the grouping method is applied to the problem of polygonal approximation of curves. In Section 4.4, some results in gray scale image segmentation are presented. In section 4.5, the results are discussed and some concluding remarks and suggestions are made.

4.2. Grouping with Hierarchical Graph Structures

A set of primitives can be represented by a *primitive adjacency graph*. The vertices of this graph are the image primitives (e.g. regions or curve parts) and its edges represent the adjacency relations between the primitives. The primitive adjacency graph plays an important part in our grouping scheme, because only prinitives which are connected by an edge may be grouped. Henceforth, a graph is indicated by the symbol G and its vertex and edge sets by the symbols V and E, respectively. If $G = (V, E)$ is a graph, a subset H of V is called *connected* (with respect to G) if, for all $x, y \in H$, there is a sequence $x = x_0, \dots, x_n = y$ such that $(x_i, x_{i+1}) \in E$ for all i. The graphs representing the results of two successive steps of the iterative bottom-up grouping procedure are called the *child graph* and the *parent graph* respectively. The parent graph represents the result of the grouping procedure applied to the child graph.

An iteration step in a bottom-up grouping procedure involves three stages.

(1) **Group selection** Non-overlapping groups of vertices in the child primitive adjacency graph are selected to be merged. These groups are chosen such that each group of vertices is a connected subset of G, and the primitives obtained by merging the primitives corresponding to the individual vertices in the group closely fit the region-model. In this chapter, groups will consist of one or two vertices.

(2) **Vertex construction** The primitive adjacency graph is transformed such that each group of vertices (the children) in the child graph maps to a single vertex (the parent) in the resulting parent graph. A parent represents a primitive which equals the union of the primitives corresponding to its children.

(3) **Edge construction** Vertex pairs of the parent graph that represent adjacent primitives are joined by edges. Two parent vertices are adjacent if and only if they have a pair of adjacent child vertices.

Formally, the iteration step can be defined as follows:

definition 4.1 *Let $G = (V, E)$ be a graph and let $H_1, \ldots, H_k$ be connected subsets of V such that $H_i \cap H_j = \emptyset$ for $i \neq j$ and $\cup H_i = V$. A graph $G' = (V', E')$ is said to result from a grouping step in G according to the groups $H_1, \ldots, H_k$ if there is a function $\pi : V \rightarrow V'$ such that*

(1) $\pi(V) = V'$;

(2) $\pi(x) = \pi(y)$ for each $x, y \in V$ if and only if there is a group H_i which contains x and y ;

(3) two vertices x' and y' in G' are neighbors ($(x', y') \in E'$) if and only if there are two vertices x and y in V such that $\pi(x) = x'$, $\pi(y) = y'$ and $(x, y) \in E$.

The first condition implies that all vertices in the parent graph are derived from the vertices of the child graph. The second condition implies that each vertex in the parent graph corresponds exactly to one of the groups H_i. The third condition describes adjacency between vertices in the parent graph.

Starting with the initial graph representing the input image a hierarchy of graphs (cf. subsection 1.4.2) is built by recursive application of the grouping process.

We now present a grouping scheme in which each pair of adjacent vertices is considered for merging. For each candidate pair, the fit of the union of the two vertices to the primitive model is computed. This information is used to select those pairs that are actually grouped, as described previously.

The selection of pairs for merging poses a transitivity problem. If, for example, three vertices x, y and z are mutually adjacent, both the pair (x, y) and the pair (y, z) are candidates for merging. However, they cannot both be selected, because y is a member of both groups, and only two vertices can be merged at a time. Each choice of a number of pairs to be grouped corresponds to the selection of a subset of edges in the primitive adjacency graph. An admissible choice of pairs, for which no transitivity conflicts occur, corresponds to a set of edges such that no vertex in the graph lies on more than one edge in the subset. In graph theory, such a set of edges is called a *matching* [36].

The *line graph* [36] is used to perform the selection of pairs to be merged.

definition 4.2 *Let $G = (V, E)$ be a graph. The line graph $L(G) = (V', E')$ is the graph for which each vertex $v' \in V'$ corresponds to an edge $(v_1, v_2) \in E$ such that two vertices v' and $w' \in V'$ are connected by an edge if and only if the corresponding edges $\{v_1, v_2\}$ and $\{w_1, w_2\} \in E$ share a common point.*

If two vertices in the line graph are connected if and only if the corresponding primitive pairs overlap, a matching corresponds to a subset of the vertices of the line graph in which no two adjacent vertices are present. Such a subset of vertices is called an independent set.

The line graph defined by the primitive adjacency graph is used to select pairs of primitives to be merged. A *merge score* is assigned to each vertex in the line graph. This merge score is equal to the error of the associated candidate pair with respect to the primitive model. Vertices which have a merge score above a threshold value t are not selected. Then, an independent subset V' of the remaining vertices is computed.

A large number of groups is indeed merged in each step if the set V' satisfies the following two properties:

(1) no two vertices in V' are adjacent;

(2) each vertex which is not in V' but has a merge score below threshold t has at least one vertex in V' as a neighbor.

In graph theory, a set satisfying these properties is called a *maximal independent set* (MIS) [36].

The following method [76, 49] can be used for the selection of a maximal independent set in the line graph. First, a numerical value is assigned to each vertex. This value can be a random number, or it can be determined by properties of the vertex. Vertices which have a larger label than all their neighbors are selected as members of the maximal independent set. Their neighbors are rejected. As a result, two neighboring vertices of the graph can not both be selected as members of the maximal independent set. It is possible that there are unselected vertices which are not adjacent to a selected vertex. In this case, all vertices that have been neither selected nor rejected are attributed a new label, and the selection procedure is repeated until each vertex has been either selected or rejected. The selection process usually converges after a few iteration steps.

In this chapter, two kinds of label assignment are investigated. In the first case, the labels are random numbers. This implies that the MIS which is computed is not influenced by vertex properties. In the second case, the label is determined by the merge score associated with each vertex: the higher the merge score, the lower the corresponding label. This implies that vertices corresponding to a low merge score have a high probability of being selected. An additional random label is used to resolve ties between vertices with equal merge scores. Using this choice of labels, the structure of the MIS which results, is adapted to the properties of the vertices. The two choices of labels described above will be referred to as adaptive and non-adaptive.

4.3. Polygonal Approximation of Curves

The grouping process described in the previous sections can be applied to the polygonal approximation of curves. In this case, the vertices of the graph represent line segments. Each vertex in the initial graph represents a line segment of pixel length. The initial graph can be derived from a chain code description [29]. The edges are defined from the adjacency relations along the curve. Each vertex which is not an end point of the curve has exactly two adjacent vertices.

Each pair of adjacent line segments is evaluated as a possible group. Two such line segments share an end point. The union of these two line segments is the line segment that connects their unshared end points. There are several measures for determining the quality of the approximation of a curve by a line segment. Four types of error measures have been investigated.

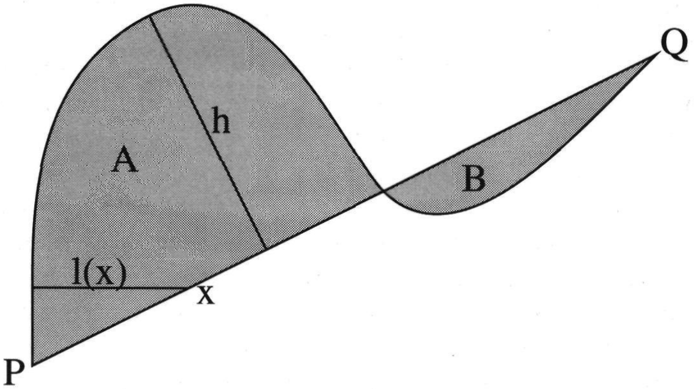


Figure 4.1: The approximation of a curve by a line segment. See the text for an explanation.

- The measure proposed by *Wall and Danielsson* [108]. The curve and the line segment cut out regions from the plane. The difference between the area of the regions to the right of the line segment and those to its left is called the *signed area* between the line and the curve. The error measure proposed by Wall and Danielsson is the absolute value of the signed area between the line segment and the curve, divided by the length of the line segment. A nice property of this measure is that the signed area between the union of two line segments and the curve can be calculated from the signed areas between the two individual line segments and the curve. Therefore, the error of a candidate pair can be computed from the properties of the constituting line segments, without having to consider the curve itself.

In figure 4.1, the Wall and Danielsson error measure is $|A-B|/PQ$, where A and B are the areas of the shaded regions and PQ is the length of the line segment.

- The measure proposed by *Borgefors* [13] is based on the closest distance of points on the line segment to points of the curve. This distance is calculated for each point on the line segment, and the root-mean-square value of this distance over the line segment is computed. The error measure of Borgefors can be computed efficiently from the distance transformation of the original curve.

In figure 4.1, the error measure defined by Borgefors is

$$\left[\int_{PQ} l^2(x)\,\mathrm{d}x/PQ\right]^{1/2}, \tag{4.1}$$

where $l(x)$ is the shortest distance from a point x on the line segment to the curve.

- The *average unsigned area* is the area between the the curve and the approximating line segment, divided by the length of the line segment. This measure is similar to the one proposed by Wall and Danielsson, but no sign information is taken into account. If the curve does not contain extreme turns, this measure equals the average distance of the curve to the line segment.

In figure 4.1, the average unsigned area is $(A+B)/PQ$, where A and B are the areas of the shaded regions and PQ is the length of the line segment.

- The *maximal distance* between the line segment and the curve is defined as the maximum (over all points on the curve) of the distance to the line segment. In figure, the maximal distance to the line segment is h.

The construction of a polygonal curve approximation starts with the construction of an initial graph that contains "line segments" of the size of a single pixel. Clustering is performed until the error measure of all pairs of adjacent line segments exceeds a threshold.

Figures 4.2 and 4.3 show the result of the polygonal curve approximation procedure with non-adaptive and adaptive computation of the MIS, respectively.

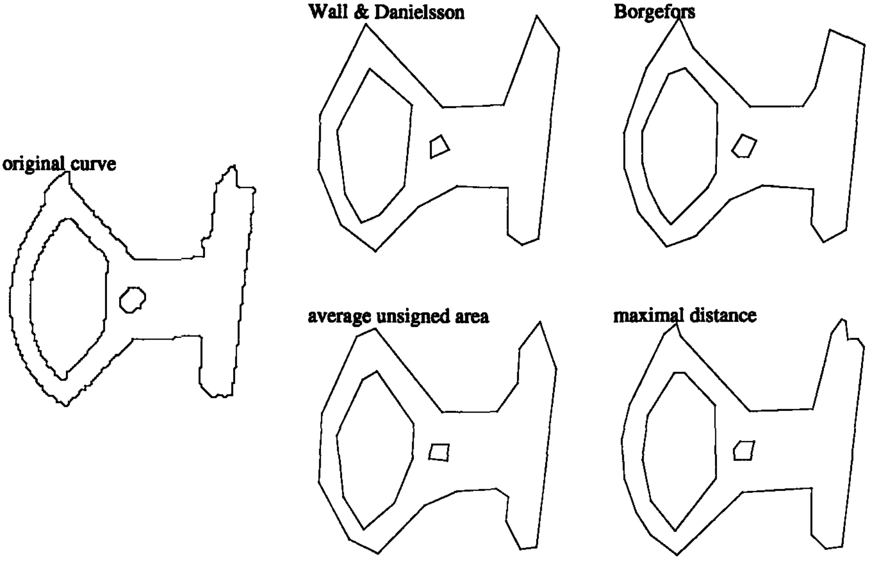


Figure 4.2: Polygonal curve approximations computed using non-adaptive MIS computation.

The threshold was 2.0 pixel sizes for all error measures. The size of the complete image was 128 × 128 pixels.

It can be seen that the approximations computed by the adaptive method follow corners in the curve more accurately. Note that the curve approximation contains a stochastic component, such that different runs of the same algorithm produce slightly different segmentation results. There are no major differences between the outcomes of different runs based on the same error measure.

The characteristics of the different error measure can be seen best from figure 4.3, because the effect of the error measure is more important for the adaptive method than for the non-adaptive one. The result from the maximal distance method gives the most accurate impression. This is caused by the fact that it uses the *maximum* distance between the curve and the approximating polygon, while the other measures are based on average values. Note that, using the Borgefors measure, the structure at the top right corner of the curve is cut, while this does not happen with the Wall and Danielsson measures. Yet the error according to the Wall and Danielsson error would be small for the segment which is found with the Borgefors measure, because there are areas between the curve and the segment on both sides of the segment. But, each segment results from joining smaller segments. These smaller segments would have a larger error according to the Wall and Danielsson error than to the Borgefors error.

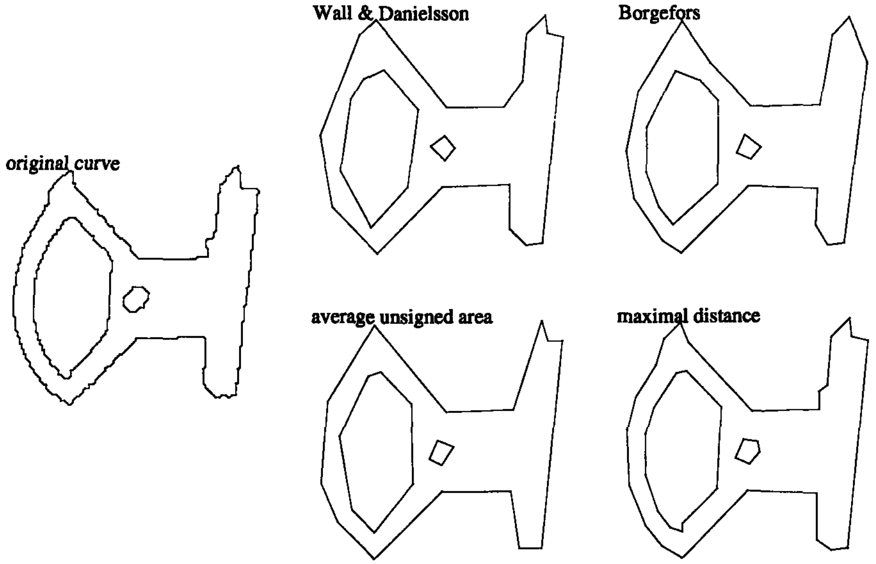


Figure 4.3: Polygonal curve approximation computed using adaptive MIS computation.

Note also that the approximation produced by the average unsigned error measure looks far less accurate than the approximation produced by the Wall and Danielsson measure, although the definitions of these measures seem very similar. This is caused by the different effects of noise on both measures. If a wiggly line is approximated by a straight line, parts of the curve will lie on both sides of the line segments. The corresponding areas cancel in the Wall and Danielsson measure, while they are added for the average unsigned are measure. Therefore, the distinction between straight but noisy lines and more significant structures is less strong with respect to the average unsigned area measure than with respect to the Wall and Danielsson measure.

4.4. Gray Level Image Segmentation

This section presents the application of the grouping procedure described in section 4.2 to grey-level image segmentation. The input image is represented by a 4-connected graph in which each vertex represents an individual pixel. The first region model we tested is the constant grey level model. The error-of-fit is the RMS value of the residue, i.e. the standard deviation within the region.

Figure 4.4 shows a synthetic image and its segmentation. The image size is 128 × 128 pixels. The image consists of bands with grey values 64, 128, 160 and 176, i.e. it contains step edges of heights 64, 32 and 16. The image is corrupted with additive Gaussian noise of $\sigma = 16$. The threshold on the merge score was 17. Note that the standard deviation within each of the bands is about 16, and that the standard deviation within the union of two adjacent bands is about 36, 23 and 18 for the left, middle and right pair of bands, respectively.

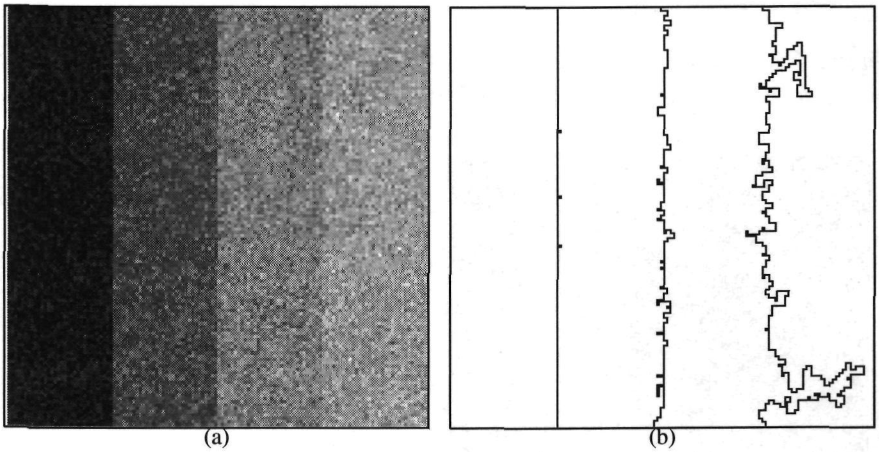


Figure 4.4: A synthetic image (a) and its segmentation (b).

Therefore, the boundary between the two bands to the right is just detectable with our method. If the merge score threshold were larger than 18, these two bands would be merged, while the individual bands would be fragmented if the threshold were lower than 16. The just detectable boundary to the right is very jagged; the other two boundaries are detected more accurately. Smoothness of a boundary is a global property. Therefore, every bottom-up segmentation procedure will have difficulties with the detection of weak smooth boundaries and accurate detection requires a post processing step.

Figure 4.5b shows the segmentation result for a 256 × 256 grey scale image (figure 4.5a), with grey values between 0 and 255. The threshold take for figure 4.5b was 20.

The next model we tested assumes a linear grey level function for each image region. Again, the RMS value error of the residue is used as the merge score. Finding the best fitting linear model is a straightforward regression problem. Figure 4.6a shows a synthetic image consisting of four bands. In each band, the grey level is a linear function of the coordinates. The regions have been chosen such that the grey levels are continuous across the region

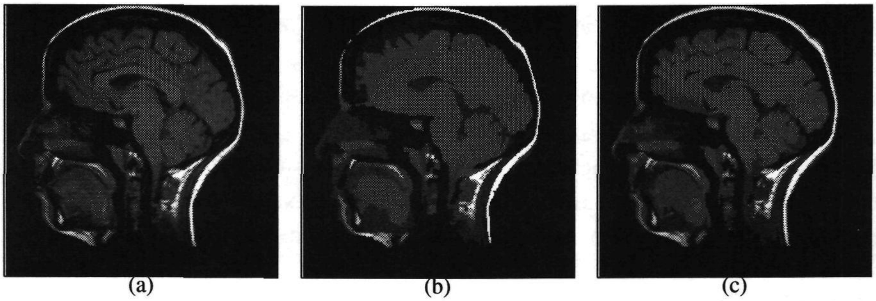


Figure 4.5: A grey scale image (a) and segmentations achieved with the standard deviation (b) and the grey level difference (c) as merge scores.

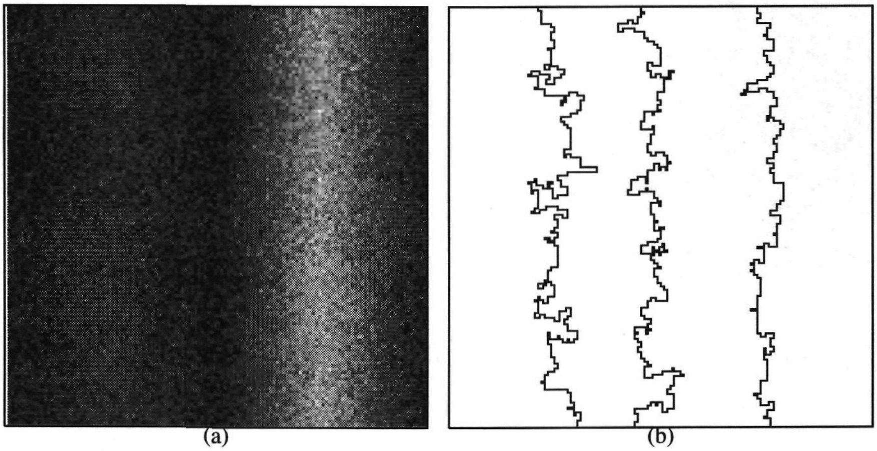


Figure 4.6: A synthetic image (a) consisting of four bands with linear grey scale functions, and a segmentation (b) based on the linear grey scale model

boundaries. The image has been corrupted by additive Gaussian noise with standard deviation 16.

Figure 4.6b shows a segmentation based on the linear grey scale model. The threshold used was 17 grey scale values. The synthetic image has been constructed such that merging two adjacent bands would produce regions with a RMS error value of about 18, 24 and 30, including the added Gaussian noise. Therefore, the two bands to the left are just discernible. If the threshold were increased, these regions would be merged; if it were decreased, individual bands would be fragmented. Note that all edges between boundaries are jagged, although the two bands to the right are clearly discernible with our criterion. This is caused by inaccurate merges which occur in the early stages of grouping: fitting a linear region to a small number of regions is not very stable. In the first stages of the grouping, information is integrated over small regions only. If small regions on both sides of the boundary between two bands are considered, the residue of a linear fit to their union is small. Only when broader regions on both sides of the edge are taken into account, will this residue be large and can the different bands be discerned. Meer *et al.* [69] suggest the use of least-median-of-squares estimators for this type of models.

Application of the linear model to natural images produces unsatisfactory results [78]. This is caused by the noise sensitivity of the linear model mentioned before, but also by the fact that linear regions can be fitted to blurred step edges, which occur frequently in natural images. This leads to severe edge dislocations.

One way to avoid problems caused by erroneous merges in the early stages of the process is to use a number of models in succession. Grouping is started with the constant grey level model. After stabilization, a linear model is assumed. As such models are more flexible, there are pairs of regions which can now be merged. After that, a third grouping step is performed assuming a quadratic region model. Figure 4.7 shows the image of a rod (a), the segmentation achieved with a constant region model (b) and the segmentation achieved through the three stage process (c). Although there has been some merging in the second and third phase, the bands of different brightness have not been detected as single regions.

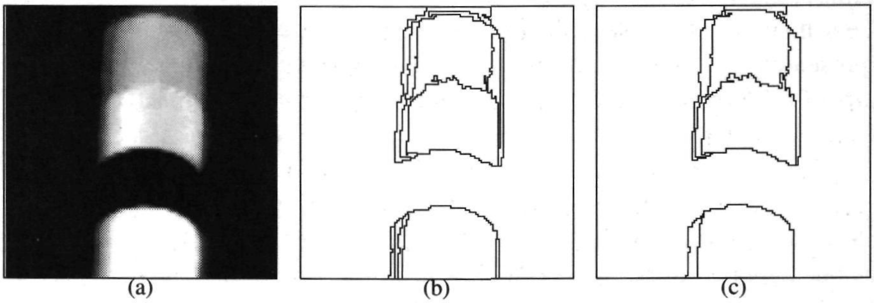


Figure 4.7: An image of a rod, a segmentation based on the constant grey level model and a segmentation based on a three stage grouping process with, successively, a constant, linear and quadratic model.

If two adjacent regions are very different in size, the standard deviation within their union will be close to the standard deviation in the largest region, even if the difference between the average grey values of the regions is large. This implies that small regions can easily be absorbed by larger ones. In order to avoid this, it is possible to use the difference between the average grey values of two regions as the merge score in stead of the standard deviation within their union. A result based on this merge score is shown in figure 4.5c. The threshold was 20 grey scale values. Note that a number of small regions is present which are not visible in figure 4.5b.

There is only one, global, threshold which determines whether or not two regions are allowed to merge. It is possible that not-so-similar regions are merged in the early stages of the process. This impedes the later stages of the segmentation. In order to avoid this problem, grouping can be started with a lower threshold and performed until stabilization. Then the threshold is increased, and grouping is performed in the primitive adjacency graph which resulted from the previous grouping procedure. Grouping is again performed until stabilization and then the threshold is increased. This can be repeated several times. In the most extreme case, just one pair of regions—namely the pair with the smallest merge score—would be merged in each step. This situation is similar to the clustering algorithm used in chapter 2.

The segmentation produced using an increasing threshold are not very different from those computed a single threshold. The adaptive construction of the MIS already encourages the "best" merges to be performed first.

4.5. Conclusions

In this chapter, a new grouping scheme for image primitives has been presented. The scheme has been described in a hierarchical graph context. Primitives are represented as the vertices of a primitive adjacency graph; compatibility of pairs of primitives is expressed through a merge score, which is derived from a primitive model.

The method is based only on the primitive adjacency graph and the merge scores associated with the edges in this graph. The merge scores are used to select pairs of adjacent primitives which are then grouped. The new technique has been applied to two type of image primitives: line segments for polygonal approximation of curves and regions for grey-level image segmentation .

In contrast to previous stochastic methods, the main role in our method is played not by the image primitives themselves, but by the adjacency and compatibility relations between them. Thus, the primitive model can influence the selection of groups of primitives which are to be merged, in a more direct way.

For polygonal curve approximation, four kinds of merge scores have been evaluated and the differences in the results have been discussed. It has been demonstrated that adaptive selection of pairs of line segments to be merged yileds better results than non-adaptive selection. The error criterion of Borgefors and the maximal distance error criterion produce the visually most pleasing results.

For grey level image segmentation, constant and linear grey level region models have been investigated. Synthetic images, corrupted with noise, have been segmented in order to assert the power of our method to separate similar regions.

Segmentation of natural images produces acceptable results, although the method poses a number of problems. These problems are caused by erroneous groupings which can occur because (1) the regions considered in the first stages of the merging process are very small and (2) because the primitive model and the merge score thresholds are global, not adapted to the image contents.

Within the present context, these problems can be reduced, but not solved completely, because these limitations are inherent to single pass, non-adaptive, bottom-up segmentation methods. In chapter 5, these problem will be attacked by using adaptive merging criteria, combining region and edge information and applying relinking.

5

Segmentation by Relinking in Irregular Structures

5.1. Introduction

In this chapter, the extension of pyramid relinking to hierarchies of graphs is developed. This approach allows the segmentation of an image into connected regions and the use of boundary information in the relinking process.

Pyramid relinking, which was originally described by Burt *et al.* [18], is a powerful and conceptually attractive method for image segmentation. In the past, a number of techniques have been proposed to reduce some of its weaknesses, such as border effects, the fixed number of regions and artifacts occurring for specific input patterns (e.g. [1, 6]). Several extensions have been made, which allow the application of relinking to other image types, such as flow fields [37] or textured images [84].

Conventional image segmentation by relinking uses a regular pyramid structure, as illustrated in figure 5.1. This is a stack of regular grids of sizes $2^n \times 2^n, 2^{n-1} \times 2^{n-1}, \ldots, 1 \times 1$. In the lowest level of the pyramid, each cell corresponds to a pixel in the image grid. Each cell in level $i + 1$ represents a cluster of cells in level i. The cells which may be contained in such a cluster form a 4×4 square in level i. These squares overlap in such a way that each cell on level i can belong to one out of four clusters. The cells in the cluster represented by a given cell are called the *children* of this cell; the representing cell is called the *parent* of its children. Each cell has one parent. Note that, for cells near the image border, the number of possible parents and children of a cell is smaller than 4 and 16, respectively.

The parent-child links induce a tree structure in the hierarchy. Thus, each cell in the pyramid represents a region in the image plane, which can be found by tracking all series of parent-child links leaving from a given cell, in a downward direction.

Segmentation by relinking is performed by iteratively updating the cluster membership of cells, i.e. by adapting parent-child links. This is done in such a

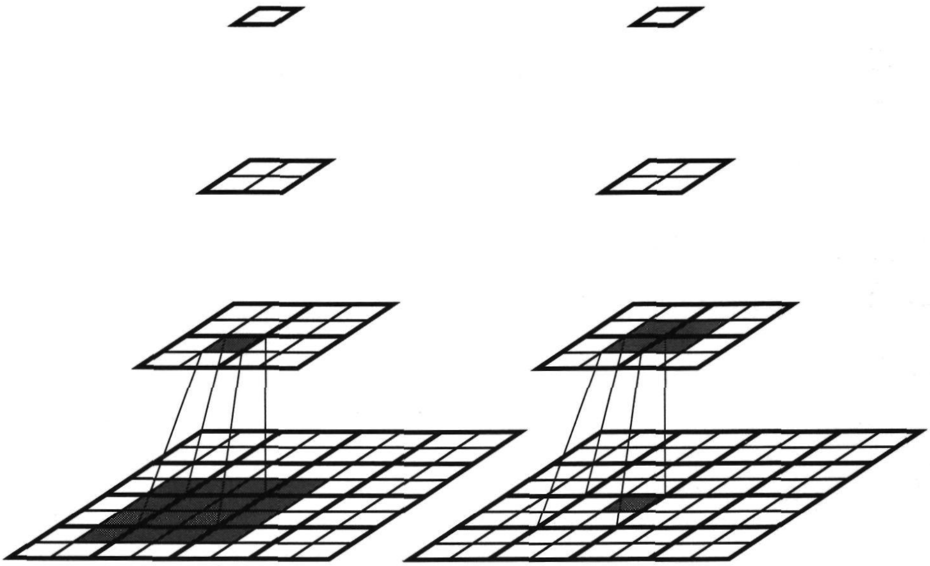


Figure 5.1: The regular pyramid structure used in conventional relinking. The lines between levels show the relative positions of cells in different levels. On the left, the sixteen shaded cells in the lower level are the candidate children for the shaded cell in the higher level. On the right, the four shaded cells in the higher level are the candidate parents for the shaded cell in the lower level.

way that the standard deviation of the grey levels of the clusters represented in each cell decreases. Thus, the relinking procedure converges towards a state in which the cluster represented by each cell is as homogeneous as possible.

In this chapter, some drawbacks of the relinking method are discussed and a relinking scheme based on hierarchies of graphs is presented, which solves these problems. The drawbacks originate from the fact that the levels in a relinking pyramid do not represent a region adjacency graph, but merely a subdivision of the image points in a predefined number of classes. Moreover, not all subdivisions corresponding to a subdivision of the image in the correct number of classes, can be represented [8].

The first problem is the fact that the clusters represented by a cell need not correspond to connected regions in the image plane. The relinking process takes the spatial structure of the image into consideration by allowing only a fixed set of possible children for each cell. The algorithm does not use connectivity: cells which are adjacent in some higher level grid of the pyramid need not represent adjacent regions in the image plane, or vice versa. This can cause the creation of regions which are scattered over the image plane and which consist of many connected components. If the pyramid structure is adapted by increasing the number of candidate children for each cell, regions can become increasingly scattered and the process becomes similar to isodata clustering of grey values, in which no account is taken of spatial structure. (See Kasif and Rosenfeld [50] for a discussion of the relations between pyramid relinking and isodata clustering.)

The second problem is caused by the regularity of the grid of cells in each level of the image grid and the associated set of 16 possible children for each cell. In such a configuration, not all possible subdivisions of the image plane can be represented, as was shown by Bister *et al.* [8]. Therefore, artifacts can occur in the segmentation of particular shapes such as elongated ones. This can be repaired by allowing irregular structures, in which the number of levels and the number of neighbors for each vertex is not fixed in advance.

The third problem is related to the first one. As the concept of a connected region can not be represented in the conventional relinking pyramid, it is also not possible to manipulate or represent the boundary between two classes or regions. Therefore, it is not possible to use information on boundaries between regions, such as length and average response of an edge detection filter, in a relinking based segmentation method.

In this chapter, we attack these three difficulties by using the hierarchy of graphs formalism. In section 5.2, we describe new relinking rules that force the regions represented by each cell to be connected.

However, this relinking strategy fails to detect strongly elongated objects, such as spirals, as a single region. Moreover, the number of regions represented by the hierarchy is fixed. In section 5.3, a method for the adaptive construction of subsequent levels in a hierarchy of graphs is presented, in which arbitrary image subdivisions can be represented.

The methods discussed in sections 5.2 and 5.3 require the application of the hierarchy of graphs formalism. In this formalism, information on the boundaries between regions can be represented as the attributes of the edges of the region adjacency graphs. In section 5.4, it is described how this possibility can be exploited for the combination of region and boundary information in order to improve the segmentation process.

In section 5.5, the integration in a single system of the methods described in the previous section is described and some segmentation results are shown.

Section 5.6 presents the conclusions of this chapter.

5.2. Connectivity Preserving Relinking

In the classical relinking method [18], the spatial arrangement of cells in a regular grid in higher levels of the pyramid does not reflect the spatial arrangement or connectivity of the regions represented by such cells. In this section, an adaptation of the relinking rules is proposed. If these new rules are used, regions are guaranteed to be connected and the adjacency relations between the regions can be represented by a region adjacency graph.

Let $G_i = (V_i, E_i)$ for $i = 0, \ldots, n$ be a hierarchy of graphs (cf. subsection 1.4.2). For each level of the hierarchy, there is a mapping $\pi_i : V_i \rightarrow V_{i+1}$ which assigns to each vertex a *parent* in the next level. The mapping $\kappa_i : V_i \rightarrow \mathcal{P}(V_{i-1})$ assigns to each vertex $v \in V_i$ its children $\{w \in V_{i-1} \mid \pi_{i-1}(w) = v\}$. Where no confusion can occur, the subscripts of π and κ are omitted.

We will use notations such as $\kappa(X)$ for $\{\kappa(v) \mid v \in X\}$, where X is a set of vertices at a fixed level. Sometimes $\kappa(v)$ will be identified with the subgraph consisting of the children of v. We will write $N(v)$ for the set of neighbors of v in the graph. If $v \in V_i$, the set $\kappa_1\kappa_2 \ldots \kappa_i(v)$ is the set of vertices in the base level graph which corresponds to the region in the image represented by vertex v. This region is called the *receptive field* of v and is denoted by $R(v)$. If $v \in V_i$, then $\pi^k(v)$ will denote $\pi^{i+k-1} \ldots \pi^{i+1}\pi^i(v)$ and $\kappa^k(v)$ will denote $\kappa^{i-k+1} \ldots \kappa^{i-1}\kappa^i(v)$.

The lowest level graph $G_0 = (V_0, E_0)$ corresponds to the image grid, which is considered as a 4-connected graph. It is also possible to use a 6-connected or 8-connected grid. The latter has the disadvantage that it is not a planar graph. This is not a problem in the present chapter, but it would be a problem for the method described in chapter 6.

The edges in the higher levels represent the adjacency structure for the receptive fields of the vertices in that level. These edges depend on the parent-child relations and on the structure of the base level graph. Two vertices v and w on level i are connected by an edge if they have children $v' \in \kappa(v)$ and $w' \in \kappa(w)$ which are connected in level $i - 1$. Recursive application of this criterion allows for the construction of all levels of the graph from the base level.

During the discussion of the connectivity of regions, the following result will be used.

Theorem 5.1 *Consider a hierarchy of graphs $G_i = (V_i, E_i)$ with parent assignments π_i and child assignments κ_i. Suppose that the edges of G_0 are fixed and that the edges in higher levels are induced by the base level and the parent assignments. Then the following statements are equivalent.*

1. *For each i and for each vertex $v \in V_i$, the subgraph of G_{i-1} induced by the vertices in $\kappa_i(v)$ is connected.*
2. *Each vertex v has a connected receptive field.*

PROOF. Suppose that the first statement holds. It will be shown with induction in the level i that each vertex $v \in V_i$ has a connected receptive field in G_0. For $i = 0$, this is obviously true: each vertex in the base level is its own receptive field. Now suppose that the first statement holds for level $i - 1$ and let $v \in V_i$. Consider two vertices w_1, w_2 in the receptive field $R(v)$ of v.

As $\kappa_i(v)$ is connected, there is a path from $\pi^{i-1}(w_1)$ to $\pi^{i-1}(w_2)$ in $\kappa(v) \subset V_{i-1}$. Denote this path by $u_1, \ldots, u_n$. For each edge (u_j, u_{j+1}) in this path there is an edge (u'_j, u''_{j+1}) in the base level with $u'_j \in R(u_j)$ and $u''_j \in R(u_j)$. By the induction hypothesis, each $R(u_j)$ is connected. As u'_j and u''_j are both in $R(u_j)$, there are paths in the base level from u''_j to u'_j. For the same reason, there are paths from w_1 to u'_1 and from u''_n to w_2 in the base level. Concatenation of these paths yields a path from w_1 to w_2 in the base level, which is entirely contained in $R(v)$. Thus, there is a path from w_1 to w_2 in the receptive field $R(v)$ and $R(v)$ is connected.

Now suppose that the second statement holds. Consider a vertex $v \in V_i$. Let w_1 and w_2 be two vertices in $\kappa_i(v) \subset V_{i-1}$. It will be shown that there is a path in $\kappa(v)$ from w_1 to w_2. Let $w'_1 \in \kappa^{i-1}(w_1)$ and $w'_2 \in \kappa^{i-1}(w_2)$ be two vertices in the base level. Then there is a path $w'_1 = u_1, u_2, \ldots, u_n = w'_2$ in $R(v)$ from w'_1 to w'_2. Consider the vertices $u'_j = \pi^{i-2} \ldots \pi^1 \pi^0(u_j)$ in V_{i-1}. For each pair (u_j, u_{j+1}), either $u'_j = u'_{j+1}$ or (u'_j, u'_{j+1}) is an edge in $\kappa(v)$. Thus, by deleting repetitions, the sequence $u'_1, \ldots, u'_n$ yields a path in $\kappa(v)$ from $w_1 = u'_1$ to $w_2 = u'_n$. ∎

The second property in theorem 5.1 (connectivity of receptive fields) is the one we are interested in; yet the first one (connectivity of the set of children of a given vertex) is the most manageable one, because it is a local property. Therefore, in the sequel, only connectivity of sets of children will be discussed.

Consider a hierarchy of graphs G_i in which each vertex represents a connected region in the base level. Suppose $\pi(v) = p_{\text{old}}$ for some $v \in V_k$, for some k. Consider the adapted stack of graphs G'_i which is constructed from G_i by putting $\pi(v) = p_{\text{new}}$ and adapting the edge structure accordingly. In a relinking process, the hierarchy of graphs is adapted repeatedly in this way, until an optimal segmentation is achieved. All vertices can be relinked to a new parent simultaneously or they can be relinked one at a time.

We now discuss, for which vertices the connectivity of the receptive field is lost by this relinking step. Edges change only in levels k and higher and the receptive fields of vertices in the levels $i \leq k$ do not change. Vertices in level $k + 1$ and vertices in levels $i > k + 1$ will be discussed separately.

5.2.1. Connectivity Preservation in the Parent Level

Suppose a vertex v in level k with parent p_{old} is relinked to a new parent p_{new}. In level $k+1$, the receptive fields of p_{old} and p_{new} change. After the relinking step,

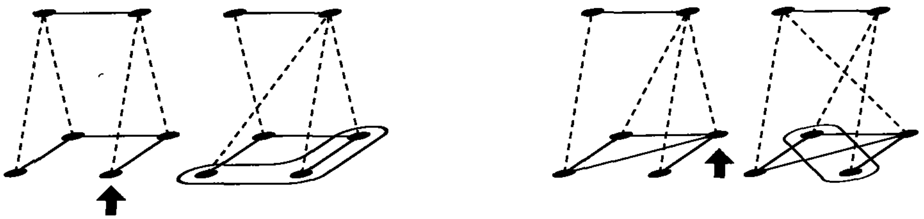


Figure 5.2: The two ways in which connectivity can be lost at the parent level by relinking. The receptive fields that become disconnected by a relinking step are marked.

the children of p_{old} are $\kappa(p_{\text{old}}) \setminus \{v\}$; the children of p_{new} are $\kappa(p_{\text{new}}) \cup \{v\}$. The receptive fields of other vertices in this level do not change.

Connectivity of receptive fields in level $k + 1$ is lost in two cases (see figure 5.2). The first case (shown to the left) is the situation where $\kappa(p_{\text{new}}) \cup \{v\}$ is not connected. This happens when v is not connected to some vertex in $\kappa(p_{\text{new}})$, i.e. when p_{new} is not the parent of some neighbor of v.

The second case (shown to the right) is the situation where removing v from $\kappa(p_{\text{old}})$ changes the number of connected components of $\kappa(p_{\text{old}})$. Such a vertex v is called an ***articulation point*** or ***cut point*** [36] of $\kappa(p_{\text{old}})$. Note that this also includes the case where v is the only child of p_{old}, in which case p_{old} would have an empty receptive field after the relinking step.

The time required for the computation of the articulation points of the subgraph $\kappa(p_{\text{old}})$ is $\mathcal{O}(|\kappa(p_{\text{old}})|)$ [94]. They need only be recomputed when $\kappa(v)$ is changed by a relinking step.

Summarizing, retaining the connectivity of the receptive fields of vertices in level $k + 1$ imposes two conditions on the relinking rules:

1. A vertex v which is an articulation point of $\kappa(\pi(v))$ may not be relinked to a new parent.
2. A vertex v may choose only a new parent from the set $\pi(N(v))$, i.e. a parent of a neighbor of v.

Wharton [109] has tried to find a connectivity preserving relinking method, but overlooked the second criterion, and the possible loss of connectivity on higher levels of the pyramid which will be discussed in the following subsection.

It may be desirable to perform relinking for many vertices of a level in parallel. Then, extra care must be taken in order not to loose connectivity: two vertices which can be relinked individually from a given situation may not always be relinked simultaneously. Only a subset of the vertices in a level can be relinked to a new parent in each step. A safe strategy is to relink a vertex v only if none of the other vertices in $\kappa(\pi(v))$ or $N(v)$ is relinked.

A suitable subset of vertices which may be relinked in parallel can be computed by a stochastic procedure, similar to stochastic decimation [68, 76] as described in chapter 4. Each vertex v for which a relinking might be performed draws a random variable from some distribution. If that random number is larger than that drawn by all the vertices in $\kappa(\pi(v)) \cup N(v)$ which may not be relinked simultaneously, the vertex v 'wins the right' to be relinked, and the other vertices in $\kappa(\pi(v)) \cup N(v)$ are prohibited to relink. New random numbers can be drawn and new vertices selected as long as there are vertices which are neither selected for relinking nor prohibited to relink.

5.2.2. Connectivity Preservation in Higher Levels

Thus far, we have ignored the connectivity of receptive fields of vertices at levels $i > k + 1$. If the pyramid is constructed bottom-up, as will be the case in section 5.3, such levels do not exist at the moment of relinking: first, levels 0 and 1 are built and the parent-child links between levels 0 and 1 are processed by relinking. Then level 2 is constructed and the parent-child links between levels 1 and 2 are processed by relinking, et cetera. In this case, no higher levels are present during the relinking steps.

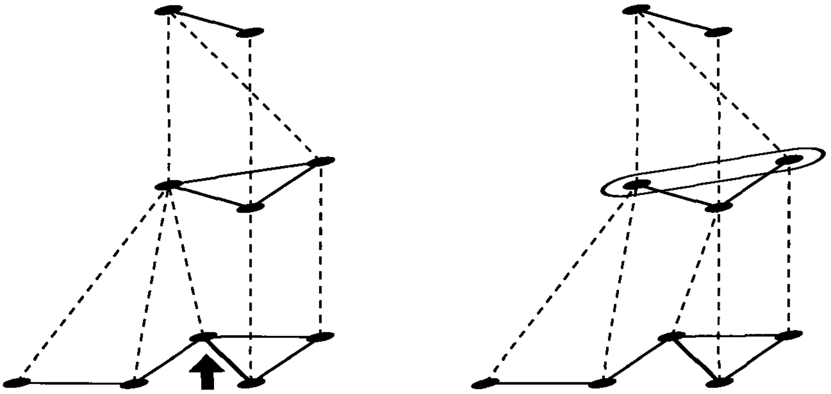


Figure 5.3: If edges in the parent level disappear by a relinking step, the connectivity in some higher level can be lost.

If relinking is performed when higher levels are present, the connectivity of receptive fields in higher levels of the pyramid can be lost, as illustrated in figure 5.3. Suppose a vertex $v \in V_k$ with parent $p_{\text{old}} \in V_{k+1}$ is relinked to a new parent $p_{\text{new}} \in V_{k+1}$. For a vertex $w \in V_{k+n+1}$, $n > 1$, the set $\kappa(w)$ is not changed by relinking. Connectivity of the receptive field of w can only be lost by the removal of edges between vertices in $\kappa(w)$. On level $k + n$, an edge which might be removed has the form $(\pi^n(v), \pi^n(v'))$, where $v' \in N(v)$. Removal of this edge may destroy the connectivity of $R(\pi^{n+1}(v))$. If the connectivity of a connected graph is lost by the deletion of a single edge, this edge is called a bridge [36]. The edge is indeed removed if all the edges in level k between $\kappa^n\pi^n(v)$ and $\kappa^n\pi^n(v')$ have v as an end point, i.e. $\kappa^n\pi^n(v) \cap N(\kappa^n\pi^n(v')) = \{v\}$. Moreover, v must not end up in the receptive field of $\kappa^n(v)$ or $\kappa^n(v')$ after the relinking step. Therefore, the condition $\pi^{n-1}(p_{\text{new}}) \notin \{\pi^n(v), \pi^n(w)\}$ must hold.

Summarizing, relinking of vertex $v \in V_k$ to $p_{\text{new}} = \pi(v')$, with $v' \in N(v)$, destroys the connectivity of the receptive field of $\pi^{n+1}(v)$ if the following conditions are satisfied:

1. $(\pi^n(v), \pi^n(v'))$ is a bridge in $\kappa\pi^{n+1}(v)$.
2. $\kappa^n\pi^n(v) \cap N(\kappa^n\pi^n(v')) = \{v\}$.
3. $\pi^{n-1}(p_{\text{new}}) \notin \{\pi^n(v), \pi^n(v')\}$.

These conditions can be tested for each vertex v and new parent p_{new}. For each vertex v this requires inspection of a set of vertices of the form $\pi^m(v)$, up to the level where $|\pi^m(N(v) \cup \{v\})| \leq 2$. Using these conditions, the set of possible parents for each vertex can be restricted in such a way that connectivity of receptive fields is not destroyed by relinking.

Just like we described in the previous subsection, it is possible to generate sets of vertices which can be relinked in parallel. Again, there are combinations of vertices which cannot be relinked simultaneously. As before, a random procedure can be used to compute such a set. In the present situation, however, vertices which may not be relinked simultaneously can be far apart, and propagation of information through the levels of the pyramid is required to compare the random labels for all mutually exclusive pairs. This makes the resulting algorithm is rather complicated.

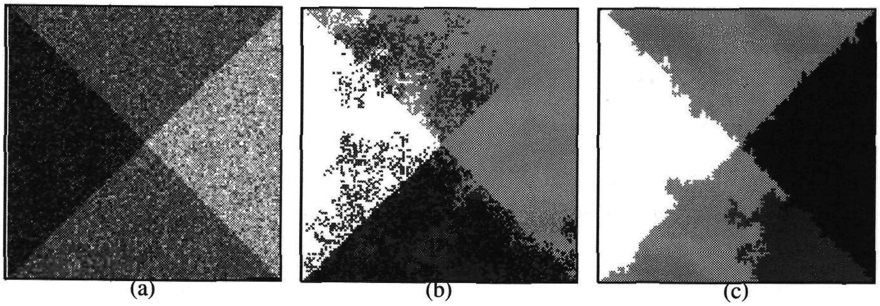


Figure 5.4: With the classical relinking algorithm, noise can scatter points of each class in the image. With the new method, classes are connected. From left to right: a synthetic image, corrupted with noise; its segmentation by the classical algorithm; its segmentation by the connectivity preserving algorithm.

5.2.3. The Relinking Method

In the previous subsections, it has been described how a set of candidate parents for a vertex can be computed. This subsection describes the relinking procedure. In the present method, vertices are visited one by one. For each vertex, a set of allowed candidate parents is computed, depending on the actual structure of the hierarchy. This set plays the same role as the fixed set of four (disregarding boundary effects) candidate parents in the classical relinking scheme. Then, a new parent is chosen from the set of allowed candidate parents. The vertex is relinked to the new parent and the graph structure and attributes of vertices are updated accordingly. This procedure is repeated until a stable configuration is reached.

The new parent p_{new} is chosen for each vertex v such that the difference between $g(v)$ and $g(p_{\text{new}})$ is minimized.

The rules which determine the selection of a new parent for a vertex can be formulated as an energy minimization problem. For each level m in the hierarchy, an energy

$$E_{\text{region}}(m) = \sum_{v \in V_m} n(v) \left[g(v) - g(\pi(v))\right]^2, \tag{5.1}$$

can be defined. Here, $n(v)$ denotes the area of the receptive field of v and $g(v)$ denotes its grey level.

The choice method used here is a steepest descent method: a vertex $v \in V_k$ is relinked to a new parent in such a way that $E_{\text{region}}(k)$ is reduced as much as possible in each step. This can cause the energy to converge in a local minimum. Spann [96] proposed a stochastic relinking algorithm, similar to simulated annealing, which tries to avoid local minima.

The relinking method can be shown to converge by an argument due to Cibulskis and Dyer [20]. Whenever a vertex in level k is relinked, the energies $E_{\text{region}}(i)$ with $i < k$ are not changed, while the energy $E_{\text{region}}(k)$ is reduced. This implies that the energy of the base level is non increasing. As there is only a finite number of possible configuration for the links between the two lower levels, these links must reach a stable configuration in a finite number of steps. When this has happened, $E_{\text{region}}(0)$ will remain constant and the energy $E_{\text{region}}(1)$ will start to decrease monotonically. Therefore, the links between G_1 and G_2 will finally reach a stable configuration, et cetera.

A different way of formulating the region based relinking criteria uses the score Q_{region} defined by

$$Q_{\text{region}} = -|g(v) - g(p_{\text{new}})|. \tag{5.2}$$

The new parent will be the one which maximizes this score. When the maximal score equals $-|g(v) - g(p_{\text{old}})|$ for all vertices, the relinking procedure has converged.

The effect of connectivity preservation in relinking can be seen from figure 5.4. The initial image contains four regions of grey values 64, 128, 128 and 192, corrupted with Gaussian noise with standard deviation $\sigma = 32$. If the image were divided in four squares along horizontal and vertical lines, the initial configuration of the parent-child links would already represent the correct segmentation, and no conclusions can be drawn from the segmentation of such an image. Therefore, the image is segmented along diagonal lines. The middle image shows, in false grey scale, the subdivision in 4 classes computed by the classical algorithm of Burt *et al.* [18]. The image to the right shows the classification into four classes, obtained by connectivity preserving relinking. In can be seen that Burts algorithm scatters the points in each class over the image, while the connectivity preserving method generates connected classes, which are less affected by noise.

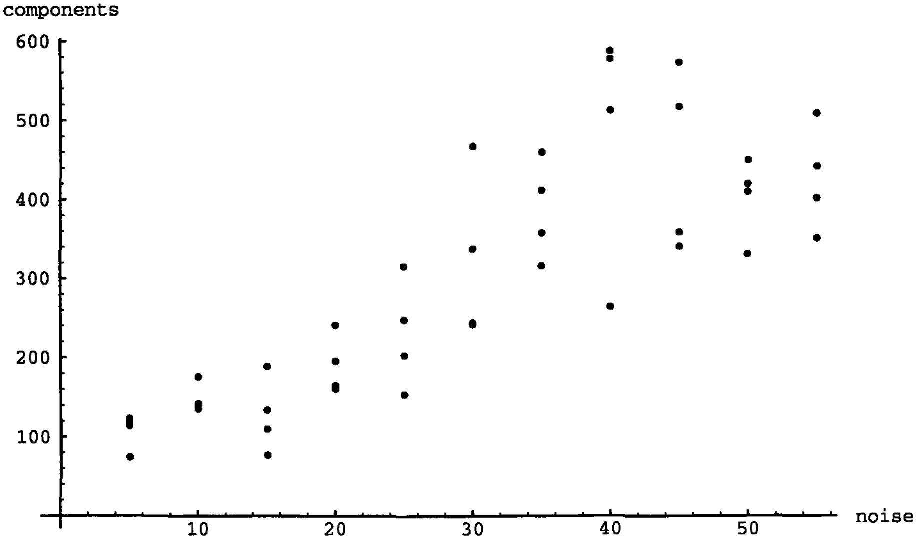


Figure 5.5: The number of connected components generated by Burt's algorithm at various noise levels

The number of 8-connected components generated by Burts relinking algorithm depends on the amount of noise added. In figure 5.5, the number of connected components is shown for a number of realizations of Gaussian noise with varying standard deviations. It can be seen that the number of connected components is large, even for very low noise levels. Moreover, the number of connected components for a given noise level assumes values in a rather broad range.

It can also be seen that the boundaries detected by the connectivity preserving method are not always smooth. This is caused by the fact that relinking is a steepest descent procedure. In order to remove the indentations present in some parts of the boundary, a number of relinking steps would have to be performed in which the difference between parent and child grey values is incremented temporarily.

5.3. Adaptive Construction of Successive Levels

As mentioned before, classical relinking suffers from a number of problems. Firstly, not all segmentations can be represented in a regular structure: elongated objects cannot be represented [8]. Moreover, the number of regions in each level is fixed. Originally, relinking was used for images containing a single light object on a dark background, for which this restriction is not a problem, but for segmentation of more general images, the number of regions must be adapted to image contents. Various authors [6, 44, 97] have described methods for root detection, i.e. the marking of cells in the pyramid as the representatives for a region in the final segmentation. This can improve the results, but the methods still operate in a regular structure.

These problems can be avoided if a hierarchy of region adjacency graphs is built level by level, adaptive to image contents. In this section, level by level construction of a hierarchy is combined with relinking. This is done as follows: from the base level G_0, the next level G_1 is constructed and the parent-child links between these levels are initialized. The parent-child links which are created in this phase have the same role as the regular structure in the classical relinking scheme: they serve as an initial configuration which is adapted by relinking. This adaptation is performed by the procedure described in section 5.2. Then the second level G_2 is constructed. The parent-child links between G_1 and G_2 are initialized and updated by relinking, et cetera.

In order to create the vertices of a new level in the hierarchy, the vertices of the lower level are partitioned into a number of connected sets or *clusters*. Each of these clusters will be the set of children of a vertex on the next level. Like in graph decimation [76], each cluster will consist of central vertex and a number of its neighbors.

In order to avoid adjacent, but dissimilar regions to be merged into a single cluster, such vertices are forbidden to become members of a single cluster. Dissimilarity of adjacent vertices is defined using an edge strength measure.

We consider two choices for the edge strength. The first one is defined by

$$S^{\mathrm{sub}}_{\mathrm{global}}(v,w) = |g(v) - g(w)| - \frac{1}{2}(\sigma(v) + \sigma(w)), \tag{5.3}$$

where $g(v)$ is the average grey value within the receptive field of a vertex and $\sigma(v)$ is the standard deviation of the grey value. The second choice is

$$S^{\mathrm{div}}_{\mathrm{global}}(v,w) = \frac{|g(v) - g(w)|}{1 + \frac{1}{2}(\sigma(v) + \sigma(w))}. \tag{5.4}$$

Both measures depend on the difference of the grey values of the regions, corrected for variations within each region. The measure S_{div} has the advantage that it is dimensionless. On the other hand, the estimation for $\sigma(v)$ and $\sigma(w)$ can be bad for small regions, yielding a large uncertainty in the value of $S^{\mathrm{div}}_{\mathrm{local}}$.

If each vertex stores the size, the average grey value and the standard deviation of its receptive field, these values can be recomputed for each vertex by considering only the corresponding values of its children after each relinking step. Therefore, local communication in the hierarchy suffices for the execution of these computations.

In the lower levels of the hierarchy, receptive fields are very small, maybe just a single pixel. Small receptive fields are relatively homogeneous, so the corresponding edge strengths are high. Therefore, the strength of the boundary between small receptive fields must be corrected for region size. This is achieved by multiplying all edge strengths with a geometry factor of the form

$$\left(1 - \frac{1}{A(v) + \delta}\right) \times \left(1 - \frac{1}{A(w) + \delta}\right), \tag{5.5}$$

where $A(v)$ is the area of the receptive field of v and δ is some small number. This factor approaches 1 for large regions, but is small for small regions. For single pixel receptive fields it is approximately equal to δ^2. Thus, survival of small noise regions to higher levels of the hierarchy is suppressed. Vertices are said to be dissimilar when the strength of the edge between them is larger than some predefined value.

Now that dissimilarity of vertices has been defined, the determination of clusters can be performed. Clusters will have the following properties:

1. For each cluster $C \subset V$, there is a center vertex $c \in C$ such that $C \subset N(c) \cup \{c\}$.
2. No cluster contains a pair of adjacent, dissimilar vertices.
3. Two center vertices c_1 and c_2 can be neighbors only if the cluster of c_1 contains a dissimilar neighbor of c_2 or vice versa.

The difference with graph decimation is that, in graph decimation, pairs of dissimilar vertices in a cluster are forbidden only if one of these vertices is the center of the cluster, while here, all pairs in a cluster are to be considered. If conventional graph decimation were used here, undesirable merges of regions can occur when there is a small region on the boundary between two large dissimilar regions, and this small region is selected as the center of a cluster.

The computation of clusters is similar to stochastic decimation, but some changes have to be made in order to avoid dissimilar vertices to be put in a single cluster. Clusters are computed by repeated application of the following steps

1. Every vertex v_i which is not yet part of a cluster is given some label λ_i.
2. Every vertex whose label is larger than that of all of its similar neighbors is selected as the center of a new cluster.
3. For each newly selected center c, a maximal subset of $N(c)$ containing no dissimilar pairs is added to complete the cluster.

The label λ_i can be a random number, but it can also be some image dependent value. Throughout this chapter, the area of the receptive field of a vertex is used as its label. An additional random number is assigned to each vertex for resolving ties between regions of equal size.

The difference with the stochastic decimation procedure is in the order of the steps. In stochastic decimation, the computation of a maximal independent set by repeated selection of local maxima is completed before the clusters are computed by assignments of neighbors; here, a number of clusters is computed in each selection of local maxima.

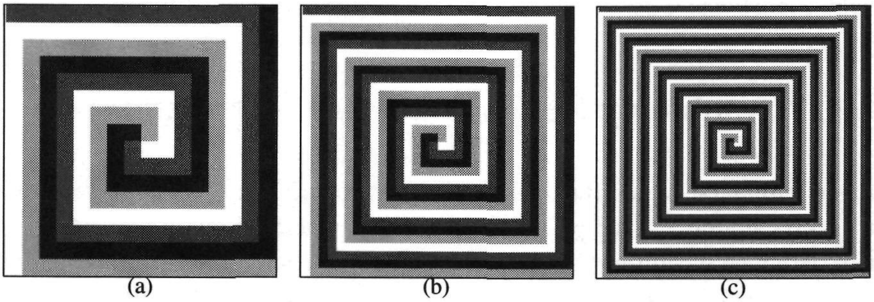


Figure 5.6: Three 128 × 128 images, each containing four regions.

The effect of adaptive construction of subsequent levels is shown by comparing the segmentations of some spiral images (figure 5.6). These are 128×128 images, each containing four regions. Figure 5.7 shows the segmentation of these images. The top row shows the segmentation according to the method

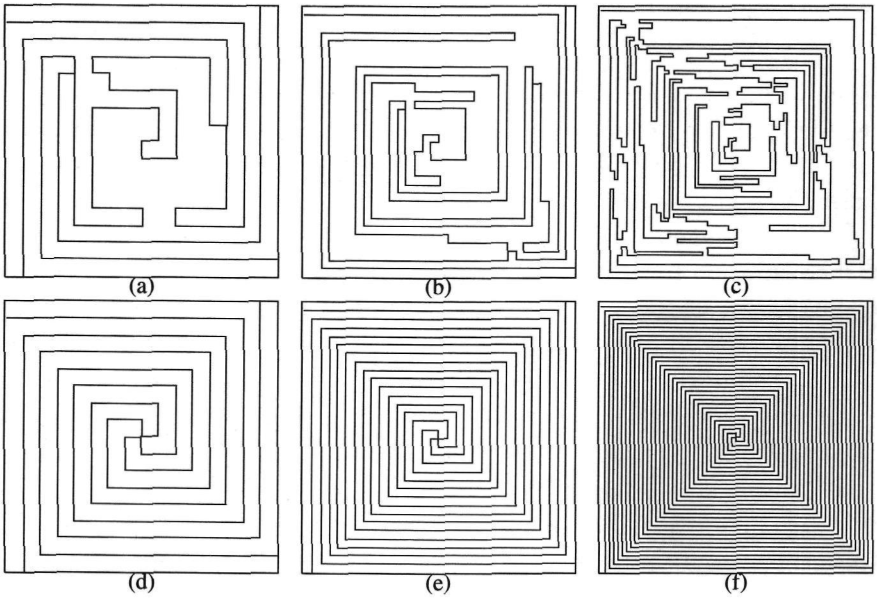


Figure 5.7: Segmentation of images with four connected regions using the regular pyramid (top row) and adaptive construction of levels (bottom row).

described in the previous section; the bottom row shows the segmentation computed with adaptive construction.

In can be seen that the regular pyramid is too rigid for the detection of strongly elongated regions. The adaptively constructed pyramid, on the other hand, finds these regions exactly.

5.4. Combining Region and Edge Information

Image segmentations can be based on a variety of criteria. Two important groups of criteria are those using properties of regions and those using properties of the boundaries between regions. In a hierarchy of graphs formalism, both types of information can be represented. In this section, it is described how region and boundary information can be combined in a relinking scheme.

Region based methods presuppose a model for homogeneous regions. It can for example be assumed that each region has a uniform grey value, or that the grey value of each pixel is drawn from some distribution, which is the same for each pixel in a homogeneous region. Models for different types of images, such as flow fields [38] or textured images [84], can also be used. In relinking methods using region information, the new parent is chosen in such a way that the fit of the receptive fields of the old and new parent vertices are improved in each step.

Segmentation methods based on boundary information detect differences between the properties of adjacent locations in the image, for example by the application of an edge detection filter. Boundaries between regions are then detected, for example by the application of a peak filter on the gradient or by the computation of the zero crossings of a second order difference operator, which correspond to the extrema of a first order difference operator. Segmentation methods using peak filters can have problems with the generation of compact regions, because if a small part of the boundary between two regions is missing, these regions will be merged. Segmentation methods based on zero crossings do generate compact regions, but have problems with situations where three regions meet in a point.

In conventional relinking schemes, there is no representation available for connected regions and the boundaries between such regions. Therefore, these schemes cannot handle boundary information. Such information can be represented in a hierarchy of graphs. This was done for the first time by Montanvert and Bertolino [75], using a randomized edge detector which can detect edges at all scales simultaneously [7].

The segmentation method presented in the previous section consists of two parts: the relinking procedure and the procedure for the adaptive construction of new levels. In both parts, the combination of region and edge information can be used.

5.4.1. Combining Region and Boundary Information for Relinking

In the previous section, we described the selection of a new parent in terms of energy minimization. The same approach will be taken here in order to incorporate boundary information in the relinking procedure.

In grey level based relinking as described in section 5.2, the energy being minimized was.

$$E_{\text{region}}(m) = \sum_{v \in V_m} n(v) \left[g(v) - g(\pi(v)) \right]^2, \tag{5.6}$$

which corresponds with a choice criterion using the score

$$Q_{\text{region}} = -|g(v) - g(p_{\text{new}})|. \tag{5.7}$$

Boundary-based segmentation evaluates regions by considering the edges surrounding each region. The response of an edge detector should be strong on the boundary of a region, while it should be weak in the interior of a region.

The strength of each edge is measured as the average response of an edge detector along the boundary represented by the edge. For the results presented here, the Sobel edge detector was used. Boundary-based image segmentation in a hierarchical structure can be performed by minimization of the sum over all regions of the average edge strength over the boundary of each region. When each edge carries its length and its average edge detector response as attributes, the average edge response of an edge can be computed recursively. When relinking is performed for a given level in a hierarchy, each vertex on the upper level corresponds to a region, and the boundaries of such regions are composed of active edges in the lower level.

Let $\eta(R(v))$ denote the average response of an edge detection filter along the receptive field of a vertex v. Our boundary based relinking criterion will be based on the minimization of the energy

$$E_{\text{boundary}}(m) = \sum_{v \in V_{m+1}} \eta(R(v)), \tag{5.8}$$

In each level of the hierarchy, the boundary around a vertex v, represented by edges between v and its neighbors, consist of boundary parts on the previous level. These boundary parts correspond to edges between vertices in $\kappa(v)$ and vertices not in $\kappa(v)$. An edge (v, w) for which $\pi(v) = \pi(w)$ is not part of some boundary in the next level; such an edge is called inactive. If $\pi(v) \neq \pi(w)$, the edge (v, w) represents a boundary fragment which is part of the boundary between the receptive fields of $\pi(v)$ and $\pi(w)$. Such an edge is called an active edge.

If a vertex v is relinked from an old parent p_{old} to a new parent p_{new}, some active edges become inactive and vice versa. The edges which become inactive are the edges (v, v') with $\pi(v') = p_{\text{new}}$; the edges which become active are the edges (v, v') with $\pi(v') = p_{\text{old}}$.

When a vertex v is relinked from p_{old} to p_{new}, the only average edge strengths which change are those of the receptive fields of p_{old} and p_{new}. Thus, the total change in the average edge strength is

$$\eta\big(R(p_{\text{old}}) \setminus R(v)\big) + \eta\big(R(p_{\text{new}})\big) - \eta\big(R(p_{\text{old}})\big) - \eta\big(R(p_{\text{new}}) \setminus R(v)\big), \tag{5.9}$$

where $\eta(\cdot)$ denotes the average edge strength along the boundary of a region.

In order to maximize the average edge strength, in each time the new parent must be chosen such that the score

$$Q_{\text{edge}} = \eta\big(R(p_{\text{new}})\big) - \eta\big(R(p_{\text{new}}) \setminus R(v)\big) \tag{5.10}$$

is maximal. When the maximal score is equal to $\eta\big(R(p_{\text{old}})\big) - \eta\big(R(p_{\text{old}}) \setminus R(v)\big)$ for every vertex v , the relinking procedure has converged. Convergence of this relinking procedure can be proven with the argument used before.

Combining region and edge information in a relinking method can be done by using a combined score

$$Q_\alpha = \alpha Q_{\text{edge}} + (1-\alpha) Q_{\text{region}}, \tag{5.11}$$

where α is a weight factor between 0 and 1. Again, the relinking method can be shown to converge with the argument used before [20]

5.4.2. Combining Region and Boundary Information for Construction of New Levels

The second part of our segmentation method is the construction of new levels, which is performed adaptively to image contents through the influence of vertex dissimilarities. It will now be shown how region and boundary information can be combined in the definition of these dissimilarities. This can be done simply by choosing a region dissimilarity measure which depends on both region and edge information.

In section 5.3 we considered the edge strengths

$$S^{\text{sub}}_{\text{global}} = |g(v) - g(w)| - \frac{1}{2}(\sigma(v) + \sigma(w)) \tag{5.12}$$

and

$$S^{\text{div}}_{\text{global}} = \frac{|g(v) - g(v)|}{1 + \frac{1}{2}(\sigma(v) + \sigma(w))} \tag{5.13}$$

These edge strengths are based on the average grey values of adjacent regions, i.e. on global properties. Local properties of the boundaries between adjacent regions can be measured by considering the average response of an edge detector along this boundary. Let $s_2(v,w)$ denote the average response of an edge detector along the boundary of the receptive fields $R(v)$ an $R(w)$ and let $s_1(v)$ denote the average response within the receptive field $R(v)$.

Then the strength of an edge, corrected for the structure within regions, can be expressed as

$$S^{\text{sub}}_{\text{local}}(v,w) = s_2(v,w) - \frac{1}{2}(s_1(v) + s_1(w)) \tag{5.14}$$

or as

$$S^{\text{div}}_{\text{local}}(v,w) = \frac{s_2(v,w)}{1 + \frac{1}{2}(s_1(v) + s_1(w))}. \tag{5.15}$$

A combined dissimilarity measure can then be defined as

$$S^{\xi}_{\alpha}(v,w) = \alpha S^{\xi}_{\text{local}}(v,w) + (1-\alpha) S^{\xi}_{\text{global}}(v,w), \tag{5.16}$$

where ξ denotes either ‘div’ or ‘sub’. In all cases, the value of α used here will be the same values as the one used previously for weighing scores in the relinking method described earlier in this section.

5.5. Results

The elements described in the previous sections have been combined in a segmentation algorithm. A hierarchy is constructed in a bottom-up order. Alternatingly, a new level is constructed by decimation of a region adjacency graph, and the parent-child links between the lower and the upper level are updated by relinking. Relinking is performed in a connectivity-preserving manner, and region and edge information are combined both in the construction of the similarity graph and during relinking. This section presents the segmentation results for some synthetic and natural images.

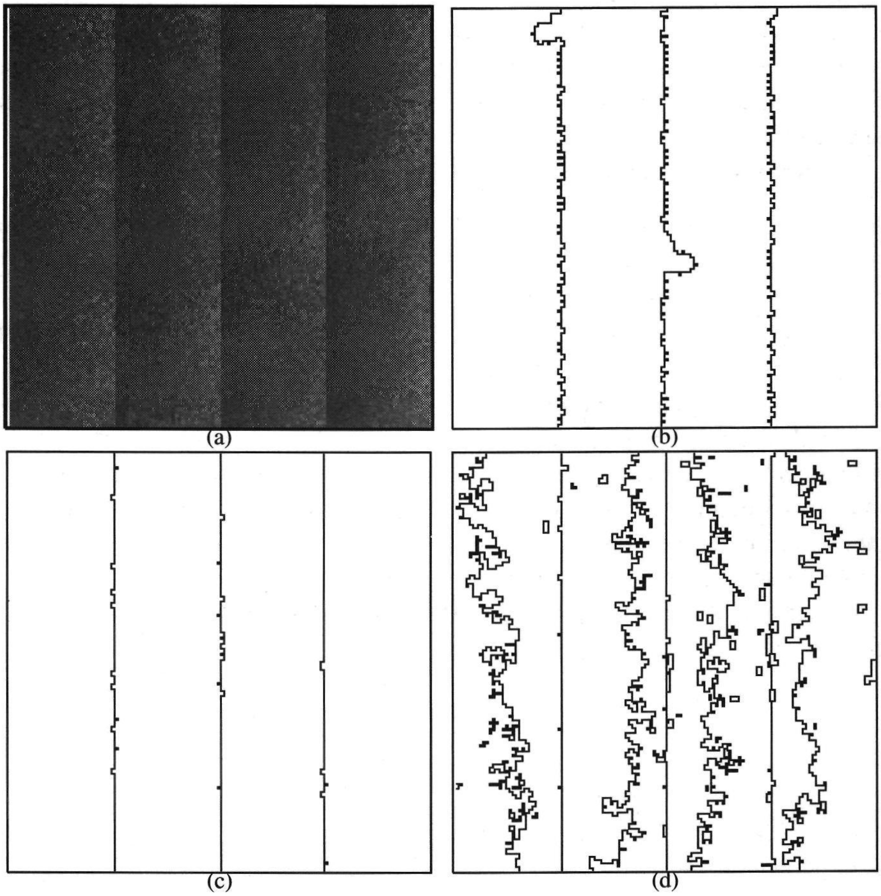


Figure 5.8: A synthetic image (a), contaminated with noise and its segmentations with $\alpha = .95$ (b), $\alpha = .50$ (c) and $\alpha = .10$ (d).

The effect of combining region and edge information is illustrated in figure 5.8. It shows an 128 × 128 image consisting of four ramps with grey values from 112 to 144, contaminated with Gaussian noise with standard deviation $\sigma = 8$. The four regions cannot be discerned based on their average grey values, because these are all the same. Therefore, boundary information must be used. On the other hand, in the lower levels, information on average grey values is more suitable for the classification of pixels, because boundary information is rather noisy for small regions. Therefore, some intermediate value of α must be used for the optimal segmentation of the ramps image.

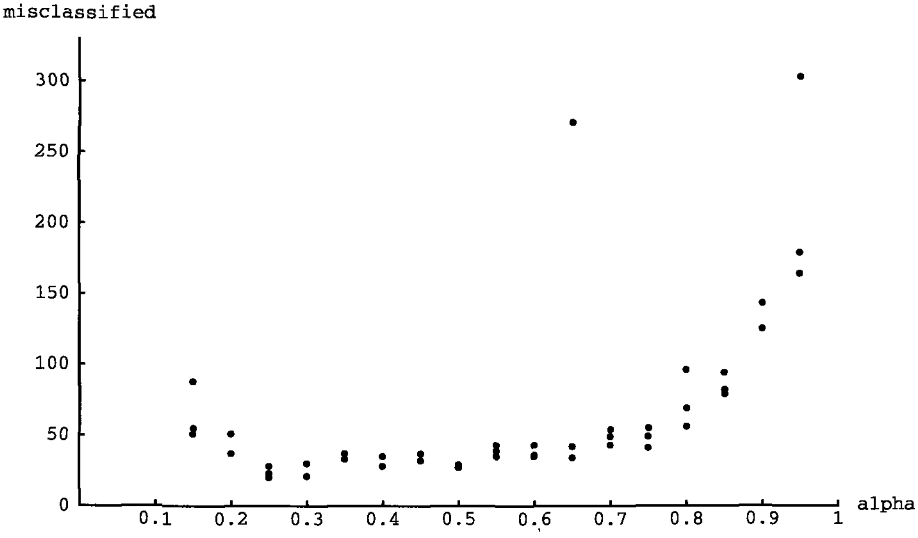


Figure 5.9: The number of misclassified pixels as a function of the parameter α with edge strength $S_{\alpha}^{\mathrm{sub}}$.

The segmentations, computed with edge strength $S_{\alpha}^{\mathrm{sub}}$, are shown for $\alpha = 0.95$, $\alpha = 0.5$ and $\alpha = 0.1$. If α is too large, the region boundaries are disturbed because of disturbances in the lower levels of the hierarchy. If, on the other hand, α is too small, the average grey level criterion dominates and the regions are split. Increasing the threshold with low values of α causes all the regions to be merged, possibly leaving some small noise segments.

Figures 5.9 and 5.10 show the number of misclassified pixels (with respect to the original four bands image) for various values of α and for different realizations of the noise. Figure 5.9 shows the situation for the strength measure $S_{\alpha}^{\mathrm{sub}}$ and figure 5.10 shows the situation for $S_{\alpha}^{\mathrm{div}}$. It can be seen that the quality of the segmentation is good for a wide range of α values. Note that, in one case in figure 5.9, the segmentation with $\alpha = 0.65$ has a large number of misclassified pixels. This is caused by a 'wrong' link in a high level of the hierarchy, in which a large receptive field is involved.

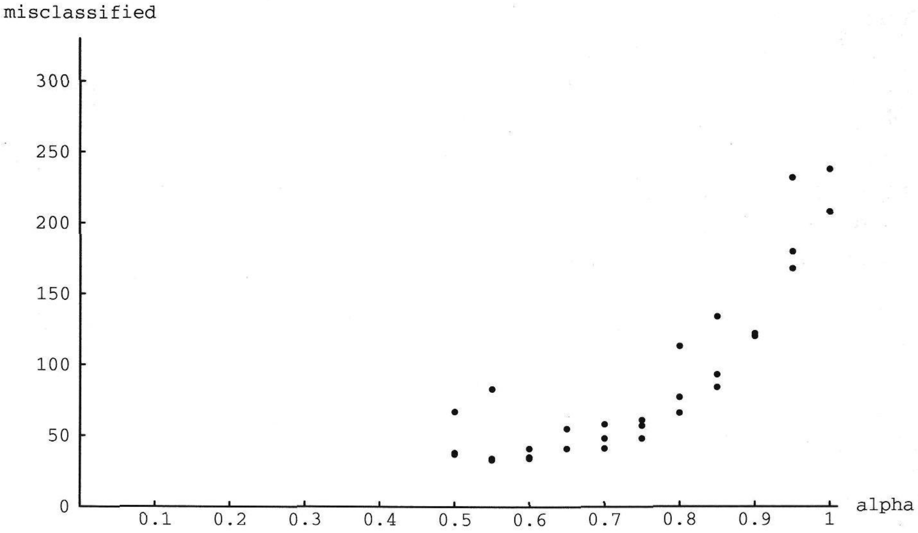


Figure 5.10: The number of misclassified pixels as a function of the parameter α with edge strength S_α^{div}.

When S_α^{div} is used, a satisfactory segmentation can be obtained with $\alpha = 1$. On the other hand, the segmentation method breaks down for $\alpha < 0.5$. A wider range of possible values for α is available when S_α^{sub} is used.

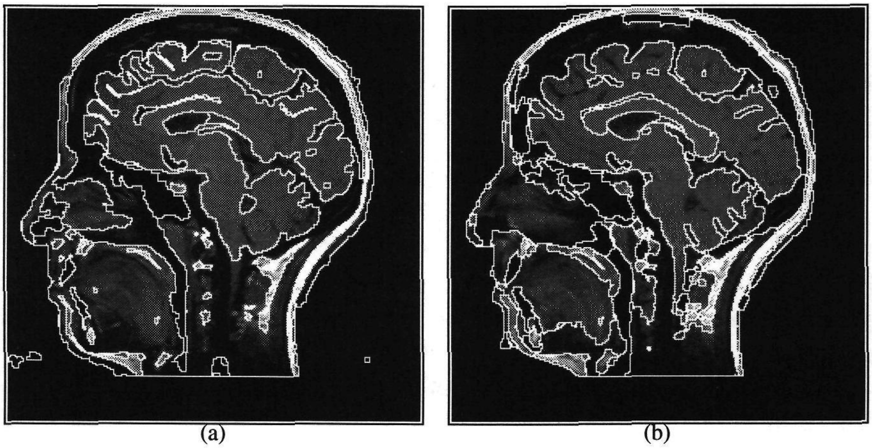


Figure 5.11: Segmentation with edges strengths $S_{0.5}^{\text{div}}$ (a) and $S_{0.5}^{\text{sub}}$ (b).

Figure 5.11 shows the results obtained for the head image, both for the edge strength $S_{0.5}^{\text{div}}$ (figure 5.11a, threshold 1.5) and the edge strength $S_{0.5}^{\text{sub}}$. (figure 5.11b, threshold 20).

Figure 5.12 shows the segmentation result for an image of muscle tissue without (a) and with (b) relinking. As the individual muscle fibers do not have a strictly homogeneous grey value, edge information must be emphasized more in the segmentation process. Therefore, $\alpha = 0.75$ was used.

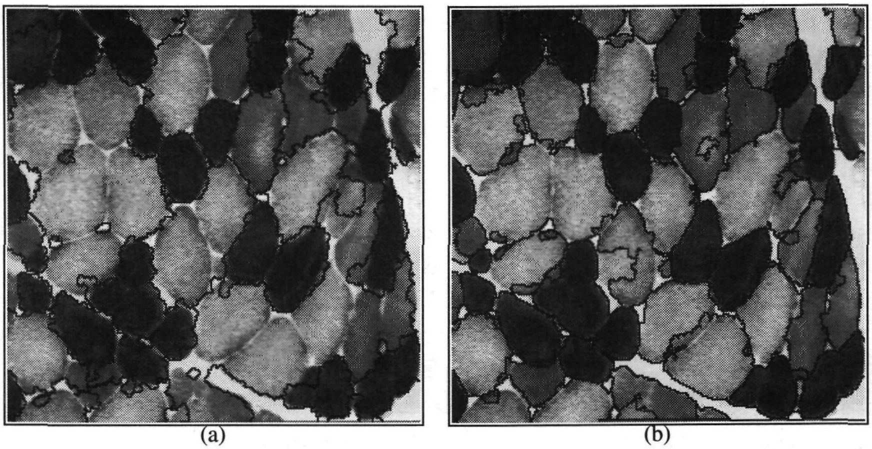


Figure 5.12: Segmentation for the muscle image without (a) and with (b) relinking. The edge strength S_{α}^{div} was used with $\alpha = 0.75$ and threshold 1.25.

It can be seen that the construction without relinking extracts some of the structures from the image, but improvement can be made by applying relinking. The segmentation obtained when relinking is used shows more individual fibers than the segmentation obtained without relinking. This can be understood as follows. Each time a level is constructed by graph decimation, the boundaries between the resulting regions are not located optimally. Therefore, the response of an edge detection filter will not be maximal on the boundaries found by the procedure, but more in the interior of some regions. Similarly, the regions found by graph decimation do not correspond exactly with the boundaries between homogeneous regions in the image. Therefore, a region detected by the decimation process can contain pixels from different regions in the image. This causes an increase in the standard deviation of the grey levels in a region.

If the boundaries are not corrected in the subsequent relinking phase, these effects reduce the edge strengths S_{α}^{div} and S_{α}^{sub}. More adjacent vertices will be similar and therefore, more regions will be merged in the construction of subsequent levels.

5.6. Conclusions

In this chapter, we have presented the extension of pyramid relinking to hierarchies of graphs. This approach solves a number of problems associated with conventional relinking methods.

The first problem we solved was the problem of connectivity preservation. Conventional relinking does not produce connected regions, but classes which may be scattered in the image plane. We have been able to adapt the relinking rules in such a way that connectivity of regions is guaranteed. When the pyramid is built level by level, two criteria (illustrated in figure 5.2) must be used to select suitable relinking steps. When relinking is performed in a complete pyramid, in which all levels are present, a third criterion (illustrated in figure 5.3) must be used in order to avoid the destruction of the connectivity of receptive fields of vertices in higher levels.

The artifacts from which relinking in regular pyramids suffers were avoided by constructing the hierarchy level by level, adaptive to image contents. The construction of a new level can be seen as a first guess for the aggregation of region primitives which is improved by the relinking procedure.

The solution to the connectivity preservation problem required the introduction of a graph structure. The edges in this graph correspond with the boundaries between receptive fields in the image plane. This provides the possibility to represent information on these boundaries in the data structure and to combine region and boundary information in the segmentation process. It has been shown that the combination of region and edge information can be useful and that the corresponding segmentation scheme is robust under the change of the parameter α.

Some segmentation results for natural images have been presented, showing that satisfactory results can be obtained in practice.

6

Top-Down Processing in Hierarchical Structures

6.1. Introduction

In the previous chapter, we have investigated methods for the description of an image by a hierarchy of region adjacency graphs. Such a description reflects the structure of the image, but some effort must be made to extract relevant information.

Each vertex in the hierarchy of graphs corresponds with a region in the image. Often a significant object in the image will coincide with such a region, but sometimes this is not the case. It is possible that an object is represented by several vertices in one level, while it is only a part of the receptive field of a vertex in the next level. Moreover, not all vertices have the same significance. Some search method is required to detect the most significant structures in the hierarchy.

In this chapter, some examples of top-down search methods in a hierarchy of graphs are discussed. These methods use a priori knowledge of image content and are therefore adapted to a particular problem. Our specific application is the detection of fibers in a microscope image of muscle tissue (figure 6.1). The individual fibers appear in the image as more or less convex regions with a uniform grey value which are bounded by a curve on which the image gradient is large.

In section 6.2, a top-down tree traversal method is described which searches for clusters of vertices which form more or less convex regions in the image plane and which fall within a given size range. The information used by this method is only of a geometric nature. The information on the grey level distribution of the image which was used for the construction of the hierarchy, is no longer used.

In section 6.3, a method for boundary refinement based on dynamic programming is presented. Dynamic programming (e.g. [21]) is a well-known technique for many optimization problems and has been used for a long time in

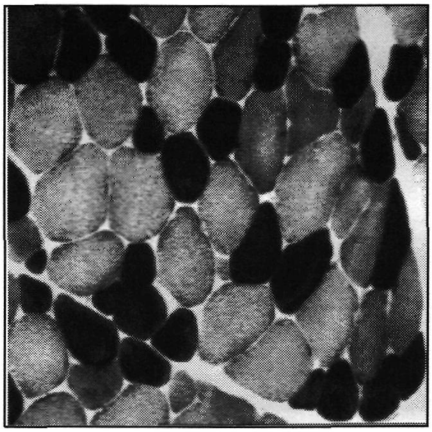


Figure 6.1: An image of muscle tissue, on which our top-down methods will be tested.

image processing for edge detection, e.g. by Montanari [73] and Martelli [64, 65]. Recent contributions have been made by Gerbrands [31] and Orange [81].

In these techniques, some initial guess for the approximate location of the curve must be supplied. In section 6.3, the boundary of the receptive field in a high level of the hierarchy is used as an initial guess, while the boundaries of receptive fields in an intermediate level of the hierarchy are used as possible parts of the exact boundary.

6.2. Top-Down Selection of Convex Objects

This section describes the detection of muscle fibers in the hierarchical structure, based on convexity. Convex regions in the image are detected by a searching method which operates in a hierarchy of region adjacency graphs. Only geometric information is used in this process, but grey value information, which was used in the construction of the hierarchy, is no longer used.

Two measures for the convexity of a set X are used. Both measures are invariant under translation, rotation and scale. The measure c_a measures the relative area of the concavities of X, while c_d measures the largest relative depth of such concavities. Let $\mathcal{C}(X)$ be the convex hull of the set X. Let $\rho(X)(\cdot)$ denote the distance transform of a set, defined by $\rho(X)(x) = \min_{y \notin X} d(x, y)$ and let $A(X)$ denote the area of a set. The area-based convexity measure is

$$c_a(X) = \frac{A(X)}{A(\mathcal{C}(X))}. \tag{6.1}$$

This measure is called the *indice de concavité en surface* (concavity index by surface) by Coster and Chermant [22]. The distance transform based convexity measure is

$$c_d(X) = 1 - \frac{\max_{x \notin X} \rho(\mathcal{C}(X))(x)}{\max_x \rho(\mathcal{C}(X))(x)}. \tag{6.2}$$

Both measures have values between 0 and 1 and the value 1 is reached only for convex sets. The measures are illustrated in figure 6.2, where the light region denotes X and the complete region its convex hull.

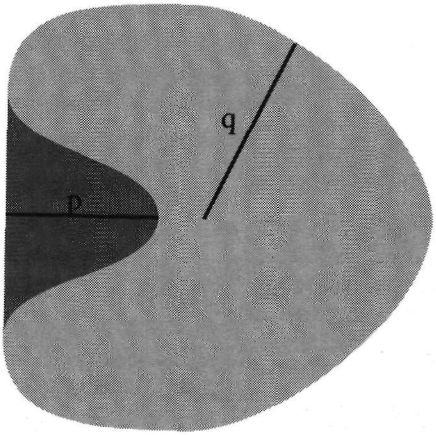


Figure 6.2: The measure $c_a(X)$ is the quotient of the light area and the total area and the measure $c_d(X)$ is 1 minus the quotient of the lengths of the segments p and q.

Fibers must not only be convex, but they must also have an area in a certain range. Therefore, the convexity of each region X is multiplied with a size factor

$$f(X) = \begin{cases} |X|/A_{\text{min}} & \text{if } |X| < A_{\text{min}} \\ 1 & \text{if } A_{\text{min}} \leq |X| \leq A_{\text{max}} \\ A_{\text{max}}/|X| & \text{if } |X| > A_{\text{max}} \end{cases} \tag{6.3}$$

where $|X|$ is the area of the region and A_{min} and A_{max} are low and high values for typical fiber areas. The value of the size factor is 1 for regions with an area between A_{min} and A_{max} and gradually decreases to 0 for regions with areas outside this range. The products of c_a and c_d with f will be called the corrected convexities.

The search procedure operates on clusters, i.e. the child sets of vertices. All connected subsets of a cluster are considered and their corrected convexities are computed. Then, the subset with the highest corrected convexity is selected. If its corrected convexity is higher than some predefined value, the region outlined by this subset is marked as a fiber, and the corresponding vertices are removed from the cluster under consideration. Then, the most convex connected subset is selected from the remaining cluster. This process continues until no more connected subsets with a given minimal convexity exist.

After all convex subsets of a cluster have been computed and removed from the cluster, the remaining vertices are considered in the next step. For each of these vertices, the child set is determined, and each of these child set is searched for convex connected subsets. This procedure is repeated in top-down order.

Figure 6.3 shows the result obtained for a hierarchy describing the muscle tissue image. Figure 6.3a is the highest level of the initial hierarchy. An undersegmentation is used here in order make the effect of convex cluster search more visible. The other two figures show the results obtained using the measures c_a and c_d with thresholds 0.8 and 0.5 for the corrected convexities, respectively. The values used for $A_{\min}$ and $A_{\max}$ were 500 and 2000.

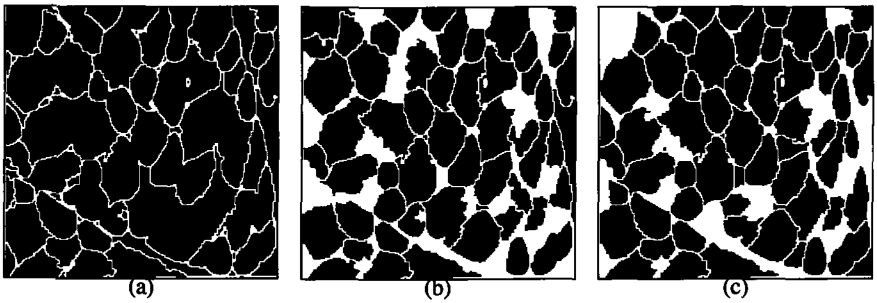


Figure 6.3: The top level of the hierarchy (a) and the clusters detected with the area measure (b) and the distance transform measure (c).

Note that the search procedure used here has rather limited possibilities: when a fiber is found at a given level in the hierarchy, no further effort is made to produce some refinement of the object by considering lower levels in the hierarchy. Moreover, all resulting regions are the union of a set vertices which are children of a common parent. In some cases, it can be impossible to describe a fiber in such a way. The method described in the next section does not have these weaknesses.

6.3. Boundary Detection by Cost Minimization

The edges in a region adjacency graph correspond with boundaries between regions in the image. By selecting a proper set of edges in the region adjacency graph, the outline of an object in the image can be constructed. This section presents an optimization approach for the computation of edge sets which correspond to fiber boundaries.

Object boundaries correspond to closed paths in the dual of the region adjacency graph. This is explained in subsection 6.3.1.

Subsection 6.3.2 presents an optimization approach for the detection of fiber boundaries. Based on our model of boundaries—more or less round curves through points with a large gradient in the image—a cost function is constructed, such that minimal cost paths correspond to fiber boundaries. Dynamic programming is used to detect optimal paths.

In subsection 6.3.3, the optimization approach is used in a hierarchical structure in order to achieve a stepwise refinement of object boundaries. In subsection 6.3.4, some results are shown and the effect of parameter choices is illustrated.

6.3.1. Boundaries as Paths in the Region Adjacency Graph

Edges in the region adjacency graph correspond to boundary parts in the image plane. In order to select series of consecutive boundary parts, a representation is required in which it can be seen, which edges of the region adjacency graph represent consecutive curves in the image plane. In the region adjacency graph, this ordering is not represented explicitly.

For this purpose, the *dual* [36] of a region adjacency graph must be constructed.

Definition 6.1 *Let G be a planar graph with vertices V and edges E. Its dual graph is denoted by $\overline{G}$ and its vertex and edge sets by $\overline{V}$ and $\overline{E}$, respectively. Each vertex in $\overline{G}$ corresponds with a face (a region surrounded by a closed path and with no interior edges) of G. Two vertices of $\overline{G}$ are connected by an edge if the corresponding faces in G share an edge.*

The faces of the dual graph correspond to vertices in the original graph. There is a one-to-one relation between the edges of G and $\overline{G}$.

Dual graphs are defined for graphs which are embedded in the plane; for regions adjacency graphs, such an embedding exists obviously. This is illustrated in figure 6.4. In the present context, it is necessary to take double edges into account. Double edges occur when the boundary between two regions consists of more than one connected component. This is the case, for example, for the two vertices marked by a $*$ in figure 6.4: the boundaries between the corresponding regions consists of two connected components.

It is useful here to consider the image grid in terms of pixels, cracks and points. In this view, pixels correspond to open squares of the form $(x, x + 1) \times (y, y + 1)$ with $x, y \in \mathbb{Z}$, in the Euclidean plane. Pairs of directly

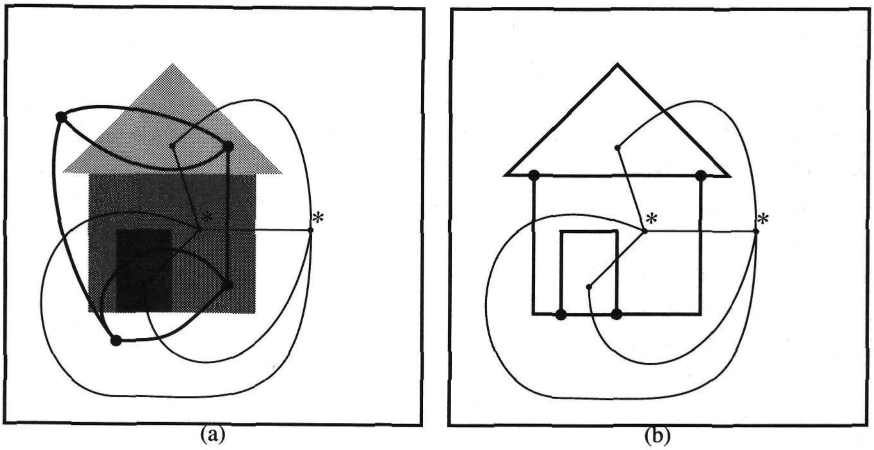


Figure 6.4: Figure (a) shows a region adjacency graph (in thin lines) and its dual (in bold lines). Note the correspondence between faces of one graph and vertices of the other one. Note also the correspondence between mutually intersecting edges of both graphs. In figure (b), the edges of the dual graph have been redrawn in the shape of the curves they represent in the image plane. There is a double edge between the vertices marked by a $*$ because the boundary between the corresponding regions consists of two connected components.

adjacent pixels (4-neighbors) meet at *cracks*, which are line segments of the form $(x, x+1) \times \{y\}$ or $\{x\} \times (y, y+1)$. Four pixels meet in points (x, y). This description corresponds with the topological notion of a *cell complex*, as noted by Kovalevsky [54].

Edges in the dual region adjacency graph correspond to boundaries between regions in the image plane. These boundaries are formed by series of cracks, connected at points. The vertices in the dual region adjacency graph correspond to points in the image plane where boundaries between regions meet.

Let $\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_n$ be a path in the dual graph and let $\overline{e}_i$ be the edge between $\overline{v}_{i-1}$ and $\overline{v}_i$. Then $\overline{e}_1, \overline{e}_2, \ldots, \overline{e}_n$ corresponds with a series of contiguous curves in the image plane.

From a hierarchy of region adjacency graphs, a hierarchy of dual region adjacency graphs can be constructed. This has been described in detail by Kropatsch *et al.* [58] for the situation where the hierarchy is built by graph decimation, but their description is also valid for more general hierarchies. The levels of the dual hierarchy are constructed by constructing the dual graph for each level independently. Each edge in a higher level of the dual hierarchy corresponds to a series of contiguous edges in a lower level. In terms of curves in the image plane, this means that each edge in a higher level represents a curve which consists of a number of consecutive parts, which are each represented by an edge in a lower level.

Consider a path in some higher level of a hierarchy of dual region adjacency graphs and the curve it represents in the image plane. This curve can also be represented by a path in some lower level of the dual hierarchy. This path can be constructed by considering all the edges occurring in the higher level, and computing the corresponding edges in the lower level graph. Concatenating all of these edges yields the desired path in the lower level.

6.3.2. Construction of the Cost Function

Closed paths in a dual region adjacency graph correspond with closed curves in the image plane. In this subsection, a cost minimization procedure will be used in order to detect closed curves which correspond to fiber boundaries. This will be done by means of a hierarchical method. This subsection describes the procedure used in each level; the next subsections describe the hierarchical procedure and present some results.

In this section, three issues must be addressed: a suitable cost function must be constructed, a set of allowed paths must be defined and an optimization algorithm must be chosen.

The optimization procedure will be carried out by dynamic programming. Using dynamic programming, we will minimize a cost function of the form

$$C(\overline{v}_0,\overline{v}_1,\ldots,\overline{v}_n) = (1-\alpha)\sum_i P(\overline{v}_i,\overline{v}_{i+1})+\alpha\sum_i Q(\overline{v}_{i-1},\overline{v}_i,\overline{v}_{i+1}). \quad (6.4)$$

We consider closed paths, so $\overline{v}_{n+i}$ must be read as $\overline{v}_i$ for $i = 1,2$. The terms $P(\overline{v}_i,\overline{v}_{i+1})$ represent the cost contributed by pairs of adjacent vertices, i.e. by edges. The terms $Q(\overline{v}_i,\overline{v}_{i+1},\overline{v}_{i+2})$ represent the cost contributed by triples of consecutive vertices, i.e. by pairs of contiguous edges. In the rest of this section, P and Q will be chosen such that P depends on the extent to which the curve follows maximal gradient paths, while Q depends on the circularity of the curve. The weight factor α can be adapted to make one of these curve characteristics more dominant. As the costs P and Q will be positive, dynamic programming can be used for the computation of a minimal cost path.

Fiber boundaries are characterized by two properties:

1. Fiber boundaries separate regions of different brightness.
2. Fiber boundaries are more or less round.

The cost function will be chosen such that this type of curve can be detected.

The terms $P(\overline{v}_i,\overline{v}_{i+1})$ will be used to express the extent to which the curve separates regions of different brightness. Each pair $(\overline{v}_i,\overline{v}_{i+1})$ corresponds with a curve in the image plane. Let $c_1, c_2, \ldots, c_k$ be the cracks that constitute this curve, where k is the length of the curve. As we are interested in the actual localization of curves in the image plane, and not in the number of edges by which they are represented, the cost function $\sum_i P(\overline{v}_i, \overline{v}_{i+1})$ should not depend on the number of parts from which a curve is constituted, but only on the layout of the curve on the image plane. Therefore, P must have the form

$$P(\overline{v}_i, \overline{v}_{i+1}) = \sum_{j=1}^{k} p(c_j), \tag{6.5}$$

where $p(c_j)$ is a measure for the contrast across a single crack. A common choice in literature for the contrast function $p(c_j)$ is $M - |\nabla f(c_j)|$, where M is some large constant and $\nabla f(c_j)$ is a measure of the image gradient at the crack; usually, it is simply the difference of the grey values of the pixels on both sides of the crack. This choice has an undesirable property: because all $p(c_j)$ must be positive, M must be larger than the largest gradient value in the image. This can be a large number, implying that most of the times, $p(c_j)$ will be quite large. Therefore, short curves will be favored, which may lead to artifacts. In our application, parts of fibers might be cut off, as a shorter path through the interior of a fiber may have a lower cost than the actual boundary, which is longer.

A more suitable choice is therefore

$$p(c_j) = \frac{1}{\epsilon + |\nabla f(c_j)|}, \tag{6.6}$$

where ϵ is some small number (in our implementation 0.1) which avoids division by zero, and $\nabla f(c)$ is an estimator for the image gradient. In our case the difference of the grey values of the pixels on both sides of the crack. With this choice, the presence of large gradients in the image does not enforce an increase of the cost of all the cracks. Moreover, this choice has a clear interpretation: a path with a given length and grey value contrast has the same cost as a path which is twice as long, but has also twice the contrast.

The second criterion for fiber boundaries is that they are more or less round. In the previous section, we used convexity as a measure. This is, however, a global measure and cannot be expressed by a cost function of the form (6.4). The roundness measure proposed in this subsection uses an estimation $c = (x_c, y_c)$ of the center of the fiber. Consider a curve segment with end points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$. If this curve part is part of a round curve around c, the vector $p_2 - p_1$ is perpendicular to the vector between c and the center $(1/2)(p_1 + p_2)$ of the line segment $p_1 p_2$. The deviation θ of the angle between the vectors $p_2 - p_1$ and $(1/2)(p_1 + p_2) - c$ from the optimal value of $\pi/2$ is a measure for the non-roundness of the curve.

This observation is used for the construction of $Q(\overline{v}_i, \overline{v}_{i+1}, \overline{v}_{i+2})$. The three vertices define two adjacent curve segments. The curve represented by the concatenation of the two corresponding edges is considered, and the direction of the line segment between its end points is compared with the optimal direction given the estimated center c. Let θ denote the deviation from the optimal angle and let l denote the length of the line segment between both end points. Then Q is defined by

$$Q(\overline{v}_i, \overline{v}_{i+1}, \overline{v}_{i+2}) = l\theta^2. \tag{6.7}$$

The square is introduced to reduce the cost of small deviations, while increasing the cost of large deviations. This allows more deviations from exact circularity than a linear cost function. The length l is used again because the cost must be proportional to the actual length of the curve in the image plane and not to the number of edges by which it is represented.

The boundary shape is evaluated using two edges instead of one. This is done in order to avoid undesirable effects which can occur if the curves represented by edges become short at the lower levels of the hierarchy. This can be especially disadvantageous in the lowest level, where edges correspond with individual cracks and only horizontal and vertical directions are possible.

We still have to define the set of allowed paths over which cost minimization must be performed. As we will work in a top-down procedure, there will be an initial coarse guess, which was computed at some higher level. This guess is some closed curve in a dual region adjacency graph. Let $X \subset \overline{E}$ be the set of edges in this path. Then a set $Y \subset \overline{E}$ of edges is constructed which lie in a strip around the coarse path. An edge $\overline{e} \in \overline{E}$ is in Y if and only there is a face in the dual region adjacency graph which is bounded by both e and some edge in X. Thus, the allowed paths all lie in a narrow strip which follows the initial guess.

The set of allowed paths is the set of all paths containing only edges in Y. One restriction must be made: in order to avoid artifacts, paths which do not go around the estimated center c must be excluded. This is done by allowing certain pairs of adjacent vertices to occur only in a particular order in the path, effectively making the edges in the dual graph directed.

6.3.3. The Hierarchical Search Method

The cost minimization approach described in the previous subsections can be used for top-down boundary refinement in a hierarchical structure. In the examples presented here, we use a hierarchy of region adjacency graphs constructed by the method presented in chapter 5.

As the initial guess for the curve boundary, the boundary of the receptive field of some vertex in the hierarchy of region adjacency graphs is used. An estimation for the center of the corresponding fiber must also be given. At present, these initial guesses are generated by a human operator, but they can also be generated automatically.

The circumference of the selected region corresponds with a closed path in the top level of the dual region adjacency graph. Top-down search is started at some intermediate level in the hierarchy, typically the third one. The curve corresponding to the initial guess is represented as a closed path in this level. Then the allowed paths are defined by selecting a set of allowed edges, and the minimal cost path is computed, yielding the first refinement of the initial guess. This path is then represented at the next lowest level, where it is again used as an initial guess, which is refined by cost minimization. This procedure is repeated, until the final curve in the base level of the hierarchy is reached.

6.3.4. Results

Some results are presented in figure 6.5. The pictures show the initial guess, generated at the top level, the allowed edges in the third level of the hierarchy, the minimal cost path detected at the third level of the hierarchy and the final path, detected at the base level. Note that the initial guess shows some large deviations from the true fiber boundary. In terms of the hierarchy, these deviations are small, because they correspond to just a few 'wrong' parent-child links. Therefore, the path refinement procedure has no difficulties in finding the correct boundary. The method described here detects almost all fibers correctly. In practice, errors occur only when the structure of the hierarchy is deformed so much that the correct fiber boundaries cannot be detected by repeatedly applying relatively small changes in successive levels.

The parameter α in equation (6.4) was chosen to be 0.01. With this choice the P and Q terms have about the same order of magnitude, and both shape and gradient information are taken into account. The value of α for which both terms have the same order of magnitude is image dependent. The contrast cost function depends on the grey values present in the image, while the shape cost function does not. Therefore, the optimal value of α is related to the contrast or the grey value range of the image. In practice, it appears that one value of α suffices for all fibers in an image.

Figure 6.6 shows the effect of modification of α. In the middle image, $\alpha = 0.01$ and the fiber boundary is detected correctly. In the left image, $\alpha = 0.002$ was used. The cost function is dominated by contrast terms and the boundary traces strong contrasts, yielding an undesirable bump on the boundary. In the right image, $\alpha = 0.05$ was used. The cost function is dominated by the shape terms and a more or less circular boundary is generated, although this one does not follow brightness edges.

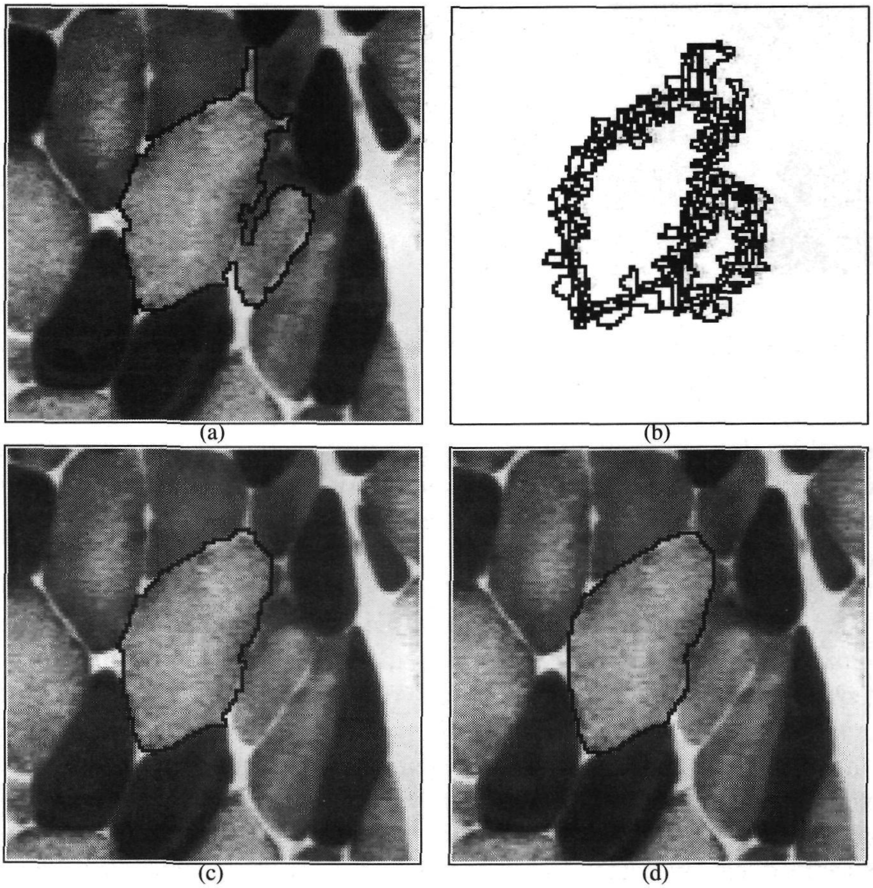


Figure 6.5: Boundary refinement by cost minimization: the initial guess (a), the set of allowed edges in the intermediate level (b), the minimal cost path at the intermediate level (c) and the final path in the base level (d).

6.4. Conclusions

If a hierarchical image description is constructed in a bottom-up fashion, some post-processing or top-down search methods are required for the extraction of image content from this description. In this chapter, some possibilities of how this can be done have been presented.

Top-down search methods require some image model or a priori knowledge of image content. We have described two methods for the detection of more or less convex objects with smooth boundaries, uniform grey values and step edges at their boundaries.

The first method searches for convex combinations of regions and uses no

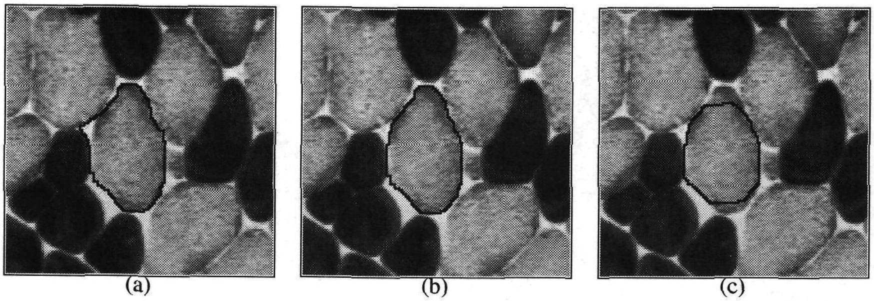


Figure 6.6: Optimal boundaries detected with $\alpha = 0.002$ (a), $\alpha = 0.01$ (b) and $\alpha = 0.05$ (c).

grey value information. Most fibers are detected reasonably well, but problems occur when a fiber is not the union of a number of children of a common parent. Moreover, no extra refinement is performed after a boundary is detected at a given level.

The second method searches for object boundaries which are optimal with respect to a cost function, which is constructed based on a priori knowledge of image content. This method requires the extension of well known curve detection methods, used on pixels grids, to region adjacency graphs. It consists of a number of optimization steps, performed in top-down order on the levels of the hierarchy. The objects are detected with satisfactory accuracy and the method is able to correct for errors in the tree structure of the hierarchy, which originate from the bottom-up procedure. There is one free parameter. It has been argued that this parameter must be chosen in such a way that the two terms in the cost function have the same order of magnitude. It has been demonstrated that changing this parameter has the expected effect on the result.

Model-based image processing techniques, such as those based on active contours or snakes [51] and parametrically deformable models [98] have become popular recently, and their combination with hierarchical methods might become a fruitful field of research.

7

Chamfer Metrics and Mathematical Morphology

7.1. Introduction

Mathematical morphology [66, 95] is an approach to image processing based on set theoretic notions such as inclusion and intersection. Recently, mathematical morphology has been extended to grey level images [95] and even complete lattices [87, 41, 40], but originally mathematical morphology dealt with binary images. Mathematical morphology models binary images as sets, by considering the foreground as a subset of the image plane. The image is analyzed by inspecting the containment or intersection relations of the image with translates of some set B, the structuring element.

The elementary operations in mathematical morphology are the dilation, the erosion, the opening and the closing. They are defined as follows:

$$\begin{array}{rrcl} \text{dilation:} & X \oplus B & = & \bigcup\{B_h \mid h \in X\} \\ \text{erosion:} & X \ominus B & = & \{h \in E \mid B_h \subseteq X\} \\ \text{opening:} & X \circ B & = & (X \ominus B) \oplus B \\ \text{closing:} & X \bullet B & = & (X \oplus B) \ominus B \end{array}$$

Here, B_h denotes the structuring element B translated over the vector h. The image space, typically IR^2 or a square grid, is denoted by E.

Mathematical morphology is a useful tool for the analysis of the geometrical structure of an image (see Heijmans [39] for a review). The geometrical properties of morphological operators, such as invariance under rotations, depend on the geometric properties, such as rotation symmetry, of the structuring elements used.

The notion of scale can be introduced in mathematical morphology by considering the operations defined by a family of structuring elements in stead of a single one. Such a family of structuring elements is typically parameterized by a scale parameter and larger scale parameters correspond to larger structuring elements.

On the discrete grid, a family of structuring elements is defined most commonly by successive dilations of a structuring element with itself. A typical basic structuring elements B would be a square, corresponding to the 8-connected grid, or a diamond, corresponding to the 4-connected grid. The corresponding family of structuring elements is then defined by $B(n) = B \oplus \ldots \oplus B$ (n terms). The scale parameter assumes positive integer values.

In the continuous case, where images are subsets of the Euclidean plane, discs of increasing radius are an appropriate choice for a family of structuring elements. The fact that these structuring elements are defined by a metric provides a natural interpretation of the scale parameter.

In practice, mathematical morphology is performed on the discrete grid. For many applications, it is favorable to use a family of structuring elements which is a good approximation of the family of Euclidean discs. Families defined by successive dilations of a single structuring elements are not good approximations of Euclidean discs. On the other hand, the family of structuring elements defined as the intersections of Euclidean discs with the discrete grid can only be used for more complicated morphological algorithms at large computational cost.

As chamfer metrics can approximate the Euclidean metric closely, spheres defined by the chamfer metric seem to be a suitable choice as a family of structuring elements, at the same time allowing the formulation of efficient algorithms. Such a choice however poses some other problems, for example caused by the fact that larger spheres are in general not invariant under an opening by smaller spheres.

The goal of this chapter and the two following ones is the integration of chamfer metrics into mathematical morphology, both by presenting new theoretical results and by providing efficient algorithms for morphological operations with chamfer discs. Efficient algorithms for morphological operations have been described by a number of authors. Groen and Foster [33] use lookup tables in order to speed up decisions based on inspection of the neighborhood of a pixel. Vincent and Schmitt [106, 93] use queues in which only those pixels which must be processed are stored; pixels which need not to be processed are ignored. Van den Boomgaard and Van Balen [11] use a decomposition of the structuring element combined with a bit-mapped storage structure for the image in order to construct efficient algorithms. The algorithms described by these authors are not applicable to operations based on the chamfer metric.

The next section presents some results on discrete metrics. Section 7.3 introduces chamfer metrics and some of their properties. This chapter ends with some conclusions. The following two chapters use the results presented in this chapter for the description of the medial axis and the opening transform based on chamfer metrics, respectively.

7.2. Discrete Metrics

In this section, discrete metrics are defined and some of their properties are described.

A *metric* on a set E is a function $d : E \times E \rightarrow [0, \infty]$ satisfying the following conditions:

(1) $d(x,y) = 0 \Leftrightarrow x = y$ for all $x, y \in E$.
(2) $d(x,y) = d(y,x)$ for all $x, y \in E$.
(3) $d(x,y) + d(y,z) \leq d(x,z)$ for all $x, y, z \in E$.

Note that we allow metrics to assume the value ∞. This is necessary because later on, metrics will be constructed for which the distance between two points is defined as the length of a shortest path between them. If there is no path between two points, the distance between these points will be ∞. For chamfer metrics, which are a special case of shortest path metrics, all distances will be finite.

Let d be a metric on a set E. Let D denote the set $\{d(x,y) \mid x, y \in E\}$; this set is called the *range* of the metric. For each $d \in D$ open and closed spheres can be defined.

Definition 7.1 *Let d be a metric on a set E and let D be its range. Let $r \in D$ and $x \in E$. The closed sphere with radius r and center x is the set*

$$\overline{B}(x,r) = \{y \in E \mid d(x,y) \leq r\}. \tag{7.1}$$

Definition 7.2 *Let d be a metric on a set E and let D be its range. Let $r \in D$ and $x \in E$. The open sphere with radius r and center x is the set*

$$B(x,r) = \{y \in E \mid d(x,y) < r\}. \tag{7.2}$$

The rest of this chapter will be concerned only with metrics which have a discrete range. Such metrics will be called discrete metrics:

Definition 7.3 *A metric d on a set E is called a discrete metric if its range $\{d(x,y) \mid x, y \in E\}$ has no limiting points.*

Thus, for each value $r > 0$ in the range D of a discrete metric d, there is a value $s \in D$ which is the largest value in D smaller than r and for each $r \in D$ there is a value $t \in D$ which is the smallest value in D larger than r. Therefore, for a discrete metric each closed sphere of radius r is also an open sphere of some larger radius s, which is the smallest number in D larger than r. Each open sphere of radius $r > 0$ is also a closed sphere of some smaller radius s which is the largest number in D such that $s < r$. In the sequel, we will sometimes write $B(r)$ for $B((0,0),r)$ and $\overline{B}(r)$ for $\overline{B}((0,0),r)$.

If X is a subset of a discrete metric space E, it is possible to define for each point the shortest distance to a point outside the set.

Definition 7.4 *Let d be a discrete metric on a set E and let D be its range. Let X be a bounded subset of E. The external distance transform ρ_X^{ext} of X is the function $E \rightarrow D$ defined by*

$$\rho_X^{\text{ext}}(x) = \min_{y \in X^C} d(x,y). \tag{7.3}$$

Note that the value of $\rho_X^{\text{ext}}(x) = 0$ for $x \notin X$. The definition implies $B(x, \rho_X^{\text{ext}}(x)) \subset X$ for all $x \in X$. Using the definition of a metric, it can be seen that $\rho_X^{\text{ext}}(x) \leq \rho_X^{\text{ext}}(y) + d(x,y)$, for all $x, y \in E$. For a function $f : E \rightarrow \mathbb{R}$, the *support* $\text{Supp}(f)$ of f is defined as the set $\{x \in E \mid f(x) \neq 0\}$. The support $\text{Supp}(\rho_X^{\text{ext}})$ of ρ_X^{ext} is the set X itself.

For each point x in a bounded subset X of a discrete metric space E, it is possible to determine the largest closed sphere centered at x and is contained in X. This leads to the notion of the internal distance transform.

Definition 7.5 *Let d be a discrete metric on a set E and let D be its range. Let X be a bounded subset of E. The internal distance transform ρ_X^{int} of X is the function $X \rightarrow D$ defined by*

$$\rho_X^{\text{int}}(x) = \max\{r \in D \mid \overline{B}(x,r) \subset X\}. \tag{7.4}$$

The internal distance transform can be calculated from the external distance transform. The value of $\rho_X^{\text{int}}(x)$ for an $x \in X$ is the largest number $r \in D$ such that $r < \rho_X^{\text{ext}}(x)$. Such a number exists because d is a discrete metric.

Note that he following equivalences describe the relation between distance transforms and the inclusion of structuring elements in an image:

$$\begin{aligned} \overline{B}(x,r) \subseteq X \quad &\Leftrightarrow \quad \rho_X^{\text{int}}(x) \geq r \\ B(x,r) \subseteq X \quad &\Leftrightarrow \quad \rho_X^{\text{ext}}(x) \geq r \end{aligned}$$

As morphological operations are based on inclusion and intersection of a structuring element with an image, these equivalences will be central in the discussion of mathematical morphology using chamfer discs.

For chamfer metrics, the external distance transform can be computed using a two-pass algorithm, while the computation of the internal distance transform requires an extra step. Therefore, the external distance transform can be handled more easily; it is used in section 9.2. The algorithms in sections 8.2, 8.3, 8.4 and 9.3 use the values of $\rho_X^{\text{int}}(x) - \rho_X^{\text{int}}(y)$ for point pairs x, y. This value is in general not equal to $\rho_X^{\text{ext}}(x) - \rho_X^{\text{ext}}(y)$ and the external distance transform cannot be used here. Note that the terms *internal* and *external distance transform* are used sometimes in the literature to denote the distance transforms of X and X^C, respectively.

Another ingredient for the algorithms to be presented later is the reconstruction. A reconstruction is the computation of a set as a union of spheres, which can be performed if the centers and radii of these spheres are given.

Definition 7.6 *Let d be a discrete metric on a set E and let D be its range. Let $f : E \rightarrow D$ be a function of bounded support. The open sphere reconstruction $R(f)$ of f is defined by*

$$R(f) = \bigcup_{x \in \mathrm{Supp}(f)} B(x, f(x)). \tag{7.5}$$

Thus, the reconstruction is a union of open spheres: an open sphere with center x and radius $f(x)$ is inserted in the set for each $x \in E$ with $f(x) > 0$. An equivalent definition for $R(f)$ is

$$R(f) = \{x \in E \mid d(x, y) < f(y) \text{ for some } y \in E\}.$$

It is also possible to define a reconstruction using closed spheres.

Definition 7.7 *Let d be a discrete metric on a set E and let D be its range. Let X be a bounded subset of E and let f be a function $X \rightarrow D$. The closed sphere reconstruction $\overline{R}(f)$ of f is defined by*

$$\overline{R}(f) = \bigcup_{x \in X} \overline{B}(x, f(x)). \tag{7.6}$$

The closed sphere reconstruction builds a set as a union of closed spheres. An equivalent definition is:

$$\overline{R}(f) = \{y \in E \mid d(x, y) \leq f(x) \text{ for some } x \in X\}.$$

If d is a metric on a set E, a point $y \in E$ is said to lie between two points $x, z \in E$ if $d(x, y) + d(y, z) = d(x, z)$. If d is the Euclidean metric, y lies between x and z if y lies on the line segment from x to z, which is the intuitive meaning of 'between'. Later we will need metrics for which, given two points x and y and a value r in the range of d, a third point z can be found at a distance r from y, such that y lies between x and z.

Definition 7.8 *Let d be a metric on a set E and let D be its range. The metric d is called extending if, for each $x, y \in E$ and $r \in E$, there is a $z \in E$ such that $d(x, y) + d(y, z) = d(x, z)$ and $d(y, z) = r$.*

7.3. Chamfer Metrics

The chamfer metric was introduced by Borgefors [12]. Her goal was the construction of a metric on the square grid which is a good approximation of the Euclidean metric and allows efficient computation. The chamfer metric is a metric defined on the square grid $\{p(1,0) + q(0,1) \mid p, q \in \mathbb{Z}\}$ or on the hexagonal grid $\{p(1,0) + q(1/2, 1/2\sqrt{3}) \mid p, q \in \mathbb{Z}\}$. These grids are illustrated in figure 7.1.

They are invariant under rotations and reflections. The group of symmetries of the square grid (leaving the origin fixed) contains four rotations, including the identity, and four line reflections. The symmetry group of the hexagonal grid contains six rotations, including the identity, and six reflections. Both the square grid and the hexagonal grid are representations of $\mathbb{Z}^2$. The only difference lies in the different symmetries of the two types of grids.

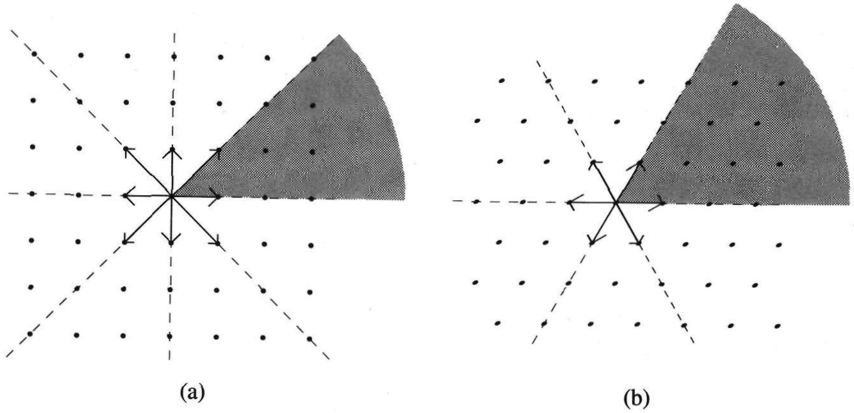


Figure 7.1: The square grid (a) and the hexagonal grid (b) with a number of primitive vectors. The primitive vectors divide the grid in a number of wedges. In each grid, one of these wedges has been shaded.

A metric on $\mathbb{Z}^2$ can be defined as follows. Let $V = (v_1, \ldots, v_k)$ be a set of vectors in $\mathbb{Z}^2$ such that $v \in V$ implies $-v \in V$. These vectors are called prime vectors. Let l be a function $V \to \mathbb{N}$ such that $l(v) = l(-v)$ and $l(v) > 0$ for all v. The numbers $l(v)$ are called the weights of the prime vectors. It is also possible to use real numbers as weights, but that is not done in this paper. This is not a real restriction: if rational weights are to be used, they can be multiplied by a suitable scaling factor, yielding integer values. In practice, operations will be performed on a bounded grid and real numbers can be approximated with sufficient accuracy by rational numbers.

If x and y are two points in $\mathbb{Z}^2$, a path from x to y is defined as a sequence $p_0, p_1, \dots, p_k$ of points such that $p_0 = x$ and $p_k = y$ and such that $p_i - p_{i-1} \in V$ for $i = 1 \dots k$. The length of this path is defined as :

$$L(p_0, \dots, p_k) = \sum_{i=1}^{k} l(p_i - p_{i-1}). \tag{7.7}$$

If $x = y$, we allow $p_0 = x = y$ as a path from x to y. The length of this path is 0.

Let $x = p_0, \dots, p_k = y$ be a path from x to y. The differences $u_i = p_i - p_{i-1}$ are called the *steps* in the path. Any permutation σ of the steps u_i yields another path $p'_0, \dots, p'_k$ from x to y given by $p'_i = p'_{i-1} + \sigma(u_i)$. Note that permutations do not affect the length of the path. Any sequence $u_1, \dots, u_k$ of prime vectors such that $\sum_{i=1}^{n} u_i = y - x$ defines a path from x to y: take $p_j = x + \sum_{i=1}^{j} u_j$.

Theorem 7.9 *With $(v_1, \dots, v_n)$ and l, L as defined above, a metric $d : \mathbb{Z}^2 \times \mathbb{Z}^2 \to [0, \infty]$ is defined by*

$$d(x, y) = \min\{L(p_0, \dots, p_k) \mid k \in \mathbb{N};\ p_0, \dots, p_k \textit{ is a path from } x \textit{ to } y\}.$$

The proof of this theorem is simple: the first two properties of a metric are satisfied by construction and the third property can be verified using concatenation of paths. Note that $d(x, y) = \infty$ if there is no path between x and y.

The construction described above defines a metric, but the chamfer metrics to be investigated in this paper will have some additional properties. Before these properties can be formulated, some definitions have to be made.

Definition 7.10 *Let $\{v_1, \dots, v_n\}$ be a set of prime vectors and let l be a weight function. Then the normalized prime vectors $\hat{v}_1, \dots, \hat{v}_n$ are defined by*

$$\hat{v}_i = v_i / l(v_i).$$

Note that the normalized prime vectors are in general not points on the grid. Let $\mathbb{R}^{\geq 0}$ denote the set of nonnegative reals.

Definition 7.11 *Let u and w be two independent vectors in $\mathbb{R}^2$. The wedge between these two vectors is the set*

$$W_{u,w} = \{\lambda_1 u + \lambda_2 w \mid \lambda_1, \lambda_2 \in \mathbb{R}^{\geq 0}\}.$$

The wedge between two vectors is indeed a wedge shaped set, bounded by two half lines in the directions of the vectors (see figure 7.1).

Definition 7.12 *Let $V = \{v_1, \ldots, v_n\}$ be a set of prime vectors. Two independent vectors $u, w \in V$ are adjacent if $W_{u,w}$ contains no prime vectors other than u and w.*

An adjacency relation is defined for normalized prime vectors in the same way as for prime vectors. Note that $\hat{u}$ and $\hat{v}$ are adjacent if and only if u and v are adjacent.

Lemma 7.13 *Let V be a finite set of vectors in $\mathbb{Z}^2$ such that V contains a pair of independent vectors and suppose that for each $v \in V$, $-v \in V$ as well, and that $-v$ is the only vector in V which is collinear with v. Then for each $x \in \mathbb{Z}^2$ there are two adjacent vectors u and w in V such that $x \in W_{u,w}$.*

PROOF. The lemma can be proved by ordering the prime vectors according to their angle with the positive x-axis. Under the given conditions, no two prime vectors have the same direction and two prime vectors are adjacent in the sense of lemma 7.12 if they are adjacent in the ordering according to their direction. Under the given conditions it can easily be seen that the plane is covered by wedges spanned by pairs of adjacent prime vectors. ∎

Two different wedges intersect only in the origin, or they have a bounding half line in common.

If $u = (u_x, u_y)$ and $w = (w_x, w_y)$ are two points in $\mathbb{Z}^2$, the *determinant* $\det(u, w)$ is defined by $\det(u, w) = u_x w_y - u_y w_x$. Note that u and w are linearly dependent if and only if $\det(u, w) = 0$.

Although a set of weighted prime vectors can define a metric as in theorem 7.9, the metrics referred to in the literature as chamfer metrics form a subclass of such metrics. The next theorem gives a formal definition of this subclass; the consequences of this definition are described in the following lemmas.

Definition 7.14 *A metric as defined above is called a chamfer metric if the following properties hold:*

(1) The set V of prime vectors contains a pair of linearly independent vectors.
(2) If $v \in V$, the only other vector in V which is collinear with v is $-v$.
(3) If u and w are two adjacent prime vectors, then $\det(u, w) = \pm 1$.
(4) The normalized prime vectors $\hat{v}_1, \ldots, \hat{v}_n$ lie on the boundary of a convex polygon.

Condition (3) in definition 7.14 implies that each grid point which lies in a wedge can be written as an integer combination of the prime vectors generating the wedge:

Lemma 7.15 *Let u and w be two adjacent prime vectors and let $p = \lambda u + \mu w \in \mathbb{Z}^2$ for some $\lambda, \mu \in \mathbb{R}^+$. Then $\det(u, w) = \pm 1$ implies λ and μ are integer.*

PROOF. From the equality

$$\begin{pmatrix} u_x & w_x \\ u_y & w_y \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} \tag{7.8}$$

it follows immediately that

$$\begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \frac{1}{\det(u,w)} \begin{pmatrix} w_y p_x - w_x p_y \\ u_y p_x - u_x p_y \end{pmatrix}. \tag{7.9}$$

Both $w_y p_x - w_x p_y$ and $u_y p_x - u_x p_y$ are integer, so λ and μ are integer if $\det(u,w) = \pm 1$. ∎

As there is a pair of independent prime vectors, it follows from lemma 7.13 that each grid point lies in some wedge. Lemma 7.15 implies that each grid point in a wedge can be written as an integer combination of the two prime vectors generating this wedge. This also immediately produces a path from 0 to each point in which only two prime vectors occur as steps. Because of translation invariance, it follows that for each $x, y \in \mathbb{Z}^2$, there is a path from x to y that contains at most two different prime vectors as steps. From property (4) in definition 7.14 it follows that such a path is a shortest path:

Lemma 7.16 *Suppose $p = m_1 u_1 + m_2 u_2$ where u_1 and u_2 are adjacent prime vectors and $m_1, m_2 \in \mathbb{N}$. Then $d(0,p) = m_1 l(u_1) + m_2 l(u_2)$.*

PROOF. The proof of this theorem is illustrated in figure 7.2 for the 2-3–metric, with $p = (2, 1)$.

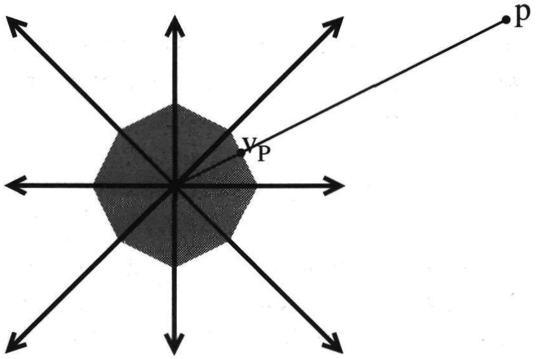


Figure 7.2: Illustration of the proof of lemma 7.16 with the 2-3–metric. The prime vectors, the polygon defined by the normalized prime vectors and the points $p = (2, 1)$ and $v_P = (2/5, 1/5)$ are indicated.

It is clear that a path P of length $m_1l(u_1) + m_2l(u_2)$ from 0 to p exists. Now let Q be a path from 0 to p which contains each prime vector v_i n_i times. Then the length of Q is $\sum_{i=1}^{n} n_il(v_i)$. We show that $L(P) \leq L(Q)$. We have

$$\begin{aligned} p &= \sum_{i=1}^{n} n_iv_i \\ &= \left(\sum_{j=1}^{n} n_jl(v_j)\right) \sum_{i=1}^{n} \left(\frac{n_il(v_i)}{\sum_{j=1}^{n} n_jl(v_j)}\right) \hat{v}_i \\ &= L(Q)v_Q, \end{aligned}$$

where

$$v_Q = \sum_{i=1}^{n} \left(\frac{n_il(v_i)}{\sum_{j=1}^{n} n_jl(v_j)}\right) \hat{v}_i$$

is a convex combination of prime vectors $\hat{v}_i$. We also have

$$\begin{aligned} p &= m_1u_1 + m_2u_2 \\ &= (m_1l(u_1) + m_2l(u_2)) \left\{ \frac{m_1l(u_1) \times \hat{u}_1}{m_1l(u_1) + m_2l(u_2)} + \frac{m_2l(u_2) \times \hat{u}_2}{m_1l(u_1) + m_2l(u_2)} \right\} \\ &= L(P)v_P, \end{aligned}$$

where

$$v_P = \frac{m_1l(u_1)}{m_1l(u_1) + m_2l(u_2)}\hat{u}_1 + \frac{m_2l(u_2)}{m_1l(u_1) + m_2l(u_2)}\hat{u}_2$$

is a convex combination of two adjacent normalized prime vectors $\hat{u}_1$ and $\hat{u}_2$.

These are two ways of writing p as a multiple of a convex combination of normalized prime vectors, so

$$L(Q)v_Q = L(P)v_P.$$

Therefore, the two convex combinations of normalized unit vectors point in the same direction. The convex combination v_P lies in the polygon. As $\hat{u}_1$ and $\hat{u}_2$ are adjacent vectors and lie on the boundary of a convex polygon, the convex combination v_P of $\hat{u}_1$ and $\hat{u}_2$ lies on the boundary of the polygon. The point v_Q lies on the line segment between the origin and v_P, so the length of v_P is at least as large as the length of v_Q. As $L(P)|v_P| = L(Q)|v_Q|$ it follows that $L(Q)$ is at least as large as $L(P)$. ∎

There are many different sets of prime vectors and weights which generate the same metric. If u and w are adjacent prime vectors, the addition of $u+w$ and $-u-w$ with weight $l(u)+l(w)$ to the set of prime vectors produces the same chamfer metric as the one generated by the original set of prime vectors. The number of shortest paths between two points is however increased by enlarging the set of prime vectors, because two steps u and w in a path can be replaced by a single step $u+w$. There is, however, the following theorem:

Theorem 7.17 *Suppose V is a set of prime vectors provided with weights, generating a chamfer metric such that the normalized prime vectors are the corners of a convex polygon. Suppose $p = m_1u_1 + m_2u_2$, where u_1 and u_2 are two adjacent prime vectors and $m_1, m_2 \in \mathbb{N}$. Then a shortest path from 0 to p contains only steps u_1 and u_2.*

PROOF. Let P and Q be paths as described in the proof of the previous theorem. Suppose that $L(P) = L(Q)$, i.e. that $v_P = V_Q$. It must be shown that the only n_i which have a nonzero value are those corresponding to u_1 and u_2. The line segment between the normalized prime vectors $\hat{u}_1$ and $\hat{u}_2$ is an edge of the convex hull of the set of normalized prime vectors and they are the only normalized prime vectors on this edge of the polygon. The equality $v_P = v_Q$ can be written as

$$\sum_{i=1}^{n}\left(\frac{n_il(v_i)}{\sum_{j=1}^{n} n_jl(v_j)}\right)\hat{v}_i = \frac{m_1l(u_1)}{m_1l(u_1)+m_2l(u_2)}\hat{u}_1 + \frac{m_2l(u_2)}{m_1l(u_1)+m_2l(u_2)}\hat{u}_2.$$

As there is only one way of writing a point on a edge of a convex polygon as a convex combination of the corners of the polygon, it follows that the two convex combinations above are the same, and that the only nonzero n_i's are those corresponding to u and w. ∎

If not all normalized prime vectors are corners of a convex polygon, different types of shortest paths can occur. Suppose for example that $v_x = (1,0)$ and $v_y = (0,1)$ are prime vectors with weight 1 and $v_{xy} = (1,1)$ is a prime vector with weight 2. Then $(0,0)$—$(1,1)$ is a shortest path from $(0,0)$ to $(1,1)$ containing a single step v_{xy}, but the path $(0,0)$—$(1,0)$—$(1,1)$ contains steps v_x and v_y and is a shortest path as well. If the weight of v_{xy} is 3, the path $(0,0)$—$(1,0)$—$(1,1)$ is even shorter than the path $(0,0)$—$(1,1)$.

The definition of chamfer metrics can be extended to higher dimensions. For $\mathbb{Z}^3$, for example, the adjacency relation is defined for triples of vectors and the determinant is replaced by a 3×3 determinant.

Some often used chamfer metrics are represented in figure 7.3. Each square containing a number corresponds with a prime vector. The position of the square relative to the center square is the prime vector and the number in the square is the weight of this prime vector. For the p-q–metric, for example, there are eight prime vectors. The prime vectors $(1,0)$, $(0,1)$, $(-1,0)$ and $(0,-1)$ are called prime vectors of type p; their weight will be denoted as p. The prime vectors $(1,1)$, $(-1,1)$, $(-1,-1)$ and $(1,-1)$ are called prime vectors of type q; their weight will be denoted as q. Often used p-q–metrics are the 2-3–metric, the 3-4–metric and the 5-7–metric.

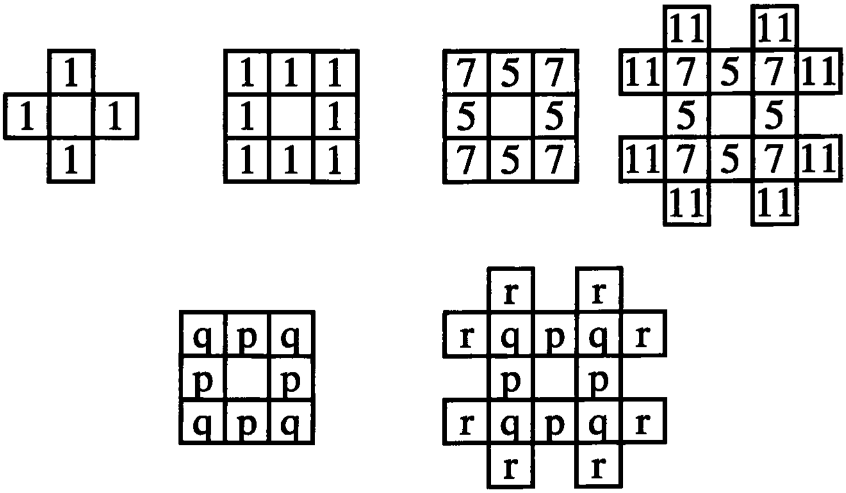


Figure 7.3: The masks describing chamfer metrics. The top line shows the masks for the city block metric, the chess board metric, the 5-7–metric and the 5-7-11–metric, respectively. Metrics defined by choosing other weights are referred to as p-q–metrics or p-q-r–metrics.

Verwer [105] has analyzed the accuracy of chamfer metrics on a square grid as an approximation to the Euclidean metric. The city block metric and the chess board metric are accurate within 17.16%. The 5-7–metric is accurate within 4.21% and the 5-7-11–metric is accurate within 1.79%.

The examples in this chapter and the next one will use the 5-7–metric and the 5-7-11–metric, although similar results can be obtained for other metrics. The reasons for this choice are the following. The city block metric and the chess board metric are not good approximations of the Euclidean metric. The chamfer metrics on the hexagonal grid have the disadvantage that there are no imaging devices producing images on a hexagonal grid. Both the 5-7–metric and the 5-7-11–metric are good approximations of the Euclidean metric. The medial axis algorithm for the 5-7-11–metric is different from the one for the 5-7–metric. Therefore, both are used in examples.

As the distance between two points in a chamfer metric is always an integer, every chamfer metric is a discrete metric. If the weights of adjacent prime vectors in the definition of a chamfer metric have greatest common divisor (gcd) 1, something more can be said. In that case, the range of the metric is the set of all but a finite number of natural numbers.

Theorem 7.18 *Let D be a chamfer metric and let D be its range. Let l_1 and l_2 be the weights of two adjacent prime vectors such that gcd$(l_1, l_2) = 1$. Then $\mathbb{N} - D$ is finite.*

PROOF. Suppose l_1 is the weight of v_1 and l_2 is the weight of v_2. For each $n_1, n_2 \in \mathbb{N}$ the vector $x = n_1 v_1 + n_2 v_2 \in W_{v_1, v_2}$ satisfies $d(0, x) = n_1 l_1 + n_2 l_2$. Therefore, D contains every number which can be written as $n_1 l_1 + n_2 l_2$ with $n_1, n_1 \in \mathbb{N}$. As gcd$(l_1, l_2) = 1$, there are integers m_1 and m_2 such that any integer t can be written as an integer combination of l_1 and l_2 in exactly the following ways [99]:

$$t = (m_1 t + l_2 k) l_1 + (m_2 t - l_1 k) l_2 \qquad (k \in \mathbb{Z}).$$

The possibilities using only nonnegative coefficients for l_1 and l_2 are found by solving the equations $m_1 t + l_2 k \geq 0$ and $m_2 t - l_1 k \geq 0$. These inequalities imply that k must lie in the segment $[-\frac{m_1}{l_2} t, \frac{m_2 t}{l_1} t]$. The length of this interval is $t / l_1 l_2$, so if $t \geq l_1 l_2$, the length of the interval is at least 1 and an integer value for k can be found in the interval. Therefore, any integer $t \geq l_1 l_2$ occurs as a distance value. Therefore, there are at most $l_1 l_2$ natural numbers which are not in the range of the metric. ∎

If it is known that all integers above a known bound are contained in the range, the range can be determined by checking the natural numbers below this bound. The range of the 5-7–metric, for example, is $\mathbb{N} - \{1, 2, 3, 4, 6, 8, 9, 11, 13, 16, 18, 23\}$. This structure of the range of a chamfer metric enables the computation of the internal distance transform from the external distance transform. For each point, the internal distance transform value of a point with external distance transform value r is the smallest $s \in D$ such that $s < r$. For r larger than a certain bound, this value is $r - 1$; for other values a lookup table can be constructed in order to facilitate easy computation.

Distance transforms and reconstructions can be computed efficiently for chamfer metrics. Both can be computed in two image scans: a forward scan, in which pixels are scanned from top to bottom and from left to right, and a backward scan, in which pixels are scanned from bottom to top and from right to left. In the forward scan, pixels values are updated using information from so-called backward neighbors; in the backward scan, pixel values are updated using information from forward neighborhoods.

The forward and backward masks associated with the 5-7–metric are shown in figure 7.4. This figure should be interpreted as follows: if (x_1, x_2) are the

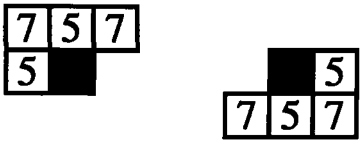


Figure 7.4: The backward and forward masks for the 5-7–metric.

coordinates of a pixel, its backwards neighbors are $n_1 = (x_1, x_2 - 1)$ and $n_3 = (x_1 - 1, x_2)$ at distance $d_1 = d_3 = 5$ and $n_2 = (x_1 - 1, x_2 - 1)$ and $n_4 = (x_1 + 1, x_1 - 1)$ at distance $d_2 = d_4 = 7$. The forward neighborhood mask should be interpreted in the same way.

The algorithms for the distance transformation and the reconstruction are well known from the literature [12] and are described here only for the sake of completeness. They use a rectangular grid where the pixels are denoted by x_i and their values by v_i.

Algorithm 7.19 *The external distance transform of an object X with respect to a chamfer metric.*

(1) Initialize: $r_i := \infty$ if $x_i \in X, r_i := 0$ otherwise.

(2) For all pixels x_i in forward scanning order:
if $\min\{r_j + d_j \mid x_j$ backward neighbor of $x_i\} < r_i$, then replace r_i by this value.

(3) For all pixels x_i in backward scanning order:
if $\min\{r_j + d_j \mid x_j$ forward neighbor of $x_i\} < r_i$, then replace r_i by this value.

Algorithm 7.20 *The computation of the internal distance transform of a bounded set X for a chamfer metric with range D.*

(1) Compute the external distance transform of X.

(2) For each $x_i \in X$: replace r_i by $\max\{r \in D \mid r < r_i\}$.

Algorithm 7.21 *The open sphere reconstruction of a function f with respect to a chamfer metric.*

(1) Initialize: $r_i := f(x_i)$.

(2) For all pixels x_i in forward scanning order:
if $\max\{r_j - d_j \mid x_j$ backward neighbor of $x_i\} > r_i$, replace r_i by this value.

(3) For all pixels x_i in backward scanning order:
if $\max\{r_j - d_j \mid x_j$ forward neighbor of $x_i\} > r_i$, replace r_i by this value.

(4) Determine the set of pixels x_i with $r_i > 0$.

Algorithm 7.22 *The closed sphere reconstruction of a function f on support X with respect to a chamfer metric.*

```
(1) Initialize: $r_i = f(x_i)$ if $x_i \in X$; $r_i := -\infty$ otherwise.
(2) For all pixels $x_i$ in forward scanning order:
    if $\max\{r_j - d_j \mid x_j$ backward neighbor of $x_i\} > r_i$, replace $r_i$ by this value.
(3) For all pixels $x_i$ in backward scanning order:
    if $\max\{r_j - d_j \mid x_j$ forward neighbor of $x_i\} > r_i$, replace $r_i$ by this value.
(4) Determine the set of pixels $x_i$ with $r_i \geq 0$.
```

7.4. Conclusions

In this chapter, the relation between mathematical morphology, geometry and scale has been pointed out and it has been argued that families of discs defined by chamfer metrics possess attractive properties as structuring elements.

In this chapter, chamfer metrics and some of their properties have been introduced. Chamfer metrics have been defined using linear combinations of prime vectors. This definition is equivalent to the more common definition based on shortest paths.

It has been established, which values can be assumed by a chamfer metric and what paths can occur as a shortest path between two points. For this purpose, the notions of adjacent prime vectors and normalized prime vectors have been introduced. The internal and external distance transform have been introduced, and their relation has been established. The connection of these distance transforms with the inclusion of structuring elements in an image will be central in the discussion of mathematical morphology using chamfer discs.

8

The Medial Axis and the Medial Line

8.1. Introduction

Binary images, modeled as subsets of the image plane, can be analyzed as a union of subsets, which are then considered to be the primitives in the image. Images can be decomposed into subsets in several ways.

Montanvert [74] uses a skeleton to partition an image into (more or less) convex subsets, which may be overlapping and which are bounded at narrow parts or *necks* in the objects. The interrelations between the sets are represented by a graph, which is derived from the skeleton. Pavlidis [83] decomposes polygons into maximal convex subsets. He considers convex polygons as more simple shapes than non-convex ones. Therefore, his decomposition is a decomposition of composed images into primitive ones. Pitas and Venetsanopoulos [85] describe a recursive morphological decomposition scheme which generates an image decomposition into non-overlapping sets. First, the largest value r is determined for which spheres of radius r are included in the image. Such spheres are then removed from the image and the remaining image is analyzed in the same way.

In this chapter, images are described as a union of maximal structuring elements. In its most general form, maximal structuring elements can be defined as follows. Let E denote the image plane (typically $\mathbb{R}^2$ or the square or hexagonal grid), on which translation is defined.

Definition 8.1 *Let $X \subseteq E$ be an image and let $\mathcal{B}$ be a set of subsets of E, called structuring elements. A structuring element $B_1 \in \mathcal{B}$ is called a maximal structuring element in X if*

(1) $B_1 \subseteq X$.

(2) $B_1 \subseteq B_2 \subseteq X$ implies $B_1 = B_2$ for all $B_2 \in \mathcal{B}$.

In the sequel, it will be assumed that the family of structuring elements consists of the translated and scaled versions of a single simple shape. In this case, there is a one-parameter family $\{B(r) \mid r \in R\}$ of structuring elements, where R is the set of 'radii' (e.g. $\mathbb{R}$ or $\mathbb{N}$) and $B(r) \subseteq B(r')$ if $r \leq r'$. The origin of the underlying space is defined to be the center of the structuring elements. The complete set of structuring elements is found by translation, i.e. $B(h,r) = B(r) + h$.

Since each structuring element has a center, it is possible to consider the set of all centers of maximal structuring elements. This set is called the *medial axis* [95].

Definition 8.2 *Let $X \subset E$ be a binary image and let $\{B(h,r) \mid r \in R, h \in E\}$ be a set of structuring elements. The medial axis M_X is the locus of the centers of maximal structuring elements contained in X.*

Each structuring element is completely defined by its center h and its radius r. Therefore, the set of maximal structuring elements present in an image is determined completely by the medial axis combined with a function which assigns to each medial axis point the radius of the corresponding maximal structuring element. This function is known as the *quench function*.

When some (very mild) conditions are satisfied, knowledge of the medial axis and the quench function—i.e. of the maximal spheres present in the image—allows for the reconstruction of the original image. When chamfer discs are used as structuring elements, reconstruction is possible whenever the image is bounded. For the continuous case, using closed Euclidean spheres, reconstruction is possible whenever the image is bounded and topologically closed.

When continuous images are considered, one often uses Euclidean discs as structuring elements. This is a suitable choice, because Euclidean discs are rotation invariant and because their relation to the Euclidean metric allows an interpretation in terms of size.

On a computer, images are commonly represented on the square grid or sometimes on the hexagonal grid. On such a grid, structuring elements can be defined by repeated dilation of a simple structuring element with itself. This is done for example by Maragos [62, 61]. This choice of structuring elements allows the construction of efficient algorithms, but does not result in an accurate approximation of the results which are obtained when using Euclidean discs in the continuous case. A better choice is therefore the use of discs defined by a *chamfer metric* [12], which can provide an accurate approximation of Euclidean metrics.

In this chapter, the results from chapter 7 are used to derive characterizations and efficient algorithms for the medial axis defined by the p-q–metric (section 8.2) and the p-q-r–metric (section 8.3). The algorithms for p-q-r–metrics are more complicated than those for the p-q–metrics but the p-q-r–metric can provide a more accurate approximation to the Euclidean metric. In section 8.4, a medial line will be presented which contains the medial axis. Such a medial line allows for reconstruction of the original object, while preserving object homotopy.

In chapter 9, the results obtained in this chapter will be used to construct algorithms for morphological operations such as size distributions, anti-size distributions, the opening transform and the pattern spectrum.

8.2. The Medial Axis for the p-q-Metric

In this section, we present the medial axis algorithm for p-q–metrics. This algorithm uses a no-upstream condition for the internal distance transform which is analogous to the no-upstream condition which holds for the medial axis in $\mathbb{R}^2$, provided with the Euclidean metric [67]. Recall that, if X is some open subset of $\mathbb{R}^2$, its distance transform ρ is the function from $\mathbb{R}^2$ to $\mathbb{R}$ defined by $\rho(x) = \inf\{d(x,y) \mid y \in X^C\}$. A point $x \in X$ is a medial axis point if and only if there is no $y \neq x$ such that $\rho(y) = \rho(x) + d(x,y)$.

The algorithm presented in this section for p-q–metrics uses the fact that these metrics are extending.

Definition 8.3 *Let d be a metric on a set E with range D. A value $r \in D$ is called an extending value if for each $x, y \in E$ there is a $z \in E$ such that $d(x,y) + d(y,z) = d(x,z)$ and $d(y,z) = r$.*

Definition 8.4 *A metric d on a set E with range D is called extending if each $r \in D$ is an extending value.*

The Euclidean metric is extending. Given x, y and r, the point z as described in the definition can be constructed using simple geometry: it is one of the intersections of the line xy with the circle with center y and radius r. Of the two intersection points, the one must be chosen for which y lies between x and z.

All p-q–metrics are extending as well. This can be shown by constructing a point z, as we did in the previous paragraph for the Euclidean metric. Let x and y be two points in $\mathbb{Z}^2$ and let $r \in D_{p-q}$. The vector $y - x$ can be written in the form $n_1v_1 + n_2v_2$, where v_1 and v_2 are adjacent prime vectors and n_1 and n_2 are non-negative integers. It can be assumed without loss of generality that $l(v_1) = p$ and $l(v_2) = q$. As $r \in D_{p-q}$, r can be written as $m_1p + m_2q$, where m_1 and m_2 are nonnegative integers. We can take $z = y + m_1v_1 + m_2v_2$. Then $z - y = m_1v_1 + m_2v_2$, so $d(y,z) = m_1p + m_2q = r$. Moreover, $z - x = (n_1 + m_1)v_1 + (n_2 + m_2)v_2$, so $d(x,z) = (m_1 + n_1)p + (m_2 + n_2)q = d(x,y) + d(y,z)$.

The following theorem presents the no-upstream condition for medial axis points for extending metrics.

Theorem 8.5 *Let d be a metric on $\mathbb{Z}^2$. Let X be a bounded subset of $\mathbb{Z}^2$. A point $x \in X$ for which $\rho(x)$ is an extending value is the center of a maximal sphere if and only if there is no $y \neq x$ such that $\rho(y) \geq \rho(x) + d(x,y)$ ("x has no upstream").*

PROOF. 'only if': suppose that there exists an $y \neq x$ such that $\rho(y) \geq \rho(x) + d(x,y)$. In that case every $z \in \overline{B}(x, \rho(x))$ satisfies

$$d(z,y) \leq d(z,x) + d(x,y) \leq \rho(x) + d(x,y) \leq \rho(y).$$

Therefore $\overline{B}(y, \rho(x))$ is a sphere containing $\overline{B}(x, \rho(x))$ and contained in X. Therefore, x is not the center of a maximal sphere.

'if': Suppose x is not the center of a maximal sphere. Then the sphere with center x and radius $\rho(x)$ must be contained in a closed sphere with center $y \neq x$ and radius $\rho(y)$. Let z be a point such that $d(y,x) + d(x,z) = d(y,z)$ and $d(x,z) = \rho(x)$. Such a points exists because $\rho(x)$ is an extending value. From $z \in \overline{B}(x, \rho(x)) \subseteq \overline{B}(y, \rho(y))$ it follows that $d(z,y) \leq \rho(y)$. From these relations it can be deduced that

$$\rho(y) - \rho(x) \geq d(z,y) - \rho(x) = d(z,y) - d(x,z) = d(x,y).$$

∎

Corollary 8.6 *Let d be an extending metric on $\mathbb{Z}^2$ and X a bounded subset. Then $x \in X$ is a medial axis point if and only if there is no $y \in X$ such that $\rho(y) \geq \rho(x) + d(x,y)$.*

This theorem provides a characterization of the medial axis points, but it cannot be used for the construction of an efficient algorithm: in order to determine whether x is a medial axis point of X, all points y must be inspected. In the case of the p-q–metric, the search can be limited to the neighbors of x.

Theorem 8.7 *Let X be a bounded subset of $\mathbb{Z}^2$, provided with a chamfer metric whose set of prime vectors is V and let $x \in X$. If there is a point y such that $\rho(y) \geq \rho(x) + d(x,y)$, then there is also an $y' \in \mathbb{Z}^2$ satisfying $y' - x \in V$ and $\rho(x) + d(x,y')$.*

PROOF. Let x and y be points as described in the theorem. Then

$$\overline{B}(x, \rho(x)) \subseteq \overline{B}(y, \rho(y)) \subseteq X.$$

There is a pair of adjacent prime vectors v_1 and v_2 such that $x - y = n_1 v_1 + n_2 v_2$ and n_1 and n_2 are non-negative. Without loss of generality, it can be assumed that $n_1 > 0$. We can now take $y' = x - v_1$. This implies $d(x,y) = d(x,y') + d(y',y)$ and $y' - x \in V$. It is now sufficient to prove that

$$\overline{B}(x,\rho(x)) \subseteq \overline{B}(y',\rho(x)+d(x,y')) \subseteq \overline{B}(y,\rho(y)).$$

The first inclusion follows from

$$d(x,p) \leq \rho(x) \Rightarrow d(y',p) \leq d(y',x) + d(x,p) \leq d(y',x) + \rho(x).$$

The second inclusion follows from:

$$\begin{gathered} d(y',p) \leq \rho(x) + d(x,y') \Rightarrow \\ d(y,p) \leq d(y,y') + d(y',p) \leq d(x,y') + d(y',y) + \rho(x) = \\ d(x,y) + \rho(x) \leq \rho(y). \end{gathered}$$

∎

Corollary 8.8 *Let X be a bounded subset of $\mathbb{Z}^2$, provided with the p-q–metric. Then $x \in X$ is a medial axis point if and only if there is no 8-neighbor y of x such that $\rho(y) \geq \rho(x) + d(x,y)$.*

This theorem suggests the following algorithm for the computation of the medial axis in the p-q–metric.

Algorithm 8.9 *The computation of the medial axis of a bounded subset X of $\mathbb{Z}^2$ with respect to the p-q–metric.*

(1) Compute the external distance transform of X.

(2) Compute the internal distance transform ρ of X from the external one.

(3) Mark all points x having no 8-neighbor y with $\rho(y) \geq \rho(x) + d(x,y)$.

This algorithm requires four image scans, and local computation only. It is possible to perform steps (2) and (3) in a single scan, but this makes the local operation to be performed in this step much more complicated, so this is not a good approach.

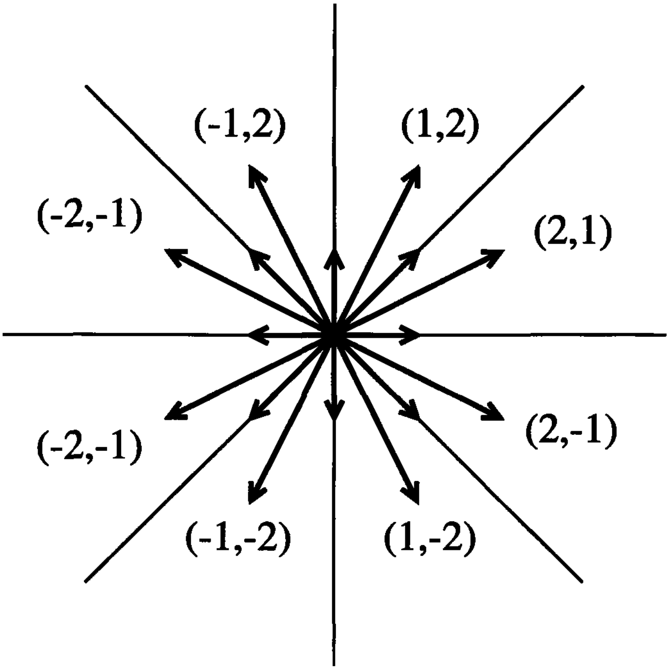


Figure 8.1: Horizontal, vertical and diagonal vectors partition the plane in eight parts, which can be labeled according the vector of type q they contain.

8.3. The Medial Axis for the p-q-r-Metric.

A p-q-r–chamfer metric can approximate the Euclidean metric more accurately than any p-q–metric [105]. On the other hand, the algorithms for the p-q-r–metric are more complicated than those for the p-q–metric. The medial axis algorithm described in the previous section can not be used, because the p-q-r–metric is not extending. This can be seen from the example in figure 8.2 for the 5-7-11–metric. The small sphere of radius 5 is contained in the larger one of radius 11, but the distance of their centers is 7, while the difference of their radii is only 6. This sections presents an algorithm which is suitable for the p-q-r–metric.

For the p-q-r–metric, there are three types of prime vectors: type p of the form $(\pm 1, 0)$ or $(0, \pm 1)$ with weight p, type q of the form $(\pm 1, \pm 1)$ with weight q and type r of the form $(\pm 1, \pm 2)$ or $(\pm 2, \pm 1)$ with weight r. Each vector of type r is adjacent to a vector of type p and a vector of type q, but no pair of vectors of type p and type q are adjacent (see figure 8.1). As a consequence, each shortest path between two points contains either vectors of type p and type r, vectors of type q and type r or vectors of a single type. The range D_{p-q-r}

of the p-q-r–metric consists of the points which can be written as $px + ry$ or $qx + ry$, for nonnegative integers x and y. Note that integers larger that qr can be written in both forms.

Like in the previous section, our aim is to determine for each point x whether there is a point $y \neq x$ such that $\overline{B}(x, \rho(x)) \subseteq \overline{B}(y, \rho(y))$. We will not try to find a criterion which decides whether this relation holds for two arbitrary points x and y. Rather, we will show that if, for a given x, such a point y exists, there is also a 16-neighbor y' of x such that $\overline{B}(x, \rho(x)) \subseteq \overline{B}(y', \rho(y'))$. It is then sufficient to check for each 16-neighbor y of x whether $\overline{B}(x, \rho(x)) \subseteq \overline{B}(y, \rho(y))$. If there is no such point, then x is a medial axis point, otherwise it is not.

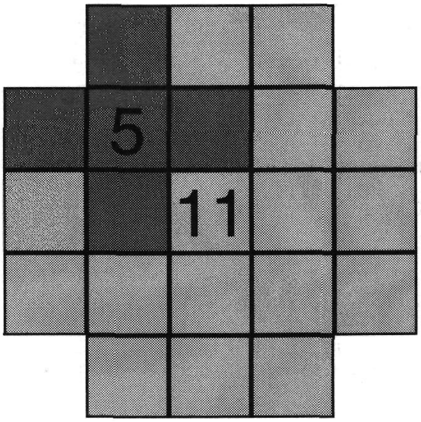


Figure 8.2: An example showing that the no-upstream criterion for the medial axis can not be applied to the 5-7-11 metric.

Theorem 8.10 *Let X be a bounded subset of $\mathbb{Z}^2$. A point $x \in X$ is a medial axis point if and only if there is no 16-neighbor y of x such that $B(x, \rho(x)) \subset B(y, \rho(y))$.*

Note that, if $\rho(x) \in p\mathbb{N} + r\mathbb{N}$ and $\rho(x) \in q\mathbb{N} + r\mathbb{N}$, $\rho(x)$ is an extending value and theorem 8.5 can be used. It can then be argued, like in the proof of theorem 8.7, that x is a medial axis point if there is no 16-neighbor y of x such that $\rho(y) \geq \rho(x) + d(x, y)$. Yet, for the p-q-r–metric, there are values of $\rho(x)$ which are in $p\mathbb{N} + r\mathbb{N}$ but not in $q\mathbb{N} + r\mathbb{N}$, or vice versa.

The proof of theorem 8.10 consists of three parts. First, some concepts are defined and some auxiliary results on convex polygons are shown. The it is show (lemma 8.17) that it is sufficient to inspect a star-shaped region, whose shape depends on $\rho(x)$. From this result, it will be derived (lemma 8.18) that it is sufficient to inspect a 16 point neighborhood.

Preparation

Before we come to the actual proof of theorem 8.10, the extension of the p-q-r–metric to the continuous plane and some result on convex sets are presented.

Definition 8.11 *Let V be the set of prime vectors used to define the p-q-r–metric and let l be the corresponding weight function. The p-q-r–chamfer metric on $\mathbb{R}^2$ is defined by*

$$d(x,y) = \inf\{\sum_{v \in V} |n_v| l(v) \mid \sum_{v \in V} n_v v = y - x, n_v \in \mathbb{R}\} \tag{8.1}$$

It is not difficult to see that the p-q-r–chamfer metric is indeed a metric, and that the "spheres" defined by this metric are polygons with 16 corners. The restriction of the p-q-r–chamfer metric defined above to the grid $\mathbb{Z}^2$ yields the p-q-r–chamfer metric on $\mathbb{Z}^2$.

Spheres in the continuous plane are defined as follows.

Definition 8.12 *For each $r \in \mathbb{R}, r \geq 0$ and $x \in \mathbb{R}^2$, the continuous sphere $S(x,r)$ is defined by*

$$S(x,r) = \{y \in \mathbb{R}^2 \mid d(x,y) \leq r\}.$$

Continuous and discrete spheres are related as follows. Suppose $x \in \mathbb{Z}^2$ and $r \in \mathbb{R}, r \geq 0$. Then

$$S(x,r) \cap \mathbb{Z}^2 = B(x, r^-), \tag{8.2}$$

where r^- is the number $\max\{s \in D \mid s \leq r\}$. This implies that, for $x \in \mathbb{Z}^2$ and $r \in D$,

$$B(x,r) = S(x,r) \cap \mathbb{Z}^2. \tag{8.3}$$

For continuous spheres, the equivalence $S(x,r) \subseteq S(y,s) \Leftrightarrow s \geq r + d(x,y)$ holds. From figure 8.2, it can be seen that this is not true for discrete spheres. The figure uses the 5-7-11–metric. The smaller sphere of radius 5 is contained in the larger sphere of radius 11, but the distance between their centers is 7, while the difference of their radii is only 6.

Definition 8.13 *The relation $\sqsubseteq$ between subsets A and B of $\mathbb{R}^2$ is defined by*

$$A \sqsubseteq B \text{ iff } A \cap \mathbb{Z}^2 \subseteq B \cap \mathbb{Z}^2.$$

Obviously, $A \subseteq B$ implies $A \sqsubseteq B$, but the reverse is in general not true.

The following lemmas describe the behavior of convex sets under scalings. The scaling $M(c,\lambda)(X)$ of a subset X of $\mathbb{R}^2$ with center $c \in \mathbb{R}^2$ and magnification factor $\lambda \in \mathbb{R}$ is defined as the set $\{\lambda(x-c) + c \mid x \in X\}$.

Lemma 8.14 *Let X be a convex subset of $\mathbb{R}^2$ and let $x \in X$, $\lambda > 1$. Then $X \subseteq M(x, \lambda)(X)$.*

Lemma 8.15 *Let P be a convex polygon. Suppose that the carriers of two of its sides AB and CD intersect in a point Z, and that B is closer to Z than A and C is closer to Z than D. Suppose λ is a scaling factor such that $B' = \lambda(B - Z) + Z$ lies between B and A and $C' = \lambda(C - Z) + Z$ lies between C and D. Then the only part of P which is not contained in $M(Z, \lambda)(P)$ is the polygon bounded by the boundaries of the polygons between B and C and between B' and C', and the segments BB' and CC'.*

This lemma is illustrated in figure 8.3. It can be proven by considering, for all lines l through Z, the intersections of l with P and $M(Z, \lambda)(P)$. These intersections are either both equal to $\emptyset$ or partially overlapping segments of l.

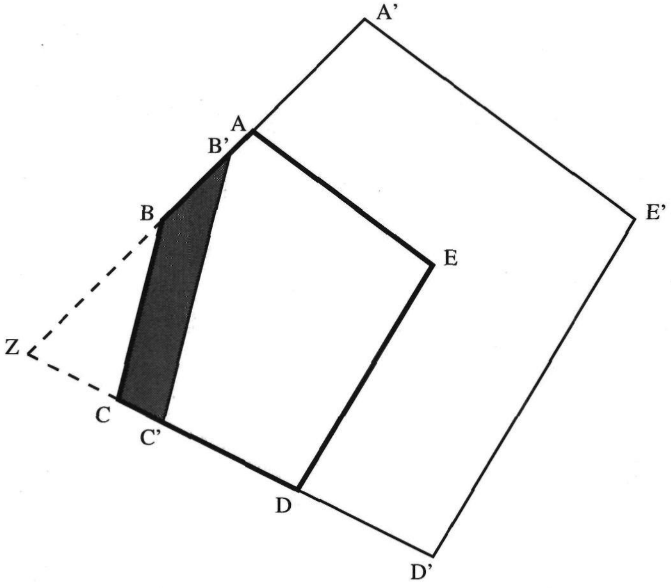


Figure 8.3: Illustration of lemma 8.15. The only part of the small polygon ABCDE which is not contained in the magnified polygon A'B'C'D'E' is the shaded area.

Restriction to a star-shaped neighborhood

We now come to the first part of the proof of theorem 8.10. In order to simplify the discussion, we will suppose that x is the origin. We divide $\mathbb{Z}^2$ in eight octants, as shown in figure 8.1. The octants will be called the $(2, 1)$–octant, the $(1, 2)$–octant, the $(-1, 2)$–octant, etc., after the prime vector of type r they contain. Note that the symmetry group D_4, which consists of four rotations (including the identity) and four reflections in horizontal, vertical and diagonal lines, maps a point in one octant to a point in each of the other octants. Therefore we can assume without loss of generality that y lies in the $(-2, -1)$–octant.

Suppose that z is a point in the $(2, 1)$–octant. Let $z_1 = z, z_2, \ldots, z_8$ be the images under D_4 of the point z. It can easily be seen that $d(y, z_i) \leq d(y, z)$ for all i. Consequently, the sphere $\overline{B}(0, \rho(0))$ is completely contained in $\overline{B}(y, \rho(y))$ if the intersection of the sphere centered at 0 with the $(2, 1)$–octant is completely contained in the sphere centered at y.

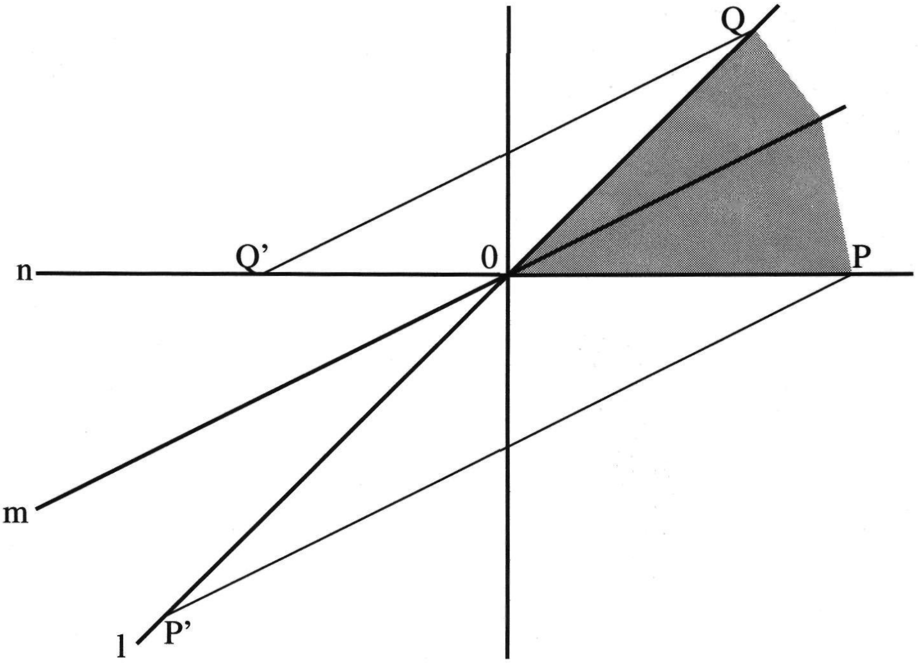


Figure 8.4: The geometry of the situation under consideration. See the text for an explanation.

As a consequence of the two previous paragraphs, we can restrict to the situation depicted in figure 8.4. Let l, m and n be the lines through the origin in the directions $(1, 1)$, $(2, 1)$ and $(1, 0)$, respectively. Let P and Q be the points $(\lfloor \rho(0)/p \rfloor, 0)$ and $(\lfloor \rho(0)/q \rfloor, \lfloor \rho(0)/q \rfloor)$ and let $P' = (-\lfloor \rho(0)/p \rfloor, -\lfloor \rho(0)/p \rfloor)$ and $Q' = (-\lfloor \rho(0)/q \rfloor, 0)$ be their projections on the lines l and n in the $(2, 1)$–direction. The shaded region represents the intersection of $\overline{B}(0, \rho(0))$ with the $(2, 1)$–octant. This region will be referred to as the *opposite part* of the sphere.

It lies above the line PP' and below the line QQ'.

Let y be a point in the $(-2,-1)$–octant with $\overline{B}(0,\rho(0)) \subseteq \overline{B}(y,\rho(y))$. We are now ready to construct a point y' near 0 such that $\overline{B}(0,\rho(0)) \subseteq \overline{B}(y',\rho(y'))$ as well.

Lemma 8.16 *Let y be a point in the $(-2,-1)$–octant for which the inclusion $\overline{B}(0,\rho(0)) \subseteq \overline{B}(y,\rho(y))$ holds. Then $\overline{B}(0,\rho(0)) \subseteq \overline{B}(y',\rho(y'))$ if y' satisfies the following conditions:*

(1) y' lies in the $(-2,-1)$–octant.

(2) $d(y,z) = d(y,y') + d(y',z)$ for every z in the opposite part of the sphere.

PROOF. It is sufficient to show that

$$\overline{B}(0,\rho(0)) \subseteq \overline{B}(y',\rho(y) - d(y,y')) \subseteq \overline{B}(y,\rho(y)).$$

As y' lies in the $(-2,-1)$–octant, the first inclusion can be proved by showing that $d(y',z) \le \rho(y) - d(y,y')$ for each point z in the opposite part, i.e. for each z in the $(2,1)$–octant with $d(z,0) \le \rho(0)$. But according to (2), we have for such points z: $d(y',z) = d(y,z) - d(y,y')$. Because $\overline{B}(0,\rho(0)) \subseteq \overline{B}(y,\rho(y))$ we know $d(y,z) \le \rho(y)$, so $d(y',z) \le \rho(y) - d(y,y')$.

The second inclusion holds because $d(z,y') \le \rho(y) - d(y,y')$ implies $d(z,y) \le d(z,y') + d(y',y) \le \rho(y)$. ∎

Lemma 8.17 *Let X be a bounded subset of $\mathbb{Z}^2$. A point $x \in X$ is a medial axis point if there is no point y with $B(x,\rho(x)) \subseteq B(y,\rho(y))$ and one of the following holds:*

(1) $y - x = (a,0)$ or $(0,a)$, with $|a| \le \rho(x)/q$.

(2) $y - x = (a,a)$ or $(a,-a)$, with $|a| \le \rho(x)/p$.

(3) $y - x = (\pm1,\pm2)$ or $(\pm2,\pm1)$.

PROOF. Without loss of generality, we can consider the situation of figure 8.4. We will construct a point y' which satisfies the conditions mentioned in lemma 8.16. The point y lies in one of five regions (see figure 8.5).

Region 1. $y = -\alpha(2,1) - \beta(1,1)$ with $\alpha \ge 0, \beta \ge \lfloor\rho(0)/p\rfloor$. Then we take $y' = P'$. Clearly, P' lies in the $(-2,-1)$–octant. We have $P' - y = \alpha(2,1) + (\beta - \lfloor\rho(0)/p\rfloor)(1,1)$, a nonnegative linear combination of $(2,1)$ and $(1,1)$. Because all points in the opposite part of the sphere lie above the line PP' and below l, it is possible for each point z in the opposite part to write $z - P'$ as a nonnegative linear combination of $(1,1)$ and $(2,1)$ as well. This implies $d(y,P') + d(P',z) = d(y,z)$.

Region 2. Suppose $y = -\alpha(2,1) - \beta(1,1)$ with $\alpha \ge 0, 0 < \beta < \lfloor\rho(0)/p\rfloor$. Then we take $y' = -\beta(1,1)$. Clearly, this y' lies in the $(-2,-1)$–octant. We have $y' - y = \alpha(2,1)$. Because the opposite part lies below the line l and above

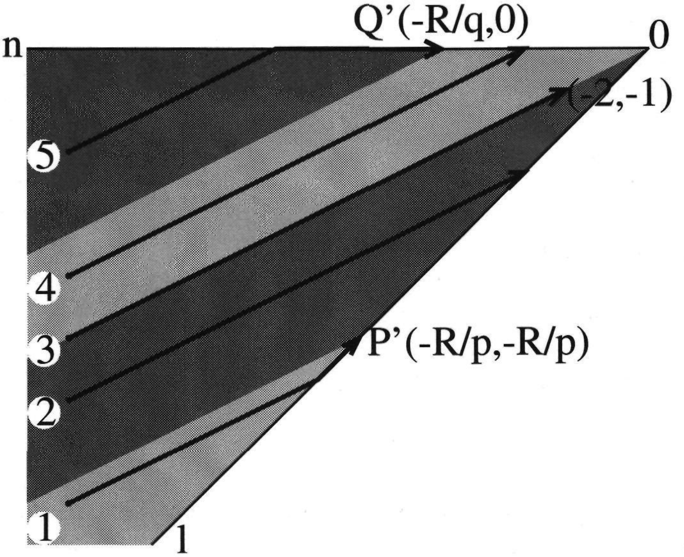


Figure 8.5: The construction of the points y' from the points y on the left. The points y' are found by moving in the $(2,1)$-direction until the line l or n is hit. If the line is hit to the left of P' or Q', move rightward along the line to one of these points. Regions 1, 2, 4 and 5 are the shaded areas; region 3 is the half line ending at $(-2,-1)$.

the x-axis, it is possible for all points z in the opposite part to write $z - y'$ as a nonnegative linear combination of either $(1,1)$ and $(2,1)$ or $(2,1)$ and $(1,0)$. In both cases, $d(y,y') + d(y',z) = d(y,z)$.

Region 3. Suppose $y = -\alpha(2,1)$ with $\alpha > 0$. Then we take $y' = (-2,-1)$. Clearly y' lies in the $(-2,-1)$–octant. We have $y' - y = -(\alpha - 1)(2,1)$. As the opposite part lies below the line l and above x-axis, it is possible write for each z in the opposite part to write $z - y'$ as a nonnegative linear combination of either $(1,1)$ and $(2,1)$ or $(2,1)$ and $(0,1)$. In both cases $d(y,y') + d(y',z) = d(y,z)$.

Region 4. Suppose $y = -\alpha(2,1) - \beta(1,0)$ with $\alpha \geq 0, 0 < \beta < \lfloor \rho(0)/q \rfloor$. Take $y' = -\beta(1,0)$, as we did for y in region 2.

Region 5. Suppose $y = -\alpha(2,1) - \beta(1,0)$ with $\alpha \geq 0, \beta \geq \lfloor \rho(0)/q \rfloor$. Take $y' = Q'$, as we did for y in region 1. ∎

We have now arrived at a local neighborhood of the origin which must be inspected in order to determine if the origin is a medial axis point. This environment contains $\lfloor \rho(0)/q \rfloor$ points in each of the four horizontal or vertical directions, $\lfloor \rho(0)/p \rfloor$ points in each of the four diagonal directions and eight points at a knights jump from the origin. Of course, similar environments for

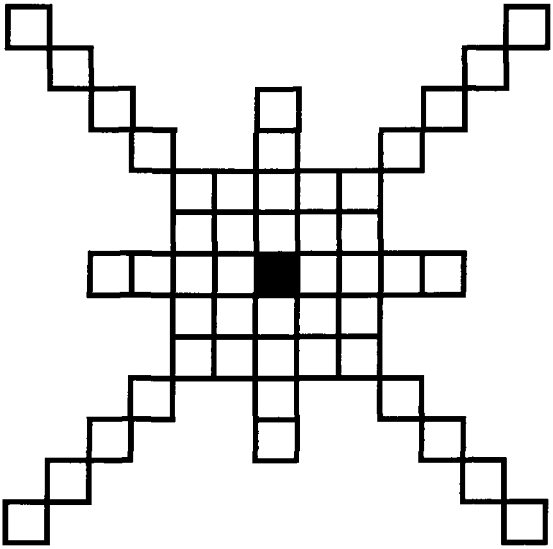


Figure 8.6: The neighborhood which must be investigated for a point with a distance transform value of 31.

other points can be found by translation. Note that the size of the environment depends on the value of $\rho(x)$. Figure 8.6 shows the environment corresponding to $\rho(0) = 31$ for the 5-7-11-metric. The environment contains horizontal and vertical branches of length $\lfloor\frac{31}{7}\rfloor = 4$, diagonal branches of length $\lfloor\frac{31}{5}\rfloor = 6$ and eight points at a knights move from the center.

Restriction to a 16 point neighborhood

We will now show that it is possible to restrict the neighborhood to sixteen points. Therefore, we must consider the case where y is a point on an axis or on a diagonal, but not a neighbor of x.

Lemma 8.18 *Let $\rho, \tau \in D$, $a \in$ IN with $a > 1$.*
(1) If $B((0,0),\rho) \subseteq B((-a,0),\tau)$, then there is an $\rho' \in D$ such that

$$B((0,0),\rho) \subseteq B((-1,0),\rho') \subseteq B((-a,0),\tau).$$

(2) If $B((0,0),\rho) \subseteq B((-a,-a),\tau)$, then there is an $\rho' \in D$ such that

$$B((0,0),\rho) \subseteq B((-1,-1),\rho') \subseteq B((-a,-a),\tau).$$

Proof. We will prove only the first part of the lemma, as the second part can be proved in a similar way. It can be assumed without loss of generality that τ is the smallest value in D for which $B((0,0),\rho) \subseteq B((-a,0),\tau)$ holds. The theorem is shown by considering continuous spheres in stead of discrete ones. We will prove that there is an $\rho' \in \mathbb{R}$ such that

$$S((0,0),\rho) \sqsubseteq S((-1,0),\rho') \sqsubseteq S((-a,0),\tau). \tag{8.4}$$

The lemma follows immediately from this relation by applying (8.2) and definition 8.13.

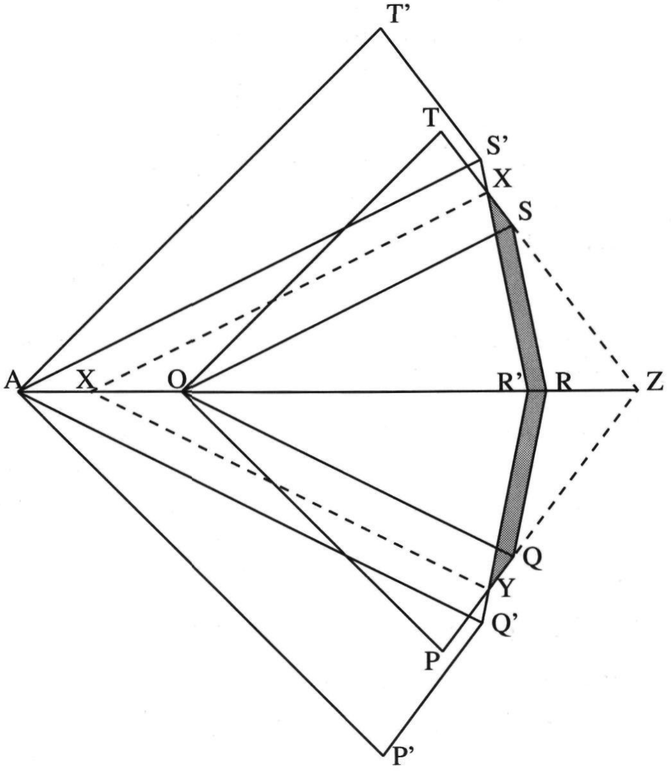


Figure 8.7: The relative positions of two spheres. See the text for an explanation.

Consider the octagons $S((0,0),\rho)$ and $S((-a,0),\tau)$. Parts of these polygons are depicted in figure 8.7. The center of the small sphere is the origin O. Some of its corners, P, Q, R, S and T, are marked. The center of the larger sphere is $A = (-a, 0)$; some of its corners, P', Q', R', S' and T', are marked.

There must be a grid point on the segment RS or on the segment ST. If S' lies below the line ST, both the segments RS and ST lie outside the larger sphere, and so does at least one grid point. This would violate the inclusion relation. Therefore, S' lies above the line ST.

If S' lies above the line RS, the smaller sphere would be included in the interior of the larger one, and it would be possible to find a value $s' < s$ for which $S((0,0),r) \subseteq S((0,-a),s')$ holds. This would imply $B((0,0),r) \subseteq B((0,-a),s'^{-})$, violating the assumption that s is the smallest value for which this inclusion holds. Therefore, S' lies below the line RS.

The boundaries of the spheres intersect in the points X and Y. The projection of these points in the $(2,1)-$ and the $(2,-1)-$direction, respectively, is the point B. Note that two degenerate cases can occur. The points R and R' can coincide, or the point S' can lie on the segment ST. In these situations, X will be chosen to be S or S', respectively.

Note that the relation $S((0,0),\rho) \subseteq S((-a,0),\tau)$ holds only if R and R' coincide. On the other hand, since $S((0,0),\rho) \sqsubseteq S((-a,0),\tau)$, the shaded area $YQRSXR'$ does not contain any points from $\mathbb{Z}^2$.

The the proof is as follows. We construct a parameterized family of spheres $S((-t,0),r(t))$ such that $r(0) = \rho$, $r(a) = \tau$ and

$$S((-t,0),r(t)) \sqsubseteq S((-t',0),r(t')) \tag{8.5}$$

if $t \leq t'$. This implies

$$S((0,0),r(0)) \sqsubseteq S((-t,0),r(t)) \sqsubseteq S((-a,0),s) \tag{8.6}$$

for any $t \in [0,a]$. Then (8.4) follows directly by taking $t = 1$ in (8.6).

The family $S((-t,0),r(t))$ is found by transforming $S((0,0),\rho)$ gradually into $S((-a,0),\tau)$. Intuitively, the family can be described as the sequence of spheres which is obtained by gradually shifting and enlarging $S((0,0),\rho)$ in such a way that S moves along the segment SX to X, and then along the segment XS' to S', while the center of the sphere moves from $(0,0)$ to $(-a,0)$.

The family consists of two parts. The first part is obtained by applying the transformation $M(Z,\lambda)$ to $S((0,0),r)$ with λ ranging from 1 to $|R'X|/|RS|$. This yields the spheres $S((-t,0),r(t))$ with t between 0 and b.

The second part of the family is obtained by applying the transformation $M(R',\lambda)$ to the sphere $S((-b,0),r(b))$, with λ ranging from 1 to $|R'S'|/|R'X|$. This yields the spheres $S((-t,0),r(t))$ with t between b and a.

It remains to be shown that the family thus obtained satisfies (8.5). For the first part of the family, this follows from lemma 8.15, combined with the fact that the shaded area in figure 8.7 contains no grid points. For the second part, increasingness with respect to $\subseteq$ follows immediately from lemma 8.14. Therefore, increasingness with respect to $\sqsubseteq$ certainly holds. ∎

Now theorem 8.10 follows directly from lemma 8.17 and lemma 8.18.

* * *

Theorem 8.10 implies that it is sufficient to consider 16 point neighborhoods in the detection of medial axis point. We will now describe the operations which are to be performed in such a neighborhood, and the resulting algorithm will be presented.

Let $x \in X$ be a point of a set $X \subset \mathbb{Z}^2$. In order to determine whether x is a medial axis point, it must be checked for each neighbor y of X in a sixteen point environment, whether

$$B(x, \rho(x)) \subseteq B(y, \rho(y)) \quad (8.7)$$

holds. There are three cases to be discerned: y is a direct (4-connected) neighbor of x, y is an indirect (8-connected but not 4-connected) neighbor of x, or y is at a knights move from x.

In the last case, $d(y,p) = d(y,x) + d(x,p) = r + d(x,p)$ for all p in the $(2,1)$–octant. Therefore, $B(x,\rho(x)) \subseteq B(y,\rho(y))$ if and only if $\rho(y) \geq \max\{d(y,p) \mid p \in B(0, \rho(0,0))\} = r + \rho(0)$.

Now suppose that y is a direct neighbor of x. It can be assumed without loss of generality that $x = (0,0)$ and $y = (-1,0)$. Then (8.7) is true if and only if $\rho((-1,0)) \geq \rho((0,0)) + \Delta\rho$, where $\Delta\rho$ is given by

$$\Delta\rho = \max\{d((-1,0),z) - \rho(0) \mid z \in B((0,0), \rho((0,0)))\} \quad (8.8)$$

Consider again figure 8.7, and suppose that A is the point $(-1,0)$.

The point z which maximizes $d((-1,0),z)$ in (8.8) is a grid point on the boundary of the larger sphere which is also included in the smaller sphere. If S' lies strictly above the line ST, z lies on the segment $R'X$, but if S' lies on the segment ST, then z can also lie on the segment XT.

In the first case, $d((0,0),z) = \max\{s \in p\mathbb{N} + r\mathbb{N} \mid s \leq \rho((0,0))\}$, so $d((-1,0),z) = p + \max\{s \in p\mathbb{N} + r\mathbb{N} \mid s \leq \rho((0,0))\}$. In the second case, $d((-1,0),z) = d((0,0),z) + r - q = \rho((0,0)) + r - q$. Using the notation $s_{(pr)} = \max\{s' \in p\mathbb{N} + r \qquad ' \leq s\}$, we find $\qquad \mathbb{N} \mid s$

$$\Delta\rho = \max(r - q, \rho(x)_{(pr)} - \rho(x) + p). \quad (8.9)$$

For all but a finite number of values for $\rho(x)$, $\rho(x) \in p\mathbb{N} + r\mathbb{N}$. In this situation, the point z which maximizes $d((-1,0),z)$ in (8.7) lies on the segment RS, therefore $\rho(x)_{(pr)} = \rho(x)$ and $\Delta\rho = p$. Therefore, a table of the values of $\Delta\rho$ can be precomputed for those values of $\rho(x)$ which are not in $p\mathbb{N} + r\mathbb{N}$, while for other values $\Delta\rho = p$ can be used.

A expression similar to (8.9) can be found for the case where y is an indirect neighbor of x. Note that for p-q-r–metrics with small values of p, q and r, we have $p + q - r = 1$, such that $\Delta\rho$ will always be equal to p or $p - 1$.

These observations lead to the following algorithm.

Algorithm 8.19 *The computation of the medial axis of a bounded subset X of $\mathbb{Z}^2$ with respect to the p-q-r–metric.*

```
(1) Compute the internal distance transform $\rho$ of $X$.
(2) For each $x \in X$:
    - If a direct neighbor $y$ of $x$ satisfies $\rho(y) \geq \max(\rho(x) + r - q, \rho(x)_{(pr)} + p)$, mark $x$ as a non-medial axis point;
    - If an indirect neighbor $y$ of $x$ satisfies $\rho(y) \geq \max(\rho(x) + r - p, \rho(x)_{(qr)} + q)$, mark $x$ as a non-medial axis point;
    - If a knights move neighbor $y$ of $x$ satisfies $\rho(y) \geq \rho(x) + r$, mark $x$ as a non-medial axis point;
    - otherwise mark $x$ as a medial axis point.
```

8.4. The Medial Axis and the Medial Line

The literature shows great confusion concerning the terms 'medial axis' and 'skeleton'. There are two types of sets which are denoted by these terms. One is the locus of centers of maximal spheres in an object, the other one is a thin subset of an object which lies (more or less) in the middle of the object and has the same homotopy as the object. In the continuous case, this does not lead to great difficulties, because usually, the differences between these two sets are very small [67].

In the discrete case, this confusion is more serious, because the locus of centers of maximal spheres often is a set with much more connected components than the object itself. In general, it can be said that authors from the field of mathematical morphology [61, 95] use the term skeleton for the locus of centers of maximal spheres, while others [4, 24, 80] call this set the medial axis and use the term skeleton for a thin set having the same homotopy as the object. There are, however, exceptions to this rule [71]. Other names occurring in the literature for the different types of sets are medial line, symmetric axis and homotopic thinning. In this thesis, the term medial axis denotes the locus of centers of maximal spheres and the term medial line denotes a thin set of the same homotopy as the object.

We repeat the following definitions from chapter 7, now specialized for the case where the structuring elements are discs defined by some discrete metric.

Definition 8.20 *Let d be a discrete metric on a set E. Let X be a bounded subset of E. A sphere $\overline{B}(x,r) \subseteq X$ is called a maximal sphere in X if, for each sphere $\overline{B}(y,s)$, the inclusion $\overline{B}(x,r) \subseteq \overline{B}(y,s) \subseteq X$ implies $x = y$ and $r = s$.*

Definition 8.21 *Let d be a discrete metric on a set E and let X be a bounded subset of E. The medial axis M_X of X is the set of centers of maximal spheres in X.*

If the medial axis of an object is given, together with the value of the external distance transform in the medial axis points, the object can be reconstructed using the equality

$$X = \bigcup_{s \in M_X} \{x \in \mathbb{Z}^2 \mid d(x,s) < \rho_X^{\text{ext}}(s)\}. \tag{8.10}$$

The restriction of the distance transform to the medial axis is sometimes called the quench function [95]. The reconstruction property can be seen as follows: for each $x \in X$, the set of spheres $B(y,s)$ containing x and contained in X is not empty because $\overline{B}(x,0) = \{x\}$, which is also an open sphere of some radius (d is discrete), is such a sphere. The set of all such spheres is partially ordered by inclusion. Because X is bounded, it has a finite number of elements. Hence, there is a maximal ball $B(y,R)$ containing x where y is included in the medial axis. Therefore, $x \in \{z \in \mathbb{Z}^2 \mid d(z,y) < \rho_X^{\text{ext}}(y)\} \subseteq \bigcup_{s \in M_X} \{z \in \mathbb{Z}^2 \mid d(z,s) < \rho_X^{\text{ext}}(s)\}$.

The inclusion in the other direction can easily be verified using the definition of the distance transform.

For a general metric, it can be difficult to compute the medial axis, but for p-q–metrics and p-q-r–metrics, algorithms 8.9 and 8.19 can be used. In general, the medial axis of an object will have more connected components that the object itself. Therefore, it is interesting to look at medial lines. A medial line S_X of a bounded subset X of $\mathbb{Z}^2$ is a subset of X which is thin, has the same homotopy as X and lies in the middle of X. The medial line can be used as a description or a compact representation of shape. A medial line of a subset of $\mathbb{R}^2$ can be defined without too much trouble [67], but the definition of a medial line for discrete sets is much more intricate.

Many authors have worked on the problem of defining a medial line for discrete sets. One approach [3, 24, 71, 74, 80] starts with choosing some "special configuration" points, such as maxima or saddle points of the distance transform, as a subset of the medial line. In general, this set will not have the correct homotopy. This is repaired by computing arcs between the points already selected in such a way that the resulting set has the correct homotopy. The difficult part is to prove that the resulting set has indeed the correct homology. Some authors provide strict correctness proofs [71, 74, 80], some [3, 24] do not.

Another approach [95, 43] is based on thinning. All object pixels are scanned in some order, and they are removed if this can be done without changing the homotopy of the object. At the end, a set of pixels remains, none of which can be removed without changing the homotopy of the set. This set is a medial line.

We present a medial line which contains the medial axis. The knowledge of the medial line, together with the value of the distance transform on the points in the medial line suffices for the reconstruction of the object.

Our medial line algorithm is based on the work of Hilditch [43]. Her algorithm is based on thinning. In the present situation, thinning is performed under the condition that a medial axis point can never be removed, even if this removal would not change the homotopy. The resulting set is sometimes called an anchor skeleton. An example of a medial line of this type is shown in figure 8.8. It can be seen that the end points of the branches of the medial axis are detached from the central part. The medial line has the same homotopy as the original object.

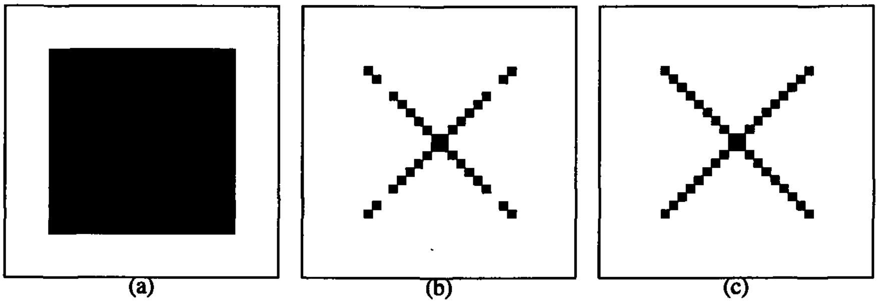


Figure 8.8: A binary object (a), its medial axis defined by the 5-7-11–metric (b) and the medial line calculated by Hilditch anchor skeletonization (c).

The author conjectures that a homotopy-preserving medial line can also be derived from the medial axis using a path-climbing algorithm but has not been able to prove the correctness of his algorithm. This algorithm is based on the work of Dorst [24] and Niblack *et al.* [80]. It is a steepest path climbing type of algorithm. Our algorithm differs with those of Dorst and Niblack *et al.* in the choice of starting points for the paths: the starting points in our algorithm are the medial axis points and those points which form a one or two pixel wide connection between larger parts of the object. Although this algorithm seems to produce medial lines of the correct homotopy, the author has not been able to find a proof for this. The problem lies in necks of an object, which are narrow connections between two wider parts of the object. It must be shown that there are centers of maximal spheres in such a narrow part, and that paths of steepest ascent leave in both directions to the wider parts of the object.

8.5. Conclusions

In this chapter, a characterization of the medial axes defined by p-q–metrics and p-q-r–metrics has been given and efficient algorithms for their computation have been derived. From the medial axis, a medial line has been derived. This medial line can be derived from the medial axis using thinning, as described by Hilditch; such a thinning provably yields a homotopy preserving medial line. It is conjectured that a second algorithm using path climbing also computes a homotopy preserving medial line.

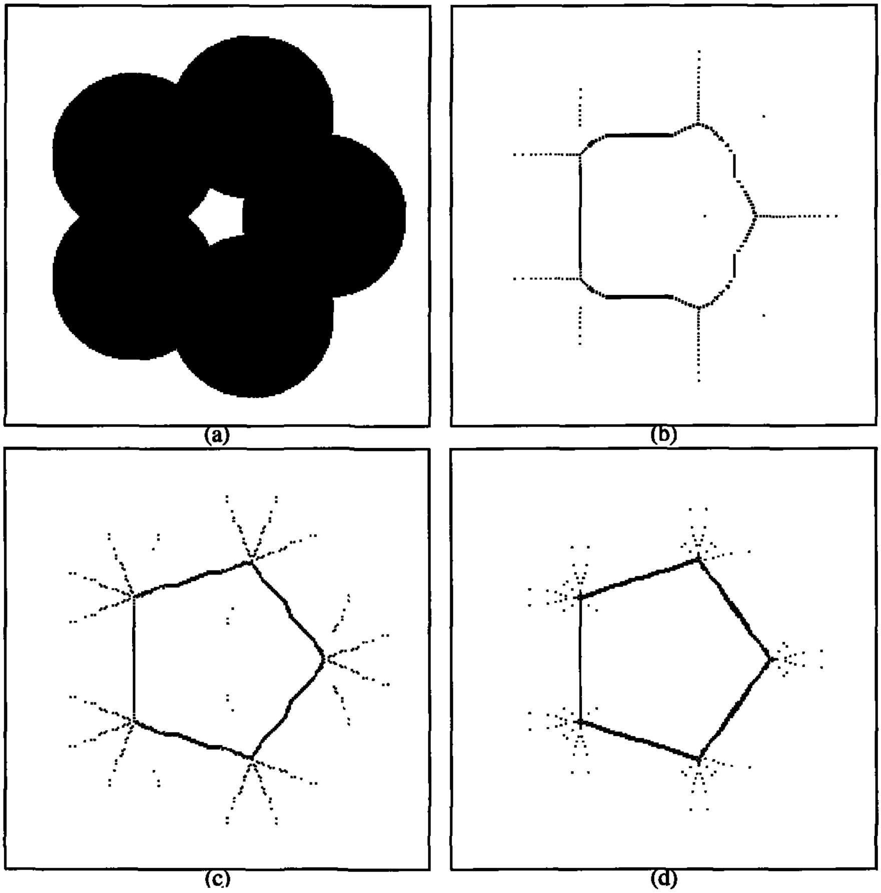


Figure 8.9: An image containing five Euclidean discs (a) and its medial axes as defined by (b) the square structuring element, (c) the 5-7–metric and (d) the 5-7-11–metric

The characterization of the medial axis defined by the p-q-r–metric is based on the fact that p-q–metrics are extending. As p-q-r–metrics are not, a more complicated analysis is needed for that case. This analysis requires the extension of chamfer metrics to the continuous plane.

These are useful results, because discs defined by chamfer metrics provide a much closer approximation to Euclidean discs than the structuring elements which are used more commonly in mathematical morphology on discrete images. Yet the computational cost of our algorithms is of the same order of magnitude as the cost of conventional medial axis algorithms. The use of chamfer discs as structuring elements provides better rotation invariance, as can be seen in figure 8.9, which is similar to the one presented by Verwer [104] in his discussion of skeletonisation based on chamfer metrics. When square structuring elements are used, artifacts can occur.

The structure elements used most commonly in mathematical morphology, the diamond and the square, can be treated within our framework, because they correspond to the city-block metric and the chess-board metric, respectively.

Recently, Borgefors [14] has presented (independently of us) a characterization of the centers of maximal discs in the 5-7-11–metric which is similar to ours. Although her results are correct, she did not prove that it is sufficient to consider only 16 neighbors for each point. As the 5-7-11–metric is not extending, some more elaborate analysis, as presented in this paper, is required. Moreover, she did not present the 'general rule' for the evaluation of neighboring pairs, described by our equations (8.8) and (8.9). Arcelli and Frucci [2] mention similar results, but do not present any details.

Now that the characterization of the medial axis in the p-q-r–metric has been completed, the next challenge ahead is the characterization for chamfer metrics with more than three types of prime vectors. Thiel and Montanvert [101] describe a procedure for the optimal choice of such prime vectors and the associated weights.

9

Openings with Chamfer Discs

9.1. Introduction

Granulometries or size distributions [66] are a morphological tool for measuring the grain sizes in an image. Their definition is based on a formalization of a sieving process. The rôle of the sieves is played by opening transforms with various structuring elements. The openings are parameterized by a size parameter and the corresponding structuring elements establish the relation between the 1-D size parameter and the notion of *size* in a higher- dimensional image.

As argued before, it is often attractive to base the notion of size on the Euclidean metric in the plane. This can be one by choosing Euclidean discs as the structuring elements defining the size distribution. For morphology on the discrete grid, one would choose the intersection of Euclidean discs with the grid as structuring elements.

Computing a size distribution with Euclidean discs can be done only at high computational cost. On the other hand, the more commonly used types of structuring elements, such as squares or diamonds, do not provide a reasonable approximation to the Euclidean disc. Again, discs defined by chamfer metrics are a good alternative.

In the next section, discs defined by chamfer metrics are used to define size distributions and their adjoint operators, the anti-size distributions. These operators will be analyzed using the results presented in the previous chapters, and efficient algorithms for their computation will be derived.

Size distributions consist of a family of operators, each of which maps one image into another. It is attractive to consider image operators which generate a condensed description of the action of a complete family of operators. Such an operator is the opening transform, an operator which maps a binary image into a grey scale image and represents the action of a complete size distribution in a single image. Another example is the pattern spectrum, which is an operating which describes a binary image by a function. This function describes the relation between the scale parameter of a size distribution and the area of the transformation of the image with the corresponding operator from the size distribution. In section 9.3, the opening transform and the pattern spectrum defined by chamfer discs are defined and efficient algorithms for their computation are defined.

9.2. Size Distributions and Anti-Size Distributions

In this section, size distributions and anti-size distributions based on the chamfer metric are constructed and efficient algorithms for performing these operations are derived.

Definition 9.1 *Let E be some set and let Λ be some totally ordered set. A size distribution [61, 95] is a family $\{\alpha_r\}_{r \in \Lambda}$ of operators mapping subsets of E to subsets of E such that for all $X, Y \subset E$, $r, s \in \Lambda$*

(1) $X \subset Y \Rightarrow \alpha_r(X) \subset \alpha_r(Y)$.

(2) $\alpha_r(X) \subset X$.

(3) $\alpha_r \circ \alpha_r = \alpha_r$.

(4) $\alpha_r \circ \alpha_s = \alpha_{\max(r,s)}$.

The first three conditions are exactly the conditions which are used to define that α_r is an opening [95]. The fourth condition describes the composition behavior of these openings. Note that the third condition is a consequence of the fourth; it is written down in order to clarify the fact that each α_r is an opening.

In the sequel, the index set Λ will be the range D of the metric under consideration. The goal is to define a size distribution based on spheres in a discrete metric. A first attempt could be to let each α_r be a structural opening [87] with the sphere $\overline{B}(r)$. This opening is defined by

$$X \circ \overline{B}(r) = \bigcup_{\overline{B}(h,r) \subset X} \overline{B}(h,r). \tag{9.1}$$

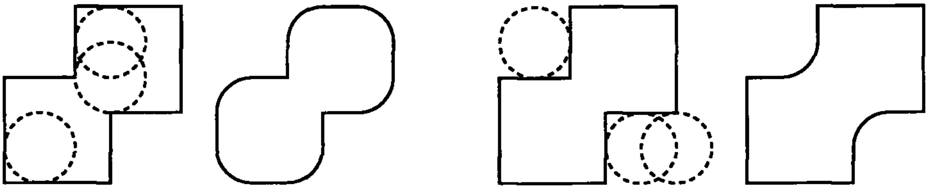


Figure 9.1: Examples of the structural opening (a) and closing (b) of a shape X with a structuring element B. The opening is found by fitting the structuring element B in the object X. The structural closing is found by fitting the structuring element B in the background of X.

The structural opening is illustrated in figure 9.1. If X is the image and B is the structuring element, $X \circ B$ is defined as the union of all translates B_h of B which fit in X. This family of structural openings satisfies the first three conditions, but not the fourth one. This can be seen from a simple example using the 5-7-metric. Let $X = \overline{B}(7)$; as $\overline{B}(7) \circ \overline{B}(5) = \overline{B}(5)$ we get that

$$(X \circ \overline{B}(7)) \circ \overline{B}(5) = \overline{B}(5) \neq X = X \circ \overline{B}(7).$$

However, it can be seen [66] that the following function family is indeed a size distribution:

$$\alpha_r(X) = \bigcup_{s \geq r} X \circ \overline{B}(s).$$

As each α_r is an opening, size distributions are not only useful in the analysis of sizes, but the individual operations can also be used as an alternative for the structural opening with a single sphere.

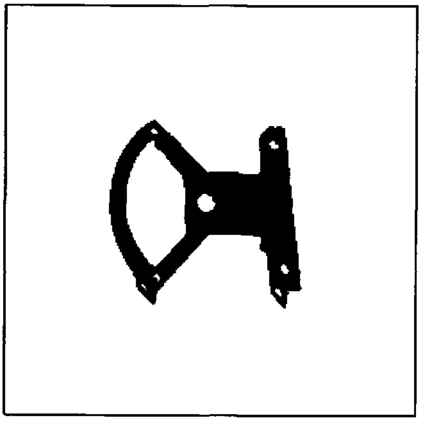


Figure 9.2: An example object for which the opening and the closing are computed.

An algorithm for the calculation of $\alpha_r(X)$ is suggested by the following theorem:

Theorem 9.2 *Let d be a discrete metric on a set E and let D be its range. Let X be a bounded subset of E and $r \in D$. Then*

$$\begin{aligned}\alpha_r(X) &= \bigcup_{\rho_X^{\text{ext}}(x) > r} B(x, \rho_X^{\text{ext}}(x)) \\ &= \bigcup_{\rho_X^{\text{int}}(x) \geq r} \overline{B}(x, \rho_X^{\text{int}}(x))\end{aligned}$$

PROOF. First suppose that $x \in \alpha_r(X)$. Then $x \in X \circ \overline{B}(s)$ for some $s \geq r$. Thus, there is a $y \in X$ such that $x \in \overline{B}(y,s) \subset X$. The second inclusion implies $\rho_X^{\text{ext}}(y) > s \geq r$, so $x \in \overline{B}(y,s) \subset B(y,\rho_X^{\text{ext}}(y)) \subset X$. Therefore, $x \in \bigcup_{\rho_X^{\text{ext}}(y)>r} B(y,\rho_X^{\text{ext}}(y))$.

Now suppose that $x \in \bigcup_{\rho_X^{\text{ext}}(y)>r} B(y,\rho_X^{\text{ext}}(y))$. Then there is a y such that $x \in B(y,\rho_X^{\text{ext}}(y)) \subset X$. Because $\rho_X^{\text{ext}}(y) > r$ it holds that $B(y,\rho_X^{\text{ext}}(y)) = \overline{B}(y,r_0)$ for some $r_0 \geq r$. We now have $x \in \overline{B}(y,r_0) \subset X$ for some $r_0 \geq d$, so $x \in X \circ \overline{B}(r_0) \subset \bigcup_{s \geq r} X \circ \overline{B}(s) = \alpha_r(X)$.

The second part can be proven in the same way. It can also be deduced from the first part by observing that $\rho_X^{\text{int}}(x) \geq r \Leftrightarrow \rho_X^{\text{ext}}(x) > r$ and $\overline{B}(x,\rho_X^{\text{int}}(x)) = B(x,\rho_X^{\text{ext}}(x))$. ∎

From this theorem, an algorithm for computing size distributions can be derived. The algorithm can be used for chamfer metrics or any other metric for which distance transforms and reconstructions can be computed. Let θ_r be the function defined by

$$\theta_r(s) = \begin{cases} 0 & \text{if } s \leq r \\ s & \text{if } s > r \end{cases}$$

Algorithm 9.3 *Computation of the size distribution α_r of a bounded subset X of $\mathbb{Z}^2$ with respect to a chamfer metric.*

(1) Calculate the external distance transform ρ_X^{ext}.

(2) Remove values smaller than or equal to r from the distance transform: let f_X^r be the function $\theta_r \circ \rho_X^{\text{ext}}$.

(3) Calculate the reconstruction of f_X^r.

This algorithm requires five image scans: two for the calculation of the distance transform, one for the removal of small values and two for the reconstruction. In figure 9.3, an example of the calculation of the opening transform, using the 5-7–metric, of the object shown in figure 9.2 is shown. Figure 9.3a shows the distance transformation of the object, from which the pixels with value 30 or less have been removed. The result in figure 9.3b is obtained by reconstruction of the modified distance transformed image.

Definition 9.4 *Let E be some set and let Λ be a totally ordered set. An anti-size distribution [95, 61] is a family $\{\phi_\lambda\}_{\lambda \in \Lambda}$ of operators mapping subsets of E to subsets of E such that for all $X, Y \subset E$, $r, s \in \Lambda$:*

(1) $X \subset Y \Rightarrow \phi_r(X) \subset \phi_r(Y)$.

(2) $X \subset \phi_r(X)$.

(3) $\phi_r \circ \phi_r = \phi_r$.

(4) $\phi_r \circ \phi_s = \phi_{\max(r,s)}$.

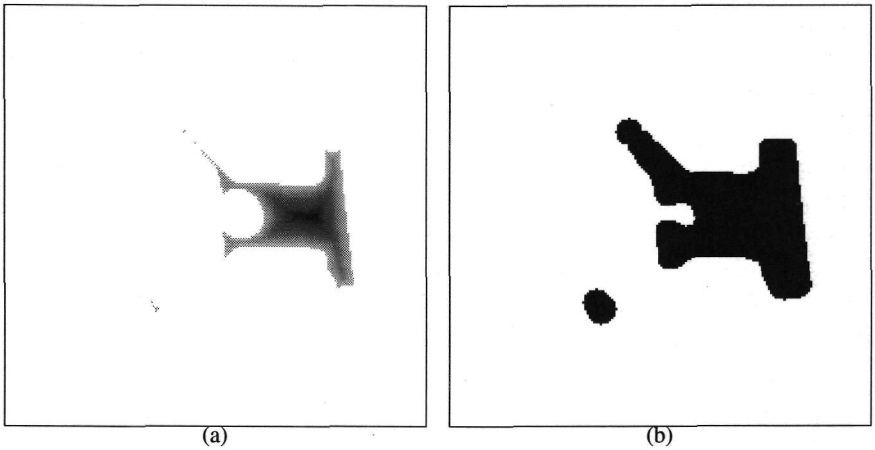


Figure 9.3: The computation of the size distribution with radius 30 using the 5-7-metric.

The first three conditions are exactly the conditions used for defining that each ϕ_λ is a closing [95]. The fourth condition describes the composition behavior of these closings. As with the definition of the size distribution, the third condition follows from the fourth. The index set Λ will be D.

The goal is to construct an anti-size distribution based on spheres in the chamfer metric. Conceptually, size distributions and anti-size distributions are very related, because the anti-size distribution of an object is equivalent to the size distribution of its complement. This observation, however, does not lead to useful algorithms, because the complement of a bounded set is not bounded. Therefore the anti-size distribution is treated here separately.

Structural closings [87] are defined by

$$X \bullet \overline{B}(r) = \bigcap_{X \subset \overline{B}_h^C} \overline{B}_h^C .$$

This operation is illustrated in figure 9.1. The structural closing of X with structuring element B is found by fitting the structuring element in the background. If a point x is contained in a translate B_h of the structuring element which does not intersect X, x is not a point of $X \bullet B$, otherwise it is.

As was the case with the structural opening and the size distribution, the structural closing satisfies the first three conditions, but not the fourth one. Analogously to the case of size distributions, a anti-size distribution can be defined by

$$\phi_r = \bigcap_{s \geq r} X \bullet \overline{B}(s).$$

Again, the anti-size distribution is not only useful for the analysis of sizes, but it also provides an alternative for the structural closing. The operation ϕ_r is the dual of the opening discussed above: $\phi_r(X) = [\alpha_r(X^C)]^C$. This duality suggests a way of calculating $\phi_r(X)$: first calculate the complement of X; calculate the opening of this set with the algorithm presented above; take the complement of the result. The problem is that X^C is not a bounded set. Therefore, this algorithm cannot be performed.

Fortunately, the algorithm can be adapted in such a way that the computation becomes finite. The algorithm for the p-q metric will be discussed. Analogous algorithms exist for other chamfer metrics. Remember that R is the open sphere reconstruction (definition 7.6).

Theorem 9.5 *Let X be a bounded subset of $\mathbb{Z}^2$, provided with the p-q–metric. Let $\rho^{\mathrm{ext}}_{X^C}$ be the distance transform of the background of X. Let V be the set $\{x \in X \mid \rho^{\mathrm{ext}}_{X^C} \leq r + q\}$. Let 1_V be the characteristic function of V. Then $\phi_r(X) = V \setminus R(1_V \cdot \theta_r \circ \rho^{\mathrm{ext}}_{X^C})$.*

PROOF. Write $\psi(X)$ for the set $V \setminus R(1_V \cdot \theta_r \circ \rho^{\mathrm{ext}}_{X^C})$. Let x be a point in $\mathbb{Z}^2$. If $\rho^{\mathrm{ext}}_{X^C}(x) > r + q$ then $x \notin V$, so $x \notin \psi(X)$. It also holds that $\overline{B}(x, r+q) \subset X^C$, so $x \notin X \bullet \overline{B}(r+q)$, so $x \notin \phi_r(X)$.

In the sequel it is assumed that $\rho^{\mathrm{ext}}_{X^C}(x) \leq r + q$. It will be shown that (1) $x \notin \phi_r(X) \Rightarrow x \notin \psi(X)$ and that (2) $x \notin \psi(X) \Rightarrow x \notin \phi_r(X)$.

(1) Suppose $x \notin \phi_r(X)$. Then there is a $y \notin X$ and an $s \geq r$ such that $x \in \overline{B}(y, s) \subset X^C$. From the second inclusion it follows that $r \leq s < \rho^{\mathrm{ext}}_{X^C}(y)$. Two cases must be discerned: (1a) $r < \rho^{\mathrm{ext}}_{X^C}(y) \leq r + q$ and (1b) $\rho^{\mathrm{ext}}_{X^C}(y) > r + q$.

(1a) If $r < \rho^{\mathrm{ext}}_{X^C}(y) \leq r + q$, then $1_V(y) \cdot \theta_r(\rho^{\mathrm{ext}}_{X^C}(y)) \neq 0$. The point y participates in the reconstruction step and $x \in B(y, \rho^{\mathrm{ext}}_{X^C}(y))$ is a point not contained in $\psi(X)$.

(1b) If $\rho^{\mathrm{ext}}_{X^C}(y) > r + q$, let P be a shortest path from x to y. Let z be the point on P which is nearest to y and satisfies $\rho^{\mathrm{ext}}_{X^C}(z) \leq r + q$. Due to the construction of the chamfer metric, this point also satisfies $\rho^{\mathrm{ext}}_{X^C}(z) > r$. Therefore, $1_V(z) \cdot \theta_r \rho^{\mathrm{ext}}_{X^C}(z) \neq 0$, so z participates in the reconstruction. Because z lies on a shortest path from x to y, we have $d(x, z) = d(x, y) - d(z, y) < \rho^{\mathrm{ext}}_{X^C}(y) - d(z, y)$. From the triangle inequality it follows that $\rho^{\mathrm{ext}}_{X^C}(y) - d(z, y) \leq \rho^{\mathrm{ext}}_{X^C}(z)$. Therefore $d(x, z) < \rho^{\mathrm{ext}}_{X^C}(z)$. As z participates in the reconstruction, this implies $x \in R(1_V \cdot \theta_r \rho^{\mathrm{ext}}_{X^C})$. Therefore $x \notin \psi(X)$.

(2) Suppose that $x \notin \psi(X)$. Then there is an $y \in \mathbb{Z}^2$ such that $\rho^{\mathrm{ext}}_{X^C}(y) > r$ and $x \in B(y, \rho^{\mathrm{ext}}_{X^C}(y))$. Therefore $x \notin X \bullet \overline{B}_{\rho^{\mathrm{int}}_{X^C}(y)}$. As $\rho^{\mathrm{int}}_{X^C} \geq r$, this implies $x \notin \phi_r(X)$. ∎

This theorem suggests the following algorithm for the computation of $\phi_r(X)$.

Algorithm 9.6 *The computation of the anti-size distribution ϕ_r of a bounded subset X of $\mathbb{Z}^2$ with respect to the p-q–metric.*

(1) Calculate $\rho^{\text{ext}}_{X^C}$ on a region which is large enough to contain V.
(2) Remove all points from the distance transform having a value smaller than or equal to r.
(3) Remove all points having a distance transform larger than $r + q$ as well.
(4) Calculate the open sphere reconstruction of the remaining function.
(5) Take the intersection of the complement of the reconstructed set with the set of points x with $\rho^{\text{ext}}_{X^C}(x) \leq r + q$.

In the first step, care must be taken at the image boundaries, because the pixels outside the image boundaries must be considered to be part of X^C. The execution of this algorithm requires five scans of a somewhat enlarged region, because the region must be large enough to contain V.

This theorem can also be interpreted in the following way: in order to compute the anti-size distribution, which is defined as an intersection of an infinite number of sets, it is sufficient to perform finitely many—at most q—structural closings and compute their intersection. Likewise, it can be seen that size distributions can be computed as the union of at most q structural openings. This observation, however, does not provide an efficient algorithm, because there is no fast algorithm for performing structural openings with spheres as structuring elements.

Figure 9.4 shows an example of the calculation of the anti-size distribution of radius 30 of the object in figure 9.2 using the 5-7–metric. Figure 9.4a shows the distance transform of the background. In figure 9.4b, only pixels with a radius between 31 and 37 have been retained. Reconstruction of figure 9.4b yields figure 9.4c. Intersecting the complement of figure 9.4c with the set of pixels which have a value less than or equal to 37 in figure 9.4a yields the result in figure 9.4d.

9.3. The Opening Transform and the Pattern Spectrum

In this section, we present the opening transform and the pattern spectrum, and an efficient algorithm for their computation, based on the medial axis.

Size distributions have been presented in the previous section. Let $(\alpha_r)_{r \in D}$ be a size distribution. Suppose $r, s \in D$ with $r \geq s$. As each α_r is an opening, $\alpha_r(X) \subseteq X$ so $\alpha_r(X) = \alpha_s \alpha_r(X) \subseteq \alpha_s(X)$ for all images X. This observation is the inspiration for the definition of the opening transform.

Definition 9.7 *Let X be a bounded subset of $\mathbb{Z}^2$ and let $\{\alpha_r\}$ be a size distribution. The opening transform A_X is the mapping from X to D defined*

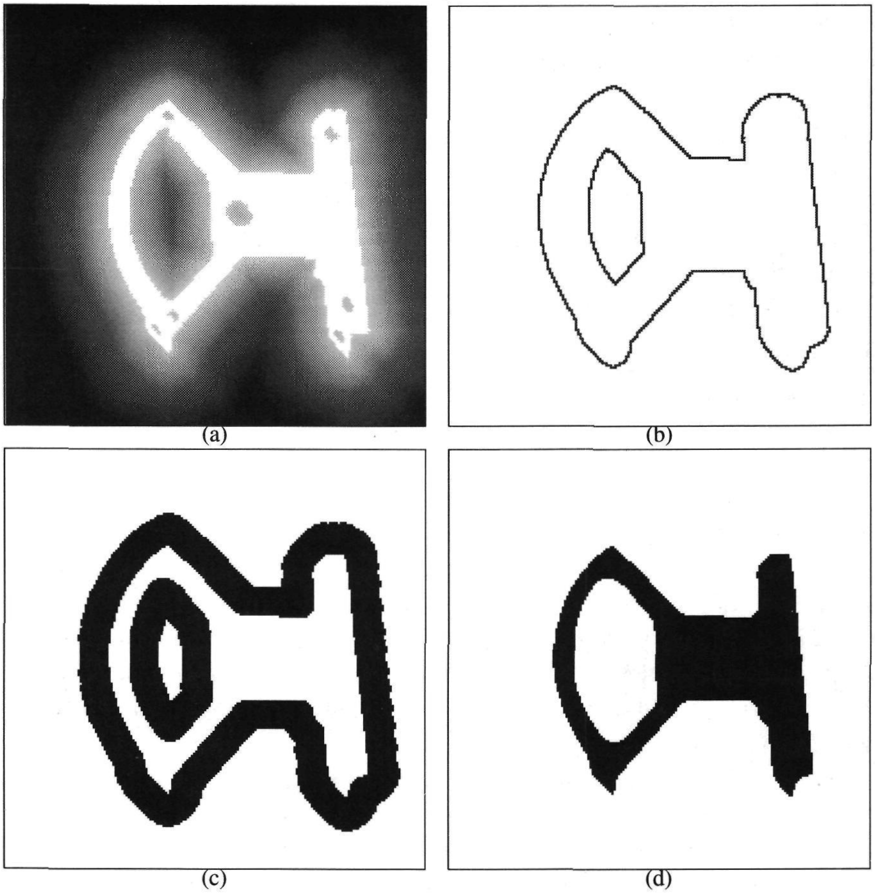


Figure 9.4: The computation of the anti-size distribution with radius 30 using the 5-7–metric.

by

$$A_X(x) = \max\{r \in D \mid x \in \alpha_r(X)\}.$$

Important information on the shape of X can be obtained by monitoring the change of $\alpha_r(X)$ as the parameter r is varied. Maragos [61] defines the pattern spectrum for subsets of $\mathrm{I\!R}^2$, provided with the Euclidean metric, as

$$p_X(r) = \frac{\mathrm{d}A(X \circ B(r))}{\mathrm{d}r}, \tag{9.2}$$

where $A(X \circ B(r))$ is the area of $(X \circ B(r))$.

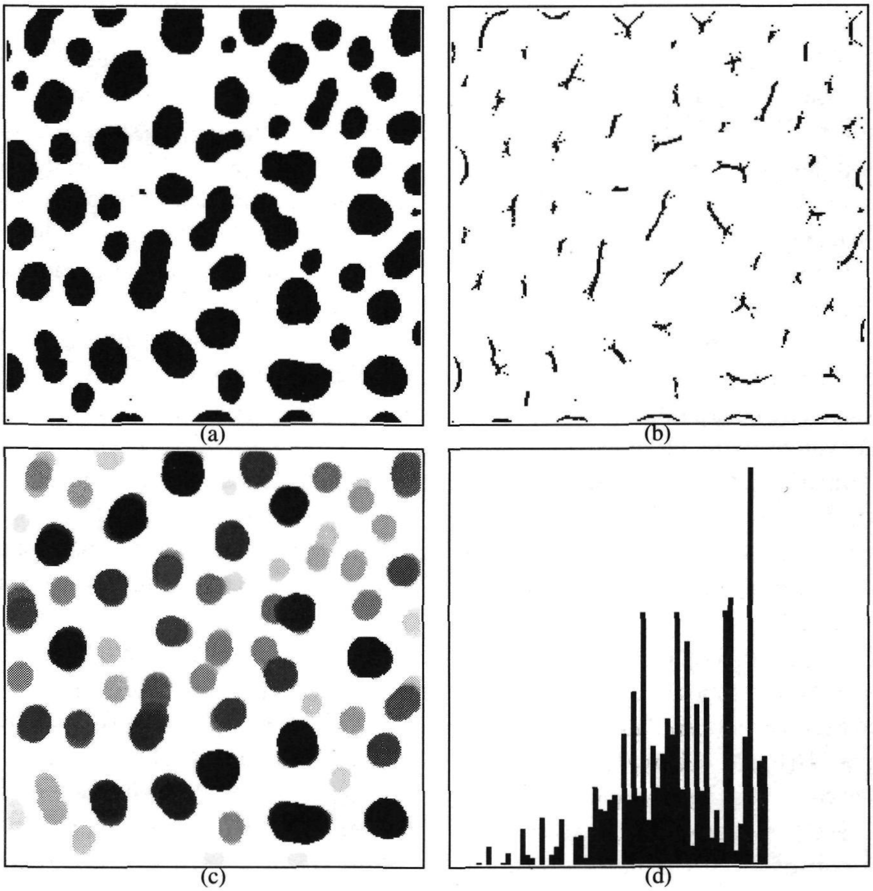
For the discrete case, this produces the following

Definition 9.8 *Let X be a bounded subset of $\mathbb{Z}^2$ and let α_r be the size distribution induced by a chamfer metric d. The the pattern spectrum [61] $p(r)$ is the function from D to $\mathbb{N}$ defined by*

$$p_X(r) = |\alpha_r(X)| - |\alpha_{r^+}(X)|,$$

where $|\cdot|$ denotes the number of points in a set and $r^+ = \min\{r' \in D \mid r' > r\}$ is the smallest element of D which is larger than r.

Note that the pattern spectrum of a set is equal to the histogram of its opening transform. It is therefore possible to compute the pattern spectrum directly from the opening transform.


(a)

(b)

(c)

(d)

Figure 9.5: A binary image, its medial axis, its opening transform and its pattern spectrum as defined by the 5-7-11 metric.

From the definitions of A_X and α_r it follows that

$$A_X(x) = \max\{r \in D \mid x \in \overline{B}(y,r) \subseteq X \text{ for some } y \in \mathbb{Z}^2\}. \tag{9.3}$$

Because for each $x \in X$, the largest disc centered at x and contained in X has radius $\rho(x)$, this can be reduced to

$$A_X(x) = \max\{\rho(y) \mid y \in \mathbb{Z}^2, x \in \overline{B}(y,\rho(y))\}. \tag{9.4}$$

Observe that for each x and y such that $x \in \overline{B}(y,\rho(y))$, there is also a point m in the medial axis M_X such that $x \in \overline{B}(m,\rho(m))$. This leads to the relation

$$A_X(x) = \max\{\rho(m) \mid m \in M_X, x \in \overline{B}(m,\rho(m))\}. \tag{9.5}$$

This relation can be used to compute the opening transform by successively inspecting all maximal spheres in a bounded object X. First, all pixels in the result image are given value zero. The medial axis is computed, and each medial axis point is inspected in turn. If m is the medial axis point being inspected, then all pixels in the sphere $\overline{B}(m,\rho(m))$ in the result image are visited. If the present value if the pixel in the result image is smaller than $\rho(m)$, the value is updated to $\rho(m)$. In the algorithm presented here, the medial axis points are sorted in order of increasing distance transform value and visited in this order. Thus, when the points in $\overline{B}(m,\rho(m))$ are being visited, a pixel in the result image can never have a value larger that $\rho(m)$, and the pixels in $\overline{B}(m,\rho(m))$ can always be assigned value $\rho(m)$, without prior inspection of the present pixel value. This strategy enhances the efficiency of the algorithm, because a comparison of pixel values in the inner loop is replaced by a sorting of the medial axis points, which must be performed just once. As $\rho(m)$ assumes only integer values smaller than some maximal value, the medial axis points can be sorted in linear time in the number of medial axis points using distribution sorting [52].

The efficiency of the algorithm depends on an efficient way of addressing all pixels in a sphere $\overline{B}(m,\rho(m))$. Certainly, computing such spheres in a two scan reconstruction algorithm would be too costly. In stead, the pixels in the sphere $\overline{B}(0,R)$, where R is the largest occurring value of $\rho(m)$, are sorted in order of increasing distance to the origin. When the sorted list is computed, the pixels in each sphere $\overline{B}(0,r)$ for $r \leq R$ can be found by taking a suitable first part of the list. The pixels in spheres with different centers can be found by translation. In practice, the pixels in $\overline{B}(0,R)$ are not sorted, but computed in the correct order, and pixel positions are represented as relative offsets in the image array in order to provide fast access to the image.

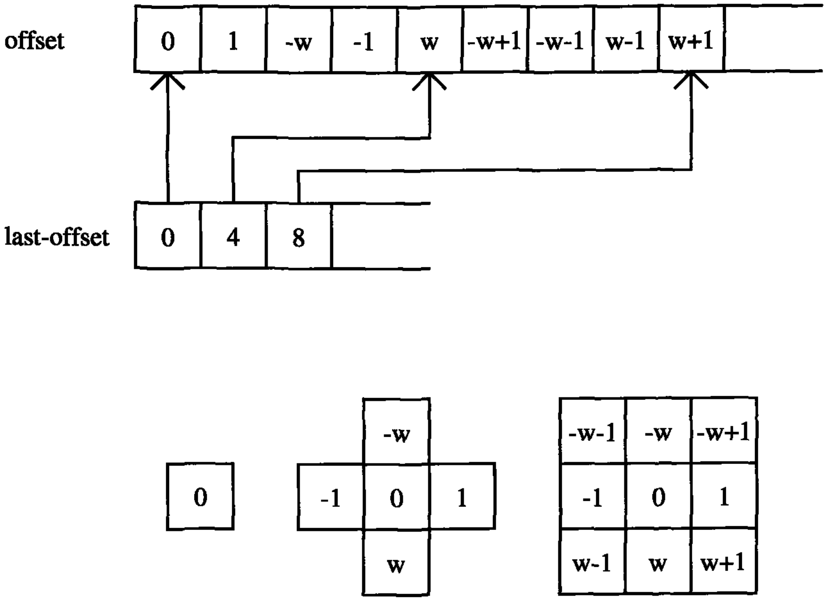


Figure 9.6: The beginning parts of the tables `offset` and `last-offset`. The closed spheres of radii 0, 5 and 7 are shown, and the relative offsets of pixels with respect to the center are shown (`w` is the width of the image grid.)

If images are stored in an array, positions of pixels in a sphere with respect to the center can be represented effectively as an offset in the array, as illustrated in figure 9.6. Two tables are required: a table `offset` with the offsets of pixels at increasing distance to the center and a table `last-offset` which contains the index of the last entry for each radius.

Summarizing, the opening transform can be computed by the following

Algorithm 9.9 *The computation of the opening transform of a bounded subset X of $\mathbb{Z}^2$ according to some chamfer metric d.*

(1) Initialize the result image to 0.

(2) Compute the distance transforms ρ_X^{ext} and ρ and the medial axis M_X.

(3) Sort the medial axis points in order of increasing distance transform value.

(4) For all medial axis points, in increasing order:

- *Set the pixels in $\overline{B}(m, \rho(m))$ to $\rho(m)$.*

As noted before, the size distribution of X is the histogram of A_X, so it can be computed from A_X in a very straightforward way.

Figure 9.5 shows a binary image, its medial axis, its distance transform and its pattern spectrum, all of them based on the 5-7-11–metric. Dark pixels correspond to large distance transform values. The execution time for the opening transform depends on the image, because the number of spheres which must be inserted, varies. For the image presented here, computation of the opening transform took 0.43s, which is two orders of magnitude faster that the 45s required for executing algorithm 9.3 for all occurring values of the radius. A similar reduction of computation time is found for other images.

9.4. Conclusions

This chapter has presented an integration of chamfer metrics and mathematical morphology. Both theoretical aspects and efficient algorithms have been discussed. As chamfer metrics on a square grid can approximate the Euclidean metric within an error of a few percent, morphological operations based on chamfer metrics such as the 5-7-metric or the 5-7-11 metric are a good approximation of morphological operations using Euclidean discs, and are in this respect to be preferred over operations using square or diamond-shaped structuring elements.

A size distribution and an anti-size distribution have been defined using chamfer metrics. The investigation of the relation between (anti-)size distributions, distance transformations and reconstruction has lead to an efficient algorithm for the computation of these operators. The algorithm requires four image scans and is based on local operation only.

Based on size distributions, the opening transform has been defined. An investigation of the relation between the opening transform, the medial axis, distance transform and reconstructions has lead to an algorithm for the computation of the opening transform based on chamfer discs. This algorithm is 50 to 100 times faster than the brute force method of applying algorithm 9.3 repeatedly for each value of the size parameter. The pattern spectrum of an object can be derived from the opening transform by histogramming.

Summarizing, the integration of chamfer metrics into mathematical morphology presented in this paper has lead to a number of morphological operators which approximate morphological operations based on Euclidean discs accurately and which can be calculated efficiently.

10

Concluding Remarks

In this thesis, image analysis methods based on hierarchical graph representations and on multi-scale mathematical morphology were investigated. In this chapter, some conclusions are drawn and some possibilities for further research are mentioned.

A major part of this thesis is concerned with the exploration of the formalism of *hierarchies of graphs* in image analysis. Hierarchies of graphs can be seen as an irregular variant of conventional pyramids or quad-trees [100] but have a richer structure. Recently, a particular type of hierarchies, called the *irregular pyramid*, has been used for image segmentation [76].

Yet there are other types of hierarchies of graphs than the irregular pyramid and there are possibilities of such structures which have not yet been explored. In this thesis, we presented some of the possibilities of hierarchies of graphs.

The building blocks for image descriptions were *primitives* (condensed representations of the spatial structure in some localized part of the image). These primitives occur as vertices of graphs in the hierarchy. This general approach enabled us, for example, to treat image segmentation an polygonal approximation of curves in a uniform manner (chapter 4).

Some important advantages of the hierarchy of graphs formalism are

- It allows for the representation of both primitives and their interrelations.
- Many different primitive types can be used.
- It allows irregular data structures and can therefore avoid artifacts caused by a superimposed regular sampling grid.
- It enables the description of topological aspects of the image representation, such as connectivity and adjacency of regions.
- It provides a suitable data structure for model based image analysis by top-down search.

A disadvantage of irregular structures is the fact that their representation in a computer can be rather memory intensive and that implementation on parallel hardware is much more complicated than for regular structures (e.g. [111]).

Properties of the primitives are represented by their *attributes* and operations on primitives only address their attributes and their location in the graph structure.

Image description by primitives and attributes is common in structural pattern recognition [30], but there collections of primitives are ordered in treelike structures whose leaves are primitives. The difference with our approach is that we aggregate groups of primitives to create new primitives of the same type, yielding a coarser or more abstract representation of the same image data. This approach enabled us to use a uniform treatment of all levels in the hierarchy.

Relations between primitives can be defined from their attributes and numerical measures for such relations can be constructed. Various functions on attributes were presented, which were used to express relations between primitives in such a way that the values of these functions could be used to guide bottom-up processing. The metric for line segments, presented in chapter 2, is such a cost function. The advantage of this metric over conventional measures for collinearity [9, 90] is the fact that it was derived from an image formation model in a systematic way. In order to evaluate its properties, the metric was used in a hierarchical clustering method. The results for both natural and synthetic images showed the robustness of the method. A possible direction for future research is the extension of the derivation of metrics from image formation models to other primitive types, e.g. for the measurement of co-circularity of curve fragments.

In chapter 3, it was shown that there is a class of cost functions, which we called increasing path distances, for which clusterings in a graph exist which are optimal with respect to this measure and for which all clusters are connected. The definition of such a cost function is based both on attributes of primitives and on the graph structure defined on these primitives. An algorithm for the computation of such clusterings was presented.

This grouping method can also be used for other types of cost functions, but then a non-optimal clustering is achieved. When the cost function, corresponding with multilevel thresholding, is used, the resulting segmentation can be seen as a compromise: the regions are connected, but each pixel is assigned to a region to which it is as similar in grey value as possible.

When primitives are manipulated only through their attributes and their location in the graph structure, algorithms can be formulated with great generality. This was done, for example, in chapter 4, where the same algorithm was used for both image segmentation and polygonal approximation of curves. The only difference in these applications was the choice of primitives and the evaluation function for the fit of pairs of primitives to a model. In this sense, our method is a general clustering method, but unlike most conventional clustering algorithms [48], it takes the spatial arrangement of the primitives into account through a graph structure. For image analysis applications, this is clearly an advantage, because of the special role of spatial structure in such applications.

In the application of our method to polygonal approximation of curves, various measures of fit were tested, some of them known from the literature. It was shown that the measure proposed by Wall and Danielsson [108] and the maximal distance measure produce the best results.

In the application of our method to image segmentation, we used the constant grey level model, which is the simplest possible choice. We also considered the linear grey level model for regions. It was explained why the segmentation of roof edges with the linear model is problematic for our method and, in fact, for any bottom-up clustering method.

A hierarchy of region adjacency graphs enables the representation of connectivity of receptive fields. This property was used in the segmentation methods presented in this thesis. In chapters 3 and 4, hierarchical graph representations were constructed in such a way that connectivity is guaranteed.

The problem of adapting an existing structure in such a way that connectivity is not lost, is more difficult. Conventional relinking segmentation methods (which are all variations on the method by Burt *et al.* [18]) do not preserve connectivity. In chapter 5, we presented a new relinking method which produces connected regions. The algorithm can be implemented on parallel hardware.

Our relinking method requires the use of a graph structure. This enables the application of both properties of regions and properties of boundaries between them, in the segmentation process. We presented a method in which both types of information can be combined. The relative importance of both information types depends on a freely adjustable parameter.

Pyramids have been used for the top-down detection of particular image features such as boundaries and for top-down template matching (see Dyer [26] for a review). On the other hand, search methods for curves based on cost minimization on a pixel grid, have been developed [31, 51, 81, 98]. We combined these approaches in a top-down curve detection procedure which performs cost minimization on successive levels of a hierarchy of graphs. This was presented in chapter 6.

We considered a particular application: the detection of fiber boundaries in a hierarchical description of a muscle tissue image. A model for muscle fibers was built and criteria for the top-down search in a hierarchical structure were derived from this model. Detection of fiber boundaries was performed by minimization of the corresponding cost function. Conventional cost minimization methods for curve localization [31, 51, 81, 98] use the pixel grid as the underlying spatial structure. We introduced the generalization of such methods to region adjacency graphs and to hierarchical optimization.

Since the cost functions used in literature [81, 73] tend to favor short curves, resulting in parts of the fibers being ‘cut off’, we had to introduce a new cost function. This cost function consists of two terms, one measuring contrast along the curve and one depending on the shape of the curve. There is one free parameter, determining the relative weight of these terms. It must be chosen such that the two terms have the same order of magnitude. The optimal value depends on image contents. The resulting method is robust and reacts in a predictable way to non-optimal parameter settings.

The results of our methods are promising and indicate that hierarchies of graphs are suitable structures for top-down search methods. A topic for future research may be the extension of other methods, such as snakes [51] or parametrically deformable models [98], to hierarchies of graphs. It is not a priori clear, which part of the analysis must be performed in the construction phase of the hierarchy, and which part in the top-down phase. This problem will also require further research.

A second important theme in this thesis was the investigation of mathematical morphology [95, 66] using chamfer discs [12] as structuring elements. The advantage of such structuring elements is the fact that they approximate the Euclidean disc more closely than the conventional square structuring elements [105]. Our investigations concerned both theoretical work on the properties of chamfer metrics and the characterization of the medial axis defined by such metrics, and the development of a number of efficient algorithms for performing operations using these structuring elements.

In chapter 7, we presented results and rigorous proofs on some properties of chamfer metrics, such as the structure of shortest paths between points. We pointed out the important difference between extending and non-extending metrics, and between the internal and external distance transform.

In chapter 8, we gave a complete characterization of the medial axes defined by p-q–metrics and p-q-r–metrics, based on a rigorous mathematical framework. Thus far, only partial results were known for the p-q-r–metric [2, 14]: for each p-q-r–metric, a number of special cases had to be worked out 'by hand'. We have presented a general description for these special cases, valid for all p-q-r–metrics.

Based on our characterization, efficient algorithms for the medial axis were derived. The computational cost of these algorithms is of the same order of magnitude as the cost for more conventional medial axis algorithms.

In chapter 9, the opening transform and the pattern spectrum [61], which are important for the measurement of the size of image structures, were investigated. Based on their relation with the medial axis, efficient algorithms for their computation were derived.

Verwer [104] has presented a thinning algorithm based on chamfer metrics and noted that the availability of such an algorithm would reduce the importance of methods based on city-block and chess-board metrics, which correspond to square structuring elements. This development will be enforced by the availability of efficient algorithms for all relevant operations in multi-scale mathematical morphology (dilations, erosions, openings, closings, size distributions, pattern spectrum) based on p-q-r–chamfer discs.

Some topics for future research are the extension of our methods to chamfer metrics with more than three types of prime vectors [101] and the construction of a connected skeleton from the medial axis. An other direction of research is the extension to grey scale morphology, where efficient algorithms for morphology with (approximations of) parabolic structuring elements are gaining interest [10, 47].

References

[1] H.J. Antonisse. Image segmentation in pyramids. *Computer Graphics and Image Processing*, 19:367–383, 1982.

[2] C. Arcelli and M. Frucci. Reversible skeletonisation by (5,7,11)-erosion. In C. Arcelli, L.P. Cordella, and G. Saniti di Baja, editors, *Visual Form-Analysis and Recognition*, pages 21–28. Plenum Press, New York, 1992.

[3] C. Arcelli and G. Sanniti di Baja. A width-independent fast thinning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):463–474, 1985.

[4] C. Arcelli and G. Sanniti di Baja. Finding local maxima in a pseudo-euclidean distance transform. *Computer Vision, Graphics and Image Processing*, 43:361–367, 1988.

[5] C. Arcelli and G. Sanniti di Baja. Ridge points in Euclidean distance maps. *Pattern Recognition Letters*, 13:237–243, 1992.

[6] S. Baronti, A. Casini, F. Lotti, L. Favaro, and V. Roberto. Variable pyramid structures for image segmentation. *Computer Vision, Graphics and Image Processing*, 49:346–356, 1990.

[7] P. Bertolino and A. Montanvert. Edge detection for biomedical images: A self-adaptive and randomized operator. In *14th Annual International Conference of IEEE Engineering in Medicine and Biology Society, Paris, France*, pages 1904–1905, 1992.

[8] M. Bister, J. Cornelis, and A. Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11:605–617, 1990.

[9] M. Boldt, R. Weiss, and E. Riseman. Token-based extraction of straight lines. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1581–1594, 1981.

[10] R. van den Boomgaard. *Mathematical Morphology: Extensions Towards Computer Vision*. PhD thesis, University of Amsterdam, 1992.

[11] R. van den Boomgaard and R. van Balen. Methods for fast morphological image transformations using bitmapped binary images. *CVGIP Graphical Models and Image Processing*, 54(3):252–258, 1992.

[12] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345, 1984.

[13] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, 1988.

[14] G. Borgefors. Centres of maximal discs in the 5-7-11 distance transform. In *Proceedings of the 8th Scandinavian Conference on Image Analysis, Tromsø*, pages 105–111, 1993.

[15] C. R. Brice and C. L. Fennema. Scene analysis using regions. *Artifical Intelligence*, 1:205–226, 1970.

[16] J.B. Burns, A.R. Hanson, and E.W. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):425–455, 1086.

[17] P. Burt and T. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[18] P. J. Burt, T. Hong, and A. Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transactions on Systems, Man and Cybernetics*, 11(12):802–809, 1981.

[19] L. Le Cam. *Notes on Asymptitic Methods in Statistical Decision Theory*. Centre de Recherces Mathématiques, Université de Montreal, 1974.

[20] J. Cibulskis and C.R. Dyer. Node linking strategies in pyramids for image segmentation. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, chapter 8, pages 109–120. Springer Verlag, Berlin etc., 1984.

[21] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Mass., 1991.

[22] M. Coster and J.-L. Chermant. *Précis d'Analyse d'Images*. Centre National de la Recherche Scientifique, Paris, 1985.

[23] J. Dieudonné. *Foundations of Modern Analysis*. Academic Press, New York, 1960.

[24] L. Dorst. Pseudo-euclidean skeletons. In *Proceedings of the Eighth International Conference on Pattern Recognition, Paris, France*, pages 286–288, 1989.

[25] R.O. Duda and P.E. Hart. Use of the Hough transform to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[26] C. Dyer. Multiscale image understanding. In L. Uhr, editor, *Parallel Computer Vision*, pages 171–213. Academic Press, Boston, 1987.

[27] T.J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-D objects using surface descriptions. In *Second International Conference on Computer Vision, Tampa, Florida*, pages 474–481, 1988.

[28] C. Fermüller and W. Kropatsch. Hierarchical curve representation. In *11th International Conference on Pattern Recognition, The Hague, The Netherlands*, pages 143–146, 1992.

[29] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. Elec. Computers*, 10:260–268, 1961.

[30] K.S. Fu. *Syntactic Methods in Pattern Recognition.* Academic Press, New York, London, 1974.

[31] J.J. Gerbrands. *Segmentation of Noisy Images.* PhD thesis, Technical University of Delft, 1988.

[32] L.D. Griffin, A.C.F. Colchester, and G.P. Robinson. Scale and segmentation of grey-level images using maximum gradient paths. *Image and Vision Computing*, 10(6):389–402, 1992.

[33] F.C.A. Groen and N.J. Foster. A fast algorithm for cellular logic operations on sequential machines. *Pattern Recognition Letters*, 2:333–338, 1984.

[34] R. Haralick and L. Shapiro. Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.

[35] R. Haralick and L. Shapiro. Glossary of computer vision terms. *Pattern recognition*, 24(1):69–93, 1990.

[36] F. Harary. *Graph Theory.* Addison-Wesley Publishing Company, 1969.

[37] R. Hartley. *Multi-Scale Models in Image Analysis.* PhD thesis, University of Maryland, 1984.

[38] R. Hartley. Segmentation of optical flow fields by pyramid relinking. Technical Report TR86, University of Maryland, Institute of Automation Research, 1984.

[39] H.J.A.M. Heijmans. Mathematical morphology as a tool for shape description. In O Ying-Lie, A. Toet, H.J.A.M. Heijmans, D.H. Foster, and

P. Meer, editors, *Shape in Picture - Mathematical Descriptions of Shape in Greylevel Images*, pages 147–176. Springer Verlag, Heidelberg, 1994.

[40] H.J.A.M. Heijmans. *Morphological Image Operators.* Academic Press, Boston, Mass., 1994.

[41] H.J.A.M. Heijmans and C. Ronse. The algebraic basis of mathematcial morphology part 1: Dilations and erosions. *Computer Vision, Graphics and Image Processing*, 50:245–295, 1990.

[42] H.J.A.M. Heijmans and A. Toet. Morphological sampling. *CVGIP Image Understanding*, 5(3):384–400, 1991.

[43] C.J. Hilditch. Linear skeletons from square cupboards. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence 4*. University Press, Edinburgh, 1969.

[44] T.H. Hong and A. Rosenfeld. Compact region extraction using weighted pixel linking in a pyramid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):222–229, 1984.

[45] S.L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the ACM*, 23:368–388, 1976.

[46] R. Hummel and R. Moniot. Reconstructions from zero crossings in scale space. *IEEE Transactions on Accoustics, Speach and Signal Processing*, 37(12):2111–2130, 1989.

[47] P. Jackaway. Properties of multiscale morphological smoothing by poweroids. *Pattern Recognition Letters*, 15:135–140, 1994.

[48] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs NJ, U.S.A., 1988.

[49] J. M. Jolion and A. Montanvert. The adaptive pyramid: A framework for 2D image analysis. *CVGIP Image Understanding*, 55(3):339–348, 1992.

[50] S. Kasif and A. Rosenfeld. Pyramid linking is a special case of ISODATA. *IEEE Transactions on Systems, Man and Cybernetics*, 13(1):84–85, 1983.

[51] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–269, London, 1987.

[52] D. Knuth. *The Art of Computer Programming Volume 3: Sorting and Searching*. Addison-Wesley, Reading Mass., U.S.A., 1973.

[53] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[54] V.A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics and Image Processing*, 46:141–161, 1989.

[55] W. Kropatsch. Curve representation in multiple resolutions. *Pattern Recognition Letters*, 6:179–184, 1987.

[56] W. Kropatsch. Image pyramids and curves, an overview. Technical Report PRIP-TR-2, Institute for Automation, Technical University of Vienna, 1991.

[57] W. Kropatsch and A. Montanvert. Regular versus irregular pyramid structures. In U. Eckhardt, A. Hübler, W. Nagel, and G. Werner, editors, *Geometrical Problems of Image Processing, Georgenthal, Germany*, pages 11–22, 1991.

[58] W.G. Kropatsch, C. Reither, D. Willersinn, and G. Wlasitz. The dual irregular pyramid. In *5th International Conference CAIP, Budapest, Hungary*, pages 31–40, 1993.

[59] L. Lifschitz and S. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):529–540, 1990.

[60] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, 1990.

[61] P.A. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):701–716, 1989.

[62] P.A. Maragos and R.W. Shafer. Morphological skeleton representations and coding of binary images. *IEEE Transactions on Accoustics, Speach and Signal Processing*, 34(5):1228–1244, 1986.

[63] D. Marr. *Vision*. Freeman, San Francisco, 1982.

[64] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*, 1:169–182, 1972.

[65] A. Martelli. An application of heuristic search methods to edge and contour detection. *Communications of the ACM*, 19:73–83, 1976.

[66] G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975.

[67] G. Matheron. Examples of topological properties of skeletons. In J. Serra, editor, *Image Analysis and Mathematical Morphology Volume 2: Theoretical Advances*. Academic Press, London, 1988.

[68] P. Meer. Stochastic image pyramids. *Computer Vision, Graphics and Image Processing*, 45:269–294, 1989.

[69] P. Meer, D. Mintz, and A. Rosenfeld. Least median of squares based robust analysis of image structure. In *Proceedings of DARPA Image Understanding Workshop, Pittsburgh, PA*, pages 231–254, September 1990.

[70] P. Meer, A. Sher, and A. Rosenfeld. The chain pyramid: Hierarchical contour processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(4):363–376, 1990.

[71] F. Meyer. Skeletons and perceptual graphs. *Signal Processing*, 16:335–363, 1989.

[72] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1:21–46, 1990.

[73] U. Montanari. On the optimal detection of curves in noisy pictures. *Communications of the ACM*, 14:335–345, 1971.

[74] A. Montanvert. Medial line: Graph representation and shape description. In *Proceedings of the Eighth International Conference on Pattern Recognition, Paris, France*, pages 430–432, 1986.

[75] A. Montanvert and P. Bertolino. Irregular pyramids for parallel image segmentation. In *Proceedings of the 16tm ÖAGM Meeting, Vienna, Austria*, pages 13–35, 1992.

[76] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tesselations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):307–316, 1991.

[77] P.F.M. Nacken. A metric for line segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:1312–1318, 1993.

[78] P.F.M. Nacken and A. Toet. Model-based bottom-up grouping of geometric image primitives. In O Ying-Lie, A. Toet, H.J.A.M. Heijmans, D.H. Foster, and P. Meer, editors, *Shape in Picture - Mathematical Descriptions of Shape in Greylevel Images*, pages 549–558. Springer Verlag, Heidelberg, 1994.

[79] R. Nevatia and K.R. Babu. Linear feature extraction and description. *Computer Graphics and Image Processing*, 13:257–269, 1980.

[80] C.W. Niblack, D.W. Capson, and P.B. Gibbobs. Generating skeletons and centerlines from the medial axis transform. In *Proceedings of the Tenth International Conference on Pattern Recognition*, 1990.

[81] C. Orange. *Supervised Boundary Formation.* PhD thesis, Technical University of Delft, 1994.

[82] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern recognition*, 26(9):1277–1294, 1993.

[83] T. Pavlidis. Analysis of set patterns. *Pattern recognition*, 1:165–178, 1968.

[84] M. Pietikäinen and A. Rosenfeld. Image segmentation by texture using pyramid node linking. *IEEE Transactions on Systems, Man and Cybernetics*, 11:822–825, 1981.

[85] I. Pitas and A.N. Venetsanopoulos. Morphological shape decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:38–45, 1990.

[86] J. Princen, J. Illingworth, and J. Kittler. A hierarchical approach to line extraction based on the Hough transform. *Computer Vision, Graphics and Image Processing*, 52:57–77, 1990.

[87] C. Ronse and H.J.A.M. Heijmans. The algebraic basis of mathematical morphology part 2: Openings and closings. *CVGIP Image Understanding*, 54(1):74–97, 1991.

[88] A. Rosenfeld. Some useful properties of pyramids. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pages 2–5. Springer Verlag, Berlin, Heidelberg, 1984.

[89] A. Rosenfeld. Arc colorings, partial path groups, and parallel graph contractions. Technical Report TR132, University of Maryland. Center for Automation Research, 1985.

[90] E. Saund. Symbolic construction of a 2-D scale space image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8), 1980.

[91] L. Scarlatos and T. Pavlidis. Hierarchical triangulation using cartographic coherence. *CVGIP Graphical Models and Image Processing*, 54(2):147–161, 1992.

[92] A. Scher, M. Schneier, and A. Rosenfeld. Clustering of collinear line segments. *Pattern recognition*, 15(2):85–91, 1982.

[93] M. Schmitt and L. Vincent. *Morphology: Algorithms and Applications.* Cambridge University Press, to appear.

[94] R. Sedgewick. *Algorithms, second edition.* Addison-Wesly Publishing Company, Inc., Reading, Mass., etc., 1988.

[95] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, London, 1982.

[96] M. Spann. Figure/ground separation using stochastic pyramid relinking. *Pattern recognition,* 24(10):993–1002, 1991.

[97] M. Spann and C. Horne. Image segmentation using a dynamic thresholding pyramid. *Pattern recognition,* 22(6):719–732, 1989.

[98] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 14:1061–1075, 1992.

[99] B.M. Stewart. *Theory of Numbers.* The Macmillan Company, New York, 1964.

[100] S.L. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing,* 4:104–119, 1975.

[101] E. Thiel and A. Montanvert. Chamfer masks: Discrete distance functions, geometrical properties and optimization. In *Proceedings og the 11th. International Conference on Pattern Recognition, The Hague,* pages 244–247, 1992.

[102] A. Toet. A morphological pyramidal image decomposition. *Pattern Recognition Letters,* 9:255–261, 1989.

[103] A. Toet, J.J. Koenderink, P. Zuidema, and C.N. de Graaf. Image analysis - topological methods. In *Information Processing in Medical Imaging, Proceedings of the 8th Conference, Brussels,* pages 306–342, 1983.

[104] B. Verwer. *Distance Transforms, Metrics, Algorithms and Applications.* PhD thesis, Technical University Delft, 1991.

[105] B. Verwer. Local distances for distance transformations in two and three dimensions. *Pattern Recognition Letters,* 12(11):671–682, 1991.

[106] L. Vincent. *Algorithmes Morphologiques a Base de Files d'Attente et de Lacets.* PhD thesis, Ecole Nationale Superieure des Mines de Paris, Fontainebleau, 1990.

[107] G.S. van der Wal. The Sarnoff pyramid chip. In *Proceedings of Computer Architecture for Machine Perception, Paris, France*, 1991.

[108] K. Wall and P.-E. Danielsson. A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics and Image Processing*, 28:220–227, 1984.

[109] S.W. Wharton. PIPS: A procedure for interactive pyramid segmentation. In *Proceedings of IGARS '88 Symposium*, pages 197–200, Edinburgh, Scotland, 13–16 September 1988. ESA Publications Division.

[110] D. Willersinn and W. Kropatsch. personal communication.

[111] S.G. Ziavras, P. Meer, and D.P. Shah. Adaptive multiresolution structures for image processing on parallel computers. *Journal of Parallel and Distributed Computing*, to appear.

[112] S.W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5:382–399, 1976.

Most of the results presented in these thesis have already been presented, or will be presented, in the following publications:

- P.F.M. Nacken. Hierarchical image structure description based on intensity extrema. In *Geometrical Problems in Image Processing*, pages 191–198, Akademie Verlag, Berlin, 1991.
- P.F.M. Nacken. Efficient algorithms for mathematical morphology based on chamfer distances. In *Mathematical Morphology and its Applications to Signal Processing*, pages 95–100, Barcelona, Spain, May 1993.
- P.F.M. Nacken. A metric for line segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1312–1318, 1993.
- P.F.M. Nacken and A. Toet. Model-based bottom-up grouping of geometric image primitives. In *Shape in Picture - Mathematical Descriptions of Shape in Grey Level Images*, pages 549–558, Springer Verlag, Heidelberg, 1994.
- P.F.M. Nacken. Chamfer metrics in mathematical morphology. *Journal of Mathematical Imaging and Vision*, accepted for publication.
- P.F.M. Nacken. Weighted metrics, the medial axis and mathematical morphology. *Journal of Mathematical Imaging and Vision*, accepted for publication.
- P.F.M. Nacken. Image segmentation by connectivity preserving relinking in hierarchical graph structures. in preparation.
- P.F.M. Nacken Top-down image analysis by cost minimization in hierarchical graph structures. in preparation.

Other publications in which Peter Nacken is author or co-author are:

- H.J.A.M. Heijmans, P.F.M. Nacken, A. Toet, and L. Vincent. Graph morphology. *Journal of Visual Communication and Image Representation*, 3(1):24–38, 1992.
- P.F.M. Nacken. Image compression using wavelets. In *Wavelets: An Elementary Treatment of Theory and Applications*, pages 81–91. World Scientific Publishing Co., Inc., Singapore, 1993.
- P.F.M. Nacken. Openings can introduce zero crossings in boundary curvature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, accepted for publication.
- E. Groen, P.F.M. Nacken, J. Bos, and B. de Graaf. Automatic determination of rotatory eye movement (ADREM). in preparation.
- P.F.M. Nacken and R. van den Boomgaard. A causality property for morphological scale space. in preparation.

Samenvatting

Het doel van beeldanalyse is het expliciet maken van de inhoud van een plaatje. De representatie van een beeld als een lijst van grijswaarden bevat alle informatie over de inhoud, maar niet in een bruikbare, toegankelijke vorm.

Nuttige informatie in een beeld heeft vaak te maken met de ruimtelijke verdeling van grijswaarden over het beeldvlak. De afmeting van een interessante structuur is echter vaak niet vooraf bekend. Daarom is het aantrekkelijk, een beeld tegelijkertijd op meerdere schalen te bekijken in een hierarchische of multi-schaal analyse.

In dit proefschrift worden twee groepen methodes voor beeldanalyse besproken. In het eerste deel wordt een aantal methodes besproken die zijn gebaseerd op het formalisme van hierarchieën van graphen. Zo een hierarchie is een stapel van graphen, die ieder een representatie van de beeldinhoud bevatten. De knopen van de graph representeren structuren in het beeld (gebieden, lijnsegmenten) terwijl de kanten de relaties tussen die structuren (aangrenzendheid van gebieden, collineariteit van lijnsegmenten) representeren. Graphen in hogere niveaus van de hierarchie beschrijven de beeldinhoud op een grover, meer abstraherend niveau.

Verschillende beeldanalysetechnieken, gebaseerd op hierarchieën van graphen, zijn ontwikkeld. Deze methodes zijn toegepast voor beeldsegmentatie, detectie van lineaire structuren en de detectie van randen van objecten.

Er is geconcludeerd dat die hierarchie van graphen een aantrekkeleijk raamwerk voor beeldanalyse is, dankzij zijn flexibiliteit in het hanteren van verschillende structuren en doordat zowel structuren als hun relaties in één representatie kunnen worden weergegeven. Beperkingen van de methodes liggen in hun 'bottom-up' karakter, maar kunnen worden verholpen door 'top-down' analyse in een latere fase van het proces.

Het tweede deel van dit proefschrift behandelt multi-schaal mathematische morfologie. Mathematische morfologie is een techniek voor beeldanalyse, die een binair beeld behandeld door er een klein testobject in de vorm van een vierkant of cirkelschijf overheen te schuiven en de interactie met het beeld (intersectie, inclusie) te beschouwen.

Multi-schaal mathematische morfologie beschouwt families van operators, die zijn gebaseerd op structurerende elementen van verschillende afmetingen. Een gebruikelijke keus hiervoor is een serie vierkanten, omdat de corresponderende operaties voor een vierkant rooster gemakkelijk kunnen worden geïmplementeerd. De invariantie van deze operaties onder rotatie is echter slecht. Het is aantrekkelijker, cirkelschijven te gebruiken, maar de corresponderende operaties zijn moeilijk te implementeren.

Gebaseerd op nieuw verworven inzichten in de eigenschappen van zogenaamde gewogen metrieken zijn methodes ontwikkeld voor multi-schaal mathematische morfologie met schijven, gedefinieerd door deze metrieken, als structurerend element. Gewogen metrieken kunnen de Euclidische metriek tot op een paar procent nauwkeurig benaderen. Dit proefschrift beschrijft algoritmen voor de elementaire morfologische operaties (dilatie, erosie, opening, sluiting), alsmede enkele meer geavanceerde operaties, namelijk de mediale as, de opening-transformatie en het 'pattern spectrum'. De kosten van deze algoritmen zijn van de zelfde orde van grootte als die van de conventienele methodes, gebaseerd op vierkante structurerende elementen.