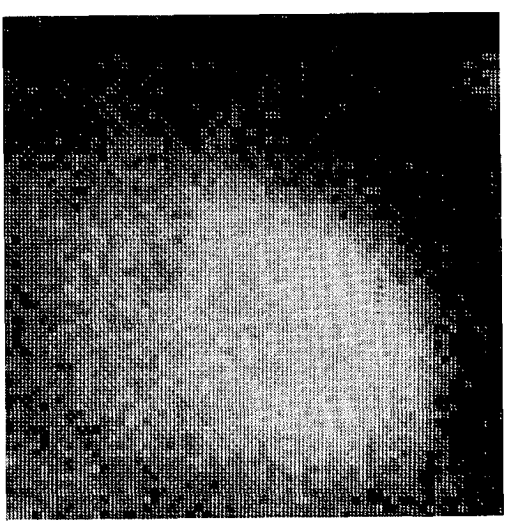
SEGMENTATION OF NOISY IMAGES



j.j. gerbrands

SEGMENTATION OF NOISY IMAGES

SEGMENTATION OF NOISY IMAGES

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus,
prof.dr. J.M. Dirken,
in het openbaar te verdedigen
ten overstaan van een commissie
door het College van Dekanen daartoe aangewezen,
op dinsdag 21 juni 1988
te 14.00 uur

door

JAN JACOB GERBRANDS

geboren te 's-Gravenhage
elektrotechnisch ingenieur

Dit proefschrift is goedgekeurd
door de promotor
PROF. IR. IJ. BOXMA

SEGMENTATIE VAN VERRUISTE BEELDEN

CONTENTS

SUMMARY

The multidisciplinary field of digital image processing is concerned with the methodology and the technology of manipulating images by means of digital computers, special purpose processors or special purpose hardware. One of the possible goals of processing an image digitally is to analyze the image contents, in order to extract information about the phenomena which are being represented by the image. Image analysis can thus be described as an image-to-data transformation, the output data being, e.g., a set of measurement values, a set of labeled objects, or even a description of the imaged phenomena.

One of the crucial steps in the analysis process is the segmentation of the image, i.e., the partitioning of the image plane into regions which are homogeneous according to some predefined criteria. The result of the segmentation stage is thus a map of the various regions, which is intended to be meaningful with respect to the imaged phenomena. The meaningfulness of the segmentation result can usually only be established by the successive analysis steps, because this requires a much higher level of abstraction.

In this thesis we discuss the segmentation of images which are severely contaminated by noise. We restrict ourself to two-dimensional monochrome still images containing distinct object regions. Although the segmentation methods discussed here are applicable to a much wider class of images, the latter restriction indicates that our segmentation problem can be described as the problem of delineating the object regions in very noisy images. The simple traditional segmentation methods show a rather poor performance in the presence of noise, mainly because the noise destroys the coherence of the image structures of interest. Obviously, the influence of the noise can be decreased, at least to a certain extent, by applying specific pre- or post-processing techniques. In the present study, we focus on attempts to improve the performance of the segmentation stage itself. Obviously, the methods discussed here can still be combined with pre- or post-processing techniques to improve the performance even further.

Because the noise destroys the coherence of the image structures, the segmentation decisions cannot be made on a pixel-by-pixel basis; the spatial context has to be taken into account. An attractive way to accomplish this is by means of sequential methods. Because sequential region-oriented methods tend to show a rather poor geometric accuracy of the resulting object region boundaries, we focus on sequential edge-oriented approaches. In particular, we discuss a dynamic programming optimal boundary detector algorithm, which facilitates the incorporation of a priori knowledge. In addition, we consider continuous relaxation labeling procedures, which are parallel though iterative procedures which also exploit the spatial context and allow the incorporation of generic a priori knowledge.

Chapter 1 of the thesis contains a concise introduction into the field of digital image segmentation. Some general notions and concepts are discussed and a short methodological review of segmentation techniques is given

In Chapter 2, we proceed with an introductory treatise of image modeling. Replacement models are described and some discretization issues are discussed. We then establish our preference for defining an edge as

a concatenation of edge pixels above the alternative of a concatenation of the pixel boundaries between neighboring pixels of adjacent regions. The estimation theoretic approach to object region delineation is discussed on the basis of some important contributions from the literature.

Chapter 3 is devoted to the dynamic programming approach to object region boundary detection. After discussing some results from the open literature, we discuss our restricted-search algorithm. In this method, the image data within a region-of-interest are geometrically transformed into a rectangular matrix. Next, a corresponding matrix of cost-coefficients is computed. Dynamic programming is used to search for the optimal path through this matrix. The resulting path is then transformed back to the image domain, thus constituting the object region boundary. This method is described and analyzed in great detail. Special attention is given to the geometric transformation. We establish both a curvature constraint and a global constraint which a boundary is to satisfy in order for our restricted-search method to be applicable to it.

In Chapter 4, we discuss continuous relaxation labeling procedures. In particular, we develop a feasible implementation of a relaxation labeling edge detector scheme. By decomposing the computations to a large extent into spatial convolutions and dyadic point operations, the proposed method is especially suited for implementation on modern digital image processing systems, which usually feature special purpose processors for these tasks. The initialization of the label probabilities is discussed in detail.

In Chapter 5, both the dynamic programming method and the relaxation labeling procedure are evaluated experimentally by measuring their performances when applied to synthetic test images with various values of the signal-to-noise ratio. The evaluation criteria include the estimated correct and false edge pixel probabilities and Pratt's figure of merit. It is shown that, especially at very low values of the signal-to-noise ratio, the dynamic programming method performs much better than the traditional methods of edge detection, even when the latter are combined with the relaxation labeling procedure.

Some possibilities for automated detection of regions of interest, as

required by the restricted-search dynamic programming algorithm, are discussed in Chapter 6. One of the proposals is to obtain a coarse pre-segmentation of the image by applying a region-oriented method. Such methods tend to correctly detect the presence of an object region but often show a rather poor performance in terms of the geometric accuracy of the region boundary.

Chapter 7 is devoted to two medical applications, in which the restricted-search dynamic programming approach has proven to be quite succesful. The first application concerns the analysis of gated blood-pool cardiac scintigrams, the second the analysis of coronary arteriograms.

Chapter 8 contains the conclusions and a general discussion of the results.

1. INTRODUCTION

In a broad sense, digital image analysis deals with the extraction of information from pictorial data by using digital techniques. One of the crucial issues in digital image analysis is the problem of image segmentation. The goal of image segmentation is to partition the image domain into regions which are homogeneous according to some predefined criteria. The segmentation result forms the basis for all subsequent measurements, object recognition, and, eventually, image interpretation.

In this thesis we discuss the digital segmentation of images which are severely contaminated by noise. We restrict the discussion to two-dimensional monochrome still images containing distinct object regions and a common background region. However, the segmentation methods to be developed here are applicable to a much larger class of images.

In this chapter, we briefly discuss some general notions and concepts. Section 1.1 contains an introductory treatment of the digital image analysis problem. Section 1.2 focusses on the segmentation stage and contains a short methodological review of segmentation techniques.

1.1. DIGITAL IMAGE ANALYSIS

Images are extremely important and widely used carriers of information, not only in everyday life, but also in medicine, remote sensing and industry, as well as in many fields of scientific research, ranging from anthropology to zoology. In the multidisciplinary field of digital image processing, both the methods and the means are being investigated and developed for the processing of such images by using digital computers or special purpose hardware. The goal to be achieved by processing an image in this way usually falls in one or more of the following categories:

1. *digital image coding* for the efficient and robust transmission or storage of images or image sequences by using data compression or data reduction techniques;
2. *digital image restoration and enhancement* to remove or reduce the effects of distortion and noise which may corrupt the image (restoration) or to amplify specific features of the image (enhancement);
3. *digital image analysis* to extract information from the image in the form of a measurement, a classification, a description or even an interpretation.

In the categories 1 and 2, the final result is again an image and the processing can be described as an image-to-image transformation. In category 3, the result is no longer an image, but data extracted from the image by the analysis process, which can thus be described as an image-to-data transformation.

In many cases, the imaging and analysis process consists of the following consecutive steps:

1. *image formation* where the physical phenomenon of interest is transformed into an image, usually by measuring the reflectance or absorption of, e.g., visible light or X-rays, or by measuring some kind of radiation originating from the object itself;
2. *image sensing* by means of an appropriate sensor, like a vidicon or CCD video camera, a γ-ray scintillation camera, etc., resulting in an electrical signal;

3. *image digitization* where the image is discretized both spatially (sampling) and in gray-value (quantization), yielding a matrix of discrete gray-values;
4. *preprocessing* by applying procedures for image calibration, restoration or enhancement to remove or reduce the effects of distortion and noise;
5. *segmentation* where the image is partitioned into hopefully meaningful regions, resulting in a region map at a symbolic level of description;
6. *region measurements* to assess characteristic region descriptors like e.g., geometric properties, texture or integrated optical density;
7. *object classification* by using techniques from the fields of statistical or structural pattern recognition;
8. *image interpretation* by analyzing the relationships between the objects in order to finally arrive at a semantic level of description of the entire image.

This general scheme is illustrated in figure 1.1.

The block scheme given in Fig. 1.1 should be regarded as a general model which describes the analysis process in a broad sense. In fact, there may be rather complicated interactions between the various processing steps which make the system much more complex than it appears in Fig. 1.1. On the other hand, some of the processing steps may be omitted in particular cases. In an analysis system intended for digital image mensuration, the processing steps 7 and 8 will obviously not be present. In fact, processing step 8 is included in Fig. 1.1 for the sake of completeness only, but really forms the object of state-of-the-art research in many laboratories, with relatively little practical results so far. This is not surprising, because the incorporation of knowledge and expert reasoning is extremely difficult, though very challenging. Knowledge-based approaches may also be of use within the various processing steps or for overall control of the system. We will not discuss these issues here. Some of our ideas concerning these possibilities can be found in Gerbrands et al. (1987) and Backer and Gerbrands (1988). A general introduction to the field of image analysis can be

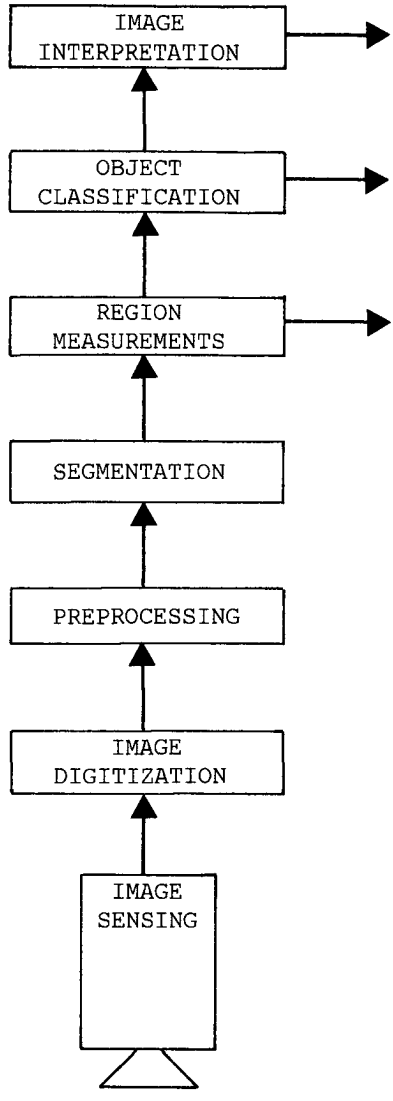


Fig. 1.1. General image analysis system.

found in many textbooks like, e.g., Duda and Hart (1973), Rosenfeld and Kak (1982), Pratt (1978), Gonzalez and Wintz (1977), Castleman (1979) and Niblack (1986).

1.2. DIGITAL IMAGE SEGMENTATION

As mentioned above, the goal of digital image segmentation is to partition the image plane into regions which are homogeneous according to some predefined criteria. The result of the segmentation stage is thus a map of the detected regions, which is hopefully meaningful with respect to the imaged objects. The term 'hopefully' is used here to stress the fact that the meaningfulness of the segmentation result can only be established by the later stages of the analysis process. This section contains a short tutorial review of the various approaches to digital image segmentation.

If for didactic purposes only, let us mentally model our two-dimensional monochrome still image as a gray-value landscape, where the gray-value of each picture element or pixel is represented by the altitude at the corresponding position in the landscape.

Consider an image containing some bright object regions in a dark background region. The dark parts in the image are represented by the valleys and low-lands in the landscape, the bright parts by mountain peaks and plateaus. If the difference between the gray-values of the object and background regions is sufficiently large in comparison with the inevitably present noise, then we can easily segment this image by applying a decision threshold to the gray-values: pixels with a below-threshold gray-value are assigned a label λ_0, indicating their membership of the background class, and pixels with an above-threshold gray-value are given the label λ_1, indicating that they belong to the object region. The thresholding procedure is followed by a connectivity analysis of the pixels with label λ_1 in order to identify the respective conglomerates. This process is called component labeling. Each conglomerate or component constitutes an object region in the segmentation mask. Many methods have been proposed for the automatic selection of a good segmentation threshold for a given image. The extension to multi-

ple decision threshold and to local or adaptive thresholds is relatively straightforward.

The dual approach of edge detection is based on the observation that there exists necessarily a transition region of intermediate gray-values between an object region and the surrounding background, i.e., a slope in the landscape. These slopes can be detected by computing some numerical approximation of the gradient vector. In the continuous case, the gradient of a two-dimensional function is a vector containing the partial derivatives of the function. In digital image processing, it is common practice to approximate a partial derivative operator by a discrete difference operator. As an example, the horizontal and vertical Sobel operators are given in Fig. 1.2. By convolving the image with these operators, we obtain the approximations to the partial derivatives. In the traditional approach of edge detection by edge enhancement/thresholding, the magnitude of the discrete gradient is computed

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Fig. 1.2. Sobel discrete difference operators

or approximated and the result is compared with a decision threshold. If at a certain pixel the magnitude of the gradient exceeds the threshold, it is assumed that the pixel lies on a significant slope of the gray-value landscape and is consequently detected as a so-called edge pixel. If the image contains distinct object regions, the idea is that the detected edge pixels will form a connected path around the gray-value mountain. When a closed path is found, a contour filling algorithm is applied to label the encompassed pixels as belonging to one region. An important variation is to look for local extrema of the magnitude of the gradient, which can be implemented by searching for zero-crossings of the second derivative. A drawback of gradient-based methods is that they tend to increase the effects of the noise, which is inevitably present in real-world imagery. This issue will be discussed in

some more detail in Chapter 2.

Gray-value thresholding methods as described above attempt to detect the regions explicitly, and the boundaries result implicitly. These are region-oriented methods. In the dual approach of edge detection, the boundaries between adjacent regions are detected explicitly, and the regions themselves result implicitly. This is usually called the edge or boundary approach. In both approaches discussed so far, the decisions are independently made on a pixel by pixel basis. Such methods are called parallel methods, because all decisions can, in principle, be taken simultaneously if the computing facilities allow that. As a result of these independent decisions, the performance of both methods is rather poor when the image is severely contaminated by noise, because the noise destroys the coherence of the structures we would like to detect. As a result, the detected regions will be fragmented or the detected boundaries broken. In addition, many spurious regions or edges will be detected.

One may attempt to solve this problem in a number of ways. The first approach is to suppress the noise by using preprocessing techniques. This can be achieved to a limited extent only, and when the image is very noisy it will be almost impossible to suppress the noise sufficiently without unacceptably affecting the image contents itself. The second approach, possibly applied in combination with the first, is to segment the image and attempt to improve the segmentation result by applying post-processing techniques. Such algorithms perform logical operations on the assigned labels in order to remove apparent label inconsistencies. Again, this can be achieved to a limited extent only. The third approach aims at the improvement of the performance of the segmentation stage itself, by incorporating some generic a priori knowledge about regions or boundaries in the segmentation algorithm, and by exploiting at least the local spatial context in which the decisions are to be made. This last approach forms the main subject of the present study.

The dichotomy of region-oriented methods versus boundary-oriented methods is not the only one possible. A second dichotomy is to distinguish parallel methods and sequential methods. Sequential methods of

digital image segmentation are characterized by the fact that the processing that is performed at a point is influenced by the results from earlier processing of other points. Combining both dichotomies we arrive at sequential region-oriented methods, commonly denoted as region-growing methods, and sequential boundary-oriented methods, generally known as edge-tracing methods.

Returning to our example of an image containing bright object regions and a dark background region, the following region-growing algorithm can be envisaged. Search for the highest peak in the gray-value landscape and consider the patch of pixels with maximum gray-value to be the initial estimate of the first region. An acceptance test is then applied to each pixel within a layer around the tentative region, and the accepted pixels are incorporated. This process is repeated until at a certain stage none of the candidate pixels is accepted, yielding the final region. The same procedure is then applied to find the next region, and so on.

It is our experience that region-growing methods tend to correctly detect the presence of the regions, but the geometric accuracy of the resulting region boundaries is often rather poor, especially when the image is severely contaminated by noise. In Chapter 3 we develop a sequential method of boundary detection, in which all possible boundaries within a predefined region of interest are considered and compared on the basis of a merit function. Maximization of the merit function, which depends on the complete boundary, results in the detection of the optimal object region boundary. The optimization technique of dynamic programming forms the heart of the method. Unlike in heuristic search techniques, the merit function is not used to guide the search. In the dynamic programming approach, all possible boundaries are evaluated implicitly and the method facilitates tracing back the optimal boundary once its end point is found. Actually, this point is called the starting point in the jargon of dynamic programming.

The dynamic programming method of boundary detection shows the usual advantages of sequential segmentation methods. A priori knowledge is incorporated in terms of the predefined region of interest and in the

choice of the merit function. The context is taken into account by evaluating the merit function along the entire boundary. The actual calculations are simple and straightforward. The method also shows the usual disadvantages of a somewhat complicated computer program and an inherent search-order dependency. However, it is felt that these are only minor drawbacks.

The parallel segmentation methods of gray-value or gradient-magnitude thresholding can also be extended in order to facilitate the utilization of generic a priori knowledge and local spatial context by combining them with a method called relaxation labeling. Relaxation labeling itself is also a parallel method, but consists of a number of iterations. Essentially, the decisions on the basis of the gray-value or the gradient-magnitude are postponed until sufficient supporting evidence is gathered from an iteratively extended neighborhood. In other words, this particular method of decision making in context inherently warrants a consistent segmentation labeling. In Chapter 4, we discuss this approach in detail and we develop a feasible implementation of a relaxation scheme for edge detection.

In Chapter 5 the methods of relaxation labeling and dynamic programming will be evaluated quantitatively by using synthetic test images with various amounts of noise. The dynamic programming method maintains a good performance, even for very low values of the signal-to-noise ratio. This is partly caused by the fact that the boundary search is restricted to a predefined region of interest. It is often possible to detect such regions of interest automatically by applying a separate algorithm to obtain a coarse presegmentation of the image. In Chapter 6 we discuss this aspect in more detail. The dynamic programming method of boundary detection which is described, analyzed and evaluated in this study is actually used in various applications. Some examples from the field of quantitative analysis of medical images are discussed in Chapter 7.

2. IMAGE MODELING

The purpose of this chapter is to discuss various aspects of image modeling in the context of digital image segmentation. The term 'modelling' is used here in rather a loose sense. Following Rosenfeld and Davis (1979), it is meant to imply the explicit formulation of the assumptions an image is to satisfy in order for a particular segmentation technique to be applicable to it. Usually, one distinguishes statistical models and spatial models. Statistical models describe the pixel population in an image or region in terms of first-order and, frequently, higher order statistics of the gray-values of the pixels. Some examples from this class of models are the gray-value distribution, the gray-value cooccurrence distribution of pairs of pixels, random field models, etcetera. The so-called spatial models describe an image in terms of regions and region properties. This type of models may include descriptions of how the image is composed of regions, statistics of region positions, orientations, sizes and shapes as well as models for region boundary curvature, etcetera. Some of the spatial models facilitate the introduction and exploitation of a priori knowledge from the specific application domain. The terms 'statistical' and 'spatial' models are somewhat misleading. Statistical models may very well consider spatial relationships

between pixels and spatial models may give statistical descriptions of region distributions or specific region properties. We prefer to use the terms 'gray-value model' and 'structural model' instead.

Given a structural model of an image, it still remains to choose gray-value models for the pixels in the regions. From this combination, it is often possible to derive a gray-value model for the image as a whole, but the reverse is usually not true. As discussed in Chapter 1, the problem to be addressed in this thesis is the segmentation of noisy two-dimensional monochrone images, containing distinct regions. In many applications, the regions will represent distinct objects or object parts in the original scene, which has led to the common practice in the field of digital image processing of using the terms 'region' and 'object' almost interchangeably. It should be noted that the term 'region' may be applied at both the signal level and the symbolic level, while the term 'object' certainly introduces semantic aspects. The purpose of the segmentation stage in an image analysis scheme is the accurate delineation of the region boundaries, which should correspond with the boundaries of the imaged objects.

2.1. REPLACEMENT MODELS

In the continuous case, a monochrome image can be described as a real valued function $f(x,y)$ of two independent real-valued variables x and y defined on the image domain D, i.e., $(x,y) \in D$. The function value $f(x,y)$ is usually referred to as the gray-value of the image at position (x,y). An image containing two distinct regions, say an object region and a background region, can be described at the symbolic level by introducing a two-valued replacement function $r(x,y)$,

$$r(x,y) \in \{0,1\}, \qquad \forall (x,y) \in D, \tag{2.1}$$

where $r(x,y)$ takes the value 1 if (x,y) belongs to the object region and value 0 elsewhere. Now let $f_0(x,y)$ and $f_1(x,y)$ denote the background image function and the object image function, respectively, where both functions are defined on the full domain D. The image containing two distinct regions can then be modeled as

$$f(x,y) = (1-r(x,y))\, f_0(x,y) + r(x,y)\, f_1(x,y). \qquad (2.2)$$

The replacement function r(x,y) constitutes a structural model of the image. Once it is known, it completely describes all geometric properties of the object region. Basically, the segmentation problem is the problem of recovering the replacement function r(x,y), given a discretized noisy version of the observable image f(x,y).

The binary replacement function r(x,y) defined in (2.1) corresponds with a partition of the image domain D into two disjoint subsets: a subset D_0 containing all positions (x,y) that belong to the background region and a subset D_1 containing the points of the object region. In the binary case, as described a.o. by Nahi and Jahanshahai (1977), the replacement function coincides with the characteristic function of the object region subset D_1:

$$r(x,y) \in \{0,1\} \qquad \forall (x,y) \in D \qquad (2.3a)$$

$$r(x,y) = 1 \text{ iff } (x,y) \in D_1, \qquad (2.3b)$$

where 'iff' stands for 'if and only if'.

Pursuing this approach, an image containing K distinct regions can be modeled by introducing K two-valued functions $\rho_k(x,y)$,

$$\rho_k(x,y) \in \{0,1\}, \qquad \forall (x,y) \in D,\ k=0,1,\ldots,K-1 \qquad (2.4a)$$

$$\rho_k(x,y) = 1 \text{ iff } (x,y) \in D_k, \qquad (2.4b)$$

where the subsets D_k, k=0,1,...,K-1, constitute a partition of the image domain into K disjoint subsets, i.e.,

$$D_k \subseteq D \qquad (2.5a)$$

$$\bigcup_k D_k = D \qquad (2.5b)$$

$$D_{k_1} \cap D_{k_2} = \emptyset \text{ if } k_1 \neq k_2. \qquad (2.5c)$$

An image containing K distinct regions can then be modeled as

$$f(x,y) = \sum_k \rho_k(x,y)\, f_k(x,y), \tag{2.6}$$

where each function $f_k(x,y)$, with $(x,y) \in D$, denotes the gray-value function of region k, k=0,...,K-1.

From a slightly different point of view, an image containing K distinct regions can be modeled by introducing a single K-valued function $\rho(x,y)$, i.e.,

$$\rho(x,y) \in \{0,\ldots,K-1\}, \quad \forall (x,y) \in D \tag{2.7a}$$

$$\rho(x,y) = k \text{ iff } (x,y) \in D_k, \tag{2.7b}$$

with the subsets D_k as defined above in (2.5). Now the image model becomes

$$f(x,y) = \sum_k \mathrm{MAX}\{0, 1 - |k-\rho(x,y)|\} \cdot f_k(x,y). \tag{2.8}$$

Both image models (2.6) and (2.8) are replacement models. The characteristic function approach of (2.6) could be generalized to continuous-valued membership functions in the sense of fuzzy set theory. The multi-valued function approach of (2.8), on the other hand, facilitates, e.g., the use of Markov random field models. In the sequel, we will restrict the discussion to a binary-valued replacement function quite frequently, in which case the characteristic function approach and the multi-valued function approach both yield the model of (2.2). Note that this does not necessarily imply a limitation to the single object case. So far, no connectivity constraints were introduced with respect to the subsets D_k. An image containing K *identically distributed* object regions may still be modeled by means of (2.1) and (2.2). Furthermore, note that a binary-valued replacement function does not necessarily imply a restriction to binary-valued images at the signal level. The nature of the image as a 2-D signal depends on the grayvalue functions $f_k(x,y)$ as well as on the replacement function. Frequently, the grayvalue functions are modeled as sample functions of stochastically independent two-dimensional random

processes. In addition, the replacement function is sometimes modeled as a sample of a random process, in which case it is assumed that the underlying processes of $\rho(x,y)$ and $f_k(x,y)$, $k=0,\ldots,K-1$, are stochastically independent. However, in a goal-oriented approach it may be more appropriate to adopt a deterministic model for the replacement function or for some of its properties.

2.2. DISCRETIZATION ISSUES

As stated in the previous section, the segmentation problem may be formulated as the problem of recovering the replacement function from a discretized version of the observable image $f(x,y)$. In general, image discretization involves aspects of image sensing, spatial sampling and amplitude quantization. Some of its effects will be discussed here in quite a qualitative way.

Let us model the continuous-space continuous-amplitude image $f(x,y)$ in terms of a discrete-valued replacement function $\rho(x,y)$ for all $(x,y) \in D$ in combination with K different constant-valued image functions $f_k(x,y)$. This model implies the presence of discontinuities at the region boundaries. Obviously, the assumed presence of ideal step edges requires infinite bandwidth of the 2-D signal $f(x,y)$ and is thus a mathematical abstraction. Leaving aside this observation, the model may still be acceptable in the continuous case. However, if the continuous image is discretized, the well-known sampling theorem definitely requires a bandwidth limitation in order to prevent aliasing. If we model the required anti-aliasing filter as an ideal low-pass filter, and if we sample the resulting signal at the minimum rate required by the sampling theorem, then the discrete image shows transition regions at the object boundaries with a width at least equal to the sampling distance. The following analysis may support this statement.

Consider a continuous image $f(x,y)$ containing one vertical ideal step edge at $y = y_0$ between two constant-valued regions. The image is filtered with an ideal low-pass anti-aliasing filter. The filtered image is then sampled by means of a square grid of Dirac pulses. This model is illustrated in Fig. 2.1. We will first restrict the analysis to the 1-D case.

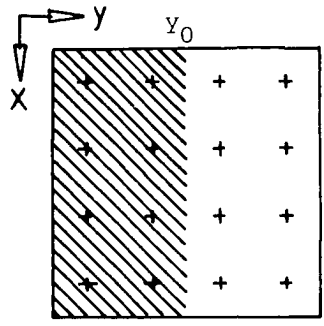


Fig. 2.1. Sampling grid on continuous image.

In the 1-D case, the continuous signal can be modeled as an ideal step function,

$$f(y) = u(y - y_0), \tag{2.9}$$

where

$$\begin{aligned} u(y) &= 0 \text{ for } y \leq 0 \\ &= 1 \text{ for } y > 0. \end{aligned} \tag{2.10}$$

This signal is filtered with an ideal low-pass filter with cut-off frequency ω_c. The output signal corresponds with the unit step response of the filter:

$$s(y) = \frac{1}{2} + \frac{1}{\pi} \mathrm{Si}\{\omega_c(y-y_0)\}, \tag{2.11}$$

where the function Si is defined as

$$\mathrm{Si}(a) = \int_0^a \frac{\sin b}{b}\, db. \tag{2.12}$$

The unit step response is sketched in Fig. 2.2. We now define the width τ of the transition region around the true edge position y_0 as the ratio between the step size and the maximum value of the derivative of the unit step response, i.e.,

$$\tau = \frac{1}{\{\frac{ds(y)}{dy}\}_{max}} = \frac{\pi}{\omega_c} . \qquad (2.13)$$

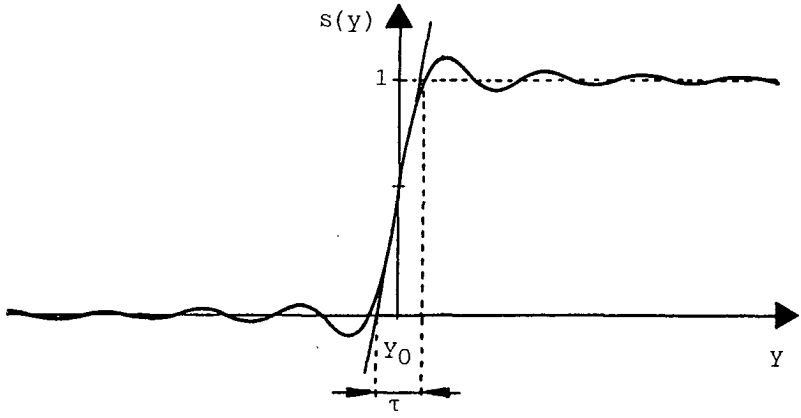


Fig. 2.2. Unit step response of ideal lowpass filter.

Sampling the bandlimited signal s(y) at the minimum rate required by the sampling theorem yields a sample distance which also equals π/ω_c. The sample positions are indicated in Fig. 2.3. for the special case where there is a sample located exactly at $y=y_0$, i.e., at the original step edge position.

Obviously, the transition region $[y_0 - \frac{\tau}{2}, y_0 + \frac{\tau}{2}]$ contains exactly one sample point, independent of the relative shift between the sampling grid and the continuous signal. In the 2-D case, one should consider both arbitrary shifts and arbitrary rotations between the 2-D sampling grid of Dirac pulses and the continuous image function. Without loss of generality, we will restrict the discussion to rotations between $-\frac{\pi}{4}$ and $+\frac{\pi}{4}$ with respect to the y-axis. Simple geometric analysis shows that the number of sample points in the transition region, counted along a row of the image, is now either one or two, depending on the rotation, the translation and the considered row.

So far, we only discussed the sampling process in terms of mathematical Dirac pulses, and neglected the effects of the physical sensing process. The actual measurement will consist of taking the weighted integral of the values of the continuous image within some finite neighborhood of the sample position. The weighting function reflects the spatial sensitivity characteristic of the sensor. For example, if there is a

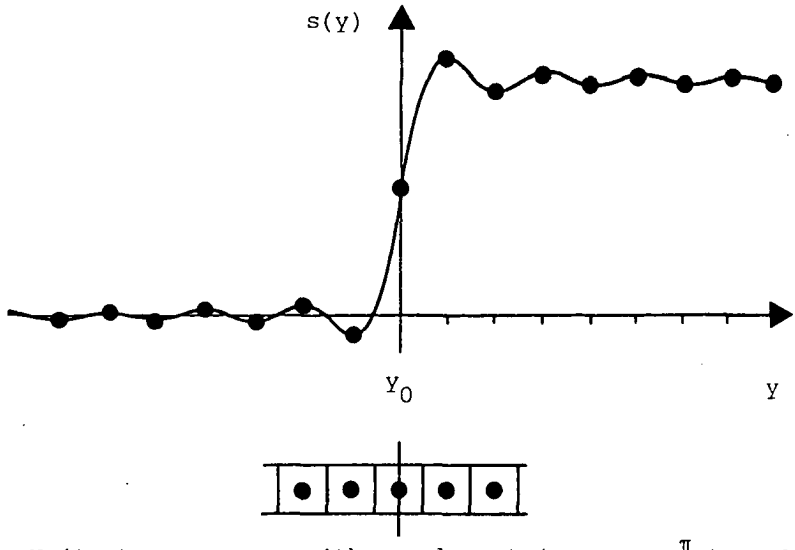


Fig. 2.3. Unit step response with samples at $(y_0 + n \cdot \frac{\pi}{\omega_c})$, $n=0,\pm1,\pm2,\ldots$

one-to-one correspondence, both in shape and in spacing, between sensor elements and pixels, and if the sensor elements show uniform sensitivity and linear response, then the pixel value will be proportional to the mean value of the signal portion which is being covered by that pixel. Now we define the transition region in the discrete image to be the set of pixels of which the values are affected by the presence of the transition region in the low-pass filtered continuous image. Simple geometric analysis now reveals that the number of pixels in the discrete transition region, counted along a row, may be as large as four. This is illustrated in Fig. 2.4.

What we learn from this rather elementary discussion is that even if the continuous image contains ideal step edges, the discrete image, which results from sensing the image and sampling according to the Shannon theorem, will contain transition regions which are quite distinct. Even in the case of a perfectly registered step edge along either the x- or the y-axis, the discrete transition region is at least one pixel wide. This observation is of importance in the development of edge detection schemes, but also in the design of artificial test images in digital computers when these test images are intended to model digitized real world images. The discrete transition regions will be even more distinct in case of oversampling, i.e. sampling well above the

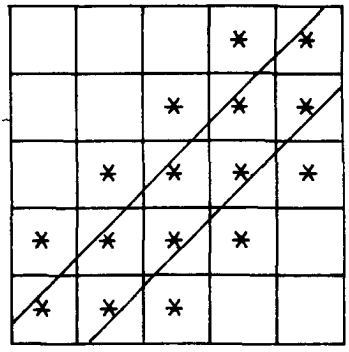


Fig. 2.4. Pixels in the discrete transition region are marked with asterisks.

Shannon rate. Oversampling is frequently used in quantitative image analysis in order to improve measurement accuracy. The spatial sampling process introduces quantization effects in any measurement problem involving the positions of points in the 2-D plane, e.g., measurements of edge location, edge length [Cf. L.Dorst, 1985] or geometric properties of objects [Cf. Ho, 1983].

The second part of the discretization process consists of quantizing the gray values into a finite set of discrete values. Obviously, this will result in quantization effects in any measurement problem involving these gray values, e.g., measurement of gradient values, integrated optical density, or texture. Unless stated otherwise, we will assume that the quantization noise may be neglected, at least in comparison with the other types of noise which are present.

A discrete image is then modeled as a real-valued function f(i,j) of two independent integer-valued variables i and j, with $(i,j) \in \{1,\ldots,N\} \times \{1,\ldots,N\}$, where i defines the row number and j defines the column number on the square grid. Following the replacement function approach of Section 2.1., a discrete image representing a continuous image containing identically distributed objects on a common background can be described at the symbolic level by introducing a ternary replacement function $\rho(i,j)$,

$$\rho(i,j) = \{0,1,2\}, \quad (i,j) \in \{1,\ldots,N\} \times \{1,\ldots N\}, \tag{2.14}$$

where $\rho(i,j)$ takes values of 0, 1 or 2 corresponding to the point belonging to the background, to an object or to the intermediate transition region. This leads to the following model:

$$f(i,j) = \sum_{k=0}^{2} \text{MAX}\{0,1 - |k-\rho(i,j)|\} \cdot f_k(i,j), \qquad (2.15)$$

with $f_0(i,j)$ representing the background function, $f_1(i,j)$ the object function and $f_2(i,j)$ the brightness function in the transition region. This function is some mixture of both the background function and the object function, but the mixing parameters are not position-invariant. The mixing parameters depend on the relative position of (i,j) with respect to the actual edge in the continuous image, on the shape of the point-spread function of the anti-aliasing filter and on the sampling. In a statistical model this would lead to stationarity problems. We will restrict the use of the model in Eq. (2.15) to some rather simple examples. It should be noted that the segmentation problem was formulated in Section 2.1 as the problem of recovering the binary-valued replacement function of the continuous image, given the discrete image. This implies that the model given in (2.15) is only of interest at intermediate levels of the analysis.

2.3. EDGE LOCATION

In the traditional gradient-type of edge detection techniques a pixel (i,j) is assigned to be an edge pixel if some numerical approximation to the magnitude of the gradient vector at (i,j) exceeds a detection threshold. In this case, a concatenation of such edge pixels constitutes an edge in the digital image. An example is given in Fig. 2.5a. Naturally, the discrete gradient masks used in these schemes are odd-sized and centered at (i,j). Common examples are the masks of Sobel and Prewitt, as mentioned in Chapter 1. An alternative is to use even-sized gradient masks and to assign the pixel boundary between two pixels to be an edge element if the magnitude of the gradient measured across this pixel boundary exceeds the detection threshold. The edge is then defined as the concatenation of pixel boundaries. An example is given in Fig. 2.5b.

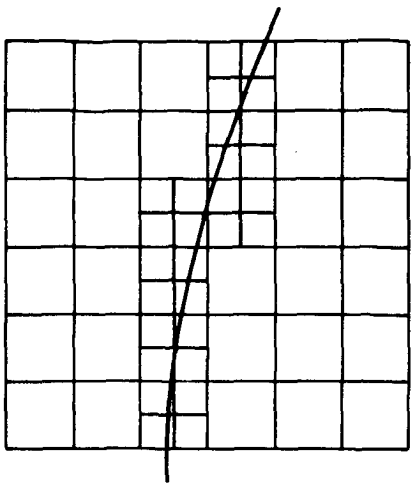


Fig. 2.5a. Edge pixels.

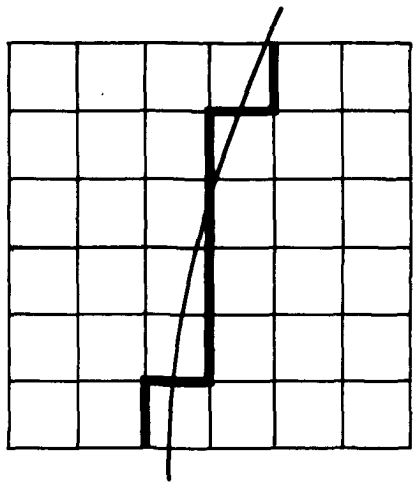


Fig. 2.5b. Pixel boundary edge.

At first sight, it may appear that the pixel boundary edge gives a more accurate estimate of the true edge location. This, however, is not true, and results from an incorrect interpretation of the concatenated-pixel edge of Fig. 2.5a. In the context of edge location, we should be more careful in defining the edge in terms of a concatenation of edge elements. In the case of odd-sized gradient masks the edge should be defined as the chain of line elements which connect the central points of successive edge pixels. An example is given in Fig. 2.6.

We will pursue our analysis with the 1-D example of Section 2.2, considering one row of an image containing an ideal vertical step edge which was filtered with an ideal low-pass anti-aliasing filter. In general, there is an unknown shift between the actual edge and the sampling grid. In the case of perfect registration, i.e., when there is a sample point located exactly at the original edge position, we define the shift φ to be zero. This situation is sketched in Fig. 2.7.

The first class of operators to be considered here contains an odd number of elements, say [+1 0 -1]. Given a specific shift between the

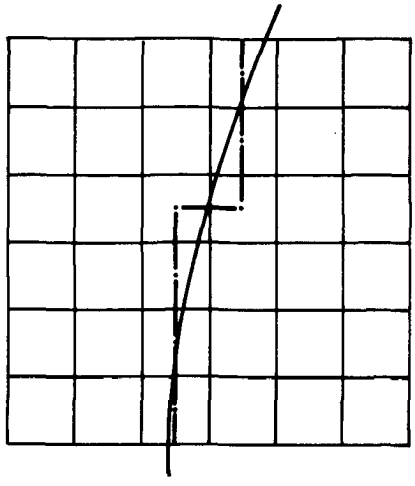
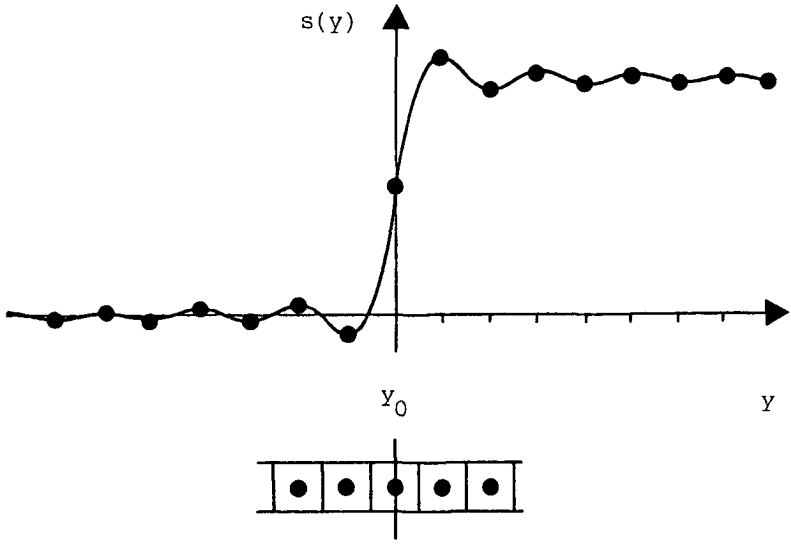


Fig. 2.6. Concatenated edge elements.



2.7. Sampled brightness function across the edge.

continuous brightness profile and the sampling grid, the sampled brightness function is convolved with this operator. The edge is detected at the point of maximum response of the discrete difference operator, and the position of the detected edge is defined as the central point of the central element. It is easy to verify that we can restrict the analysis to shifts between the continuous function and the sampling grid in the interval $[-h/2,+h/2]$, where $h = \pi/\omega_c$ is the sampling distance or grid constant. We assume the shift φ to be a random variable which is uniformly distributed between $-h/2$ and $+h/2$. The difference ε between the actual edge position y_0 and the detected position j_0 varies linearly with φ, and is thus also uniformly distributed. It is easy to see that ε is also limited to the interval $[-h/2,h/2]$, which leads us to

$$E(\varepsilon) = 0 \tag{2.16}$$

$$\mathrm{var}(\varepsilon) = h^2 / 12. \tag{2.17}$$

Now we turn to even-sized operators, say [+1 -1], and select as the detected edge position the central pixel boundary at maximum response. Simple analysis reveals that the position error is again uniformly distributed between $-h/2$ and $+h/2$, so the expected error and the variance of the error are the same as for the odd-sized operator. The analysis given here is similar to the one given by Ho (1983) for the case of segmentation by gray-value thresholding at the 50% level of the step size of the ideal edge, and the results are identical. The resulting position error variance is also equal to the variance of the quantization error which results when a 1-D signal with a uniformly distributed amplitude is quantized with a uniform quantizer, which emphasizes the fact that the spatial sampling process introduces quantization effects in any measurement problem involving the positions of points in the 2-D plane. This is also discussed in a somewhat different context in Van Otterloo and Gerbrands (1978).

From the above analysis we conclude that there is no difference in the edge location accuracy between an even gradient operator and its odd counterpart which contains an additional zero-valued central coefficient.

There is, however, a difference with respect to high-frequency noise sensitivity.

Let us first consider the frequency behavior of the discrete difference operator [1 -1]. Its impulse response is

$$h(j) = \delta(j) - \delta(j-1). \tag{2.18}$$

By using the z-transform in negative powers of z we arrive at the following transfer function:

$$H_1(z) = 1 - z^{-1}. \tag{2.19}$$

The Fourier modulation transfer function (MTF) is obtained by substituting $z = e^{jv}$ and computing the magnitude:

$$|H_1(v)| = |2 \sin \frac{v}{2}|. \tag{2.20}$$

This function is sketched for $0 \leq v \leq \pi$ in Fig. 2.8.

Now we turn to the operator [1 0 -1]. This odd operator can be decomposed into the concatenation of the previous discrete difference filter [1 -1] and a low-pass filter [1 1], because

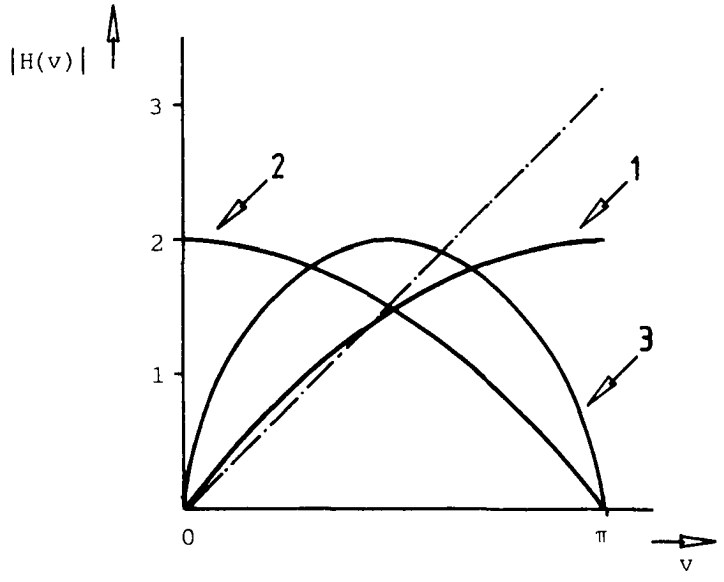


Fig. 2.8. Modulation transfer functions of the filters [1 -1] (1), [1 1] (2) and [1 0 -1] (3).

$$[1\ 1] * [1\ -1] = [1\ 0\ -1], \tag{2.21}$$

where * denotes convolution. Accordingly, the MTF $|H_3(v)|$ of the filter [1 0 -1] can be obtained by multiplying $|H_1(v)|$ with the MTF of the filter [1 1]:

$$|H_2(v)| = |2 \cos \frac{v}{2}|, \tag{2.22}$$

yielding

$$|H_3(v)| = |4 \sin \frac{v}{2} \cos \frac{v}{2}| = |2 \sin v|. \tag{2.23}$$

Both MTF's are also sketched in Fig. 2.8. From this figure, it is quite obvious that the filter [1 0 -1] has similar differentiating behavior for the lower frequencies as its counterpart [1 -1]. The higher frequencies, where the noise usually dominates the signal, are suppressed instead of amplified. Similarly, if we compare the even operator [1 1 -1 -1] and its odd counterpart [1 1 0 -1 -1], the decompositions

$$[1\ 1\ -1\ -1] = [1\ 1] * [1\ 0\ -1] \tag{2.24a}$$

and

$$[1\ 1\ 0\ -1\ -1] = [1\ 1\ 1] * [1\ 0\ -1] \tag{2.24b}$$

show the presence of additional low-pass filters [1 1] and [1 1 1] with MTF's

$$|H_4(v)| = |H_2(v)| = |2 \cos \frac{v}{2}| \tag{2.25a}$$

and

$$|H_5(v)| = |1 + 2 \cos v|, \tag{2.25b}$$

respectively. Again, this supports the observation of improved noise suppression of the odd operator in comparison with its even counterpart. Note that the extra zero-element can be introduced at no additional computational cost.

The above 1-D analysis directly translates to the 2-D case as the

common 2-D discrete difference operators can be decomposed into 1-D operators. For example, the horizontal Prewitt operator can be decomposed as follows:

$$\begin{bmatrix}1 & 0 & -1\\ 1 & 0 & -1\\ 1 & 0 & -1\end{bmatrix} = \begin{bmatrix}1\\ 1\\ 1\end{bmatrix} \otimes [1\ 0\ -1] \qquad (2.26)$$

where $\otimes$ denotes the Kronecker product of matrices. We recognize a low-pass smoothing filter along the columns in combination with a discrete difference operator as discussed above.

On the basis of this filter analysis approach we express a preference for odd-sized discrete difference operators, which naturally leads to a preference for pixel edges over pixel boundary edges. In addition, the use of pixel edges simplifies practical implementation as it prevents us from being forced to use both the original sampling grid and the pixel boundary grid in one image processing system.

2.4. PERTINENT LITERATURE

In this section we will briefly describe some of the replacement models from the pertinent literature. These models are presented here merely as examples; the treatise is certainly not claimed to be exhaustive.

Nahi and Jahanshahi (1977) discuss the boundary estimation problem for images containing a single object region within the background region. The digital image $f(i,j)$, with $(i,j) \in \{1,\ldots,N\} \times \{1,\ldots,N\}$, is modeled by means of a binary-valued replacement function $r(i,j)$:

$$f(i,j) = (1-r(i,j))\ f_0(i,j) + r(i,j)\ f_1(i,j). \qquad (2.27)$$

Here, $f_0(i,j)$ and $f_1(i,j)$ denote the background image function and the object image function, respectively. The replacement function is heavily constrained. First of all, the object region is assumed to be 'horizontically convex'. This means that if pixels (i_1,j_1) and (i_2,j_2) are both elements of the object region subset D_1, and $i_1=i_2$, then any pixel (i,j) with $j=\gamma j_1+(1-\gamma)j_2$, $0 < \gamma < 1$, is element of D_1. In other words, if two pixels on a row of the image belong to the object region, then

every pixel between them belongs to the object region. Now let m_1 and m_2 denote the first and the last rows of the image containing any object region pixels, if we scan the image row by row from top to bottom. The constraint of horizontal convexity implies that on each row ℓ, $m_1 \leq \ell \leq m_2$, a unique first object point (ℓ,a_ℓ) and a unique last object point (ℓ,β_ℓ), with $\beta_\ell \geq a_\ell$, can be defined as

$$a_\ell: (\ell,a_\ell) \in D_1 \wedge (\ell,a_\ell-1) \in D_0 \tag{2.28a}$$

$$\beta_\ell: (\ell,\beta_\ell) \in D_1 \wedge (\ell,\beta_\ell+1) \in D_0. \tag{2.28b}$$

This model is illustrated in Fig. 2.9.

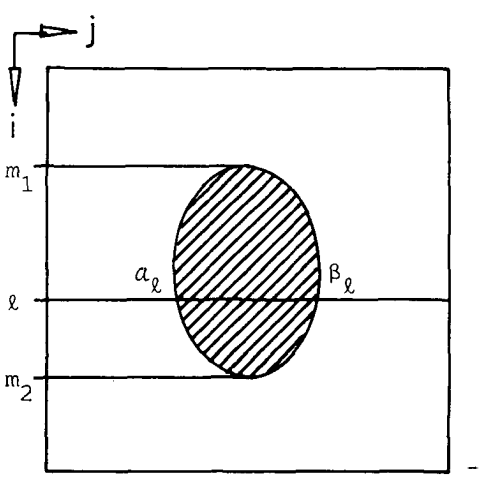


Fig. 2.9. The model of Nahi and Jahansahai

The statistics of the replacement function are then formulated in terms of the statistics of m_1, m_2 and the sequence $\{\omega_{m_1},\ldots,\omega_{m_2}\}$, where

$$\omega_\ell = (a_\ell,\beta_\ell) \quad m_1 \leq \ell \leq m_2. \tag{2.29}$$

This sequence is modeled as a first-order Markov sequence, i.e.,

$$p(\omega_\ell|\omega_{\ell-1},\omega_{\ell-2},\ldots,\omega_{m_1},m_1,m_2) =$$

$$p(\omega_\ell|\omega_{\ell-1},m_1,m_2), \tag{2.30}$$

in order to express the dependency of the object boundary points on consecutive rows. It is assumed that

$$p(\omega_\ell|\omega_{\ell-1},m_1,m_2) = p(\omega_\ell|\omega_{\ell-1},m_1) \tag{2.31}$$

which can be developed a little further into

$$p(\omega_\ell|\omega_{\ell-1},m_1) = p(a_\ell,\beta_\ell|a_{\ell-1},\beta_{\ell-1},m_1)$$

$$= p(a_\ell|a_{\ell-1},\beta_{\ell-1},m_1).p(\beta_\ell|a_\ell,a_{\ell-1},\beta_{\ell-1},m_1). \tag{2.32}$$

The density functions $p(m_1)$, $p(m_2/m_1)$, $p(a_\ell|a_{\ell-1},\beta_{\ell-1},m_1)$ and $p(\beta_\ell|a_\ell,a_{\ell-1},\beta_{\ell-1},m_1)$ are supposed to be given. Nahi and Jahanshahi then develop an estimation procedure to obtain a set of maximum a posteriori estimates for the unknowns m_1, m_2, a_ℓ and β_ℓ ($m_1 \leq \ell \leq m_2$), given a noisy observed image $g(i,j)$:

$$g(i,j) = f(i,j) + n(i,j). \tag{2.33}$$

Here, $f(i,j)$ is the noise-free image defined in Eq. (2.27), and $n(i,j)$ is an independent zero-mean Gaussian white noise field.

It is of crucial importance to note that the estimation procedure requires the background image function $f_0(i,j)$ and the object image function $f_1(i,j)$ in the replacement model to be known a priori. This severe restriction diminishes the applicability of the Nahi-Jahanshahi estimation procedure to the more general image segmentation problem. Surprisingly, it is suggested in the original paper that some other image segmentation procedure would be required to obtain estimates of the image functions $f_0(i,j)$ and $f_1(i,j)$ in the replacement model. It is interesting to dwell a little more upon some additional assumptions. First, it is assumed that the maximum width of the object is known:

$$L = \max_\ell \ (\beta_\ell - a_\ell). \tag{2.34}$$

Second, in the numerical derivation of the estimates, the conditional densities of Eq. (2.32) are replaced by $p(a_\ell|\hat{a}_{\ell-1},\hat{\beta}_{\ell-1},m_1)$ and $p(\beta_\ell|\hat{a}_\ell,\hat{a}_{\ell-1},\hat{\beta}_{\ell-1},m_1)$, respectively, where $\hat{a}_\ell$, $\hat{a}_{\ell-1}$ and $\hat{\beta}_{\ell-1}$ are esti-

mates of α_ℓ, β_ℓ and $\beta_{\ell-1}$. These conditional density functions are then chosen to be Gaussian density functions with mean values $\hat{\alpha}_{\ell-1}$ and $\hat{\beta}_{\ell-1}$ and with both variances equal to $|\hat{L}_{\ell-1}-L|^2$, where $\hat{L}_{\ell-1} = \hat{\beta}_{\ell-1}-\hat{\alpha}_{\ell-1}$. This means that the model simplifies to simple probabilistic expectation windows centered on the estimated boundary points on the previous row of the image.

The apparent theoretical elegance of the model is somewhat affected by these simplifying assumptions. This applies to the model used by Nahi and López-Mora (1978) as well. Their model is very similar, as both the object region width and the geometrical center position are assumed to satisfy first-order Markov-processes with respect to consecutive lines through the 'horizontally convex' object region. Main points of criticism towards both models are the constraint of 'horizontal convexity' and the lack of connectivity in a two-dimensional sense along the boundary. By searching for the first and last pixel of the object region on each row, the edge definition used here implicitly is the pixel boundary edge discussed in Section 2.3. The successive vertical edge elements are modeled explicitly, but the necessary horizontal connections result implicitly, as illustrated in Fig. 2.10.

A similar approach is used by Cooper et al. (1980) and Cooper and Sung (1983). Again, the image is described by a binary replacement model. The contour of the simply connected object region is modeled as the concatenation of pixel boundary edge elements. Instead of considering a model for the complete contour, the concept of local windows is introduced. Each window contains a part of the object region boundary, such that the boundary enters the window on one side, passes through the window as a simple connected edge and leaves the window through the opposite side. An example is given in Fig. 2.11.

Four cases may be distinguished, with the edge running from the left-hand side to the right-hand side, or vice versa, and with the edge passing from the top to the bottom, or conversely. The models for all cases are analogous. If the edge runs from the top to the bottom through the window, the boundary process is treated as a sequence of vertical edge elements, γ_ℓ, $\ell=1,\ldots,M$, where M is the number of rows. Each row ℓ

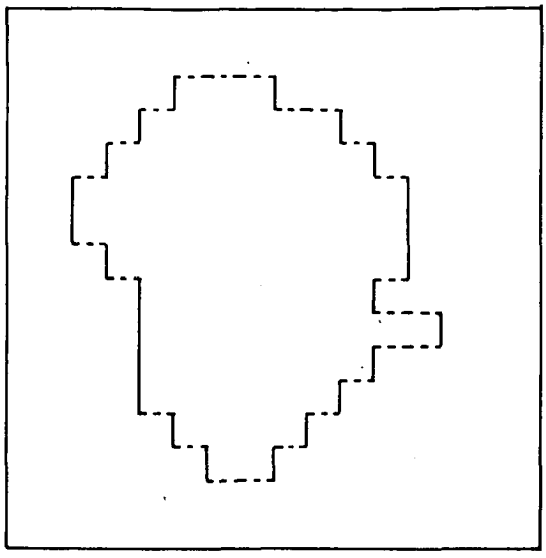


Fig. 2.10 Pixel boundary contour of object region based on Markov-model of vertical edge-elements.

contains exactly one edge element, its position being denoted as γ_ℓ^i, $i=0,\pm1,\ldots,\pm\frac{C}{2}$, where C+1 is the number of columns. An example is given in Fig. 2.12 for M=7 and C=11. Cooper c.s. discuss one particular model in detail, where the sequence $\{\gamma_\ell\}$ is modeled as a first-order Markov sequence with quantized Gaussian transition probabilities. Notice the similarity with the models of Nahi discussed above. Cooper c.s. develop a maximum likelihood boundary estimation procedure which is based on dynamic programming. This approach will be discussed in detail in the next chapter. Finally, the detected boundary segments from consecutive windows are seamed together to obtain a global boundary for the object region.

An alternative way of looking at the above model is to let the states of a Markov chain $\{x_\ell\}$ represent the geometric end points of the edge elements γ_ℓ. The transition from a particular state x_ℓ^i to a state

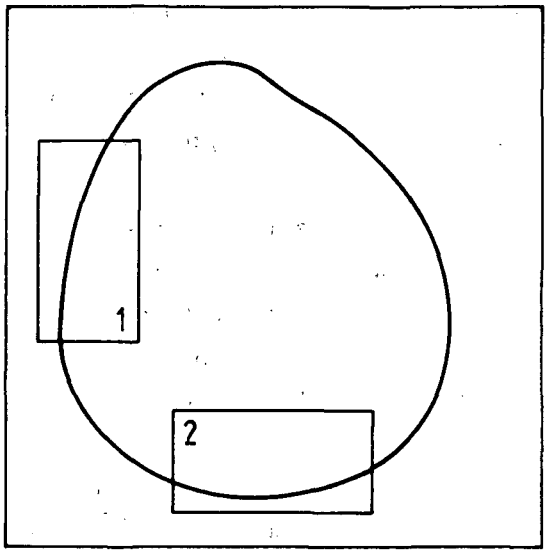


Fig. 2.11. The concept of local windows: examples of a vertical window(1) and a horizontal window (2).

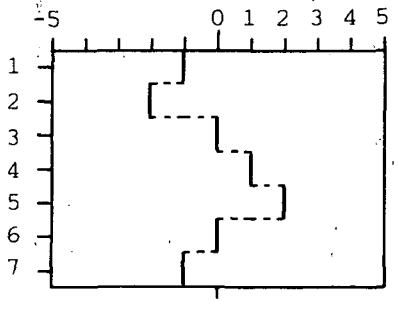


Fig. 2.12. Boundary in window, M=7, C=11.

$x^j_{\ell+1}$ corresponds with a unique sequence of $|i-j|$ horizontal edge elements, followed by one vertical element. Elliott and Srinivasan (1981) and Scharf and Elliott (1981) use a different geometric interpretation of the states of the Markov chain. There, the states correspond with the end points of edge segments terminating at the boundaries of successively larger rectangular blocks of pixels $R_\ell, \ell=1,\ldots,\ell_{max}$. Each rectangle is of size ℓ x $(2\ell-2)$ as denoted in Fig. 2.13. Not all boundary segments are

allowed. The key constraints are that edge segments departing one rectangle are not allowed to reenter it and that the edge segment between any two states x_{ℓ}^{i} and $x_{\ell+1}^{j}$ does not pass through another state located on the boundary of $R_{\ell+1}$. The possible state locations are depicted in Fig. 2.13 for ℓ=1,2,3,4. The number of possible state locations is 1 when ℓ=1, 3 when ℓ=2, and 4ℓ-3 when $\ell \geq 3$. This means that the matrix $P_{\ell,\ell+1}$ of state transition probabilities is of size (4ℓ+1)x(4ℓ-3), when $\ell \geq 3$. For example, $P_{6,7}$ is of size 25 x 21. To estimate these probabilities from a given class of images is quite impossible. Instead, Elliott c.s. choose the matrices of transition probabilities a priori. The choices are based on some heuristics related to the shapes of the corresponding boundary segments, i.e., the expressed preference for locally smooth and low curvature boundaries leads to relatively high values of the belonging transition probabilities.

It is interesting to note that the authors finally select transition matrices which effectively limit a boundary sequence from moving more than a distance of one pixel between consecutive states while equally favoring the straight path $\{x_1^1,x_2^2,x_3^5,x_4^7\}$ and both diagonal paths $\{x_1^1,x_2^3,x_3^7,x_4^{10}\}$ and $\{x_1^1,x_2^1,x_3^3,x_4^4\}$, with the states x_{ℓ}^{i} as depicted in Fig. 2.13. In a sense this implies that a relatively complex Markov model is

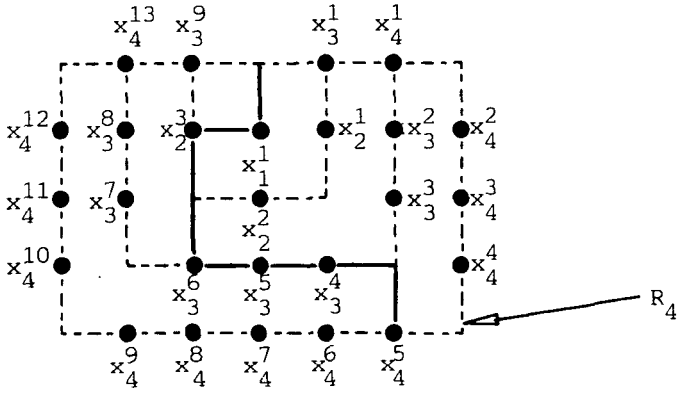


Fig. 2.13. Rectangular blocks R_{ℓ} (dotted) and possible state locations x_{ℓ}^{i} (heavy dots) for ℓ=1,2,3,4. The drawn line corresponds with the sequence $\{x_1^1,x_2^3,x_3^6,x_4^5\}$.

used to incorporate some very simple geometric notions. It also means that there is no truly probabilistic mechanism; a similar result could be obtained by using deterministic weighting functions.

A less restricted Markov model for the object region boundary is described by Cooper and Elliott (1978) and, in extended form, by Cooper (1979). Again, the boundary is modeled as a concatenation of directed pixel boundary edge elements. The i-th edge element in the sequence is denoted by t_i, and q successive elements, e.g., $\{t_{i-q},\ldots,t_{i-1}\}$, comprise a state vector for the q-th order Markov process. An edge element is not permitted to coincide with but be directed in the opposite direction to its immediate predecessor. Then, given $\{t_{i-q},\ldots,t_{i-1}\}$, t_i can be any of three edge elements t_i^1, t_i^2 or t_i^3, and there are three state transition probabilities for the three possible states $\{t_{i-q+1},\ldots,t_i^j\}$. In the above references, the case q=8 is discussed in detail. The state transition probabilities are chosen to be decreasing functions of the magnitude of the angle θ_i^j between a vector running from the beginning of element t_{i-7} to the end of t_{i-4}, and a vector running from the beginning of element t_{i-3} to the end of t_i^j. This is illustrated in Fig. 2.14.

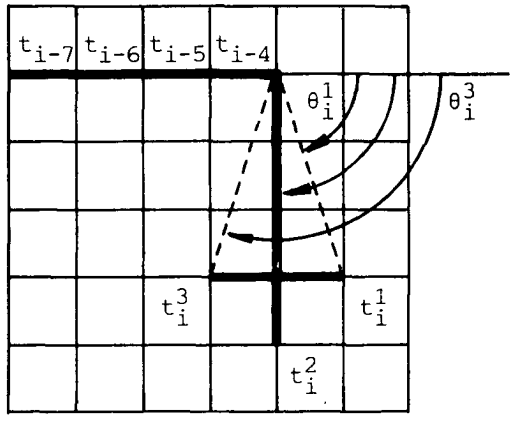


Fig. 2.14. The arrows indicate the angles θ_i^j associated with the three choices for edge element t_i^j, j=1,2,3.

This particular choice of the state transition probabilities originates from the fact that Cooper and Elliott explicitly attempt to give a maximum likelihood reformulation of the deterministic cost-function minimization approach as described by Martelli (1976). There, the angle θ_i^j is used as a measure of curvature and the cost-function comprises both the curvature along the boundary and the magnitude of the gray-value gradient accross the boundary. The deterministic cost-function minimization approach will be discussed extensively in Chapter 3. In the next section we will give an example of the maximum likelihood estimation of Markov-process object region boundaries.

2.5. MAXIMUM LIKELIHOOD BOUNDARY ESTIMATION

In this section we will give an example of the use of a Markov model for the object region boundary in combination with a maximum likelihood formulation of the boundary detection problem. The example given here is due to Cooper (1979).

Consider a digital image $f(i,j)$, with $(i,j) \in \{1,\ldots,N\}x\{1,\ldots,N\}$, which is modeled by a binary-valued replacement function $r(i,j)$ as in Eq. (2.2.):

$$f(i,j) = (1-r(i,j))\, f_0(i,j) + r(i,j)\, f_1(i,j). \tag{2.35}$$

Both the background image function $f_0(i,j)$ and the object region image function $f_1(i,j)$ are assumed to be constant-valued functions, f_0 and f_1, respectively, with $f_1-f_0 > 0$. The image is degraded by additive zero-mean Gaussian noise:

$$g(i,j) = f(i,j) + n(i,j). \tag{2.36}$$

The elementary edge elements are the pixel boundaries between adjacent pixels as discussed in Section 2.3. Each pixel boundary in the image is assigned a value by means of discrete difference operators. The entire set of these values is referred to as the edge map, and one can derive an expression for the joint likelihood of an edge map and a hypothesized object region boundary.

Denote the value of the edge element between pixels (k,ℓ) and (m,n)

by $e(k,\ell,m,n)$, where

$$e(k,\ell,m,n) = g(m,n) - g(k,\ell) \tag{2.37}$$

and it is assumed that $k=m+1$ and $\ell=n$ (vertical neighbors) or $k=m$ and $\ell=n-1$ (horizontal neighbors). For convenience we subscript the edge elements in a hypothesized boundary by a single subscript, i.e., e_i denotes the value of the i-th edge element in a sequence comprising a hypothesized boundary. Similarly, let $s(k,\ell,m,n)$ denote the value for an edge element in the noise free image $f(i,j)$, and let s_i denote the value of the i-th such edge element in a hypothesized boundary. Lastly, denote the i-th edge element in a hypothesized boundary by t_i as in the last paragraph of Section 2.4., and we also adopt the same q-th order Markov model for the object region boundary, with state transition probabilities

$$P_B(t_i,\ldots,t_{i-q+1}|t_{i-1},\ldots,t_{i-q}). \tag{2.38}$$

The likelihood of a particular object region boundary of length L is

$$P(L).P_B(t_q,\ldots,t_1) \cdot \prod_{i=q+1}^{L} P_B(t_i,\ldots,t_{i-q+1}|t_{i-1},\ldots,t_{i-q}) \tag{2.39}$$

where $P(L)$ is the a priori probability of having L elements in a hypothesized boundary and $P_B(t_q,..,t_1)$ is the likelihood of the first q elements. The likelihood of edge element value $e(k,\ell,m,n)$ is a Gaussian distribution

$$p_G(e(k,\ell,m,n)) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(e(k,\ell,m,n))^2}{2\sigma^2}\right] \tag{2.40}$$

if the edge element is not part of the boundary, and is

$$p_G(e_i-s_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(e_i-s_i)^2}{2\sigma^2}\right] \tag{2.41}$$

if the edge element is part of the boundary, indeed. The joint likelihood of the edge map and a hypothesized object region boundary is the

product of the likelihood of the boundary and the conditional likelihood of the edge map, given the boundary. Assuming statistical independence the log likelihood is

$$\mathcal{L}_{E,B} = \ell n \;\; P(L) + \ell n \; P_B(t_q,\ldots,t_1) +$$

$$+ \sum_{i=q+1}^{L} \ell n \; P_B(t_i,..,t_{i-q+1} | t_{i-1},\ldots,t_{i-q}) +$$

$$+ \sum_{i=1}^{L} \ell n \; p_G(e_i - s_i) + \sum_{(k,\ell,m,n)} \ell n \; p_G(e(k,\ell,m,n)) \qquad (2.42)$$

where the last summation is taken over the set of edge elements (k,ℓ,m,n) which do not belong to the boundary. In Cooper (1979) it is shown that (2.42) can be simplified to

$$\mathcal{L}_{E,B} = \text{constant} + P(L) - \sum_{i=q+1}^{L} c(t_i,\ldots,t_{i-q+1}) +$$

$$+ \sum_{i=1}^{L} \frac{s_i e_i}{\sigma^2} - \sum_{i=1}^{L} \frac{s_i^2}{2\sigma^2} \qquad (2.43)$$

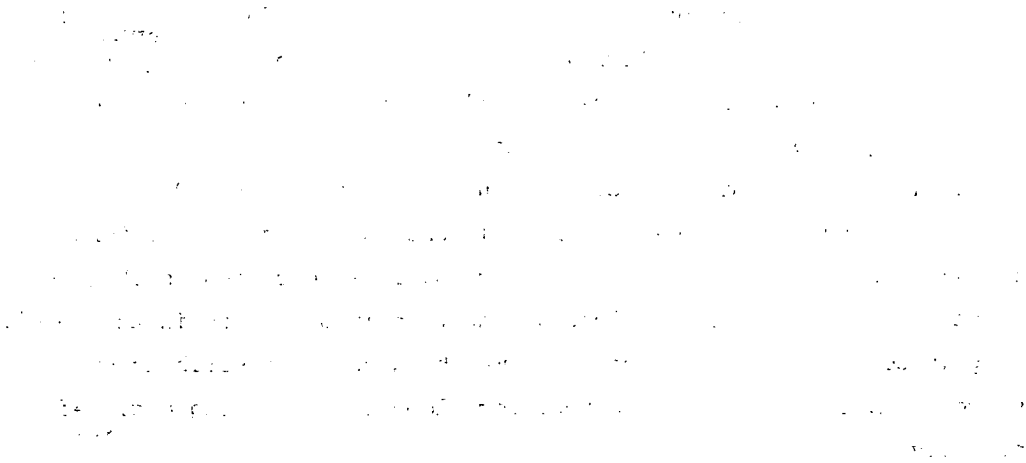
where $c(t_i,\ldots,t_{i-q+1})$ denotes a curvature-like function of the sequence $\{t_{i-q+1},\ldots,t_i\}$ as was discussed in the final paragraph of the previous section. The term 'constant' contains all terms of the log likelihood which are not influenced by the choice of the elements t_i. Equation (2.43) is maximized through appropriate choice of the variables t_i, $i=q+1,\ldots,L$.

It should be noted that the assumption of statistical independence of the edge values does not hold for edges which are related in the sense that some common pixel value is involved in the computation of the edge values. In particular, the edge elements of the same pixel are partially correlated. This problem is discussed in Cooper (1979) and appears to be non-significant.

Formally, all the picture data are involved in Eq. (2.43). However, the only data explicitly involved in the estimation are the edge elements along a hypothesized boundary. As a consequence of this observa-

tion, the maximization of the log likelihood can be done sequentially by using heuristic search techniques or dynamic programming. For reasons of computational costs, this search will have to be restricted to a predefined region of interest within the image.

Finally, it should be noted that the likelihood in (2.39) is correct for a boundary of variable length L. If one is particularly interested in simple closed object region boundaries, it appears that such global constraints (simple curve, closed) cannot be incorporated in the model. As pointed out by Cooper, the model may be good for describing subsegments, but is not quite correct for globally describing a closed boundary.



3. THE DYNAMIC PROGRAMMING APPROACH

This chapter deals with the detection of object region boundaries by applying an optimization technique commonly referred to as dynamic programming. The assumed or required properties of a hypothesized boundary are embedded in a deterministic cost-function and the boundary detection problem becomes the problem of searching for a minimum-cost solution. It will be shown that such solution can be found by applying well-known algorithms from the field of operations research. For reasons of computational complexity, it is attractive if not necessary to restrict the search space as much as possible. In the method to be developed here, this is accomplished by restricting the search for the object region boundary to a predefined region of interest in the image domain. The definition of a region of interest is application-dependent, and examples will be discussed later. The image data in the region of interest are transformed into a rectangular matrix. Next, a corresponding matrix of cost-coeffcients is computed and dynamic programming is used to find the optimal path through this matrix. The resulting path is then transformed back to the original image domain, thus constituting the object region boundary.

The method to be developed here draws heavily upon the early work of Montanari (1971) and Martelli (1972, 1976), which will be outlined in Section 3.1. The method itself will be described in Section 3.2. The issue of resampling the image data within the arbitrarily shaped region of interest into a rectangular matrix is the subject of Section 3.3. The constraints will be formulated which a boundary is to satisfy in order for our method to be applicable to it. The chapter is concluded with an example in Section 3.4 and some final remarks in Section 3.5.

3.1. PERTINENT LITERATURE

The idea of using dynamic programming in image analysis and pictorial pattern recognition is usually attributed to Kovalevsky (1967), who applied the method to the problem of separating noisy images of lines of typewritten characters into the individual symbols. Other applications, as discussed in Kovalevsky (1980), include the recognition of straight lines in binary images and the recognition of typewritten and handprinted characters.

Independently, and almost simultaneously, a similar approach appeared in the literature on information theory when Viterbi (1967) published his algorithm for the asymptotically optimal decoding of convolutional codes. The Viterbi algorithm has found use in many information theoretic problems ever since, as it can be applied to any problem concerning the maximum a posteriori probability estimation of the state sequence of a finite-state discrete-time Markov process observed in memoryless noise. It is curious to note that the Viterbi algorithm was only later shown to be a shortest-route algorithm of a type long known in the field of operations research as a variant of dynamic programming, as is mentioned by Forney (1972, 1973) in his extensive treatment of the algorithm. If one adopts a finite-state Markov model for the object region boundary, then the applicability of the Viterbi algorithm to the boundary estimation problem is obvious from our discussion in Section 2.5.

A major contribution to the dynamic programming approach is given by Montanari (1971), who used the technique for the optimal detection of

low-curvature curves in noisy pictures. Consider an ideal digital image $f(i,j)$, with $(i,j) \in \{1,\ldots,N\} \times \{1,\ldots,N\}$. The image contains a discrete curve, which consists of pixels with constant gray-value $f_c(.,.)$, against a uniform background of pixels with gray-value $f_b(.,.)$, with $f_c > f_b$. The connectivity of the curve is based on the notion of 8-adjacency. Two pixels $p = (i_p,j_p)$ and $q = (i_q,j_q)$ are 8-adjacent or 8-neighbors if the chessboard distance $d_8(p,q)$ between p and q is equal to 1:

$$d_8(p,q) = \max\{|i_p-i_q|,|j_p-j_q|\} = 1. \tag{3.1}$$

With this definition, any point which is not in the first or last row or column of the image has exactly eight neighbors. An 8-connected curve is a sequence of pixels $p_1,\ldots,p_L$ such that p_ℓ is an 8-neighbor of $p_{\ell-1}$, $2 \leq \ell \leq L$. A general treatment of these concepts of digital geometry may be found, a.o. in Rosenfeld and Kak (1982). The problem discussed in detail by Montanari (1971) is to recover a curve of given length L when the image $f(i,j)$ is corrupted by additive noise:

$$g(i,j) = f(i,j) + n(i,j). \tag{3.2}$$

The noise $n(i,j)$ is assumed to be a signal-independent, zero-mean, uncorrelated and identically distributed noise field.

To recover the curve from the noisy image $g(i,j)$, we have to find the actual values of the unknown variables $p_1,\ldots,p_L$. Here, assigning a value to the variable p_ℓ means specifying its coordinates in the image domain, with $p_\ell = (i_\ell,j_\ell) \in \{1,\ldots,N\} \times \{1,\ldots,N\} = \mathbb{N}^2$. Due to constraints like the connectivity constraint (3.1), the set of admissable values of the variable p_ℓ may be considerably smaller:

$$p_\ell(i_\ell,j_\ell) \in \mathbb{N}_\ell^2 \subseteq \mathbb{N}^2. \tag{3.3}$$

For example, $\mathbb{N}_\ell^2$ is the 8-neighborhood of $p_{\ell-1}$ except $p_{\ell-2}$, in which case $\mathbb{N}_\ell^2$ contains seven points only, with $\ell > 2$. Montanari (1971) proposes a solution to the curve detection problem which is based on the determination of the optimal curve with respect to a given merit func-

tion $\Phi_*(p_1,\ldots,p_L)$. The optimization process consists of two steps:

I. Find the maximum value Φ_*^m of the merit function $\Phi_*(p_1,\ldots,p_L)$, with $p_\ell \in \mathbb{N}_\ell^2$, $1 \leq \ell \leq L$; (3.4a)

II. Find the set of values $\{p_1^m,\ldots,p_L^m\}$ for which the maximum Φ_*^m is achieved. (3.4b)

In general, the optimization problem can be solved by complete exhaustion only, and the computational burden is prohibitive. Assume that we are searching for a curve of length L=30 in an image of size N=64, and assume that p_1 and p_2 are predefined as $p_1=(32,32)$ and $p_2=(31,33)$. Let the cardinality of all subsets $\mathbb{N}_\ell$, $2 \leq \ell \leq L$, be equal to 7 as mentioned above. In the absence of any additional constraints, the number of possible curves would be in the order of 7^{28} ($\approx 4.6 \times 10^{23}$).

The optimization problem can be greatly simplified by a proper choice of the merit function Φ_*. Particularly, if the merit function is defined as the sum of a number of terms, and each term depends on a small subset of the variables $\{p_1,\ldots,p_L\}$ only, then a multistage optimization procedure like dynamic programming can be used (Bellman and Dreyfus (1962)). An example of particular relevance to our discussion is the following merit function:

$$\Phi_*(p_1,\ldots,p_L) = \Phi_1(p_1,p_2) + \ldots + \Phi_{L-1}(p_{L-1},p_L). \tag{3.5}$$

In this case, the merit function Φ_* can be evaluated in terms of accumulating merit functions $\varphi_\ell(p_\ell)$ by means of the following recursion:

$$\varphi_1(p_1) = 0, \tag{3.6a}$$

$$\varphi_{\ell+1}(p_{\ell+1}) = \max_{p_\ell \in \mathbb{N}_\ell^2} [\varphi_\ell(p_\ell) + \Phi_\ell(p_\ell,p_{\ell+1})],$$

$$\forall p_{\ell+1} \in \mathbb{N}_{\ell+1}^2, \quad \ell=1,\ldots,L-1. \tag{3.6b}$$

At the final stage we obtain the maximum value Φ_*^m of the merit function $\Phi_*(p_1,\ldots,p_L)$ by means of:

$$\Phi_*^m = \max_{p_L \in \mathbb{N}_L^2} \varphi_L(p_L), \tag{3.7}$$

which solves part I of the optimization problem as stated in (3.4a). At the same time, we obtain the first point of the curve:

$$p_L^m = \arg \max_{p_L \in \mathbb{N}_L^2} \varphi_L(p_L). \tag{3.8}$$

The second point to be found is p_{L-1}^m, which is given by

$$p_{L-1}^m = \arg \max_{p_{L-1} \in \mathbb{N}_{L-1}^2} [\varphi_{L-1}(p_{L-1}) + \Phi_{L-1}(p_{L-1}, p_L^m)]. \tag{3.9}$$

In this way, all points p_ℓ^m, $\ell=L-1,\ldots,1$, can be found by recursively applying

$$p_\ell^m = \arg \max_{p_\ell \in \mathbb{N}_\ell^2} [\varphi_\ell(p_\ell) + \Phi_\ell(p_\ell, p_{\ell+1}^m)], \quad \ell=L-1,\ldots,1. \tag{3.10}$$

A simple way to achieve this goal is to store, for each value of $p_{\ell+1}$, the value of p_ℓ for which the maximum is obtained while evaluating Eq. (3.6b). In other words, when recursively applying (3.6b) we do not only store the intermediate cumulative merit values $\varphi_{\ell+1}(p_{\ell+1})$, but also the pointers $m_{\ell+1}(p_{\ell+1})$ for which the maximum values are achieved:

$$m_{\ell+1}(p_{\ell+1}) = \arg \max_{p_\ell \in \mathbb{N}_\ell^2} [\varphi_\ell(p_\ell) + \Phi_\ell(p_\ell, p_{\ell+1})],$$

$$\forall p_{\ell+1} \in \mathbb{N}_{\ell+1}^2, \ \ell=1,\ldots,L-1. \tag{3.11}$$

Part II of the optimization problem, as stated in (3.4b), is then solved by tracing through the array of pointers by means of the recursion:

$$p_L^m = \arg \max_{p_L \in \mathbb{N}_L^2} \varphi_L(p_L), \tag{3.12a}$$

$$p_\ell^m = m_{\ell+1}(p_{\ell+1}^m), \quad \ell=L-1,\ldots,1. \tag{3.12b}$$

If all sets of admissable values $\mathbb{N}_\ell^2$ are of the same cardinality $||\,\mathbb{N}_c^2\,||$ the brute-force evaluation of Eq. (3.5) would require in the order of (L-1) additions for each of the $||\,\mathbb{N}_c^2\,||^L$ possible curves, and $||\,\mathbb{N}_c^2\,||^L$ tests to select the optimal curve. The total computational effort would thus be in the order of L x $||\,\mathbb{N}_c^2\,||^L$ additions and tests. The dynamic programming approach requires in the order of L x $||\,\mathbb{N}_c^2\,||^2$ additions and tests only. We arrive at this estimate by noticing that the main computational burden lies in the recursion (3.6), where we have to compute the maximum over $||\,\mathbb{N}_c^2\,||$ possible values of p_ℓ for all $||\,\mathbb{N}_c^2\,||$ possible values of $p_{\ell+1}$ and for all L stages. The penalty for the decrease in the number of computations is an increase in memory requirements. Depending on the actual implementation, we have to store in the order of $||\,\mathbb{N}_c^2\,||^2$ values of the accumulating merit functions within a step of the recursion. In addition, we have to store an array of size L x $||\,\mathbb{N}_c^2\,||$ containing the pointers $m_\ell(p_\ell^m)$ of Eq. (3.12). However, the increased memory requirement is not considered to be of major importance when modern computer facilities are applied.

An example discussed in detail by Montanari (1971) concerns a merit function with the following structure:

$$\Phi_*(p_1,\ldots,p_L) = \Phi_2(p_1,p_2,p_3) + \ldots + \Phi_{L-1}(p_{L-2},p_{L-1},p_L). \tag{3.13}$$

The aim is to find the curve with maximum accumulative gray-value and minimum curvature along the curve. The curvature in point p_ℓ is computed from the difference between the chain-code of the vector connecting $p_{\ell-1}$ and p_ℓ and the chain-code of the vector connecting p_ℓ and $p_{\ell+1}$. Denoting the gray-value of point $p_\ell = (i_\ell,j_\ell)$ by $g(p_\ell)$ and the curvature at p_ℓ by $c\,(p_{\ell-1},p_\ell,p_{\ell+1})$, the following merit function is used:

$$\Phi_*(p_1,\ldots,p_L) = \sum_{\ell=1}^{L} g(p_\ell) - q \sum_{\ell=2}^{L-1} c(p_{\ell-1},p_\ell,p_{\ell+1}), \qquad (3.14)$$

where q is a weighting coefficient, with constraints to assure the connectivity of the curve and to limit the angle between successive chain-code vectors to $\pi/4$ radians. This effectively reduces the cardinality of the set $\mathbb{N}^2_{\ell+1}$ to three for a given pair of values for $p_{\ell-1}$ and p_ℓ.

One of the drawbacks of the method, as mentioned by Montanari (1971), is the fact that it is very hard to incorporate constraints of a more global nature into the method. Especially, if we are looking for a curve which is both simple and closed, there will be interaction between all points of the curve and the merit function cannot be decomposed like in (3.5) or (3.13). We will address this problem in Section 3.2.

Drawing heavily upon the work of Montanari (1971), Martelli (1972) proposes a similar approach to the problem of detecting edges in noisy pictures. The considered edges are the pixel boundary edges discussed in Section 2.3, i.e., an edge is a concatenation of pixel boundaries. These cracks between 4-adjacent pixels are the elementary edge elements. When the edge is the boundary separating two regions which have different constant gray-values, discrete difference operators with an even number of elements can be used to obtain a numerical approximation to the gray-value gradient accross the crack, as has been discussed in Section 2.3. A good edge will be one for which the sum of these gradient values along the edge is high. By substituting the gradient values for the gray-values in the merit function of Montanari (1971) described above, the application of the dynamic programming optimization procedure is straightforward. However, Martelli (1972) prefers to formulate the problem as the problem of searching for the minimum-cost path through a weighted graph, and to use the well-known A* heuristic search algorithm from the field of artificial intelligence (Nilsson (1971)).

The nodes of the graph correspond with the elementary edge elements. There exists a directed arc between two nodes if the corresponding two edge elements can be consecutive elements of an edge in the image domain,

i.e., they are adjacent. To each edge element a cost coefficient is assigned defined by:

$$c(i_p,j_p;i_q,j_q) = Q - |g(i_p,j_p)-g(i_q,j_q)|, \qquad (3.15)$$

where g(.,.) is the pixel gray-value defined in (3.2), the pixels (i_p,j_p) and (i_q,j_q) are 4-neighbors and Q is a large positive constant, e.g., the maximum possible value of the gradient. In the graph, these cost coefficients are assigned to the arcs directed to the corresponding nodes. An example of part of a gray-value image and the corresponding graph is given in Fig. 3.1. If an arc is directed from node $p_{\ell-1}$ to node p_ℓ, then node p_ℓ is called a successor of node $p_{\ell-1}$ and node $p_{\ell-1}$ is said to be a parent of node p_ℓ.

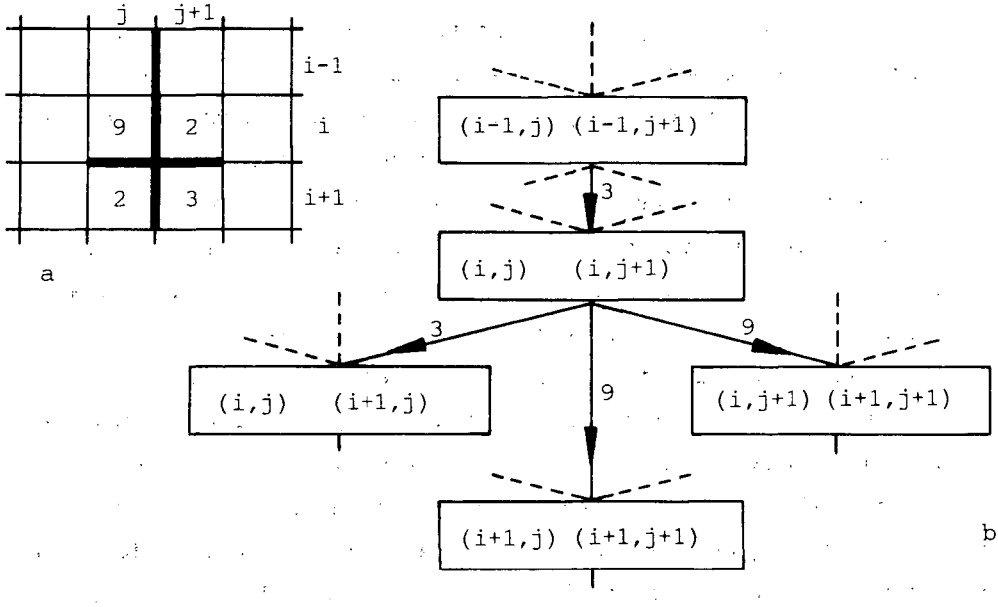


Fig. 3.1. a) Gray-value image. b) corresponding weighted directed graph for Q=10 (not all arcs have been drawn).

A sequence of nodes $p_1,\ldots,p_L$ is called a path from p_1 to p_L if p_ℓ is a successor of $p_{\ell-1}$, $2 \leq \ell \leq L$, and the sum of the cost coefficients of the directed arcs in the path is the cost of the path. The algorithm A* may now be used to find the path with minimal cost from a given starting node s to any member of a set of goal nodes. Let $\Gamma(p_\ell)$ be an estimate of the cost of a minimal cost path from s to a goal node which is constrained to go through p_ℓ. The estimate $\Gamma(p_\ell)$ can be expressed as the sum of the estimate of the cost of the optimal path from s to p_ℓ and an estimate of the cost of the minimal cost path from p_ℓ to a goal node:

$$\Gamma(p_\ell) = \hat{\gamma}(p_\ell) + \hat{h}(p_\ell). \qquad (3.16)$$

For $\hat{\gamma}(p_\ell)$ we take the cost of the lowest cost path from s to p_ℓ found so far. Heuristics from the problem domain can be utilized to evaluate $\hat{h}(p_\ell)$. The node with the smallest value of Γ found so far is hypothesized to be on the optimal path. In the algorithm A* this node is expanded next. Expanding a node means that the function Γ is evaluated for all successors of the current node. It is well-known that if $\hat{h}(p_\ell)$ is a lower bound on the cost of the optimal path from p_ℓ to any goal node, then algorithm A* will find the optimal path between the starting node s and a goal. If we choose $\hat{h}(p_\ell) \equiv 0$, the algorithm is called the uniform cost algorithm. A description of the algorithm A* may be found in Nilsson (1971), and will be omitted here.

One of the arguments of Martelli (1972) to prefer the algorithm A* to dynamic programming is that the term $\hat{h}(p_\ell)$ in Eq. (3.16) can be used to incorporate heuristic knowledge from the application domain. A second argument is that dynamic programming is a 'blind' procedure in the sense that even in the absence of noise all possible paths are evaluated, while a best-first expansion as in A* finds the optimal path almost directly. A final argument is that a proper choice of the heuristic term $\hat{h}(p_\ell)$ guarantees to find the optimal solution.

The experiments reported by Martelli (1971) concern the problem of finding the minimal cost edge through a noisy image, where the edge is

known to start in the first row and to end in the last row. Apparently, the term $\hat{h}(p_\ell)$ is set identical to 0, so no heuristic information is used. To speed up computation time, a 'pruning' mechanism is introduced. Suppose that the currently active node corresponds with an edge element on row k in the image, then all nodes corresponding with edge elements on row k' with $k' \leq k-3$ will be deleted from the search, i.e., they will never be expanded. This pruning mechanism invalidates any claim of optimality. In addition, Martelli (1972) observes that for low values of the signal-to-noise ratio, the computation time becomes independent of the signal-to-noise ratio. In other words, the algorithm becomes 'blind', like the dynamic programming procedure.

In Martelli (1976) the method is extended to the problem of finding the closed boundary of an object region. Based on a priori knowledge about the size and position of the object, a number of starting nodes are selected ad hoc. The goal nodes are defined dynamically during the search in such a way that the corresponding contour in the image is closed. In addition, the smoothness of the curve is taking into account by assigning an extra cost coefficient to the curvature of the contour, like in Eq. (3.14). The curvature estimate, however, is now based on the angle between two vectors which locally approximate the hypothesized edge. This procedure was described in more detail in Section 2.4 while discussing the results of Cooper and Elliott (1978) and Cooper (1979).

3.2. A RESTRICTED-SEARCH ALGORITHM

The applicability of the dynamic programming approach can be greatly improved if there are ways to effectively restrict the search, because this would yield a significant reduction in both computation time and memory requirements.

An obvious possibility is to confine the search to some predefined region of interest in the image domain. In many applications, particularly those concerning object mensuration and inspection, there is sufficient generic knowledge about the object in order to define such a region of interest a priori. Some examples will be given in Chapter 7. In other cases, it may be possible to obtain a region of interest from

a coarse pre-segmentation of the image. An example of this approach, which involves a low-resolution or multi-resolution region oriented segmentation algorithm, will be discussed in Chapter 6. The idea of using local windows proposed by Cooper et al. (1980) and Cooper and Sung (1983), which was discussed in Section 2.4, is in fact very similar. Their main objective, however, is the possibility of processing the various windows in parallel.

A second possibility of search reduction is to limit the number of admissable values of the variables p_ℓ, $\ell=1,\ldots,L$, in the optimization problem (3.4). As an example, the curvature constraint in (3.14) used by Montanari (1971), effectively reduces the number of possible successors $p_{\ell+1}$ to three for each considered pair of points $p_{\ell-1}$ and p_ℓ. The method we propose here is to transform the image data in the arbitrarily shaped region of interest into a rectangular matrix. An example of a region of interest which is bounded by two concentric circles is given in Fig. 3.2a and the corresponding transform matrix in Fig. 3.2b. The hypothetic region boundary indicated in Fig. 3.2a corresponds with the edge in Fig. 3.2b. In this particular case, the transform

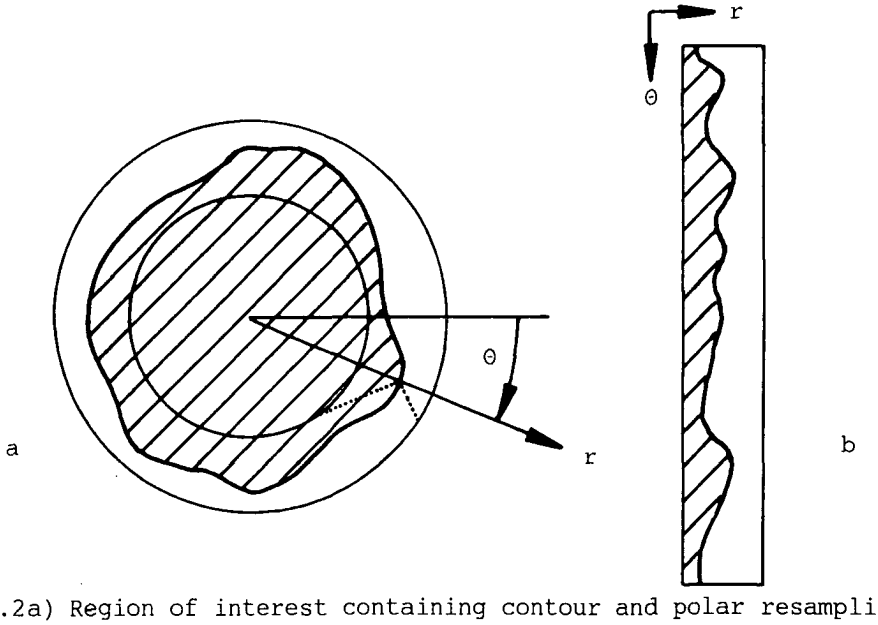


Fig. 3.2a) Region of interest containing contour and polar resampling.
b) Transform domain with corresponding edge.

involved is a straightforward polar coordinate transform of the data. The resampling process will be discussed in more detail in Section 3.3. For our discussion here, it suffices to state that it is assumed that each row of the transform matrix contains exactly one point of the transformed object region boundary. Note that this assumption is essential to the method proposed here.

Following our discussion in Section 2.3, we will consider an edge to be an 8-connected sequence of pixels in the transform domain, and as each row of the transform matrix contains exactly one boundary pixel, the number of possible successors $p_{\ell+1}$ given a hypothesized point p_ℓ is limited to three (or two, if p_ℓ is in the first or last column of the matrix). The reverse holds as well: apart from border conditions, a hypothesized point $p_{\ell+1}$ has three potential predecessors p_ℓ. This is illustrated in Fig. 3.3. The connectivity constraint in the transform domain effectively introduces a limitation regarding the curvature of the object region boundary: in the continuous case, the next boundary point is restricted to lie within a beam-shaped expectation window of $-\pi/4$ to $+\pi/4$ radians with respect to the tangent of the circle through the current boundary point. This is illustrated by the dotted lines in Fig. 3.2a.

Fig. 3.3. Each hypothesized boundary point has three possible successors (a) and three possible predecessors (b).

So, we state our problem as finding the optimal sequence of boundary points $p_1,\ldots,p_L$ in the transform matrix, where the path is known to start on the first row of the matrix and to end on the last row of the

matrix and each row contains exactly one boundary point. In other words, the values of the parameter ℓ, $\ell=1,\ldots,L$, correspond with the row numbers and the values of the variables p_ℓ correspond with the column numbers. An example of such a path is given in Fig. 3.4. The special case where the object region boundary in the image domain is a closed path will be discussed later.

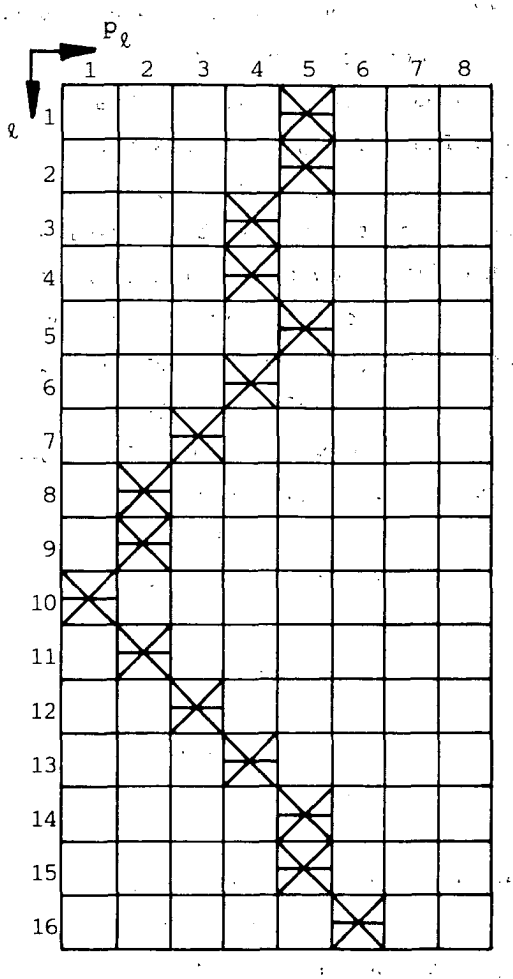


Fig. 3.4. Sequence of connected boundary pixels;
$(p_1,\ldots,p_{16}) = (5,5,4,4,5,4,3,2,2,1,2,3,4,5,5,6)$.

For the time being, we will neglect the geometrical transformation of the image data in the region of interest into a rectangular matrix and consider the transform matrix to be our input image. For our treatment of the method, we will adopt the following image model. The input image is assumed to be corrupted by signal-independent uncorrelated identically-distributed Gaussian noise $n(\ell,p_\ell)$

$$g(\ell,p_\ell) = f(\ell,p_\ell) + n(\ell,p_\ell). \tag{3.17}$$

The noise-free image $f(\ell,p_\ell)$ is modeled with a binary replacement function, i.e., it contains a connected object region with constant gray-value f_0 and a connected background region with a different but also constant gray-value f_b. The two regions are separated by a connected transient region of intermediate gray-values, as discussed in Section 2.2. This transient region contains the object region boundary, which is known to start on the first row of the image and to end on the last row of the image and each row contains exactly one boundary point.

From the gray-value image $g(\ell,p_\ell)$ we compute a matrix $c(\ell,p_\ell)$ of cost coefficients. In the context of the image model adopted here, it is quite natural to relate the cost coefficients to gradient values measured along the rows of the image $g(\ell,p_\ell)$. In terms of the original region of interest this means that we measure the gradient perpendicular to the hypothesized object region boundary. Note that this utilization of a priori information is another advantage of the method proposed here. It should also be mentioned that for the image model described, the use of gradient values in the cost coefficients results in a great similarity between our minimum-cost approach and the maximum-likelihood approach proposed by Cooper (1979), which has been outlined in Section 2.5. In view of the discussion in Section 2.3, it will not be surprising that we select an odd-sized discrete difference operator to obtain a numerical approximation of the gradient along the rows of the gray-value image. The operator can be a one-dimensional operator, say [1 0 -1], or a two-dimensional operator corresponding to, e.g. the masks of Sobel or Prewitt. If we denote its response at position

(ℓ,p_ℓ) by $g'(\ell,p_\ell)$, we now define the cost coefficient at (ℓ,p_ℓ) as

$$c(\ell,p_\ell) = Q - |g'(\ell,p_\ell)|. \tag{3.18}$$

Here, Q is again a large positive constant, which assures the cost coefficients to be positive for all values of ℓ and p_ℓ. This only serves the purpose of facilitating the shortest-path graph search formulation, as advocated by Martelli (1972). Obviously, we would arrive at the same solution if we replace the minimum-cost search in terms of $c(\ell,p_\ell)$ by a maximum-gradient search in terms of $|g'(\ell,p_\ell)|$ directly.

Following the notation of the previous section, the total cost of a path $(p_1,\ldots,p_L)$ is now defined as

$$\Phi_*(p_1,\ldots,p_L) = \sum_{\ell=1}^{L} c(\ell,p_\ell), \tag{3.19a}$$

with the connectivity constraint

$$|p_{\ell+1} - p_\ell| \leqq 1. \tag{3.19b}$$

The connectivity constraint can be incorporated into the cost functions by defining

$$\Phi(p_\ell,p_{\ell+1}) = c(\ell+1,p_{\ell+1}) \text{ if } |p_{\ell+1} - p_\ell| \leqq 1,$$

$$= \infty \qquad \text{elsewhere,} \tag{3.20}$$

for all p_ℓ, $p_{\ell+1} = 1,\ldots,P$ and all $\ell=0,\ldots,L-1$. The value $\ell=0$ corresponds with an extra stage as illustrated in Fig. 3.5, which is connected to all pixels on the first row. In terms of graph searching, this stage is the starting node. Definition (3.19) is now rewritten as

$$\Phi_*(p_1,\ldots,p_L) = \sum_{\ell=0}^{L-1} \Phi(p_\ell,p_{\ell+1}). \tag{3.21}$$

The cost function can be evaluated in terms of accumulating cost functions $\varphi(p_\ell)$ by means of the following recursion:

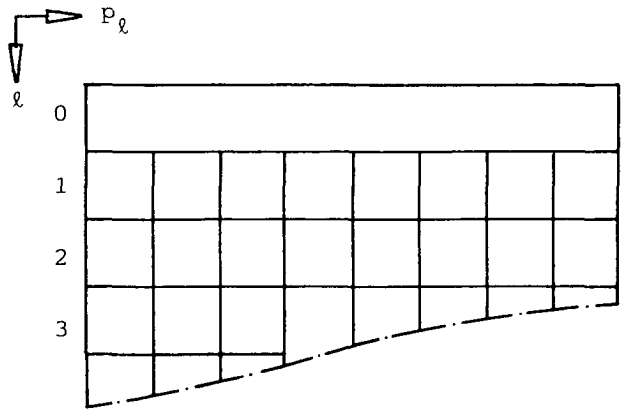


Fig. 3.5. Additional stage for ℓ=0.

$$\varphi(p_0) = 0, \tag{3.22a}$$

$$\varphi(p_{\ell+1}) = \min_{p_\ell} [\varphi(p_\ell) + \Phi(p_\ell, p_{\ell-1})],$$

$$p_\ell, p_{\ell+1} = 1,\ldots,P;\ \ell=0,\ldots,L-1. \tag{3.22b}$$

The accumulating values $\varphi(p_\ell)$ are stored in a cumulative cost matrix, which is of the same size LxP as the input image. In addition, the pointers $m_{\ell+1}(p_{\ell+1})$ for which the minimum values are achieved in (3.22b) are stored:

$$m_{\ell+1}(p_{\ell+1}) = \arg\min_{p_\ell} [\varphi(p_\ell) + \Phi(p_\ell, p_{\ell+1})],$$

$$p_\ell, p_{\ell+1} = 1,\ldots,P;\ \ell=0,\ldots,L-1. \tag{3.23}$$

Another matrix of size LxP is required to store the pointers. In fact, the first row of this matrix may be omitted because all elements point to the imaginary stage p_0. The optimal path is then found by scanning the matrix of pointers by means of the recursion:

$$p_L^m = \arg\min_{p_L} \varphi(p_L), \tag{3.24a}$$

$$p_{\ell}^{m} = m_{\ell+1}\,(p_{\ell+1}^{m}), \qquad \ell = L-1,\ldots,1. \tag{3.24b}$$

The following example illustrates the procedure.

□ Example

Consider the following gray-level matrix $g(\ell,p_{\ell})$:

$\rightarrow p_{\ell}$

$\ell\downarrow$							
	2	5	4	7	14	17	16
	5	6	6	17	12	18	17
	7	9	3	12	15	14	16
	3	5	1	7	13	16	18
	7	4	6	14	17	13	15
	5	6	8	13	17	15	14
	4	3	7	10	12	18	19
	5	4	2	6	14	16	12

$g(\ell,p_{\ell})$

Applying the operator [1 0 -1] yields $g'(\ell,p_{\ell})$:

$\rightarrow p_{\ell}$

		1	2	3	4	5
$\ell\downarrow$	1	-2	-2	-10	-10	-2
	2	-1	-11	-6	-1	-5
	3	4	-3	-12	-2	-1
	4	2	-2	-12	-9	-5
	5	1	-10	-11	1	2
	6	-3	-7	-9	-2	3
	7	-3	-7	-5	-8	-7
	8	3	-2	-12	-10	2

$g'(\ell,p_{\ell})$

The matrix of cost coefficients $c(\ell,p_{\ell}) = 15 - |g'(\ell,p_{\ell})|$ becomes:

$\rightarrow p_\ell$

$\ell\downarrow$	1	2	3	4	5
1	13	13	5	5	13
2	14	4	9	14	10
3	11	12	3	13	14
4	13	13	3	6	10
5	14	5	4	14	13
6	12	8	6	13	12
7	12	8	10	7	8
8	12	13	3	5	13

$c(\ell,p_\ell)$

The cumulative cost matrix $\varphi(p_\ell)$ and the matrix of pointers are computed recursively, starting from the top. The first row of $\varphi(p_\ell)$ is equal to the first row of the cost matrix $c(\ell,p_\ell)$ and the pointers can be omitted. The results are:

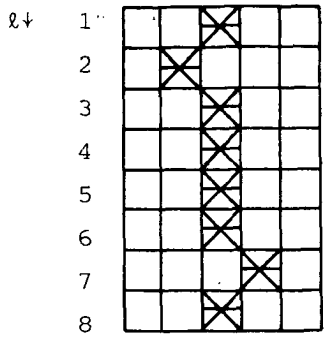
$\rightarrow p_\ell$

$\varphi(p_\ell)$ $\ell\downarrow$	1	2	3	4	5
1	13	13	5	5	13
2	27	9	14	19	15
3	20	21	12	27	29
4	33	25	15	18	37
5	29	20	19	29	31
6	32	27	25	32	41
7	39	33	35	32	40
8	45	46	35	37	45

$\rightarrow p_\ell$

$m_\ell(p_\ell)$ $\ell\downarrow$	1	2	3	4	5
1	-	-	-	-	-
2	1	3	3	4	4
3	2	2	2	3	5
4	1	3	3	3	4
5	2	3	3	3	4
6	2	3	3	3	4
7	2	3	3	3	4
8	2	2	4	4	4

The minimum of $\varphi(p_L)$ is found at p_8=3, so the first point of the boundary is the pixel (8,3). The entry in the matrix of pointers at position (8,3) is 4, which implies that the second point is the fourth pixel on the previous row, i.e., pixel (7,4). The complete path is found as (8,3),(7,4),(6,3),(5,3),(4,3),(3,3),(2,2),(1,3). The path is given below:



p_ℓ^m

■

The approach described here was originally developed for the delineation of the left ventricle in gated bloodpool cardiac scintigrams, and was first published in Gerbrands et al. (1981). Major contributions to this development can be found in Lie (1979) and Hoek (1980). This application, as well as various others, will be described in detail in Chapter 7.

The left ventricular region in bloodpool scintigrams can be described as a compact blob which is almost circular, and the polar coordinate transform is particularly suited for such blobs. However, one is interested in finding an object region boundary which is not only simply connected, but closed as well. This requirement introduces an additional connectivity constraint:

$$|p_1 - p_L| \leq 1. \tag{3.25}$$

So, the first and the last points of the boundary are required to be connected. However, this implies that the transform matrix and the derived matrices must be regarded as periodic in the ℓ-direction, which results in a non-serial optimization problem. In other words, when there is interaction between stage 1 and stage L, the decomposition of the cost or merit function (Cf. Eq. (3.5)) is not possible and the above described method cannot be applied. The formal solution to this problem is to apply the method for a predefined fixed starting point on the first row, say $p_1=k$. So, the path is forced through point k on

the first row, and is forced through an adjacent point on row L as well, i.e., Eq. (3.24a) is replaced by

$$p_L^m = \arg \min_{p_L=k-1,k,k+1} \varphi(p_L). \tag{3.26}$$

This procedure is then repeated for all possible starting points on the first row. From these P possible solutions one selects the closed path with minimal costs. However, the computation time would increase with a factor P, and in practice P may be in the order of 25. One may be willing to settle for a suboptimal solution by applying the closed path algorithm only once, by selecting one very good point on the first row. An example will be discussed in Chapter 7.

3.3. GEOMETRIC TRANSFORMATION

The proposed method of object boundary detection requires the image data in a swath around the hypothesized boundary to be transformed into a rectangular matrix. The odd-shaped region of interest should be straightened in such a way that each row of the transform matrix contains exactly one boundary pixel. In this section the geometric transformation will be discussed.

The straightening operation consists of two steps. The first step concerns the spatial transformation, which specifies the relationships between the pixel coordinates i and j in the original discrete image domain, and the coordinates ℓ and p_ℓ in the transform matrix. The general definition of the spatial coordinate transform is

$$g_t[\ell,p_\ell] = g_0[\xi_1(\ell,p_\ell),\xi_2(\ell,p_\ell)]. \tag{3.27}$$

Here, $g_0[.,.]$ denotes the original image and $g_t[.,.]$ denotes the transform matrix. The functions $\xi_1(\ell,p_\ell)$ and $\xi_2(\ell,p_\ell)$ uniquely specify the coordinate transform. In general, integral values for ℓ and p_ℓ will correspond with non-integral values for $\xi_1(\ell,p_\ell)$ and $\xi_2(\ell,p_\ell)$, whereas the discrete image $g_0[.,.]$ is known at integer coordinates i and j only.

So, the second step in the straightening operation concerns the gray-value interpolation or approximation. Various schemes will be discussed in the second half of this section.

As a final remark, it is noted that in the above we implicitly selected the so-called pixel filling approach, i.e., we start with an integer coordinate in the transform domain, compute its non-integral 'preimage' in the original domain and perform some kind of interpolation in the original image. The dual approach is called 'pixel carry-over'. There, one starts with an integer coordinate in the original image domain, computes its 'image' in the transform domain and distributes the original gray-value over the adjacent discrete positions. This more cumbersome approach will not be considered here. The reader is referred to Castleman (1979) for a general treatment of geometric transformations.

Spatial transformation

Now we turn our attention to the spatial coordinate transform, especially in relation to the desired property of the transform matrix to contain exactly one boundary point on each row. As this issue mainly concerns the geometric properties of the object region boundary, we will first look at the continuous case. So we model the boundary as a continuous curve in the original xy-domain, and we look for relationships between geometric properties of the curve and the spatial transformation. The first transformation to be discussed here is the polar coordinate transform.

Assume that the object region boundary is modeled as a circle of radius R centered at (x_c,y_c), as illustrated in Fig. 3.6a. The actual curve is not necessarily a perfect circle, and even if it is a circle, its radius and central point may deviate from the model parameters R and (x_c,y_c), respectively. So we take as a region of interest the swath between two concentric circles with common central point (x_c,y_c) and radii $R - \frac{B}{2}$ and $R + \frac{B}{2}$, respectively. This region of interest is given in Fig. 3.6b.

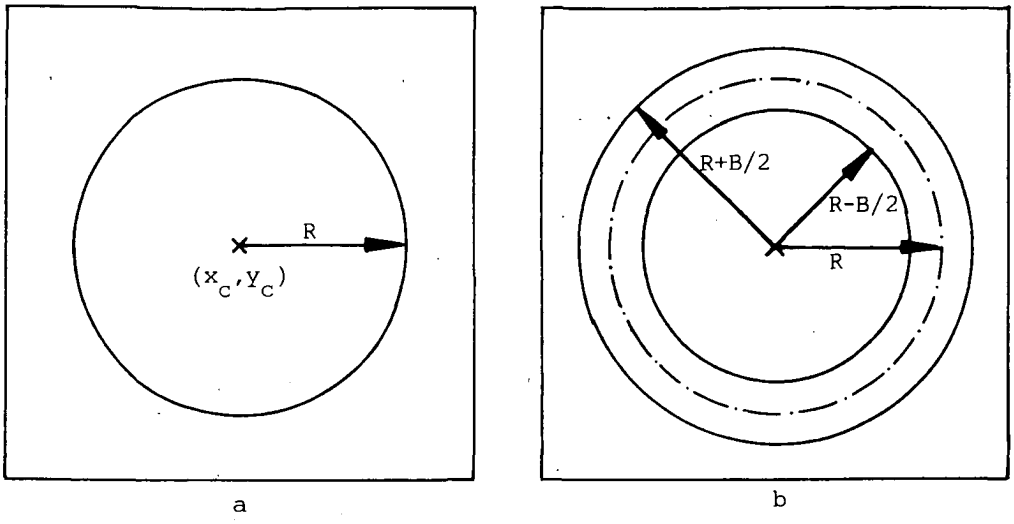


Fig. 3.6. Object region boundary (a) and region of interest (b), centered around the boundary (—. —. —. —).

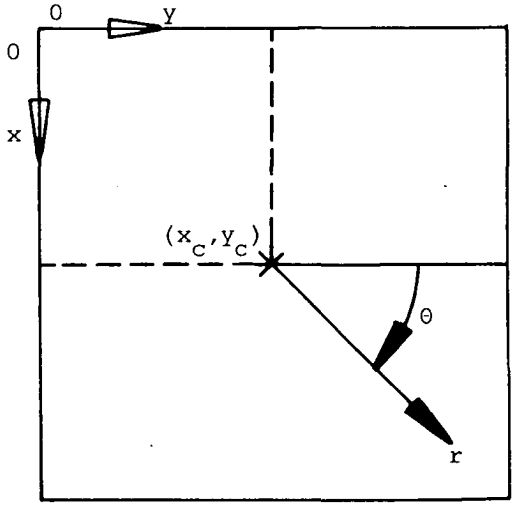


Fig. 3.7. Definition of polar coordinates.

Now let us define the polar coordinate system with respect to the xy-system as indicated in Fig. 3.7. The coordinate transform of Eq. (3.27) is then expressed as

$$\xi_1(\theta,r) = x_c + r \sin(\theta), \tag{3.28a}$$

$$\xi_2(\theta,r) = y_c + r \cos(\theta), \tag{3.28b}$$

for $0 \leq \theta \leq 2\pi$ and $R - \frac{B}{2} \leq r \leq R + \frac{B}{2}$. The variable θ corresponds with the variable ℓ in the dynamic programming formulation, and r substitutes for p_ℓ. The transformed swath is given in Fig. 3.8, together with the transform of the model circle with radius R.

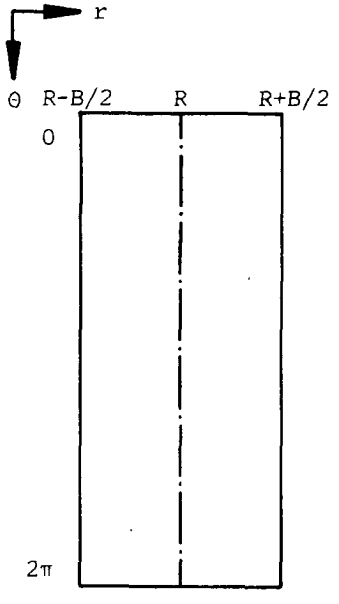


Fig. 3.8. Transformed swath from Fig. 3.6b, and the transform of the model circle with radius R (–·——·——·——).

Obviously, any circle with central point (x_c,y_c) and radius R' in the interval $[R - \frac{B}{2},R + \frac{B}{2}]$ will be transformed into a vertical straight line at r=R' in the transform domain. In addition, the transform for any circle with radius R but with a central point not coinciding exactly with (x_c,y_c) may be derived analytically (Cf. Jongeling (1987)).

However, these results are not of particular interest to our discussion here, except for the observation that for each value of θ there is only one value of r corresponding to the transformed curve. This implies that the dynamic programming method described in Section 3.2 can be applied.

The applicability of the polar coordinate transform is certainly not limited to circular object regions. The transform may be applied to any convex object region which contains the point (x_c,y_c). A common definition of convexity of planar figures is that any straight line in the plane has at most two points in common with the boundary. This implies that any straight line through any interior point of the convex region has exactly two points in common with the boundary. So we can perform the polar transformation along all halflines through (x_c,y_c), which will each contain exactly one boundary point.

The constraint of convexity is still too strict. In effect, the polar transform can be applied to any contour with the property that all halflines through the predefined point (x_c,y_c) contain exactly one boundary point. Such contours are called starshaped, as mentioned by Cohen and Boxma (1983) in their treatise of conformal mapping problems. Obviously, it may be extremely difficult or impossible to find a legitimate central point (x_c,y_c) if we are only given our noisy image. In Chapter 7 we will discuss an application where it is possible indeed, because the object regions in question are quite compact blobs, although not necessarily convex.

It may be illustrative to take an opposite point of view and consider the set of all curves which can be transformed correctly, given the central point (x_c,y_c) of the polar transformation. Without proof, it is stated that any connected region which contains (x_c,y_c) as an interior point, and which has a continuously differentiable boundary, and with the property that at no point along the boundary the tangent to the boundary contains (x_c,y_c), can be transformed with a polar transform with central point (x_c,y_c). An example is given in Fig. 3.9.

When the object region is not at least starshaped with respect to some point P, the polar coordinate transform cannot be applied. However, there are many other cases in which it is not attractive to use the

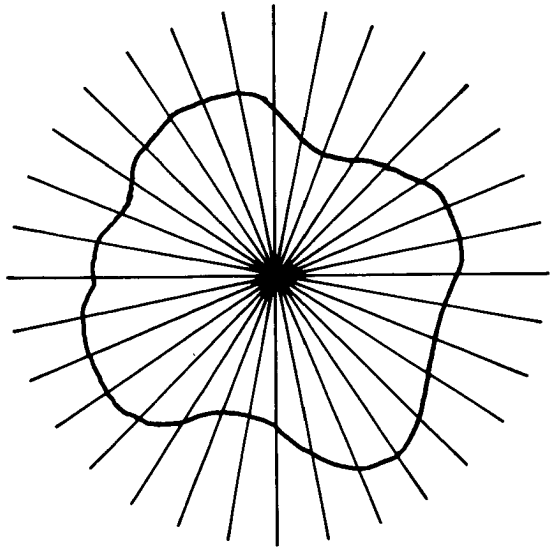


Fig. 3.9. Example of boundary of non-convex region which can be transformed with the polar transform.

polar transformation. As stated in Section 3.2, the cost coefficients in the dynamic programming optimization are computed from the magnitudes of the gradients along the rows of the transform matrix. In the image domain, we would like this to correspond with gradients measured in directions which are more or less perpendicular to the local object region boundary. Now let us take a second look at Fig. 3.6b. The region of interest given there may also be described as a swath positioned symmetrically around the hypothesized boundary. So if the object regions in question are not very compact, but quite elongated or odd-shaped instead, it would be more attractive to transform the image data in the swath by means of a spatial transformation which transforms the hypothesized boundary into a straight line. Such transformation would automatically fulfil our requirement of measuring gradients perpendicu-

lar to the hypothesized boundary. In addition, the method is no longer restricted to closed contours, but can be applied to any edge in the image, provided a suitable region of interest can be defined. The approach is illustrated in Fig. 3.10.

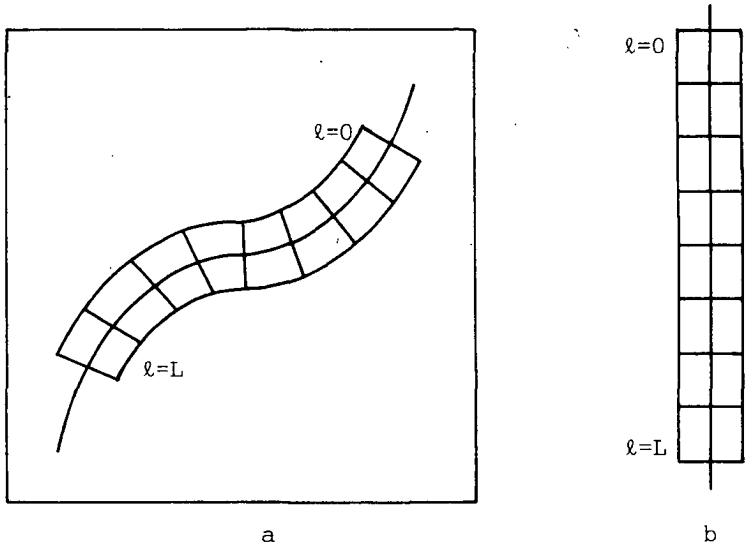


Fig. 3.10. Hypothesized boundary and region of interest in the image domain (a) and straightened region of interest by means of spatial transformation (b).

The dynamic programming method developed in Section 3.2. requires that the transformed region of interest $g_t(\ell,p_\ell)$ contains exactly one boundary point for every value of the parameter ℓ. In addition, the spatial transformation should be a one-to-one mapping between coordinates in the original image domain and in the transform domain. Locally, this is assured by limiting the width B of the region of interest to twice the minimum radius of curvature R_{min} along the curve:

$$B \leq 2.R_{min}. \tag{3.29}$$

This is illustrated in Fig. 3.11. Because the curvature C itself is

defined as the inverse of the radius of curvature,

$$C = \frac{1}{R}, \tag{3.30}$$

the width of the region of interest should be inversely proportional to the maximum curvature $C_{max} = (R_{min})^{-1}$:

$$B \leq \frac{2}{C_{max}} \quad . \tag{3.31}$$

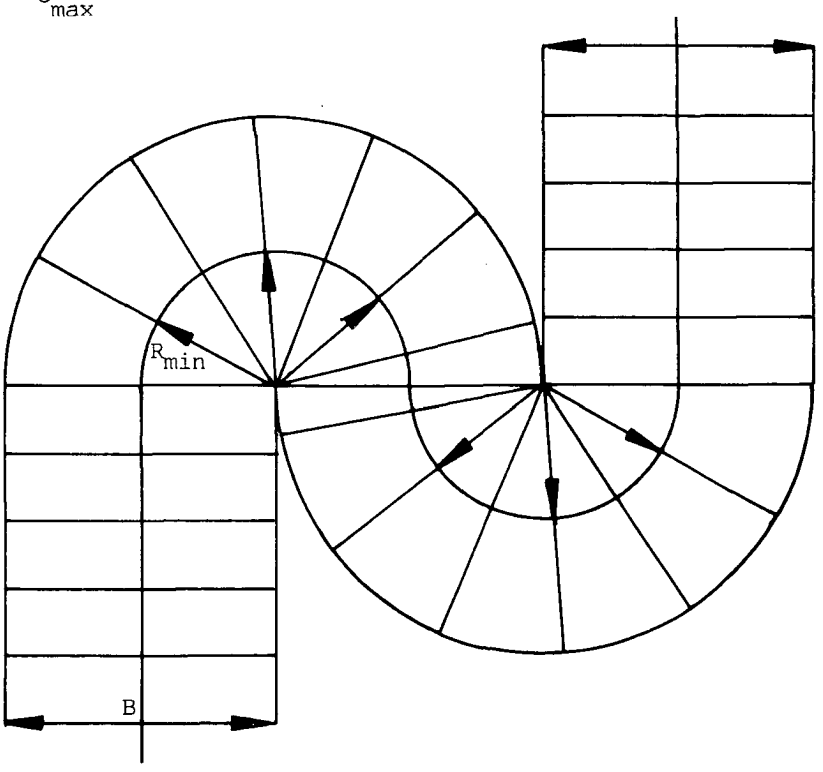


Fig. 3.11. Region of interest at section of the curve with maximal curvature for the case $B = 2(C_{max})^{-1}$.

From Fig. 3.11 it is clear that constraint (3.31) is sufficient only in the case that the path length of the section with maximal curvature is less than πR_{min}. In other words, the smallest osculating circle coincides with the curve over less than its perimeter divided by two.

There is one exception, when the curve itself is a circle with radius R_{min}, in which case the polar transform should be applied. This analysis leads to a constraint which assures a unique, non-overlapping region of interest in a global sense. If the curve itself is not a circle, and we consider two points on the curve which are separated by a path length of more than πR_{min}, than their Eucledean distance should be at least equal to the width B of the region of interest. This is illustrated in Fig. 3.12.

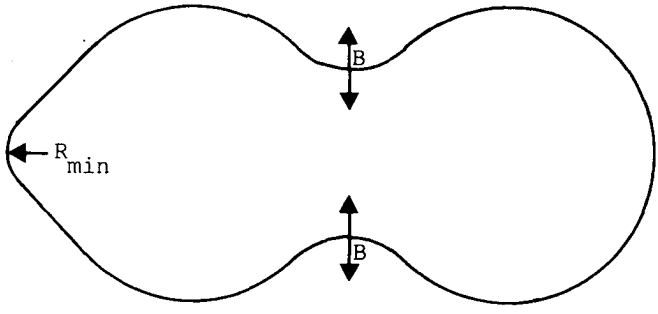


Fig. 3.12. Global constraint of non-overlapping region of interest.

Finally, it is noted that this analysis in the continuous case allows us to make a remark about the sampling density along the hypothesized boundary. In van Otterloo and Gerbrands (1978) it is proven that the sample distance along a curve should not exceed a value of $\pi.(C_{max})^{-1}$. So if C_{max} is known, as well as the length L of the curve, then the number of rows in the transform matrix should be at least equal to $L.\pi^{-1}.C_{max}$ in order to obtain a faithful representation. This result cannot be extended to the case where the image $g_0(.,.)$ itself is already discrete.

Gray-value interpolation

The second step in the geometric transformation to straighten the region of interest concerns the approximation of gray-values. As has been pointed out before, the spatial coordinate transform

$$g_t[\ell,p_\ell] = g_0[\xi_1(\ell,p_\ell),\xi_2(\ell,p_\ell)] \tag{3.32}$$

will, in general, lead to non-integral values for the functions $\xi_1(\ell,p_\ell)$ and $\xi_2(\ell,p_\ell)$, whereas the gray-values of the discrete image $g_0[.,.]$ are known at integer coordinates i and j only. So some kind of interpolation is necessary to approximate the gray-values at the real-valued coordinates [x,y], where

$$x = \xi_1(\ell,p_\ell), \tag{3.33a}$$

$$y = \xi_2(\ell,p_\ell). \tag{3.33b}$$

Theoretically, the gray-value at any coordinate [x,y] can be recovered from the sampled image if the original continuous image was sampled in accordance with the well-known sampling theorem. In the case of sampling exactly at the Shannon rate, gray-level reconstruction at intermediate positions requires the use of sinc-functions if the Fourier spectrum of the continuous image covers a rectangle in the 2-D Fourier domain. Straightforward implementation of this theoretical result would obviously be restricted to truncated sinc-functions, which is possible, but certainly not quite feasible. In the case of sampling well above the Shannon rate, the sampling theorem allows the use of filters other than the ideal low-pass filter which corresponds with sinc-function interpolation. However, direct implementation would be cumbersome in all cases.

In the image processing literature, many local interpolation schemes have been proposed with much larger practical value. The problem of interpolation is not only encountered in relationship with geometrical transformations, but also in edge detection and digital image magnification. Early contributions are due to Prewitt (1970) and Hueckel (1971). More recently, the facet model was introduced by Haralick and Watson (1981), which is explicitly used to compute the zero crossings of second derivatives by Haralick (1984). Comparative image magnification studies are reported by Troxel and Lynn (1978) as well as by Schrei-

ber and Troxel (1985). Many of the proposed schemes utilize higher-order interpolation functions, but it is well-recognized that bilinear interpolation may be sufficient in many practical cases, especially when image magnification is not the issue at hand.

In our experiments, which will be discussed in Chapter 5, we have restricted ourself to three simple interpolation schemes. The first is the well-known bilinear interpolation scheme illustrated in Fig. 3.13. In the second scheme, the interpolation is replaced by simply taking the average over the four neighboring sample points with integer coordinates. Finally, zero-order interpolation will be considered where the gray-value of the nearest neighbor sample point is assigned to the real-valued coordinate [x,y] of Eq. (3.33). In all cases, the resulting edge pixels as detected in the transform matrix are mapped back to the nearest neighbor in the original image domain.

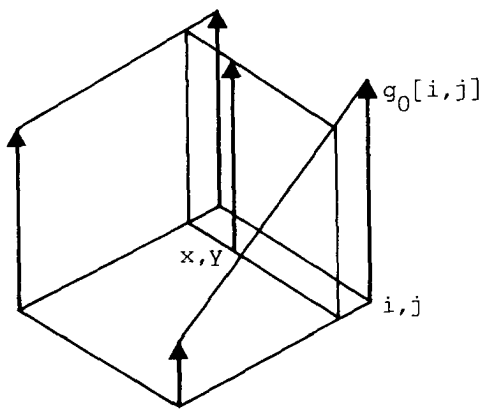


Fig. 3.13. Bilinear interpolation of gray-value $g_0[x,y]$ from four samples at integral coordinates (adapted from Castleman (1979)).

3.4. EXAMPLE

An experimental software package has been implemented on a VICOM digital image processing system. The package contains software modules

for the creation of synthetic test images, the interactive definition of a region of interest, the geometric transformation of the region of interest into a rectangular matrix, the dynamic programming optimal path procedure, the inverse geometric transformation and the quantitative evaluation of the result.

Figure 3.14 shows a synthetic test image. The image contains two regions with a difference in mean value of h. The image is corrupted by additive uncorrelated Gaussian noise with variance σ^2, independent from the signal. The signal-to-noise ratio, defined as the ratio h^2/σ^2, is equal to 1. The width of the transition region between both regions is approximately three pixels. The image size is 128 x 128 pixels. Figure 3.15 shows the interactively indicated region of interest.

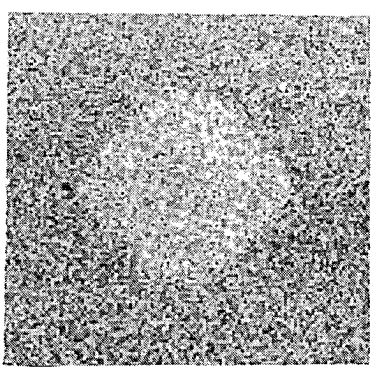


Fig. 3.14. Synthetic test image.

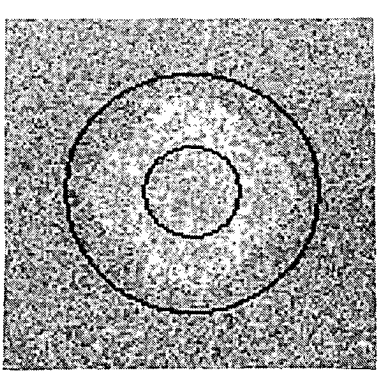


Fig. 3.15. Region of interest.

Figure 3.16 shows the region of interest straightened by the geometric transform. The gray-values are obtained by bilinear interpolation. The size of the transform matrix is 26 x 86.

Fig. 3.16. Straightened region of interest.

The cost coefficients are obtained from the gradient values using a discrete difference operator with coefficients [-1 -2 0 2 1]. The cost coefficients may be displayed as gray-values as in Fig. 3.17.

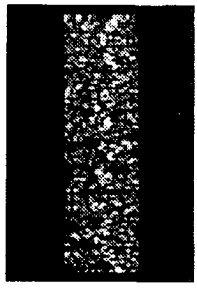


Fig. 3.17. Matrix of cost coefficients.

The result of the dynamic programming optimal path procedure is displayed superimposed in the cost matrix in Fig. 3.18, in the straightened region of interest in Fig. 3.19 and in the original image in Fig. 3.20. The edge as extracted from the noise-free original image is dis-

played in Fig. 3.21 for comparison. The quantitative evaluation of the procedure will be discussed in Chapter 5.

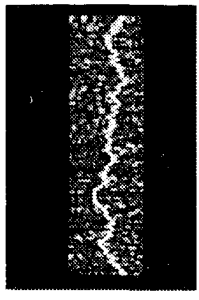


Fig. 3.18. Detected path (cost matrix)

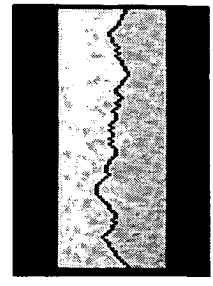


Fig. 3.19. Detected path (straightened ROI)

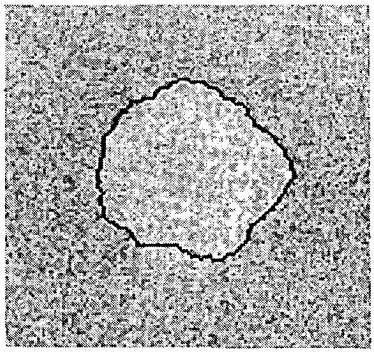


Fig. 3.20. Final boundary

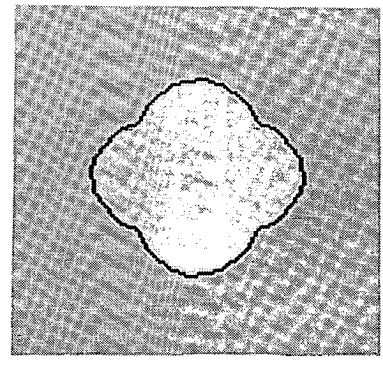


Fig. 3.21. Boundary detected in noise-free image.

3.5. CONCLUDING REMARKS

The method of object region boundary detection developed in this chapter is based on the idea of applying a dynamic programming search algorithm to a straightened region of interest. Each row of this matrix should contain exactly one boundary pixel. This greatly simplifies the dynamic programming algorithm in comparison with the case where dynamic programming is applied in the original image domain. In comparison with traditional parallel edge detection schemes, it should be noted that dynamic programming optimizes a merit function evaluated along the entire boundary, which introduces a notion of global optimality. Using the gradient values in the cost coefficients locates the desired boundary at the points of maximum gradient values, and not at points of high gradient values as in traditional parallel schemes. This makes the proposed method in a sense more related to the method of second derivative zero-crossings proposed by Haralick (1984). The use of a predefined region of interest facilitates the exploitation of problem-oriented a priori knowledge. Straightening the region of interest solves the problem that in the original image domain shorter boundaries are easily favoured over longer boundaries in a minimum-cost approach. Examples of ways to define regions of interest may be found in Chapters 6 and 7.

4. THE CONTINUOUS RELAXATION APPROACH

Image segmentation can be described as a pixel labeling or classification problem, because the ultimate goal of segmentation is to assign a label to each and every pixel. The label indicates to which one of the various image components or regions the pixel belongs. This observation implies that one may attempt to straightforwardly apply traditional techniques from the field of statistical pattern recognition to the image segmentation problem. An obvious example is image segmentation by gray-value thresholding. When a single threshold is used, the label set Λ contains two labels, λ_0 and λ_1, referring to the classes of background region pixels and object region pixels, respectively. Many threshold selection procedures have been proposed and a general treatment may be found in Rosenfeld and Kak (1982). From the point of view of statistical pattern recognition, the problem may be formulated in the framework of Bayes decision theory. If we choose the Bayes loss functions for misclassifications to be symmetric between the classes, this results in assigning a pixel the label λ_k if the a posteriori probability of λ_k, given the pixel gray-value, is maximum over all k. In the traditional approach of thresholding, these decisions are made independently on a

pixel by pixel basis. In the presence of noise, this may result in many misclassifications. In the sequel, we will describe such errors as local inconsistencies in the labeling. For the sake of completeness, note that the thresholding procedure must be followed by a connectivity analysis or component labeling procedure to identify the various connected components.

The traditional method of edge detection consists of thresholding some numerical approximation of the magnitude of the gradient. Such methods can be formulated in the same Bayes framework by considering two labels, e.g., λ_e and λ_{ne}, refering to the class of edge pixels and the class of non-edge pixels, respectively. Again, the decisions are traditionally made independently on a pixel by pixel basis, which will usually result in incomplete boundaries and numerous falsely detected edge pixels in the interior and background regions.

In many practical situations, one attempts to cope with the problem of inconsistencies in the label assignment by applying pre- or post-processing techniques. Prefiltering the image with a possibly non-linear noise suppression filter (see, e.g., Biemond and Gerbrands (1979)) may greatly facilitate the task of segmenting the image. On the other hand, given a tentative segmentation labeling, the result may be cleaned by applying cellular logic operations like dilations, erosions, openings, closings and neighborhood voting mechanisms. We will not discuss these approaches here. Instead, we will focus on mechanisms which allow the use of contextual information and a priori knowledge within the decision procedure itself.

The sequential method of dynamic programming boundary detection discussed in Chapter 3 may be viewed as an example of an approach to image segmentation which utilizes both a priori knowledge and spatial context. There, a priori knowledge is used in the definition of the region of interest and in the boundary model. Spatial context is exploited by evaluating a merit function along the entire hypothesized boundary, which introduces in a sense a notion of global optimality. In effect, all data in the region of interest are used to reach one overall decision. In this chapter we discuss parallel, iterative procedures which aim at the

same goal and are known as relaxation labeling procedures.

In Section 4.1. relaxation labeling procedures will be introduced. Section 4.2 is devoted to a feasible implementation of a relaxation labeling edge detection scheme on an image processing system featuring a spatial convolution processor. The initialization of the iterative procedure is discussed in Section 4.3. An example is given in Section 4.4, and concluding remarks are made in Section 4.5.

4.1. RELAXATION LABELING PROCEDURES

The idea of relaxation labeling procedures first appeared in the image processing literature in a paper by Rosenfeld, Hummel and Zucker (1976), although Waltz (1975) could be mentioned as an earlier source. The basic idea behind relaxation labeling procedures is that if we have a set of entities which are to be labeled, and we have a set of tentative labels for each entity, then we can attempt to utilize the relationships among the entities to find a label assignment such that a) exactly one label is assigned to each entity, and b) the labels of related entities are mutually compatible. In the present context, the entities to be labeled are the pixels in the image. However, the same mechanism can be applied to a variety of problems like scene interpretation, graph matching, histogram analysis, etc.

In the discrete case, the algorithm proposed by Rosenfeld et al. (1976) is a parallel version of the algorithm discussed by Waltz (1975) and a solution for the general consistent labeling problem formulated by Haralick and Shapiro (1979). Because some of the concepts in the continuous relaxation labeling procedure, which will be discussed later, are based on the discrete case, we will outline the discrete model first.

Let $A = \{a_1,\ldots,a_n\}$ be the set of entities to be labeled and $\Lambda = \{\lambda_1,\ldots,\lambda_m\}$ the set of possible labels. Let $\Lambda_i \subseteq \Lambda$ be the set of labels tentatively assigned to entity a_i, $1 \leq i \leq n$. For each pair of entities (a_i,a_j), where $i \neq j$, let $\Lambda_{ij} \subseteq \Lambda_i \times \Lambda_j$ be the set of compatible pairs of labels, i.e. if $(\lambda,\lambda') \in \Lambda_{ij}$ then it is possible that a_i has label λ and a_j has label λ'. By a labeling $L = \{L_1,\ldots,L_n\}$ of A we mean an assignment of a set of labels $L_i \subseteq \Lambda_i$ to each $a_i \in A$. The labeling L

is called consistent if, for all i,j, we have

$$(\{\lambda\} \times L_j) \cap \Lambda_{ij} \neq \emptyset, \qquad \forall \lambda \in L_i, \tag{4.1}$$

where $\emptyset$ denotes the null set. For $i \neq j$, this means that for each pair of entities (a_i, a_j) and each label $\lambda \in L_i$ there exists a label $\lambda' \in L_j$ such that $(\lambda, \lambda') \in \Lambda_{ij}$. Rosenfeld et al. (1976) then prove that there exists a greatest consistent labeling L^∞ such that a) L^∞ is consistent and b) for any consistent labeling L we have $L \subseteq L^\infty$, which means that $L_i \subseteq L_i^\infty$, $1 \leq i \leq n$. In addition, the authors give an algorithm to find L^∞. The algorithm starts with the initial labeling $L^0 = \{\Lambda_1, \ldots, \Lambda_n\}$. Let L^k be the labeling at the k-th iteration. To obtain L^{k+1}, discard from each L_i^k any label λ such that $(\{\lambda\} \times L_j) \cap \Lambda_{ij} = \emptyset$ for some j. In other words, the label λ at a_i is kept only if, for every a_j, there is at least one label λ' at a_j which is compatible with λ at a_i. The algorithm stops when $L^{k+1} = L^k$, which is the case if and only if L^k is consistent. The authors prove that the algorithm stops after a finite number of repetitions at the greatest consistent labeling L^∞. It should be pointed out that L^∞ may be null. Note that the algorithm stops when all compatibility constraints are satisfied. A second algorithm is then applied to L^∞ to find all unambiguous labelings, if one exists. A labeling is called unambiguous if it is consistent and assigns only a single label to each entity.

In the discrete case, a label λ is assigned to an entity a_i or it is not, i.e., either $\lambda \in L_i$ or $\lambda \notin L_i$, and the labels λ and λ' of related entities a_i and a_j are either compatible or they are not, i.e. $(\lambda, \lambda') \in \Lambda_{ij}$ or $(\lambda, \lambda') \notin \Lambda_{ij}$. The continuous relaxation labeling procedures generalize the discrete case by allowing the introduction of weights in both the label assignment and the compatibility between labels. From one point of view, the weights in the label assignment may represent fuzzy class membership, and the quantitative compatibility coefficients can be used to adjust the label weights. However, in many cases the pixel classes in the segmentation problem are considered to be mutually exclusive and exhaustive. Given the noisy image, there may be uncertainty about the

correct label assignment, but the unknown true label assignment is not fuzzy. It has become common practice in continuous relaxation labeling to treat the label assignment weights as if they were probabilities. A label λ_ℓ is assigned to pixel a_i with a weight $P_i(\lambda_\ell)$, where

$$P_i(\lambda_\ell) \geq 0, \qquad \ell = 1,\ldots,m, \tag{4.2a}$$

$$\sum_{\ell=1}^{m} P_i(\lambda_\ell) = 1. \tag{4.2b}$$

Again, the weights are adjusted by using the compatibility coefficients between labels of related pixels in an iterative updating scheme, and the adjusted weights are required to conform to (4.2a) and (4.2b) at all times. As a result, the label assignment weights behave like probabilities, but they are not necessarily based on an explicitly formulated or implicitly assumed probabilistic model. We will nevertheless often refer to the label assignment weights as probabilities. By using this terminology, we refer to $(P_i(\lambda_1),\ldots,P_i(\lambda_m))$ as the probability vector $\vec{P_i}$, and to $\{\vec{P_1},\ldots,\vec{P_n}\}$ as a probabilistic labeling L. A labeling L is called consistent when, for all a_i and λ_ℓ, $P_i(\lambda_\ell)$ equals the value of some specified function $F_{i\ell}$ of the probabilities $P_j(\lambda_1),\ldots,P_j(\lambda_m)$, for all j. Starting with an initial labeling L^0, the iterative continuous relaxation labeling scheme computes for all a_i and λ_ℓ the updated probability

$$P_i^{k+1}(\lambda_\ell) = F_{i\ell}(L^k), \tag{4.3}$$

where k is the iteration number. If this process converges to a stable limit L^∞, the resulting labeling is consistent, i.e.,

$$P_i^\infty(\lambda_\ell) = F_{i\ell}(L^\infty). \tag{4.4}$$

The notion of consistency employed here may be interpreted as follows. A labeling is called consistent when the actual label assignment weights

and the weights computed from the rest of the labeling coincide. In other words, the actual label probability of pixel a_i is equal to the same probability as estimated or predicted from the spatial context. Furthermore, let the updating algorithm stop when, for all a_i and λ_ℓ,

$$P_i^{k+1}(\lambda_\ell) = P_i^k(\lambda_\ell). \tag{4.5}$$

It then follows that if the algorithm stops, it stops at a consistent solution.

It may be very difficult to establish the convergence properties of a given updating rule (4.3), and even when convergence can be proven the limit labeling is not necessarily useful. For example, Rosenfeld et al. (1976) describe a linear updating scheme which certainly converges, but to a limit labeling which does not depend on the initial labeling L^0. In practice, updating schemes are used for a few iterations only, and formal convergence properties are not of particular importance. Instead, ad hoc updating rules have been designed which show some desired properties. The basic idea is that $P_i(\lambda_\ell)$ should be increased by the update rule if high probability labels at related pixels a_j are highly compatible with λ_ℓ at a_i. Conversely, $P_i(\lambda_\ell)$ should be decreased if other highly probable labels are not compatible with λ_ℓ at a_i. Finally, low probability labels at related pixels a_j should have little influence on $P_i(\lambda_\ell)$.

An ad hoc non-linear updating rule which shows this behavior is the original operator proposed by Rosenfeld et al. (1976). Let $C_{ij}(\lambda_\ell,\lambda_h)$ denote the compatibility between the label λ_ℓ at pixel a_i and the label λ_h at pixel a_j, with

$$C_{ij}(\lambda_\ell,\lambda_h) \in [-1,1]. \tag{4.6}$$

Here, $C_{ij}(\lambda_\ell,\lambda_h) = 1$ indicates high compatibility, a value -1 indicates complete incompatibility and a value 0 indicates indifference. One may wish to assign weight coefficients d_{ij} to the various related pixels a_j, usually depending on their distance to a_i, with

$$\sum_j d_{ij} = 1, \tag{4.7}$$

where $\sum_j$ denotes a summation over all related pixels a_j. In pixel labeling schemes, it is common practice to consider the 8-adjacent neighbors of a_i only. The label support from the related pixels a_j is now defined as

$$\Delta_{ki\ell} = \sum_j d_{ij} \sum_h C_{ij}(\lambda_\ell,\lambda_h) P_j^k(\lambda_h), \tag{4.8}$$

where $\sum_h$ denotes a summation over all labels h=1,...,m. The label probabilities, for all λ_ℓ and for all pixels a_i, are then adjusted by the update rule

$$P_i^{k+1}(\lambda_\ell) = \frac{P_i^k(\lambda_\ell)\,[1 + \Delta_{ki\ell}]}{\sum_\ell P_i^k(\lambda_\ell)\,[1 + \Delta_{ki\ell}]} . \tag{4.9}$$

Note that $\Delta_{ki\ell} \in [-1,1]$, due to Eqs. (4.2), (4.6) and (4.7). As a result we have

$$P_i^{k+1}(\lambda_\ell) \geq 0, \qquad \ell = 1,\ldots,m, \tag{4.10a}$$

$$\sum_\ell P_i^{k+1}(\lambda_\ell) = 1, \tag{4.10b}$$

which shows that the label assignment weights behave like probabilities.

Since the publication of the original paper by Rosenfeld et al. (1976), numerous authors have discussed various aspects of continuous relaxation labeling procedures. Early contributions towards a theoretical analysis are papers by Zucker (1976), Zucker and Mohammed (1978a) and Zucker, Krishnamurty and Haar (1978). Some of the attempts to incorporate relaxation labeling in a general Bayesian framework by considering the compatibility coefficients as conditional probabilities can be found in Peleg (1980) and Haralick (1983). An alternative approach is to analyze relaxation labeling procedures in terms of optimization theory. This approach is based on considering the norm of the difference between the

current probability vector $\vec{P}_i$ and the label assignment probability vector as predicted by the local neighborhood. Major contributions are the papers by Ullman (1979), Faugeras an Berthod (1981), Zucker, Leclerc and Mohammed (1981), and Mohammed, Hummel and Zucker (1983). Hummel and Zucker (1983) develop a new relaxation procedure based on variational calculus. They establish a local convergence result for this operator and show that some of the traditional relaxation updating rules are approximations to their new operator. In particular, the authors show that there is strong agreement between the ad-hoc updating rule given in Eqs. (4.6-4.9) above and the new updating rule which results from the optimization approach. As a result, we conjecture that the differences may be neglectable, especially when only a few iterations are considered, as is common practice in digital image processing.

The compatibility coefficients between the labels of related pixels have often been modeled as conditional probabilities or correlation coefficients, or in terms of the mutual information between label pairs. In the context of curve enhancement, Peleg and Rosenfeld (1978) propose to estimate such statistical quantities from one image or a set of images. However, if the images are very noisy, the estimated compatibility coefficients will be hardly reliable. In a sense, one tries to correct the unreliable data-driven label probabilities by using equally unreliable data-driven compatibilities. In the goal-oriented approach advocated here, it seems more appropriate to correct the data-driven labeling by using model-driven compatibility coefficients, chosen on the basis of a priori knowledge. This approach has also been taken by Schachter et al. (1977). Because we will use these compatibility coefficients in the sequel, they are described in detail.

Let α be the direction of the edge at pixel a_i, β the direction of the edge at pixel a_j, and γ the direction of the line joining a_i to a_j, and D the chessboard distance between a_i and a_j as defined in Eq. (3.1). Let λ_e and λ_{ne} denote the labels for the classes of edge pixels and non-edge pixels, respectively. Schachter et al. (1977) then define the following compatibility coefficients:

$$C_{ij}(\lambda_e,\lambda_e) = C_1 \cdot \cos(\alpha-\gamma) \cdot \cos(\beta-\gamma) \cdot 2^{-D}, \tag{4.11a}$$

$$C_{ij}(\lambda_e,\lambda_{ne}) = C_2 \cdot \min [0,-\cos(2\alpha-2\gamma) \cdot 2^{-D}], \tag{4.11b}$$

$$C_{ij}(\lambda_{ne},\lambda_e) = C_3 \cdot (1 - \cos(2\beta-2\gamma)) \cdot 2^{-(D+1)} \tag{4.11c}$$

$$C_{ij}(\lambda_{ne},\lambda_{ne}) = C_4 \cdot 2^{-D}. \tag{4.11d}$$

The compatibility coefficient (4.11a) expresses that parallel and perpendicular edges have no effect on one another, that collinear edges reinforce each other and that anti-collinear edges weaken each other. Similar comments can be given for the other coefficients in (4.11). In the next section, we will describe a feasible implementation of a relaxation labeling edge detection scheme based on these compatibility coefficients.

4.2. A FEASIBLE IMPLEMENTATION

The practical use of continuous relaxation labeling procedures in pixel labeling segmentation schemes is severely hampered by the large amount of computations involved. Let us assume that the initial labeling L^0 has already been determined from the input image, where

$$L^0 = \{\vec{P}_1^0,\ldots,\vec{P}_n^0\} = \{(P_1^0(\lambda_1),\ldots,P_1^0(\lambda_m)),\ldots,(P_n^0(\lambda_1),\ldots,P_n^0(\lambda_m))\}. \tag{4.12}$$

Following the traditional approach of edge detection by edge enhancement/thresholding (Cf. Abdou and Pratt (1979)), these probabilities will be computed from the response of some discrete gradient estimator. Let us furthermore assume that the compatibility coefficients $C_{ij}(\lambda_\ell,\lambda_h)$ have been predetermined on the basis of a priori knowledge. Let $||N_i||$ denote the number of directly related pixels a_j in the neighborhood N_i of pixel a_i. Computation of the updated probabilities of all m labels requires in the order of $||N_i||.(m^2+m)$ additions and $||N_i||.m^2$ multiplications per pixel a_i and per iteration k, because Eqs. (4.8) and (4.9) have to be evaluated iteratively for all labels and for all pixels. Usually, the neighborhood N_i is restricted to the set of 8-adjacent neigh-

bors of a_i and the influence of related pixels just outside this neighborhood is assumed to propagate to a_i in the subsequent iterations. If we only consider the labels λ_e and λ_{ne}, denoting the classes of edge pixels and non-edge pixels, respectively, the number of additions and multiplications is in the order of 80, per pixel and per iteration. It would thus be of great practical interest to find a feasible implementation. In this section we develop a continuous relaxation edge detection scheme which is particularly suited for implementation on a commercially available digital image processing system featuring special processors for [3x3] convolutions and point operations like addition of complete images, or multiplication of images on a pixel-by-pixel basis.

Consider the relaxation labeling procedure defined in Eqs. (4.8) and (4.9) with the constraints (4.6) and (4.7). Equation (4.8) may be rewritten as:

$$\Delta_{ki\ell} = \sum_j \sum_h d_{ij}\, C_{ij}\,(\lambda_\ell,\lambda_h)\, P_j^k(\lambda_h)$$

$$= \sum_h \sum_j d_{ij}\, C_{ij}\,(\lambda_\ell,\lambda_h)\, P_j^k(\lambda_h). \qquad (4.13)$$

In the method to be developed here, the probabilities $P_j^k(\lambda_h)$ are stored in m image memories, one for each label λ_h, h=1,...,m. If the compatibility coefficients $C_{ij}(\lambda_\ell,\lambda_h)$ are predefined spatially invariant constants, the summation over the neighbors j in (4.13) can be computed by a convolution processor, and the intermediate results for the various labels λ_h can again be stored in m image memories. Summation over the labels λ_h completes the computation of the label support term $\Delta_{ki\ell}$, and can be accomplished by a point processor. We now turn to the evaluation of the update rule:

$$P_i^{k+1}(\lambda_\ell) = \frac{P_i^k(\lambda_\ell)\,[1 + \Delta_{ki\ell}]}{\sum_\ell P_i^k(\lambda_\ell)\,[1 + \Delta_{ki\ell}]} \quad . \qquad (4.14)$$

The numerator can again be computed for all pixels a_i 'in parallel' by

using the point processor, and stored in an image memory. After this has been done for all labels λ_ℓ, $\ell=1,\ldots,m$, the denominator in (4.14) is computed. In principle, this can be accomplished by means of the point processor, just as the final evaluation of the fraction. Basically, this implementation requires in the order of $2m^2+3m$ convolutions and point operators, each of which takes less than 40 ms on an image processing system equiped with hardware convolution and point processors in combination with special buses for data transport. Furthermore, it should be noted that the above processing steps yield the updated probabilities for all labels and all pixels, i.e., for the entire image! In this analysis we neglect the unavoidable overhead due to, e.g., the loading of coefficients, and the intermediate scaling operations required to limit the data at all times to the dynamic range (pixel depth) of the image memories. The scaling operations and the limited accuracy of the hardware processors form a potential threat to the overall accuracy.

When we attempt to apply the above method to the edge detection problem, we immediately discover a flaw in the argument. The compatibility coefficients in an edge reinforcement scheme naturally depend on the directions of the edge elements at related pixels. This makes them data-dependent and thus spatially variant. As a consequence, the idea of using convolutions collapses altogether. To circumvent this problem, we propose to introduce various labels for edge pixels instead of using a single label λ_e. The various labels are indicative for the directions of the hypothesized edge elements running through the pixels. To maintain computational tractability of the relaxation procedure, the number of labels is necessarily limited, corresponding with a coarse quantization of directions. Here, we propose to subdivide the original label λ_e into four labels, λ_N, λ_E, λ_S and λ_W, corresponding with the four winds of a compass. The label λ_N (north) denotes an edge element pointing upwards along the columns of the image, with a low gray-value on its lefthand-side. Similar definitions apply to the other labels, as illustrated in Fig. 4.1. In addition, we still consider the label λ_{ne} for non-edge pixels, and the following relationships are defined:

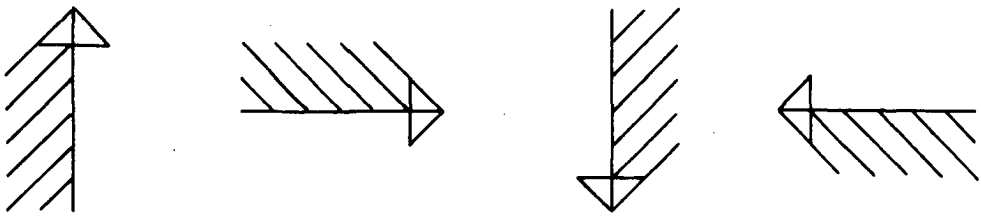


Fig. 4.1. Four subclasses of edge pixels are distinguished, labeled by λ_N (north), λ_E (east), λ_S (south) and λ_W (west).

$$P_i^k(\lambda_e) = P_i^k(\lambda_N) + P_i^k(\lambda_E) + P_i^k(\lambda_S) + P_i^k(\lambda_W), \tag{4.15}$$

$$P_i^k(\lambda_{ne}) = 1 - P_i^k(\lambda_e), \tag{4.16}$$

for all pixels a_i. In this way, the image-dependent information about the edge directions is incorporated in the probabilistic label assignment. The compatibility coefficients, on the other hand, can be defined and evaluated a priori, yielding a set of data-independent spatially invariant values. We can thus pursue the idea of utilizing spatial convolutions within the edge reinforcement relaxation labeling procedure. Furthermore, we are released from the obligation to adjust the hypothesized edge direction at each pixel by means of a separately applied edge direction relaxation procedure, as proposed by Schachter et al. (1977).

The compatibility coefficients are evaluated beforehand by using the formulas given in (4.11) except for the fact that $C_{ij}(\lambda_e,\lambda_{ne})$ and $C_{ij}(\lambda_{ne},\lambda_e)$ are chosen to be similar, as proposed by Sommen (1979). Furthermore, the multiplicative distance factors are absorbed by the weighting coefficients C_1, C_2, C_3 and C_4:

$$C_{ij}(\lambda_e,\lambda_e) = C_1 \cos(\alpha-\gamma) \cdot \cos(\beta-\gamma), \tag{4.17a}$$

$$C_{ij}(\lambda_e,\lambda_{ne}) = C_2 \cdot \min[0,-\cos(2\alpha-2\gamma)], \tag{4.17b}$$

$$C_{ij}(\lambda_{ne},\lambda_{e}) = C_3 \cdot \min[0,-\cos(2\alpha-2\gamma)], \tag{4.17c}$$

$$C_{ij}(\lambda_{ne},\lambda_{ne}) = C_4. \tag{4.17d}$$

For example, the compatibility coefficient $C_{ij}(\lambda_N,\lambda_N)$ for the righthand neighbor a_j of pixel a_i is given the value zero, because $\alpha=\pi/2$, $\beta=\pi/2$, $\gamma=0$ and Eq. (4.17a) yields the value $C_1.\cos(\pi/2).\cos(\pi/2)=0$. The complete set of compatibility coefficients is given in Fig. 4.2, without the overall weighting coefficients C_1, C_2, C_3, C_4. For obvious reasons, the schemes of Figs. 4.2b-d are rotated versions of the schemes given in Fig. 4.2a.

½	1	½	-½	0	½	-½	-1	-½	½	0	-½
0	0	0	0	0	0	0	0	0	0	0	0
½	1	½	½	0	-½	-½	-1	-½	-½	0	½
$C(\lambda_N,\lambda_N)$			$C(\lambda_N,\lambda_E)$			$C(\lambda_N,\lambda_S)$			$C(\lambda_N,\lambda_W)$		

0	-1	0
0	0	0
0	-1	0

$C(\lambda_N,\lambda_{ne})$

Fig. 4.2a. Compatibility coefficients between label λ_N at the central pixel a_i and all labels at all 8-neighbors a_j.

-½	0	½	½	0	½	½	0	-½	-½	0	-½
0	0	0	1	0	1	0	0	0	-1	0	-1
½	0	-½	½	0	½	-½	0	½	-½	0	-½
$C(\lambda_E,\lambda_N)$			$C(\lambda_E,\lambda_E)$			$C(\lambda_E,\lambda_S)$			$C(\lambda_E,\lambda_W)$		

0	0	0
-1	0	-1
0	0	0

$C(\lambda_E,\lambda_{ne})$

Fig. 4.2b. Compatibility coefficients between label λ_E at the central pixel a_i and all labels at all 8-neighbors a_j.

$-\frac{1}{2}$	-1	$-\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{1}{2}$
0	0	0	0	0	0	0	0	0	0	0	0
$-\frac{1}{2}$	-1	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
$C(\lambda_S,\lambda_N)$			$C(\lambda_S,\lambda_E)$			$C(\lambda_S,\lambda_S)$			$C(\lambda_S,\lambda_W)$		

0	-1	0
0	0	0
0	-1	0

$C(\lambda_S,\lambda_{ne})$

Fig. 4.2c. Compatiblity coefficients between label λ_S at the central pixel a_i and all labels at all 8-neighbors a_j.

$\frac{1}{2}$	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
0	0	0	-1	0	-1	0	0	0	1	0	1
$-\frac{1}{2}$	0	$\frac{1}{2}$	$-\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
$C(\lambda_W,\lambda_N)$			$C(\lambda_W,\lambda_E)$			$C(\lambda_W,\lambda_S)$			$C(\lambda_W,\lambda_W)$		

0	0	0
-1	0	-1
0	0	0

$C(\lambda_W,\lambda_{ne})$

Fig. 4.2d. Compatibility coefficients between label λ_W at the central pixel a_i and all labels at all 8-neighbors a_j.

0	-1	0	0	0	0	0	-1	0	0	0	0
0	0	0	-1	0	-1	0	0	0	-1	0	-1
0	-1	0	0	0	0	0	-1	0	0	0	0
$C(\lambda_{ne},\lambda_N)$			$C(\lambda_{ne},\lambda_E)$			$C(\lambda_{ne},\lambda_S)$			$C(\lambda_{ne},\lambda_W)$		

1	1	1
1	0	1
1	1	1

$C(\lambda_{ne},\lambda_{ne})$

Fig. 4.2e. Compatibility coefficients between label λ_{ne} at the central pixel a_i and all labels at all 8-neighbors a_j.

Note that the central elements in all sets of compatibility coefficients in Fig. 4.2 are set to zero, which implies that the label probabilities at the central pixel a_i are not allowed to strengthen or weaken themselves. The label support should be based on evidence emerging from the local context. Furthermore, note that the coefficients given in Fig. 4.2 should be scaled properly to prevent overflow. In most image processing systems it is required that the sum of the magnitudes of the coefficients in any [3x3] convolution window does not exceed a value of one. Here, the scaling is dictated by the coefficients of the window $C_{ij}(\lambda_{ne},\lambda_{ne})$ in Fig. 4.2e, yielding a scaling factor of 8 for all coefficients in all windows given in Fig. 4.2. In addition, the results must be scaled with properly chosen weighting coefficients C_1, C_2, C_3 and C_4 according to Eq. (4.17). To limit the naturally existing dominance of the non-edge label λ_{ne}, the coefficients C_3 and C_4 must be chosen considerably smaller than C_1 and C_2.

The implementation of Eqs. (4.13) and (4.14) in terms of spatial convolutions and point operations may be summarized as follows. Suppose the current label probabilities are stored in image memories denoted as $[P^k(\lambda_N)]$, $[P^k(\lambda_E)]$, $[P^k(\lambda_S)]$, $[P^k(\lambda_W)]$ and $[P^k(\lambda_{ne})]$. Initialization issues will be discussed in the next section. The computation of the numerators of Eq. (4.14) for the labels $\lambda_\ell = \lambda_N,\lambda_E,\lambda_S,\lambda_W,\lambda_{ne}$ is described by the following pseudo-code:

Algorithm 1:

FOR $\lambda_\ell = \lambda_N,\lambda_E,\lambda_S,\lambda_W,\lambda_{ne}$ DO

BEGIN

$[A]:= [P^k(\lambda_N)]*[C(\lambda_\ell,\lambda_N)]$;

$[B]:= [P^k(\lambda_E)]*[C(\lambda_\ell,\lambda_E)]$;

$[C]:= [P^k(\lambda_S)]*[C(\lambda_\ell,\lambda_S)]$;

$[D]:= [P^k(\lambda_W)]*[C(\lambda_\ell,\lambda_W)]$;

$[E]:= [P^k(\lambda_{ne})]*[C(\lambda_\ell,\lambda_{ne})]$;

$[F]:= [A]+[B]+[C]+[D]+[E]$;

$[G_\ell]:= [P^k(\lambda_\ell)]+[P^k(\lambda_\ell)].[F]$

END;

where [C(.,.)] denote the windows given in Fig. 4.2 after proper scaling, [A],...,[G_ℓ] denote image memories, * denotes spatial convolution and . denotes point-by-point multiplication. The final scaling of Eq. (4.14) is described by the following pseudo-code:

Algorithm 2:

[G]:= [G_N]+[G_E]+[G_S]+[G_W]+[G_{ne}];

FOR $\lambda_\ell = \lambda_N, \lambda_E, \lambda_S, \lambda_W, \lambda_{ne}$ DO

[$P^{k+1}(\lambda_\ell)$]:= [G_ℓ] ÷ [G] ;

where ÷ denotes point-by-point division of the entries in the image memories [G_ℓ] and [G]. Algorithms 1 and 2 constitute one iteration of the relaxation labeling procedure. After the last iteration, the edge probabilities are computed as [$P^{k+1}(\lambda_e)$] = [1] - [$P^{k+1}(\lambda_{ne})$].

When these algorithms are implemented on a commercially available digital image processing system, special attention should be given to the computation of the matrices [F], [G_ℓ] and [G] to prevent overflow. In many cases it is more attractive to compute these matrices without the use of the special hardware and image memories in order to avoid the problems of limited word-length.

4.3. INITIALIZATION

The continuous relaxation labeling procedure described in the previous section basically contains three constituting elements: the initial probabilities for the various labels, the compatibility coefficients between label pairs at related pixel pairs, and the update rule itself. Having discussed the update rule and the choice of the compatibility coefficients in the previous section, we now turn our attention to the evaluation of the initial probabilities.

Focussing on the edge detection problem, and following the discussion in Section 4.2, the matrices [$P^0(\lambda_N)$], [$P^0(\lambda_E)$],[$P^0(\lambda_S)$],[$P^0(\lambda_W)$] and [$P^0(\lambda_{ne})$], containing the initial probabilities for the various labels for all pixels, will have to be evaluated. This task can again be accom-

plished by using the convolution and point processors present in modern digital image processing systems. Based on the comparative study of Abdou an Pratt (1979), we will use a [3x3] Sobel operator to obtain estimates of the first derivatives in the horizontal and vertical directions. The magnitude $|\Delta_i|$ of the gradient vector in pixel a_i may be approximated by the sum of the magnitudes of the numerical estimates of the partial derivatives Δ_{xi} and Δ_{yi}:

$$|\Delta_i| = |\Delta_{xi}| + |\Delta_{yi}|. \tag{4.18}$$

Schachter et al. (1977) define the initial probability $P_i^0(\lambda_e)$ for the edge label λ_ℓ at pixel a_i as

$$P_i^0(\lambda_e) = \frac{|\Delta_i|}{\max_{i'} |\Delta_{i'}|}, \tag{4.19}$$

where the maximum in the denominator is taken over the gradient magnitudes in the entire image. Alternatively, one could consider the maximum over some large neighborhood of pixel a_i, at the expense of increased computational complexity. In our case, we subdivide the label λ_e into the four labels $\lambda_N, \lambda_E, \lambda_S$ and λ_W. The probabilities for the labels λ_N and λ_S are evaluated as follows. The output Δ_{yi} of the horizontal derivative operator is decomposed into its positive part and its negative part according to:

$$\Delta_{yi}^{+} = \Delta_{yi} \qquad \text{if } \Delta_{yi} > 0,$$

$$= 0 \qquad \text{elsewhere,} \tag{4.20a}$$

and

$$\Delta_{yi}^{-} = -\Delta_{yi} \qquad \text{if } \Delta_{yi} < 0,$$

$$= 0 \qquad \text{elsewhere.} \tag{4.20b}$$

This operation can be accomplished by means of the point processor, yielding intermediate matrices $[\Delta_Y^+]$ and $[\Delta_Y^-]$, respectively. The probabilities $P_i^0(\lambda_N)$ and $P_i^0(\lambda_S)$ are now evaluated as

$$P_i^0(\lambda_N) = \frac{\Delta_{yi}^+}{\max_{i'} |\Delta_{i'}|} \tag{4.21a}$$

and

$$P_i^0(\lambda_S) = \frac{\Delta_{yi}^-}{\max_{i'} |\Delta_{i'}|} \; . \tag{4.21b}$$

Again, this is done for all pixels a_i yielding the matrices $[P^0(\lambda_N)]$ and $[P^0(\lambda_S)]$. The matrices $[P^0(\lambda_E)]$ and $[P^0(\lambda_W)]$ are computed analogously on the basis of Δ_{xi}^+ and Δ_{xi}^-, respectively. Note that

$$|\Delta_{xi}| = \Delta_{xi}^+ + \Delta_{xi}^- \tag{4.22a}$$

and

$$|\Delta_{yi}| = \Delta_{yi}^+ + \Delta_{yi}^- . \tag{4.22b}$$

Substitution in Eqs. (4.18) and (4.19) leads us to the observation that

$$P_i^0(\lambda_e) = P_i^0(\lambda_N) + P_i^0(\lambda_E) + P_i^0(\lambda_S) + P_i^0(\lambda_W), \tag{4.23}$$

as required in Eq. (4.15), and the probabilities for all pixels a_i for the non-edge label λ_{ne} are evaluated by means of

$$[P^0(\lambda_{ne})] = [1] - [P^0(\lambda_e)]. \tag{4.24}$$

Basically, this completes our description of the evaluation of the initial probabilities. There are, however, two issues which need some attention.

By looking at the update rule (4.9), it is quite obvious that a zero probability for any label cannot be altered by the relaxation process. In the context of segmentation of noisy images, a zero probability for any of the labels $\lambda_N, \lambda_E, \lambda_S$ and λ_W can very well be caused by the noise. In order to facilitate the label adjustment process, all initial label probabilities are restricted to the interval $[\varepsilon, 1-4\varepsilon]$, where ε is some

small positive value, say 0.05. This can easily be incorporated in the point operators of Eq. (4.21).

The second issue is of a more fundamental nature. By defining the initial edge probabilities as in (4.19), the value $P_i^0(\lambda_e) = 1$ is assigned to the pixel(s) at which the magnitude of the gradient attains its maximum value. At all other pixels, the magnitude of the gradient is transformed into a probability by scaling $|\Delta_i|$ linearly. In effect, this implies that we are inclined to assign the non-edge label λ_{ne} to any pixel whose gradient magnitude is less than half the maximum value of the gradient magnitude occurring in the image. The scaling is defined entirely by the maximum gradient magnitude, independently from all other gradient magnitudes. A more subtle mechanism can be envisaged to derive the edge probabilities from the gradient magnitudes.

In Section 4.1, the probabilistic terminology was introduced rather carefully in relationship to the continuous label assignment weights. However, we now proceed in this direction by comparing the problem of evaluating the initial label probabilities with a two-class problem in statistical pattern recognition. The two classes are indicated by the labels λ_e and λ_{ne}, and we would like to evaluate the label probabilities, *given* the observed gradient values. In this approach, the wanted label probabilities actually play the role of a posteriori probabilities. If the class-conditional probability density functions of the gradient magnitude were known, as well as the a priori probabilities $P(\lambda_e)$ and $P(\lambda_{ne})$, we could compute the a posteriori probability for the label λ_e as illustrated in Fig. 4.3. This figure has been adapted from Duda and Hart (1973).

However attractive this approach may seem, it cannot be applied to estimate the edge probabilities on the basis of gradient magnitudes, as obtained from noisy images. The main reason for this failure is the fact that in real-world images the number of non-edge pixels greatly exceeds the number of edge pixels. In a sense, the number of background and interior pixels is proportional to the total image area N^2, and the number of edge pixels is proportional to the image size N. As a result, any observed histogram of gradient magnitude values is dominated by the back-

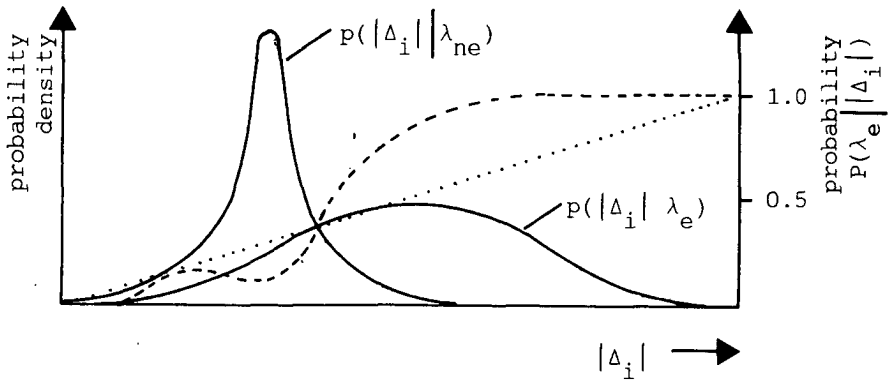


Fig. 4.3. Class-conditional probability density functions (———) of gradient magnitude, the computed a posteriori probability (-----) for the case $P(\lambda_e) = \frac{1}{3}$, and the linear scaling (.....) according to Eq. (4.19). Adapted from Duda and Hart (1973).

ground population. An example is given in Fig. 4.4. There is virtually no hope to estimate both class-conditional probability density functions and the priori probabilities from such an observed histogram. The following approximative model may give us some insight concerning the behavior of the a posteriori probability $P(\lambda_e | |\Delta_i|)$ of the label λ_e, given the observed magnitude of the gradient. Let us assume that the probability density function of the gradient magnitude for the non-edge label is triangular on the interval [0,a]:

$$p(|\Delta_i| | \lambda_{ne}) = \frac{2(a-|\Delta_i|)}{a^2} \qquad 0 \leq |\Delta_i| \leq a,$$

$$= 0 \qquad \text{elsewhere.} \qquad (4.25)$$

Furthermore, let us assume that the probability density function of the gradient magnitude for the edge label is uniformly distributed over the full dynamic range. If we set the dynamic range equal to one, we have

$$p(|\Delta_i| | \lambda_e) = 1 \qquad 0 \leq |\Delta_i| \leq 1,$$

$$= 0 \qquad \text{elsewhere.} \qquad (4.26)$$

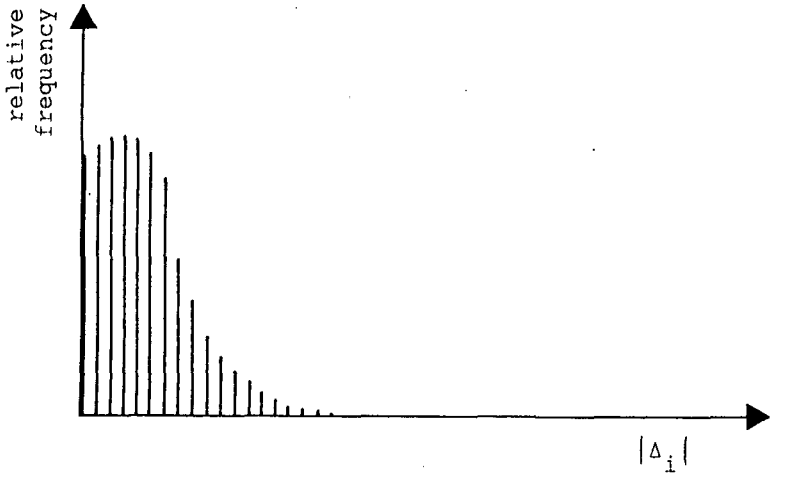


Fig. 4.4. Histogram of gradient magnitude values.

Both class-conditional probability density functions are sketched in Fig. 4.5 for the case a = 0.4.

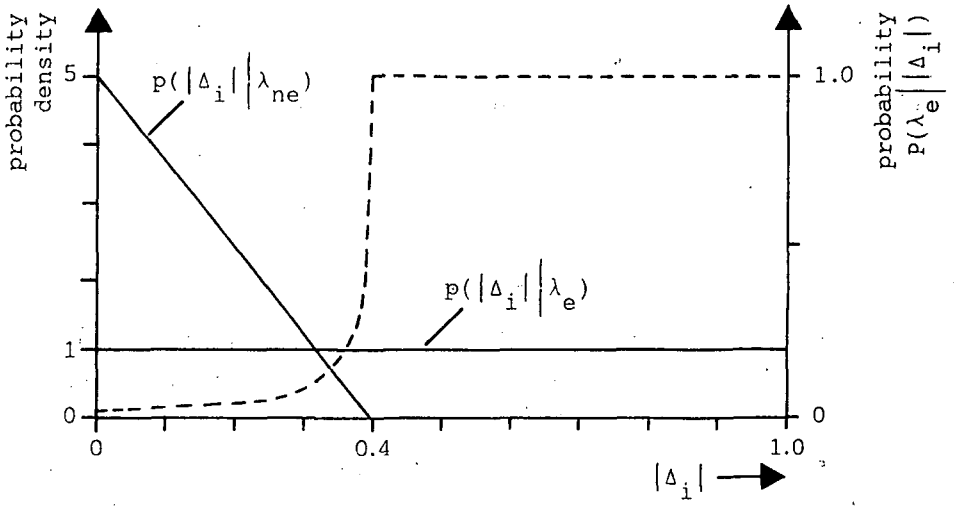


Fig. 4.5. Class-conditional probability density functions of the gradient magnitude, and the computed a posteriori probability (-----) for the case $P(\lambda_e) = 0.1$ and a = 0.4.

By applying the well-known Bayes theorem, we obtain

$$P(\lambda_e \mid |\Delta_i|) = \frac{p(|\Delta_i| \mid \lambda_e).P(\lambda_e)}{p(|\Delta_i| \mid \lambda_e).P(\lambda_e)+p(|\Delta_i| \mid \lambda_{ne}).P(\lambda_{ne})} \quad . \tag{4.27}$$

On the interval [a,1] this expression simply yields $P(\lambda_e \mid |\Delta_i|) = 1$. On the interval [0,a] we have

$$P(\lambda_e \mid |\Delta_i|) = \frac{P(\lambda_e)}{P(\lambda_e) + 2a^{-2}(a-|\Delta_i|) \; . \; (1-P(\lambda_e))} \quad . \tag{4.28}$$

The behavior of the a posteriori edge probability is given in Fig. 4.5, for the case $P(\lambda_e) = 0.1$ and $a = 0.4$.

In practice, one can attempt to estimate the parameter a and the probability $P(\lambda_e)$ from the histogram of gradient magnitude values. $P(\lambda_e \mid |\Delta_i|)$ can then be evaluated by using (4.28) and be used as the initial edge probability in the relaxation scheme. So in fact, Eq. (4.19) would be replaced by Eq. (4.28). However, we feel that we would attach too much value to the simple model introduced above. The only conclusion we draw here is that it seems worthwile to investigate the influence of an ad-hoc non-linear relationship, similar to the dotted line in Fig. 4.5, between gradient magnitude value and initial edge probability. We will return to this point in Chapter 5.

4.4. EXAMPLE

The continuous relaxation scheme is applied to the gray-value image of Fig. 4.6. The initial edge probabilities are evaluated by using Eqs. (4.18) and (4.19). The result is shown in Fig. 4.7a, where the displayed gray-values linearly represent edge probabilities, i.e., $P_i^0(\lambda_e) = 0$ is represented by black and $P_i^0(\lambda_e) = 1$ by white. Fig. 4.7b contains a binary image, resulting from thresholding Fig. 4.7a. Figures 4.8a and 4.8b show the result of the continuous relaxation procedure after one iteration. The constants in Eq. (4.17) were chosen as $C_1 = 1.0$, $C_2 = 1.0$, $C_3 = 0.5$ and $C_4 = 0.5$. The results after three iterations are given in Fig. 4.9.

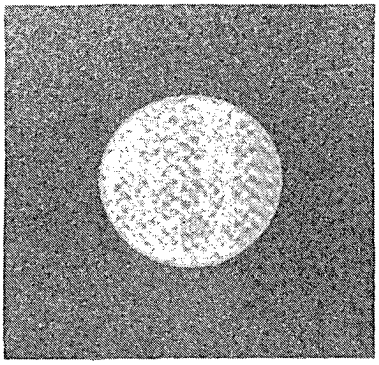


Fig. 4.6. Testimage of size [256x256], 8 bit gray-values.

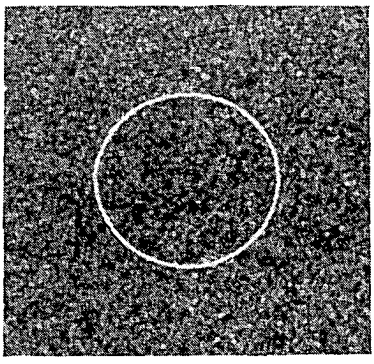


Fig. 4.7a. Initial probabilities.

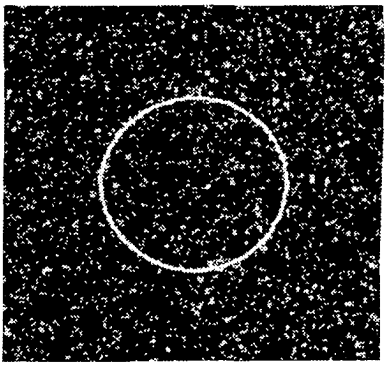


Fig. 4.7b. Result of thresholding.

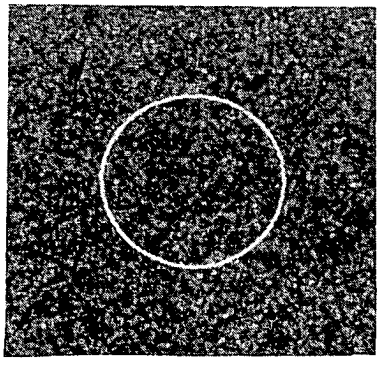


Fig. 4.8a. Result of first iteration.

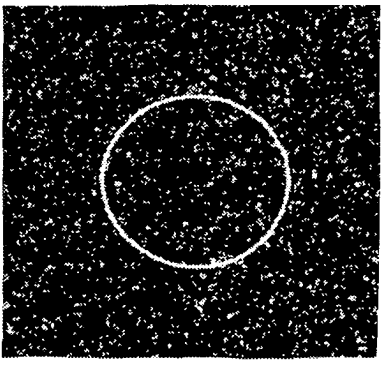


Fig. 4.8b. Result of thresholding.

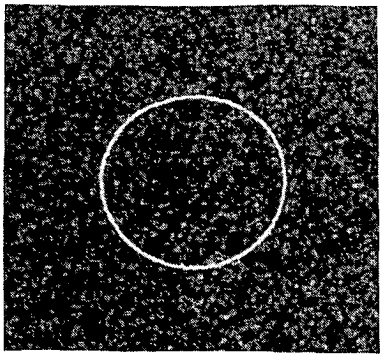


Fig. 4.9a. Result of third iteration.

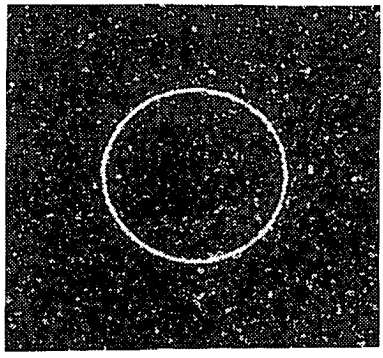


Fig.4.9b. Result of thresholding.

4.5. CONCLUDING REMARKS

Continuous relaxation labeling may be viewed as a mechanism to diminish the occurrence of segmentation errors by utilizing a priori knowledge and spatial context in an iterative and parallel updating scheme. In this scheme, weight coefficients are assigned to the various tentative labels of each pixel. These weights are adjusted on the basis of evidence emerging from the local neighborhood, expressed in terms of compatibility coefficients. In this chapter, we have taken the position that the compatibility coefficients are to be defined on the basis of generic a priori knowledge.

Straightforward application of continuous relaxation labeling procedures is severely hampered by the large amount of computations involved. In this chapter we have described a new implementation of the edge reinforcement scheme of Schachter et al. (1977). In our method, the label weights are adjusted by merely applying spatial convolutions and spatially invariant point operations, which can be performed quite efficiently on modern digital image processing systems. This is achieved basically by transfering the data dependent (and thus spatially variant) components of the compatiblity coefficients to the label weights. This necessitates the use of intermediate labels for elementary edges running in various directions. Here only four edge directions have been distin-

guished, but the method can be extended to more directions at the expense of additional computational effort.

In summary, we have reformulated the relaxation labeling procedure in a concatenation of image processing steps. Each of these steps can be applied to an entire image at video speed in state-of-the-art image processors. The approach is certainly not restricted to the problem of edge detection, but can be developed for any pixel labeling relaxation scheme, provided that the compatiblity coefficients can be made spatially invariant.

5. QUANTITATIVE EVALUATION

This chapter deals with the quantitative evaluation of the performance of the sequential dynamic programming method of boundary detection as developed in Chapter 3, and the performance of the parallel, iterative method of continuous relaxation as developed in Chapter 4.

The segmentation phase constitutes only one of many subsequent stages in a general image analysis system and its performance should ideally be evaluated in terms of overall system performance, like the accuracy of specific measures in the case of an image mensuration system, or the error probability in the case of a pictorial pattern recognition system. However, the same argument applies to all other stages of the complete system, and it is virtually impossible to vary over a number of possible methods and parameters for the various stages in order to investigate overall system performance. For instance, the dynamic programming method of object boundary detection may eventually yield a good estimate of object area, when a specific algorithm is used as an object area estimator. This does not exclude the possibility that a specific non-linear preprocessing filter, in combination with a specific thresholding segmentation

procedure and a particular area estimator eventually produces a better estimate. Within the scope of this study, however, it is only feasible to isolate the segmentation link from the complete image analysis processing chain and to evaluate its performance by studying its response to a predefined input.

5.1. METHODOLOGY

Results of the analysis and comparison of edge detector performance by using quantitative measures have been reported by Fram and Deutsch (1975,1978), Pratt (1978), Abdou and Pratt (1979), Shaw (1979) and Peli and Malah (1982). The common approach is to generate synthetic test images containing two regions of different but constant gray-levels, separated by an ideal step edge or a ramp edge. The ideal test image is then corrupted by adding signal-independent identically-distributed uncorrelated pseudo-random noise with an approximately Gaussian amplitude distribution. The noisy test image is the input of the segmentation module and its output can be compared with the original ideal image. The quantitative measures proposed by Fram and Deutsch (1975) can only be applied to images containing one vertical edge. The measure introduced by Shaw (1979) is not particularly transparent and has received little attention. Here, we follow the evaluation study of Abdou and Pratt (1979). One of the measures used is the figure of merit proposed by Pratt (1978), which is also included in the study by Peli and Malah (1982).

Abdou and Pratt (1979) evaluate the performance of parallel edge detectors of the traditional gradient enhancement/thresholding type. A continuous idealized luminance edge is defined as a planar ramp discontinuity between two regions having different but constant gray-levels. The ideal edge can be described by its position (x,y), orientation angle θ, base amplitude b, contrast h, and slope width τ, as illustrated in Fig. 5.1. Note that the gray-values within the transition region are obtained by linearly interpolating between the values b and $b+h$. The digital test image is generated by simulating an ideal spatial sampling process in combination with uniform gray-value quantization. The noise-free image can be used to study the sensitivity of any edge detector scheme to edge

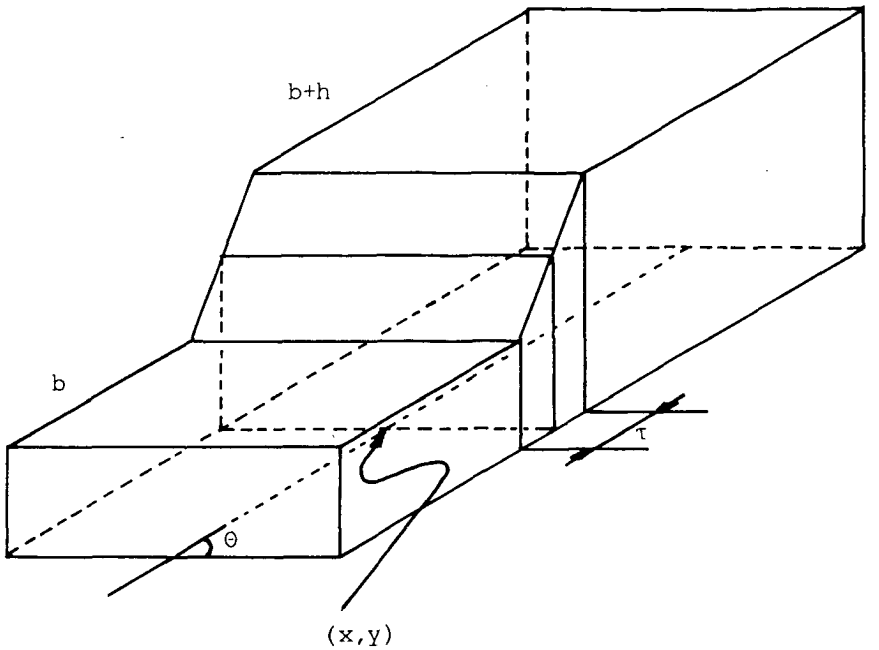


Fig. 5.1. Edge model.

position and orientation with respect to the sampling grid. Presently, we are merely interested in the performance of edge detectors in the presence of noise. The ideal digital test image is thus degraded by additive noise as described above. In the context of this discussion, the signal-to-noise ratio SNR is defined as:

$$SNR = \frac{h^2}{\sigma^2} , \qquad (5.1)$$

where σ^2 is the variance of the additive noise. Edge detector performance can now be evaluated by estimating the probabilities of correct and false edge pixel detection, similar to ROC-analysis in statistical detection theory. This is achieved by assigning a label 'true edge' (λ_{te}) to all pixels in the transition region and the label 'true no-edge' (λ_{tne}) to all other pixels in the ideal digital test image. Similarly, the pixels in the segmentation output are labeled 'detected edge' (λ_{de}) or 'detected no-edge' (λ_{dne}). The detection probability is then defined

as

$$P_D = P(\lambda_{de}|\lambda_{te}) = \frac{n(\lambda_{de},\lambda_{te})}{n(\lambda_{te})} , \tag{5.2}$$

where $n(\lambda_{de},\lambda_{te})$ denotes the number of correctly detected true edge pixels, and $n(\lambda_{te})$ denotes the total number of true edge pixels. A similar notation is used in the definition of the false positive probability:

$$P_F = P(\lambda_{de}|\lambda_{tne}) = \frac{n(\lambda_{de},\lambda_{tne})}{n(\lambda_{tne})} . \tag{5.3}$$

It should be noted that P_D and P_F can also be obtained analytically in the case of an ideal step edge corrupted by additive Gaussian noise for the traditional gradient/thresholding edge detector schemes (Cf. Abdou (1978)). This is not possible for the dynamic programming boundary detector and the continuous relaxation labeling method, where P_D and P_F can only be evaluated experimentally.

The probabilities P_D and P_F are useful performance measures for edge detectors, especially when we would find P_D=1 and P_F=0. These measures, however, do not distinguish between the various types of errors that may occur. Suppose a specific edge detector yields a large number of detected edge pixels scattered throughout the image plane. This may result in P_D=0 and P_F=1. The same values would be obtained for an edge detector which yields a connected edge parallel to the original edge, but shifted just outside the transition region. The latter result is evidently much more attractive for further processing. The figure of merit (FOM) of Pratt (1978) distinguishes between the various types of errors by taking into account the error distance d(i) between the position of the detected i-th edge pixel and the straight line representing the original edge:

$$FOM = \frac{1}{MAX\{n(\lambda_{te}),n(\lambda_{de})\}} \sum_{i=1}^{n(\lambda_{de})} \frac{1}{1+\gamma d^2(i)} , \tag{5.4}$$

where $n(\lambda_{te})$ and $n(\lambda_{de})$ are the number of true and detected edge pixels,

respectively, and γ is a scaling constant chosen to be $\gamma = \frac{1}{9}$ to provide a relative penalty between smeared edges and isolated, but offset, edges. Note that a value for the figure of merit FOM=1 will be achieved when all true edge pixels have been detected correctly and no spurious edge pixels occur. When not all of the true edge pixels have been detected, the scaling term $MAX\{n(\lambda_{te}),n(\lambda_{de})\}$ results in a lower value of FOM, even without the occurrence of spurious edge pixels. When spurious edge pixels are detected, the FOM assumes a value $0 < FOM < 1$, due to the distances d(i).

Abdou and Pratt (1979) have evaluated the performance of traditional gradient-type edge operators (Sobel, Prewitt, etc.) on the basis of test images containing either one vertical edge or one diagonal edge. The considered signal-to-noise ratios, as defined in Eq. (5.1), ranged from SNR=1 to SNR=100. In the sequel, we will apply their evaluation method to the dynamic programming boundary detector and to the relaxation labeling procedure.

5.2. EVALUATION OF DYNAMIC PROGRAMMING

The quantitative evaluation of the dynamic programming method of boundary detection to be discussed in this section is based on a study of van der Hoeven (1985) and has been the subject of a paper by Gerbrands et al. (1986a).

The dynamic programming method of boundary detection requires the definition of a region of interest (ROI). In practical applications this can often be achieved by using some automated procedure, as will be discussed in Chapters 6 and 7. In our experiments we indicated the region of interest interactively by entering two points of the straight midline of a rectangular ROI, roughly parallel with the true edge. The images containing a vertical edge are of size [64x64]. In the case of a diagonal edge, the test images are of size [128x128], but the analysis is restricted to a central window as indicated in Fig. 5.2. This window contains again 64 edge pixels. In both cases, the image data in the ROI are transformed into a rectangular matrix of size [64x25]. The grid constants of the original image and the transformed image are the same. In

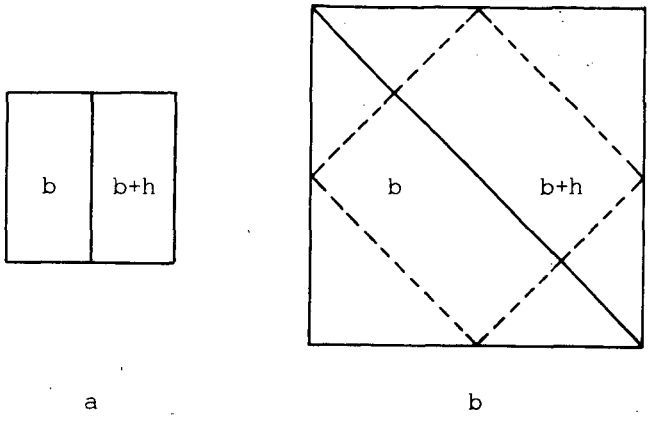


Fig. 5.2.a. Lay-out of test image containing vertical edge.
b. Lay-out of test image containing a diagonal edge and evaluation window (---).

the geometric transformation, three interpolation schemes have been considered: zero-order or nearest neighbor 'interpolation', unweighted averaging of the nearest four pixel values and true linear interpolation of the nearest four pixels. Because of its success in many medical applications to be discussed later, the cost coefficients are computed by considering both the first-order and second-order discrete differences along the rows of the transform matrix. The cost coefficients are defined as the inverse of a convex combination of the magnitudes of the first-order difference g'(i,j) and the second-order difference g"(i,j):

$$C(i,j) = M - \alpha|g'(i,j)| - (1-\alpha)|g''(i,j)|, \qquad (5.5)$$

where M is some large positive constant. We have experimented with various difference operators:

$$\frac{1}{6}\,[-2 \quad -1 \quad 0 \quad 1 \quad 2], \qquad (5.6a)$$

$$\frac{1}{6}\,[-1 \quad -2 \quad 0 \quad 2 \quad 1], \qquad (5.6b)$$

$$\frac{1}{8}\,[-1 \quad -3 \quad 0 \quad 3 \quad 1], \tag{5.6c}$$

$$\frac{1}{2}\,[-1 \quad 0 \quad 1], \tag{5.6d}$$

and with various values for the coefficient α. The dynamic programming optimal path algorithm is applied to the [64x17] central part of the cost matrix only, because the boundary strips contain unreliable data due to the convolution with the discrete difference operators. The detected path is transformed back to the original image domain by using nearest pixel position assignment. Connectivity is not assured in the image domain, and possible holes are closed by linear interpolation. The final result is then evaluated in terms of the detection probability P_D, given in (5.2), the false positive probability P_F, given in (5.3), and Pratt's figure of merit FOM, defined in Eq. (5.4) above. These quantitative measures can be used to select the optimal choices for the interpolation method, the discrete difference operator and the weighting coefficient α in the cost function (5.5). Again, it is virtually impossible to consider all combinations. One has to settle for a reasonable number of experiments, yielding a possibly suboptimal selection. In all experiments, the width of the continuous transition region was chosen equal to the pixel size in the images.

Selection of difference operator and α

Experiments were carried out for the vertical edge test image, nearest neighbor interpolation and one realization of the noise field only, with SNR-values 1, 5, 10, 20 and 50, to select the best choice for the discrete difference operator from the set given in (5.6a-d), and the best value for the weighting coefficient α in the cost function (5.5). In the first series of experiments, the ROI was aligned perfectly with the true edge. Based on the computed values of P_D, P_F and FOM, the filters given in (5.6a) and (5.6d) were discarded. The best choices for the coefficient α for the two remaining filters were α=1.0 for filter (5.6b) and α=0.9 for filter (5.6c) . In the second series of experiments we introduced an angle between the midline of the ROI and the true edge.

Angles between 4 and 14 degrees were considered using the best choices for α from the previous experiment. As an example, for SNR=1, the FOM-value for the filter given in (5.6b) decreased almost monotonically from FOM=0.77 to FOM=0.71 with increasing angle. For the filter (5.6c) the range is FOM=0.75 to FOM=0.40. For larger values of the signal-to-noise ratio, the difference between both filters is less obvious, but the preference for the filter [-1 -2 0 2 1] is quite strong. Its accompanying value for the weighting coefficient is α=1.0. The latter result indicates that the cost coefficients depend on the first-order discrete differences only. In a sense, this points towards a relationship between the minimum cost approach and the maximum likelihood approach discussed shortly in Chapters 2 and 3. In the maximum likelihood approach, the likelihood functions are obtained from the sums of the pixel values on either side of the hypothesized edge. The first-order discrete difference operator does something very similar, but considers equal numbers of pixels on either side. This weak relationship only holds in the case of additive Gaussian noise.

Selection of interpolation method

The second issue to be discussed here is the choice of the interpolation method in the geometric transformation of the region of interest prior to the computation of the cost coefficients. Three interpolation schemes have been considered: zero-order or nearest neighbor interpolation, unweighted averaging of the values of the nearest four pixels, and true linear interpolation of the values of the nearest four pixels. All experiments were carried out with the filter [-1 -2 0 2 1] and the value α=1.0 in the cost function (5.5). The first experiment concerned the test image with the vertical edge, and a perfectly aligned ROI. The experiment was performed for five different realizations of the additive noise field. The signal-to-noise ratio was equal to one in all cases. Table 5.1 gives the measured mean values and standard deviations of the FOM, P_D and P_F for each of the considered interpolation schemes.

The entire experiment was then repeated for regions of interest which

were not in perfect alignment with the original edge. Table 5.2. contains the observed mean values and standard deviations of the FOM, P_D and P_F for an alignment error of 14 degrees, as illustrated in Fig. 5.3.

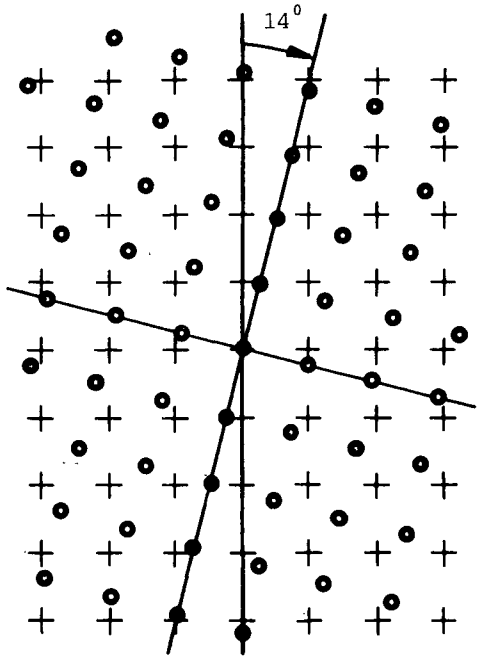


Fig. 5.3. Resample positions (⊙) with respect to the original pixels (+) for an alignment error of 14 degrees.

Alignment errors of 3.7, 7.2, 10.6 and 14.0 degrees were considered. Table 5.3. contains the average values of the means of FOM, P_D and P_F, obtained by averaging over all five angles ranging from zero to 14 degrees.

In almost all experiments the zero-order or nearest neighbor interpolation scheme laggs somewhat behind the other schemes, conformable to our intuition. A comparison between the other schemes apparently favors the unweighted averaging of the nearest four neighbors above the true bilinear interpolation scheme. This counter-intuitive result can be explained by observing that unweighted averaging has, in general, a larger noise-suppression effect than true interpolation. In the case of a per-

FOM		P_D		P_F		interpolation scheme
mean	s.d.	mean	s.d.	mean	s.d.	
0.80	0.10	0.30	0.13	0.01	0.00	zero-order
0.89	0.04	0.36	0.06	0.01	0.00	average
0.79	0.10	0.29	0.13	0.01	0.00	linear interpolation
perfectly aligned ROI						SNR=1

Table 5.1. FOM, P_D and P_F for various interpolation schemes over five noise realizations with a perfectly aligned ROI.

FOM		P_D		P_F		interpolation scheme
mean	s.d.	mean	s.d.	mean	s.d.	
0.63	0.13	0.21	0.14	0.01	0.00	zero order
0.73	0.10	0.25	0.08	0.01	0.00	average
0.67	0.10	0.21	0.10	0.01	0.00	linear interpolation
misalignement of ROI by 14 degrees						SNR=1

Table 5.2. FOM, P_D and P_F for various interpolation schemes over five noise realizations with ROI misaligned by 14 degrees.

FOM	P_D	P_F	interpolation scheme
average mean	average mean	average mean	
0.74	0.26	0.01	zero order
0.78	0.26	0.01	average
0.76	0.26	0.01	linear interpolation
average over five angles			SNR=1

Table 5.3. Average values of mean FOM, P_D and P_F over five angles of ROI ranging from 0°-14°.

fectly aligned ROI, the resampling positions coincide with the original pixel positions, and bilinear interpolation becomes identical with zero-order interpolation. The slight discrepancies in Table 5.1 originate from the fact that in one of the experiments a different noise realization was used. Other evidence supporting the noise-suppression argument is given by the observation that in the case of a misaligned ROI true interpolation performs better than zero-order interpolation, as can be seen in Table 5.2, and that the performance of true interpolation exceeds the performance of unweighted averaging at higher values of the signal-to-noise ratio. For example, an experiment over only one noise realization with SNR=20 and five different angles of the midline of the ROI yielded an average value of the FOM of 0.95 for zero-order interpolation, a value of 0.92 for unweighted averaging and a value of 0.96 for true interpolation.

The score of the detection probability P_D follows the same pattern as the score of the figure of merit. The false positive probability P_F is extremely low in all cases. This is caused by the fact that the dynamic programming method of edge detection developed here, always detects a prescribed number of edge pixels in the transform domain, independent from the signal-to-noise ratio. In the traditional approach of gradient magnitude thresholding, P_F is greatly influenced by the selection of the threshold. In our case, the relatively small number of false positives is divided by the large number of 'true no-edge' pixels in order to obtain an estimate of P_F according to Eq. (5.3). This attractive property of the dynamic programming approach is a rather obvious consequence of the fact that we confine the algorithm to search for a single connected boundary within the region of interest. In the sequel, we will concentrate on performance evaluation in terms of the figure of merit.

From these experiments it is concluded that first-order interpolation by unweighted averaging of the values of the nearest four pixels is the method of choice to be used in the geometric transformation of the image data within the region of interest, particularly for low values of the signal-to-noise ratio.

FOM-evaluation: vertical edge

The test image containing a vertical edge was corrupted by additive noise with values of SNR equal to 1, 5, 10, 20, 50 and 100. For SNR=1 and SNR=5, five noise realizations were considered and the resulting FOM-values averaged. In all cases, five orientation angles of the interactively indicated region of interest were considered and the results averaged. The final values are represented in Fig. 5.4 as dots connected by the drawn line. For purpose of comparison, the FOM-values of the traditional Sobel operator are also given in Fig. 5.4, connected with the dotted line. These values were obtained by optimizing the decision threshold with respect to the resulting value of the figure of merit, and correspond with the results reported by Abdou and Pratt (1979).

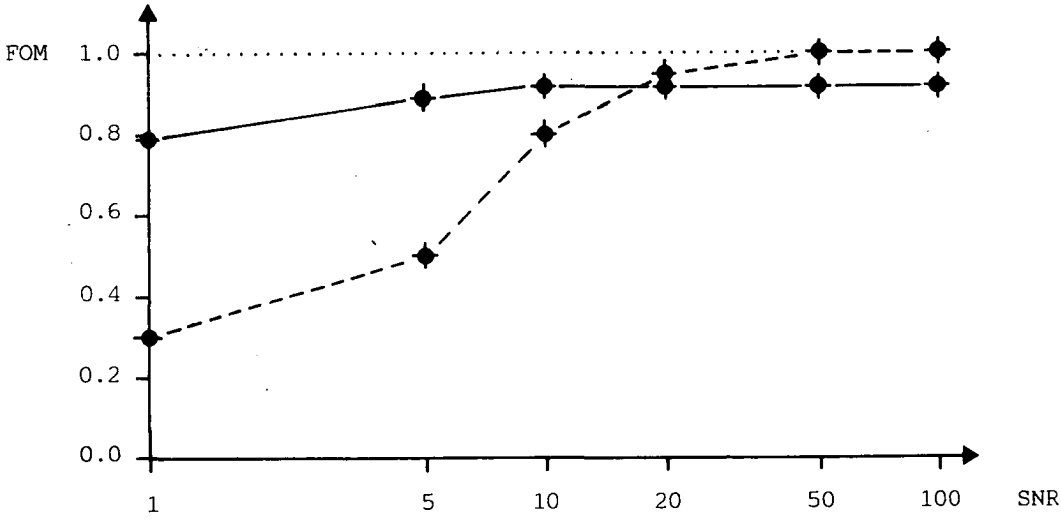


Fig. 5.4. FOM-values for the dynamic programming approach (——) and the Sobel square root operator (----) for various values of SNR, vertical edge.

FOM-evaluation: diagonal edge

The experiments described above were repeated for a test image containing a diagonal edge. The results, averaged over four orientation angles of the interactively indicated region of interest, are given in

Fig. 5.5. The quantitative analysis is restricted to a [64x64] diagonal window as illustrated in Fig. 5.2b.

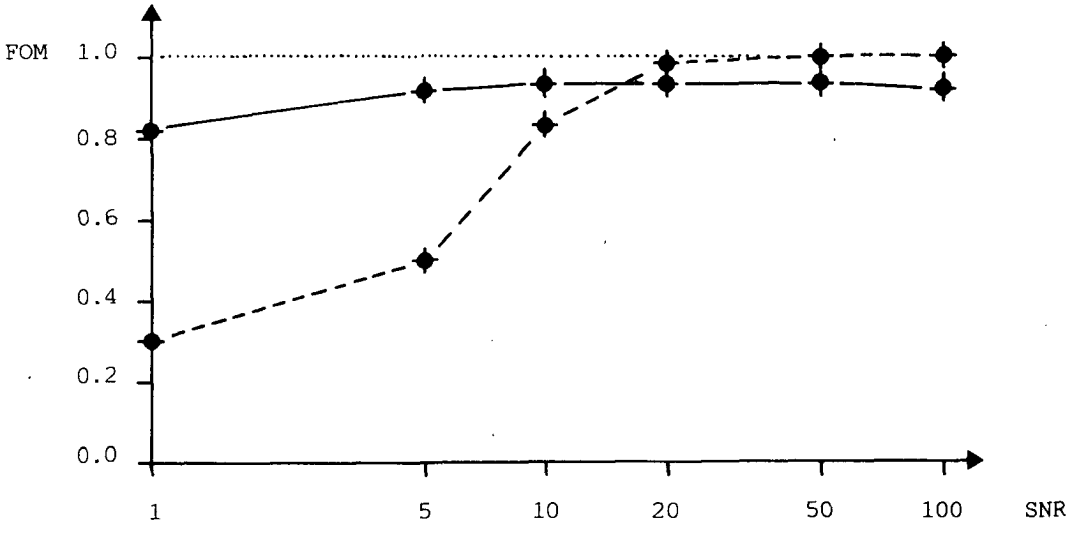


Fig. 5.5. FOM-values for the dynamic programming approach (——) and the Sobel square root operator (----) for various values of SNR, diagonal edge.

FOM-evaluation: circular boundary

The experiments were repeated for a synthetic test image of size [64x64] containing a circular region with gray-value b+h and with center (32,32) and radius 10, surrounded by a background with gray-value b. The digitization of the continuously defined original image was simulated, and the digital image was corrupted by additive noise with SNR-values of 1, 5, 10, 20, 50 and 100. For SNR=1 and SNR=5, five noise realizations were considered and the results averaged. In all cases, three different ways were considered to define a circular region of interest, and the results were averaged. The first case concerns a region of interest centered at the true center at position (32,32), with radius 20. In the second case, the radius of the region of interest was increased to 30.

In the third case, a region of interest of radius 20 was defined, but with an eccentric midpoint at position (32,34). In all cases, the image data within the region of interest were geometrically transformed by using a polar transformation along a number of radii. On the basis of experimental evidence, the number of radii was set to 2R, where R is the radius of the ROI in terms of the grid constant. Supporting evidence for this choice can be found in Jongeling (1987) in the context of object measurements. After detecting the optimal path in the polar domain, the edge pixels were transformed back to the original image domain and a closed boundary was obtained by linear interpolation.

There is a slight complication in the definition of the number of true edge pixels $n(\lambda_{te})$ used in the evaluation of the figure of merit (5.4). In the previous cases, this number was obtained by counting the number of pixels in the transition region in the noise-free digital test image, i.e., the number of pixels with a value not equal to either b or b+h. In the case of a circular object region with a continuously defined transition region width equal to the grid constant, this would lead to a rather large value of $n(\lambda_{te})$. As a consequence, a one pixel wide connected boundary within the digital transition region would yield a rather low value of the figure of merit, because not all true edge pixels were detected. Instead, we define $n(\lambda_{te})$ to be the minimum number of true edge pixels which constitute a closed boundary.

The experimental results for the test image containing the circular object region are given in Fig. 5.6.

FOM-evaluation: amended cost function

When the geometric transformation precisely matches with the shape of the object region, the object region boundary becomes a straight line in the transform domain. In the more general case, the transformed boundary deviates from a straight line. An obvious example is a circular object region boundary in combination with a polar transformation. The appearances of the true object region boundary are given in figure 5.7 for the cases of a correctly centered and an incorrectly centered polar origin. Analytical expressions for such curves have been derived by Jonge-

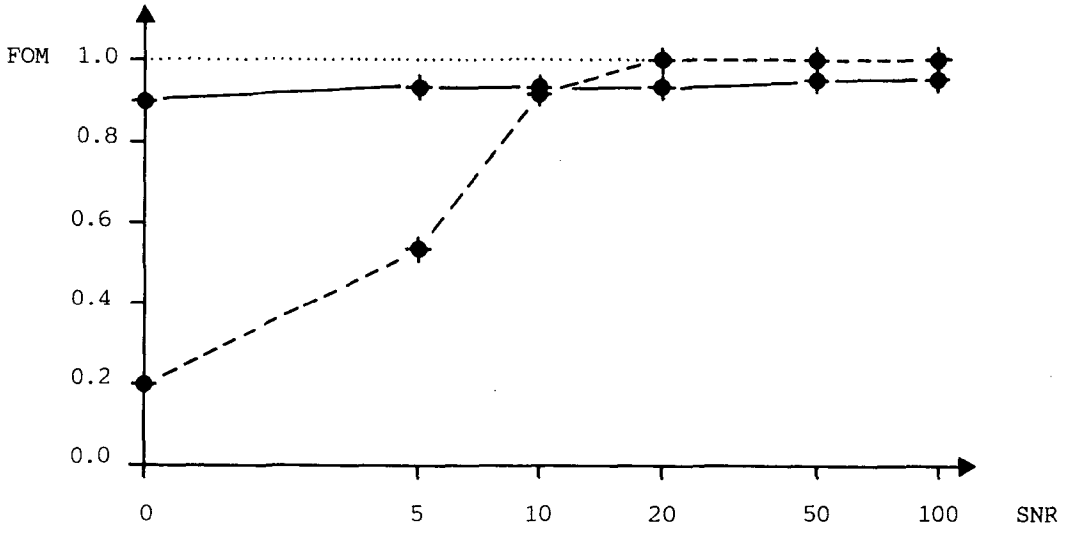


Fig. 5.6. FOM-values for the dynamic programming approach (——) and the Sobel square root operator (----) for various values of SNR, circular object region.

ling (1987).

If one has great faith in the definition of the ROI, one may incorporate a bonus for straight paths in the dynamic programming cost matrix. Similar to the use of Markov transition probabilities as described in Chapter 3, this can be achieved by introducing extra penalties for both diagonal predecessors. The cost function defined in (3.20) is amended in the following way:

$$\begin{aligned} \Phi'(p_\ell,p_{\ell+1}) &= c(\ell+1,p_{\ell+1}) && \text{if } p_{\ell+1}-p_\ell=0, \\ &= c(\ell+1,p_{\ell+1}) + c_d.M && \text{if } |p_{\ell+1}-p_\ell|=1, \\ &= \infty && \text{elsewhere,} \end{aligned} \tag{5.7}$$

for all $p_\ell,p_{\ell+1}=1,\ldots,P$ and all $\ell=0,\ldots,L-1$, where M is the same large positive constant as used in Eq. (5.5.), and $0 \leq c_d \leq 1$. Experiments

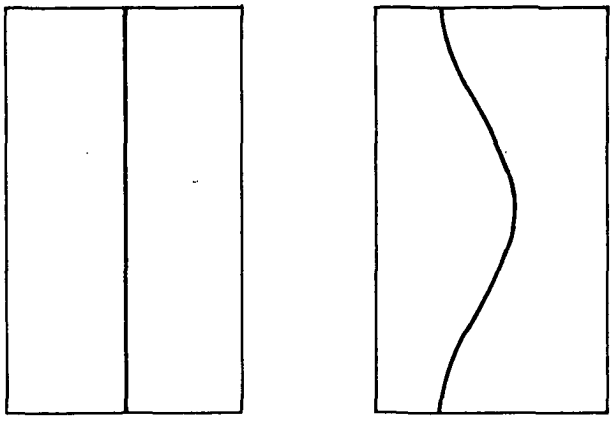


Fig. 5.7. Transformed object region boundaries for a concentric (a) and an eccentric (b) polar origin.

have been performed for the test image containing a circular object region, as described above, and values of the signal-to-noise ratio SNR=1 and SNR=2. All three interpolation schemes were considered, as well as various offsets between the centerpoint of the object region and the polar transform origin. For a detailed description we refer to Conijn (1987). Here, we mention the main results.

In the case of a perfectly centered ROI, the experiments led to the predictable observation that a value of the figure of merit FOM=1.00 can be achieved by any interpolation scheme when the value of the weighting coefficient c_d is increased sufficiently. In fact, for c_d=1, the dynamic programming optimal path algorithm constitutes an unnecessarily complicated line fitting algorithm.

In the case of an eccentric ROI, it was found that the unweighted averaging interpolation scheme performed slightly better than true bilinear interpolation. In general, a minor improvement of the performance is achieved for small values of the weighting coefficient c_d. Increasing the value of c_d even further results in a decrease of the performance. In that case, the preference for straight segments is too strong. As an example, the results for an offset of four pixels of the polar origin

are given in Table 5.4, in terms of the mean and standard deviation of the figure of merit, computed on the basis of five noise realizations, with SNR=1.

It is concluded that the introduction of a bonus for straight segments is not very useful.

c_d	FOM	
	mean	s.d.
0.00	0.89	0.05
0.05	0.91	0.04
0.10	0.91	0.04
0.15	0.90	0.05
0.20	0.90	0.05
0.25	0.89	0.05
0.30	0.89	0.05
0.35	0.89	0.05
0.40	0.89	0.06
0.45	0.88	0.07
0.50	0.88	0.07
0.60	0.77	0.14
0.70	0.67	0.18
0.80	0.65	0.17
0.90	0.58	0.13
1.00	0.56	0.13

Table 5.4. Figure of merit for circular object region, SNR=1, unweighted averaging interpolation, eccentric ROI, for various values of the bonus for straight segments.

5.3. EVALUATION OF RELAXATION LABELING

In this section we discuss the quantitative evaluation of the continuous relaxation labeling scheme described in Chapter 4. Given the implementation proposed here, three issues need our attention. The first issue concerns the choice of the weighting coefficients C_1, C_2, C_3 and C_4 in the formulas of the compatibility coefficients (4.17). Two sets

of coefficients have been considered. The first set was chosen to be identical to the set used by Schachter et al. (1977). The second set of coefficients was particularly designed for very noisy images, in the sense that the no-edge label λ_{ne} was allowed to dominate, in an attempt to prevent the occurrence of spurious edges in the interior and background regions. The second issue to be considered refers to the number of iterations of the relaxation labeling algorithm. In the context of this study, we are not focussed on establishing formal convergence properties of the relaxation mechanism. Instead, we are more interested in the performance of the relaxation algorithm when a limited number of iterations is applied. Here, we consider the application of 1, 2, 3, 6 and 9 iterations. The third issue which needs our attention is the decision threshold, i.e., following the relaxation labeling procedure we still have to decide which pixels are detected as edge pixels and which are not. In the context of Bayes decision theory, it would seem natural to choose the threshold at $P_i(\lambda_e)=0.5$. However, the experimental results were quite disappointing. Instead, we interactively selected the threshold such that the figure of merit on the basis of the initial probabilities $P_i^o(\lambda_e)$ was maximized. The same threshold was subsequently used after the selected number of iterations. Consequently, the results of this quantitative evaluation study can also be interpreted in terms of relaxation labeling as a post-processing mechanism following the application of the traditional Sobel gradient operator, because Abdou and Pratt (1979) also selected the threshold of the Sobel operator which maximized the resulting value of the figure of merit.

Following our discussion of Section 4.3, the initial probabilities $P_i^o(\lambda_e)$ were computed from the gradient magnitudes by applying a non-linear scaling. The top 0.5% of the pixels with the highest values of the gradient magnitudes were assigned an initial probability $P_i^o(\lambda_e)=1$ and the pixels with intermediate values were scaled linearly, as depicted in Fig. 5.8. This initial edge probability was subsequently divided between the labels λ_N, λ_E, λ_S and λ_W and all probabilities were restricted to the interval [0.05,0.80] as discussed in Chapter 4.

In the original publication by Schachter et al. (1977) separate relax-

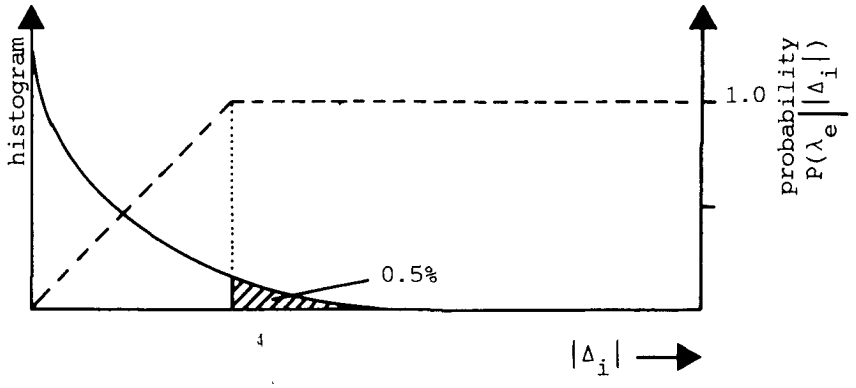


Fig. 5.8. Non-linear scaling of gradient magnitude values to obtain initial edge probabilities.

ation schemes are applied to the edge probabilities and to the hypothesized edge directions. It has been stated before that this is not necessary in the relaxation scheme proposed in Chapter 4.

Experimental results

Experiments were performed on the basis of the synthetic test image containing a vertical edge and corrupted by additive noise with SNR=1, 5, 10, 20, 50, 100. The first set of weighting coefficients was similar to the original set used by Schachter et al. (1977): C_1=1.000, C_2=0.1240, C_3=C_4=0.005. The second set allows the no-edge label to dominate: C_1=1.000, C_2=1.000, C_3=C_4=0.500. Increasing C_3 and C_4 even further tends to devour the true edges. In all cases, the false positive probability was in the order of 0.01. For SNR=50 and SNR=100, the effects of the relaxation labeling procedure were neglectable. For SNR=1, there was a slight improvement due to the relaxation procedure, but even after 9 iterations the results were still useless. The results from the other experiments are summarized in Tables 5.5 - 5.7. For the case SNR=10 and the second set of compatibility weighting coefficients we experimented with a larger number of iterations. It turned out that the results became stable after 18 iterations, with P_D=0.625 and FOM=0.765.

iteration	C_1=1.000, C_2=0.1240 C_3=C_4=0.005		C_1=1.000, C_2=1.000 C_3=C_4=0.500	
	P_D	FOM	P_D	FOM
0	0.312	0.453	0.312	0.453
1	0.312	0.446	0.312	0.459
2	0.312	0.446	0.312	0.459
3	0.312	0.446	0.312	0.458
6	0.312	0.446	0.312	0.472
9	0.328	0.447	0.312	0.471
SNR=5				

Table 5.5. Results of relaxation labeling for SNR=5.

iteration	C_1=1.000, C_2=0.1240 C_3=C_4=0.005		C_1=1.000, C_2=1.000 C_3=C_4=0.500	
	P_D	FOM	P_D	FOM
0	0.484	0.694	0.484	0.694
1	0.484	0.685	0.484	0.685
2	0.531	0.699	0.484	0.685
3	0.562	0.700	0.484	0.693
6	0.578	0.705	0.500	0.710
9	0.594	0.703	0.531	0.741
SNR=10				

Table 5.6. Results of relaxation labeling for SNR=10.

The above results were obtained for one realizaiton of the additive noise field only, but additional experiments have shown that the relative improvements depicted in Tables 5.5 - 5.7 are typical for the relaxation labeling procedure in its present implementation. In addition, despite the relative performance improvement obtainable by applying the relaxation labeling procedure, the results are still quite useless. Especially at SNR=1, we observed an improvement of the figure of merit from 0.140 to 0.172 after 9 iterations, but the finally detected edge points

were still scattered throughout the image plane.

iteration	C_1=1.000, C_2=0.1240 C_3=C_4=0.005		C_1=1.000, C_2=1.000 C_3=C_4=0.500	
	P_D	FOM	P_D	FOM
0	0.953	0.874	0.953	0.874
1	0.953	0.874	0.953	0.874
2	0.953	0.874	0.953	0.878
3	0.953	0.874	0.953	0.908
6	0.969	0.876	0.953	0.919
9	0.969	0.876	0.969	0.920
SNR=20				

Table 5.7. Results of relaxation labeling for SNR=20.

5.4. CONCLUDING REMARKS

In this chapter we evaluated the dynamic programming method of boundary detection as developed in Chapter 3, and the parallel, iterative method of continuous relaxation as developed in Chapter 4. From the experimental performance evaluation, particularly in terms of the figure of merit proposed by Pratt (1978), we conclude that at lower values of the signal-to-noise ratio, say SNR < 10, the dynamic programming method shows much better performance than the traditional method of gradient magnitude thresholding, even if the latter method is used in combination with a relaxation labeling procedure to prevent the occurrence of inconsistent label assignments to related pixels. In addition, we conclude that for images corrupted by additive Gaussian noise, the cost function in the dynamic programming approach depends on the first derivative only. The best discrete gradient operator from the observed set is [-1 -2 0 2 1], and the best interpolation method from the considered alternatives is unweighted averaging of the nearest four pixels. Incorporation of Markov-like boundary properties in the cost function does not yield a considerable improvement of the performance. Obviously, these conclusions are valid only within the context of the

experiments carried out here, but the examples in Chapter 7 show the wide applicability of the dynamic programming approach to problems where other methods fail.

6. ROI-DETECTION

The sequential search method of object delineation by dynamic programming as developed in Chapter 3, heavily relies on the possibility to define a restricted region of interest (ROI) within the image. Given such a region of interest, the image data within the ROI are transformed geometrically into a rectangular matrix, which then forms the input for the dynamic programming algorithm. In specific applications, it may very well be possible to design a goal-oriented procedure to detect the ROI automatically. Examples of this approach will be given in Chapter 7. In this chapter, the problem of ROI-detection is discussed in a somewhat more general setting. In Section 6.1 we treat the case of a compact blob-like object region which allows the use of the polar transformation. In Section 6.2 we shortly discuss the possibility of local gradient extrema tracing. Section 6.3 is devoted to ROI-detection by pre-segmenting the image with a region-oriented segmentation method.

6.1. COMPACT BLOB REGIONS

Let us consider a continuous image of size NxN which can be described

by a binary replacement model, and which contains a single object region with a circular boundary of radius R, centered at coordinate (x_c, y_c). The background region has constant gray-value b and the object region value b+h, as illustrated in Fig. 6.1a. In the noise-free case, the projection P(y) of the data on the y-axis has the value Nb for $0 \leq y \leq y_c - R$ and for $y_c + R \leq y \leq N$, and the value $Nb + 2h(R^2 - (y - y_c)^2)^{\frac{1}{2}}$ for $y_c - R \leq y \leq y_c + R$, as depicted in Fig. 6.1b.

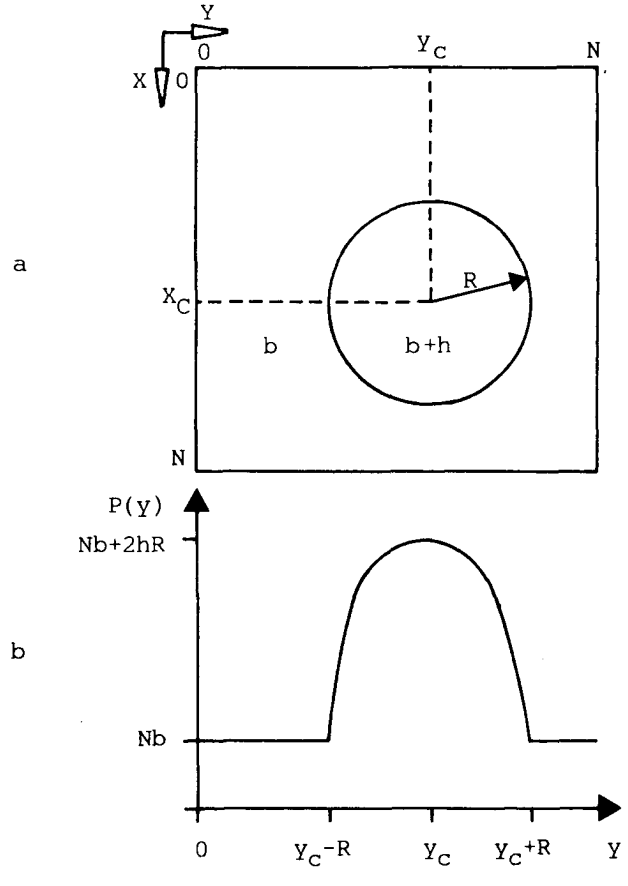


Fig. 6.1. Image containing circular object region (a) and its projection on the y-axis (b).

Obviously, when the image is heavily contaminated by noise the projection profile will be far from the ideal shown in Fig. 6.1b, but the projection mechanism tends to reduce the noise considerably. In many cases it will be possible to obtain a fairly good estimate of the value of y_c and a rough estimate of the radius R from the discrete, noisy projection profile. Repeating this procedure for the projection on the x-axis produces an estimate for x_c, and the polar transformation can be performed. This method can be applied to any object region with a compact blob-like shape, but will fail when the image contains more objects which overlap in the projection profiles.

6.2. GRADIENT EXTREMA TRACING

When the shape of the object region is such that the use of the polar transformation is not obvious, or when the image contains a number of object regions, one is forced to attempt to establish a centerline of the ROI. The image data within a swath along this centerline is then transformed geometrically, the transformation being defined as locally perpendicular to the centerline. Ideally, this centerline would coincide with the true but unknown object region boundary. The problem, however, is not circular as it seems, because it suffices to obtain a rough estimate of the boundary. The approach is hierarchical: the coarse estimate is used to define the ROI and the dynamic programming method produces the final estimate. One can envisage a number of ways to obtain a centerline for the ROI. In this section we outline the approach of tracing gradient extrema.

Tracing gradient extrema is an established method of sequential edge detection (Cf. Rosenfeld and Kak (1982)). First, a starting point is defined, e.g., by selecting a local gradient extremum of acceptably high magnitude. In the direction perpendicular to the gradient direction a search window is defined, and the pixel with the highest gradient is selected as the next point, if its magnitude is above an acceptance threshold. This process is repeated until no new point can be added to the emerging edge.

In a somewhat different context, this type of local maxima tracer has

been investigated by Van Ommeren et al. (1986). Given the present point and the current direction, the next point is selected in a process which they call "beam-like forward looking". In this process, the sums of the gradient magnitudes along three hypothesized continuations of the edge are computed, and the first point k_θ, θ=1,2,3, of the maximum sum continuation is selected. An example is given in Fig. 6.2 for a look-ahead of five pixels. At termination of the sequential edge tracking process, the detected path is smoothed and the resulting smooth curve is used as the centerline of the ROI.

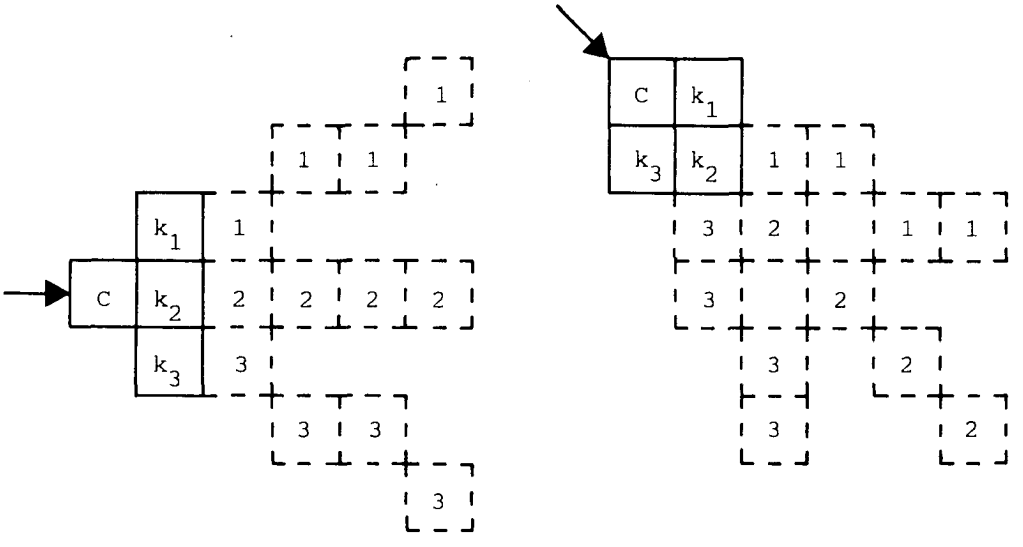


Fig. 6.2.Beam-like search windows with a look-ahead of five pixels (adopted from Van Ommeren (1984)).

Predictably, the detected path will not be correct in all cases. At present, visual inspection of the result is recommended. It is envisaged that the inspection task can be automated in the future by utilizing knowledge-based systems, following some of the ideas proposed by Cheng et al. (1988).

It is strongly recommended to apply the above method to a low-resolution version of the image. The use of a low-pass filter does not only suppress both the noise and spurious details, it also limits the spatial frequency range. As a result, fewer samples are required for a faithful

representation. When the following point spread function is used:

$$\frac{1}{144}\begin{bmatrix} 1 & 3 & 4 & 3 & 1 \\ 3 & 9 & 12 & 9 & 3 \\ 4 & 12 & 16 & 12 & 4 \\ 3 & 9 & 12 & 9 & 3 \\ 1 & 3 & 4 & 3 & 1 \end{bmatrix},$$

the resulting image may be subsampled by a factor of three. Note that the filter can be decomposed into two filters of size [3x3], and each of these is separable into two one-dimensional filters of sizes [3x1] and [1x3], respectively. This facilitates extremely fast implementations (Cf. Groen (1988)).

Because the low-resolution image can be represented by fewer pixels, its use has the additional advantage of a decreased computational burden. The main point, however, is the amplification of the hierarchical aspects: the coarse estimate is obtained at low resolution and is subsequently used to obtain a final estimate at the full resolution. This approach is extended in a region-oriented direction in the next section.

6.3. REGION DETECTION

The maximum gradient tracing algorithm proposed in the previous section, as well as the dynamic programming method itself, basically search for the discontinuity between adjacent regions in the image. In the dual approach, region-oriented segmentation methods attempt to detect the regions explicitly, and the edges result implicitly. In general, region-oriented methods tend to correctly detect the presence of the regions, but the geometric accuracy of the implicitly resulting edges is poor. Combination of the region information and the boundary information, obtained by separate processing modules, may produce a segmentation result which is better than the result which can be obtained by either a region-oriented method or an edge-oriented method. This is precisely the approach taken in the design of general purpose image interpretation or image understanding systems, as proposed, a.o., by Hanson and Riseman

(1978) and Levine and Nazif (1985). In this section we discuss the possibility to use a region-oriented segmentation method in order to define the region of interest as required by the dynamic programming boundary method. This constitutes merely a concatenation of processing modules, in contrast with the approach in computer vision where the results of both processing modules are combined at a higher level in the image interpretation scheme.

The simplest region-oriented segmentation method is constituted by simply thresholding the gray-values. Indeed, in some very simple situations, it is possible to achieve a reasonable tentative segmentation. The results may be improved by applying cellular logic operations like openings and closings, and extraction of the boundaries from the binary segmentation mask yields the wanted tentative centerlines of the ROI's. The method will fail in the case of more complex images with a poor signal-to-noise ratio and spatially varying levels for the various regions. Sequential region-growing methods of the split-and-merge type have proven to yield acceptable segmentation results, even at very low values of the signal-to-noise ratio.

The idea behind split-and-merge procedures is usually attributed to Horowitz and Pavlidis (1976) and has been treated extensively by Pavlidis (1977). Consider an image of size $[2^L x 2^L]$. This image can be divided into its four quadrants, and each of the quadrants can again be divided into its four quadrants, and so on. This process can be described in terms of the quartic picture tree (QPT), where the node at level 0 represents the entire image. Each node in the tree has four sons. The split-and-merge algorithm starts with a tentative partitioning of the image plane into non-overlapping squares of size $[2^{L-k} x 2^{L-k}]$, corresponding with level k in the QPT. We then start with the merging operations. Some uniformity predicate is defined, and if a quadruplet of sons of the same father node can be merged to form a uniform or homogenous region, they are deleted from the tree. First, this is done for all sons at the starting level k, then the procedure is repeated for all nodes without any sons at level k-1, and so on. If no more merging operations can be accomplished, we return to the remaining nodes at the starting level k,

where each node is put to a test to decide whether it is homogeneous or not, according to the uniformity predicate. If it is not, the node is expanded into its four sons, corresponding with a splitting operation on the square in the image domain. This process continues as long as necessary, possibly down to the pixel level.

After all possible merges, and all necessary splitting operations, the image is segmented in terms of squares of various sizes. The smallest squares may be single pixels. All pixels in one square are assumed to belong to one single region, because evidently no further splitting was necessary.

The final stage of a split-and-merge algorithm is called grouping. In this stage, any adjacent squares are put to a test to decide whether they can be joined and still form a homogeneous region according to the uniformity predicate. This process is repeated until all squares are assigned to a region, and constitutes the pure region growing aspect of the algorithm.

In the past, we developed a split-and-merge algorithm along the lines sketched above for a specific application in the field of remote sensing, i.e., the segmentation of side-looking airborne radar (SLAR) images of the earth's surface for vegetation classification purposes, as described in Gerbrands and Backer (1983). Based on the physics of the SLAR-imaging system, the most obvious uniformity predicate to be used was a test on the variance of the gray-values within a split-and-merge square, although the variance criterion does not yield a legitimate uniformity predicate as defined by Pavlidis (1977). However, the effects of using this illegitimate criterion have been shown to be negligible, as reported by Gerbrands and Backer (1984). In effect, very few criteria do yield a legitimate uniformity predicate, and none of these is very satisfactory in the practice of image processing. As a result, we propose to use the variance criterion, despite its theoretical drawbacks.

The split-and-merge segmentation can be used to find a region of interest for the dynamic programming boundary detector in the following way. When all possible merging operations and all necessary splitting operations have been carried out, then, in general, the small squares

will be concentrated near the transition regions between adjacent homogeneous regions. We can then create a binary image, where the pixels of any small region are labeled 'one' and all pixels of larger regions are labeled 'zero'. The decision threshold can be determined from the histogram of the sizes of the squares after the merging and splitting operations. In this binary image, isolated squares of "ones" can be omitted, but any concatenation of squares of "ones" defines a possible region of interest. By computing the skeleton of such a concatenation, the centerline of a possible region of interest can be established. This approach has been the subject of preliminary experiments performed by Starink (1986) and de Kok (1987). Figure 6.3 shows a synthetic test image, figure 6.4 shows the binary image containing the small squares only, and figure 6.5 shows the skeleton of the significant region, superimposed on the original test image.

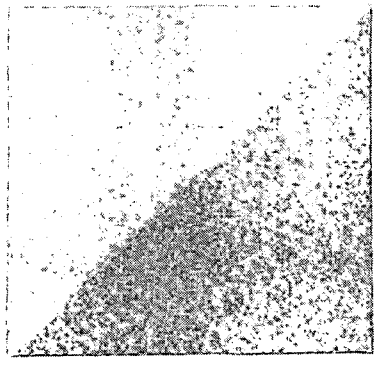


Fig. 6.3. Synthetic test image.

The method of ROI-detection described above is certainly promising, but further investigations are needed to establish its true value. In addition, it should be noted that split-and-merge algorithms, in combination with cluster analysis and a continuous relaxation scheme applied to the labels of regions, may produce a segmentation result which is quite acceptable in itself. This procedure has been reported by Gerbrands, Backer and Cheng (1986). Finally, it is remarked that the approach of split-and-merge algorithms can be generalized to pyramidal

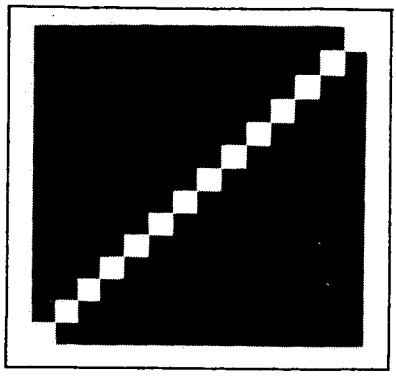


Fig. 6.4. Map of small split-and-merge squares.

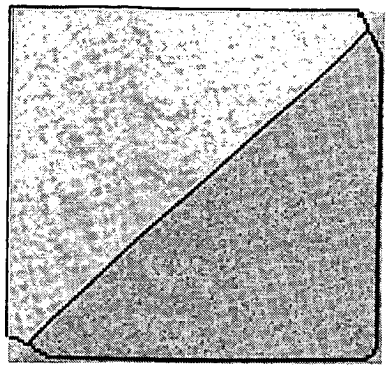


Fig. 6.5. Skeleton of significant region superimposed on original image.

structures with various relationships between father-nodes and son-nodes. Preliminary experiments in the field of digital image segmentation by using pyramidal structures have been reported by Eijlers (1987).

6.4. CONCLUDING REMARKS

In this chapter we discussed various possibilities to automatically define a region of interest, as a prerequisite for the subsequently applied dynamic programming boundary detector. The supporting experimental evidence is rather limited, but the general conclusion is that in many cases it will be possible to find a region of interest. The success rate of the discussed approaches certainly depends on the characteristics of the imagery which is to be segmented. The methods discussed in Sections 6.2 and 6.3 draw heavily upon the idea of 'planning', as introduced by Kelly (1971), in the sense that analysis at some higher level in a multi-resolutional pyramidal data structure facilitates the ultimate analysis at the bottom level of the full resolution image representation. This powerful mechanism does not only find application in the field of image analysis, but also in the field of digital image coding (motion analysis, subband coding) and seems to be closely related to popular models of human visual perception.

7. APPLICATIONS

The dynamic programming method of object region boundary detection was originally developed for the delineation of the left ventricle of the human heart in Technetium-99m radioisotope images, in a cooperative research project between Delft University of Technology and the Thoraxcenter of the Erasmus University in Rotterdam. Early publications describing this application are the papers by Lie et al. (1981) and Gerbrands et al. (1981). Due to its success, the method has since been applied to various other segmentation problems in the field of medical image analysis. In this chapter we describe how the dynamic programming method is used in two of the applications which have already proven their value in clinical practice. In Section 7.1 we discuss the delineation of the left ventricle in Technetium-99m gated blood-pool scintigrams. Section 7.2 is devoted to the detection of the boundaries of arterial segments in coronary cineangiograms. Section 7.3 contains a discussion of some typical aspects of these applications.

7.1. Tc-99m GATED BLOOD-POOL SCINTIGRAMS

Technetium-99m (Tc-99m) gated cardiac blood-pool scintigraphy has been accepted in clinical practice as a noninvasive technique for the assessment of left ventricular (LV) function. With this technique the red blood cells are labeled with the radioactive isotope Tc-99m and the emitted radiation is measured with a gamma-camera. The scintillations at each position in the flat crystal of this camera are registered and the accepted counts accumulated in a matrix in computer memory. In a gated study, the cardiac cycle is divided into a fixed number of time intervals and a corresponding matrix is defined for each interval. The counts are scheduled to the matrices depending on their time of arrival with respect to the R-top of the ECG. An example of the first eight scintigrams of a 20 frames study is given in Fig. 7.1. The images are of size [64x64]. The more or less circular blob in each image represents the activity of the LV. The total number of counts within the left ventricular activity structure, after appropriate background correction, provides a direct measure for instantaneous left ventricular volume. The upper-left image is at end-diastole, where the LV volume is maximal. The lower-right image is at end-systole, the LV volume being minimal.

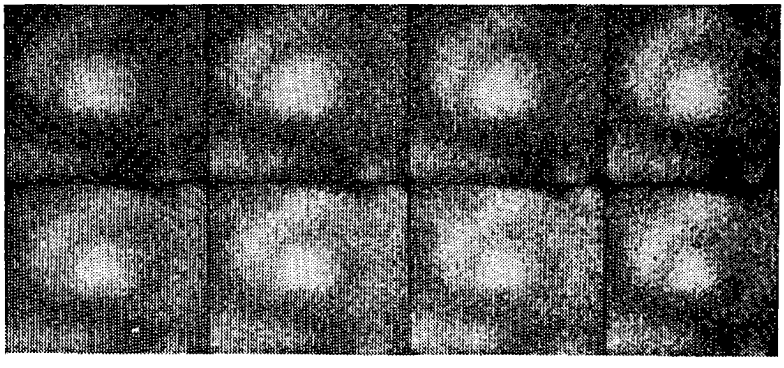


Fig. 7.1. Eight images of a Tc-99m gated blood-pool study.

Studies of this type are performed to obtain the ejection fraction, which is defined as the change in volume of the LV from end-diastole to end-systole relative to the maximum volume. Computation of the ejection fraction (EF) requires the delineation of the LV boundary and the definition of a background region. This is achieved by the dynamic programming method as described by Lie et al. (1981) and Gerbrands et al. (1981), inspired by the ideas of Martelli (1972, 1976) and Ashkar and Modestino (1978). Specific details concerning the actual implementation are excerpted from Reiber et al. (1983) and are given here.

Definition of LV center

To allow contour detection in polar coordinates, the LV center must be determined. This is done automatically in the first frame of the study. In this frame the 64 sums from the columns and rows are computed and smoothed with an unweighted 5-point operator. A provisional center position is found by searching the row and column sums at the right lower corner of the image for the first local maximum values above certain row and column thresholds, respectively. The threshold value, THRES, for the row or column sum has been found empirically to be:

$$THRES = (MAX-MIN) \times 4 + MIN,$$

where MAX and MIN denote the absolute maximum and minimum values between the 11th and 54th positions of the particular vector, respectively. Figure 7.2 shows the first frame of a gated blood-pool study with plots of the row and column sums. Following the computation of this initial center position, a 21- by 21-pixel area, centered around this position, is defined. Next, the total count within a submatrix of size 7x7 is determined for each of the possible positions of the submatrix within the 21x21 area. The center position of this submatrix at the location with the maximal total number of counts is then assumed to be the approximate center of the LV activity structure. This provisional center does not necessarily coincide with the geometric center of the left ventricle or with its center of gravity. This does not pose a serious problem for the contour-detection algorithm, since this approxi-

mate center will be updated by the contour algorithm itself. The provisional center simply provides a starting point for further analysis. In case the algorithm fails to find a reasonable center position, the user may correct the position with the joystick of the computer system.

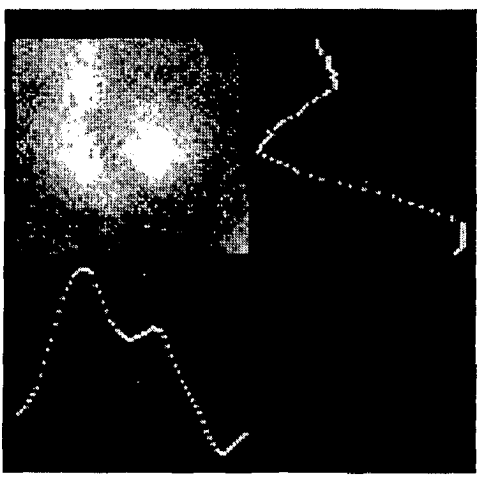


Fig. 7.2. Automated definition of approximate center of LV activity distribution from column and row sums.

Contour detection

The polar representation is obtained by sampling the original image along 64 radii from the provisional center of the LV. Along each radius 32 samples are taken, with sample distances equal to the pixel spacing in the x,y matrix. The value of a pixel in the polar matrix is defined by the average of the 3x3 neighborhood of the pixel closest to the sample point in the original image. This polar representation is shown in Fig. 7.3 (top center) together with the original image (top left). The radial distance is plotted along the horizontal axis and the angular position counterclockwise with respect to the 3 o'clock direction in the x,y matrix is plotted along the vertical axis.

The detection of the edges in the polar image is achieved by means

of a second-derivative operator applied along the horizontal lines in the polar image. The second-derivative image is obtained by applying a first-derivative operator twice. Finally, the cost matrix is defined as the inverse of the second-derivative image. The top right image in Fig. 7.3 is the first-derivative image and the bottom left image the cost matrix. The displayed brightness levels in the cost matrix are proportional to the cost coefficients. The band in which the contour is to be detected is characterized by low costs (low brightness levels) as can be seen in this image.

The minimal-cost contour is the minimum-cost path from the bottom to the top in this cost matrix (Fig. 7.3, bottom center). Retransforming to Cartesian coordinates and connecting the 64 contour positions results in a continuous contour (Fig. 7.3, bottom right).

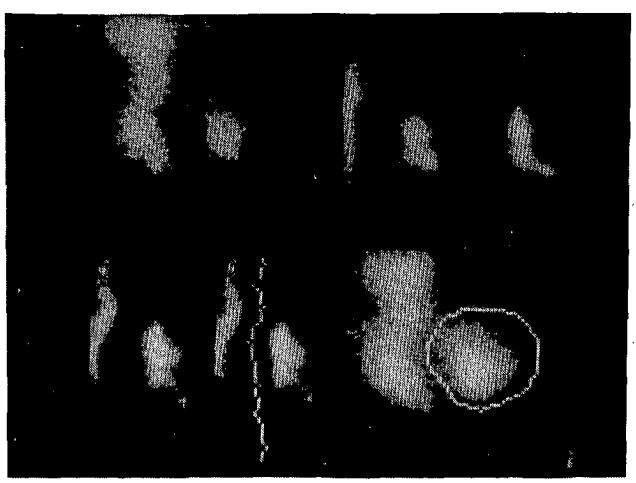


Fig. 7.3. Top images represent (left to right) original image, polar image, and first-derivative image. Bottom images represent (left to right) cost matrix, detected contour superimposed on cost matrix, and image with detected contour superimposed.

It is clear that the detected contour will depend on the initially computed center position. Accordingly, this center position is now re-

placed by the centroid of the activity distribution within the initially generated contour and the contour-detection procedure is repeated from this centroid. Repetition of this procedure will converge toward an optimal center position, but in practice a single iteration suffices, since the deviation from the optimal position is then usually not more than 1 pixel in x- and/or y-direction. One iteration is therefore taken to give the final contour. The computation time for the foregoing procedure is only 12 seconds.

If the user does not agree with the displayed contour, he may correct it or draw a new one with the joystick.

7.2. CORONARY CINEANGIOGRAMS

Coronary angiography plays an important role in the clinical management of patients with ischemic heart disease. In this procedure, roentgen contrast agent is selectively injected into one of the main coronary arteries via a catheter and X-ray cineangiograms are obtained with a cinecamera. These films provide the clinicians with accurate information about the morphology of the coronary arterial tree, thus creating the possibility to investigate the presence, the extent, the severity and the functional significance of coronary obstructions. In this section we shortly describe the delineation of selected arterial segments. At present, in clinical practice, the region of interest for the contour detection algorithm is indicated interactively. The user indicates a number of central points in the selected segment, such that the straight line segments connecting consecutive pairs of these points are well within the artery. An example is given in Fig. 7.4.

The indicated piece-wise linear curve is then smoothed, yielding the tentative centerline of the segment. Two separate regions of interest are defined by swaths of image data left and right of the tentative centerline. The data in each swath is transformed geometrically by means of a transformation which is approximately perpendicular to the local centerline. The dynamic programming method is then used to detect the lefthand and righthand boundaries of the arterial segment. The cost function involved is given in Eq. (5.5), with a value of α=0.5. The detected paths

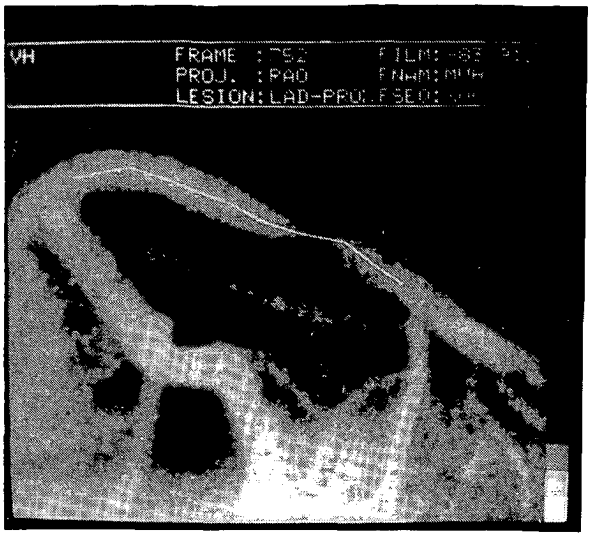


Fig. 7.4. Interactively indicated line segments.

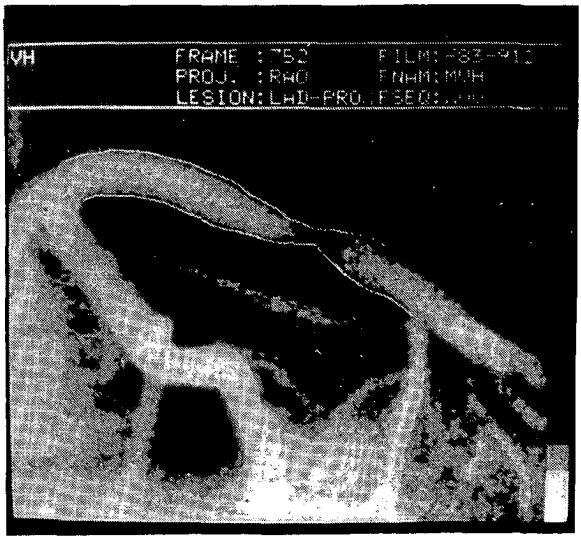


Fig. 7.5. Detected boundaries of arterial segment.

in the transform domain are mapped back to the original image domain, closed by linear interpolation and smoothed. The final centerline is then computed from both boundaries and the complete procedure is repeated to diminish the influence of the human interaction in the definition

of the ROI. The final result for the example of Fig. 7.4 is given in Fig. 7.5.

The same procedure is applied to the tip of the catheter as observed in the angiogram. Because the true dimensions of the catheter are known, this measurement creates the possibility to calibrate all subsequent measurements in millimeters. The boundary data are corrected for pincushion distortion and the digital gray-values in the arterial segment are calibrated by using a densitometric procedure described in Reiber et al. (1983a). Various quantitative measurements are then obtained, like the percentages diameter and area narrowing, the extent of the occlusion, the roughness of the boundary, etc. Descriptions can be found in Reiber et al. (1982) and Kooijman (1982). Some of these measurements are illustrated in Fig. 7.6.

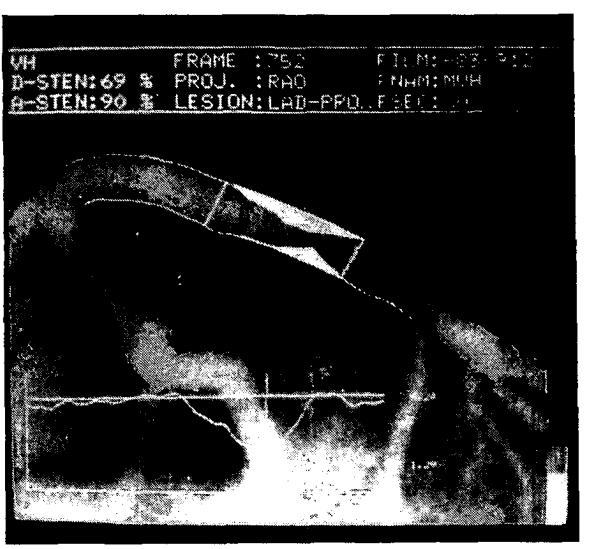


Fig. 7.6. Detected contours, diameter function, computed diameter and area narrowing, and estimated original size of artery.

The complete procedure has been evaluated extensively, as reported by Reiber et al. (1984a) and Reiber et al. (1984b). Here, it suffices to state that the accuracy and precision of the boundary detection pro-

cedure as assessed from cinefilms of perspex models of obstructed arterial segments filled with contrast agent were -30 μm and 90 μm, respectively. The variability of the complete procedure in terms of absolute arterial dimensions was less than 0.12 mm for clinical cineangiograms.

Related material can be found in Reiber et al. (1986) and Reiber et al. (1986a). Recent experimental results are described in Reiber et al. (1987).

7.3. DISCUSSION

In this section we discuss some typical aspects of the applications described above, starting with the delineation of the left ventricular region in the Tc-99m scintigrams. It is noted that the procedure to define the origin of the polar coordinate system resembles the method proposed in Section 6.1. In this application the situation is more complex, which is especially due to the presence of the right ventricular region. The interpolation method used in the polar transformation is zero-order or nearest neighbor interpolation. Note that we discarded this method in our quantitative evaluation study as described in Section 5.2. In this application, however, nearest neighbor interpolation is used in combination with a [3x3] uniform preprocessing filter. In Chapter 5, we did not consider such additional operations, because we were focussing on the potential differences in the performance of the various interpolation schemes. In an additional experiment, we applied the evaluation method of Chapter 5 to the combination of a [3x3] uniform smoothing filter and zero-order interpolation. The results are given in Fig. 7.7 for the test image containing a circular object region. From this experiment we conclude that the interpolation method used in this application compares favorably with the other schemes considered.

The next topic concerns the cost function used in the dynamic programming optimal path algorithm. The cost function used here is based on the values of the second derivative only, which is in contrast with our findings in Chapter 5. It should be noted, however, that the images considered here do not conform to the replacement model. The left ventricular region, as observed in the planar scintigram, represents a pro-

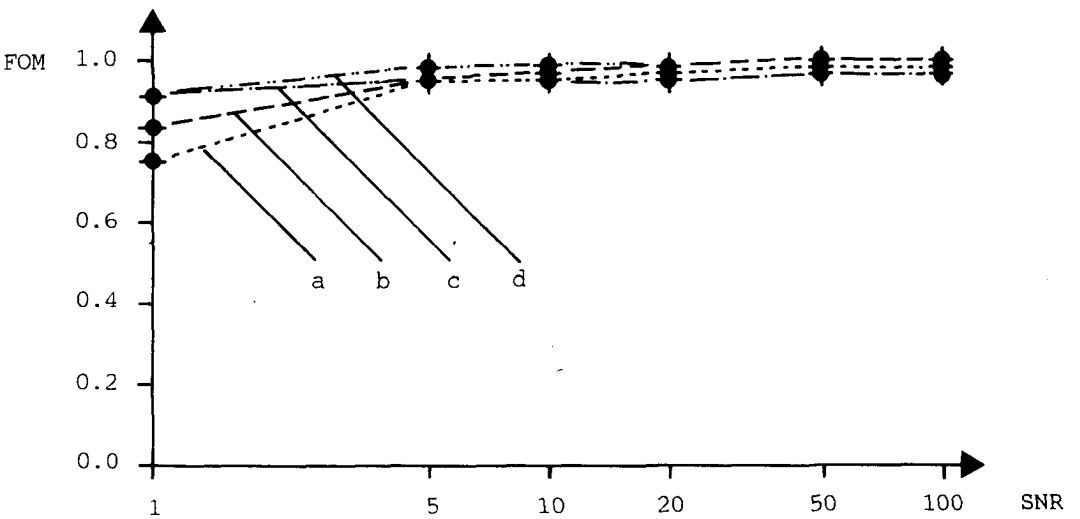


Fig. 7.7. FOM-values for various interpolation schemes: zero-order (a), true bilinear interpolation (b), unweighted averaging (c) and the combination of a [3x3] uniform filter and zero-order interpolation (d), for various values of SNR and a circular object region.

jection of the three-dimensional activity distribution in the left ventricle. If we model the three-dimensional left ventricle as a homogeneous sphere, then a cross-section of the brightness function in the projection image will look like the curve in Fig. 7.8 due to the low resolution of the imaging system. Using the first-derivative values in the cost function would yield a contour which does not encompass the complete left ventricular region, since the maxima of the first derivative function occur at the inflection points of the activity distribution. The use of the second derivative results in a contour at a sufficiently shifted position.

The clinical evaluation of the complete procedure, including various ways to compute the ejection fraction, is discussed in detail by Reiber

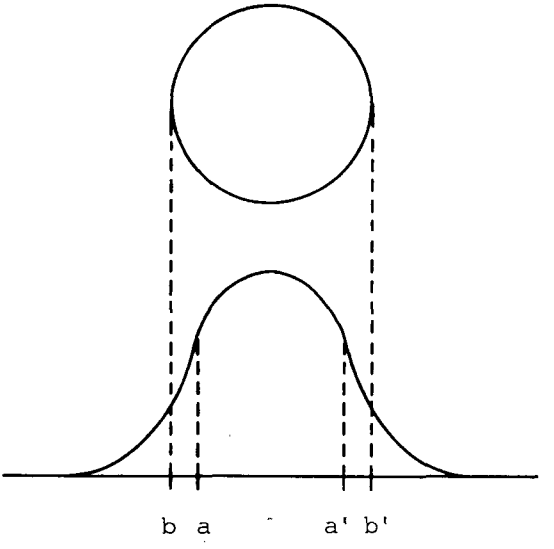


Fig. 7.8. Maximum gradient magnitude positions (a,a') and required boundary positions (b,b').

(1983). Here, it suffices to state that the contour detection scored a success rate of 92% in a total of 100 patient studies, in the sense that visual inspection of the detected contours led to interactive corrections by the user in only 8% of the cases. The complete procedure of ejection fraction estimates compares favorably with other methods in a study performed by Reiber (1984).

Now we turn our attention to the second application, i.e., the delineation of arterial segments in coronary cineagiograms. Again, nearest neighbor interpolation is used in the geometric transform to straighten the region of interest. In the present implementation, a [5x5] median filter is applied as a preprocessing filter. The noise suppression effects of median filters are similar to those of linear filters. The advantage of this type of non-linear filters is that the edges in the image are not blurred. On the contrary, they tend to be enhanced by the filtering operation. As a result, it is conjectured here that the combination of a [5x5] median filter and nearest neighbor interpolation will

perform as least as good as the combination of a [3x3] linear filter and the same interpolation scheme, as discussed above. Concerning the cost function, it is observed that in this application both the first derivative and the second derivative are used. In the light of the above remarks concerning the scintigraphic images, the difference may be explained by the fact that the blur introduced by the X-ray imaging system is much less than the blur caused by the low resolution scintillation camera.

In Chapter 5, we studied the performance of boundary detection schemes for the case of uniform regions corrupted by signal-independent additive white Gaussian noise. In relation to the applications discussed here, the evaluation method has been applied in a limited experiment to synthetic test images with a Poisson gray-value distribution. The test image containing a diagonal edge was used, and the two regions were assigned Poisson distributions with parameters λ_1 and λ_2, respectively. These distributions were approximated by pseudo-random Gaussian noise processes with parameters $\mu_1=\sigma_1^2=\lambda_1$ and $\mu_2=\sigma_2^2=\lambda_2$. The signal-to-noise ratio was arbitrarily defined as:

$$SNR_P = \frac{(\lambda_1-\lambda_2)^2}{\frac{1}{2}(\lambda_1+\lambda_2)}, \qquad (7.1)$$

the numerator representing the squared contrast between both regions in terms of the difference of the means, and the denominator representing the average of the two noise variances. The resulting FOM-values are given in Fig. 7.9. The optimal value for the weighting coefficient in the cost function of Eq. (5.5), as assessed from some preliminary experiments, was $\alpha=1.0$. The results are averaged over five noise realizations. Details of this experiment can be found in Bedawi (1988) and are omitted here.

Due to its success in both applications discussed here, the method of dynamic programming boundary detection is also used for object delineation in Thallium-201 myocardial perfusion scintigrams. Early results are discussed by Lie et al. (1981). The method has been in clinical use for

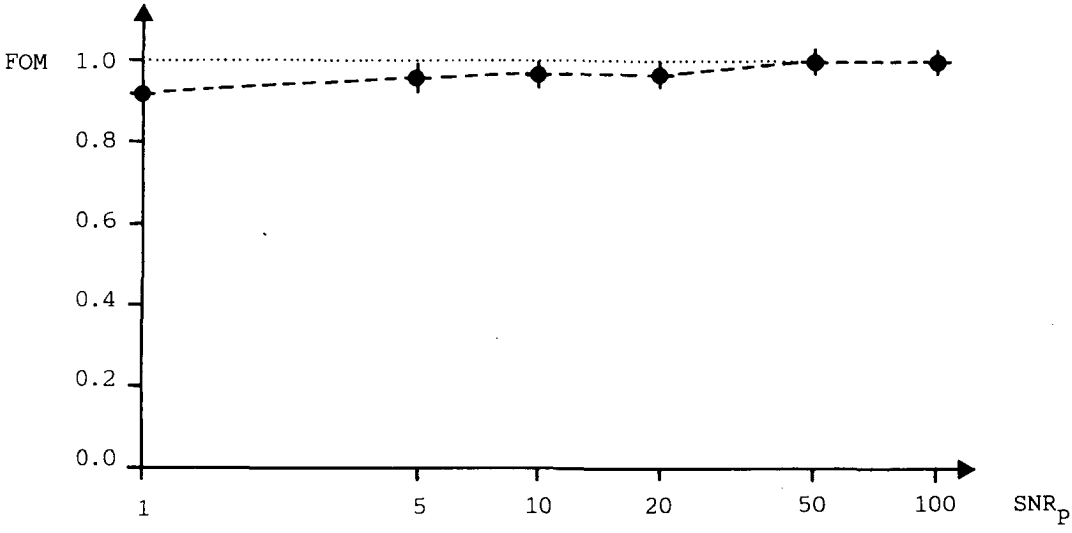


Fig. 7.9. FOM-values for pseudo-Poisson gray-value distributions for various values of SNR_p, diagonal edge.

a number of years. Detection of the complete coronary tree and frame-to-frame analysis of coronary cinefilms are the subject of current research, and preliminary results have been reported (van Ommeren (1986)). Extension of the method to delineation of the left ventricular region in both angiographic and echocardiographic images is also being investigated. The research described in this Chapter is the object of a cooperative project with the Thoraxcenter of the Erasmus University in Rotterdam.

8. DISCUSSION

In the previous chapters, we discussed the segmentation of images which are severely contaminated by noise. Because the noise destroys the coherence of the image structures we would like to detect, elementary segmentation methods like gray-value thresholding or gradient-magnitude thresholding show a rather poor performance for low values of the signal-to-noise ratio. Consequently, one has to resort to more elaborate methods, which possibly include extensive pre- or post-processing of the input image or the segmentation result. In the present study, we focused on attempts to improve the performance of the segmentation stage by exploiting the spatial context and by incorporating generic a priori knowledge. As we are particularly interested in the accurate delineation of the object regions, we concentrated on edge-oriented methods.

In Chapter 2, we established our preference for defining an edge as a concatenation of edge pixels above the alternative of a concatenation of the boundaries between neighboring pixels of adjacent regions. The choice between both edge definitions is similar to the choice between using either the chain code or the crack code as boundary descriptors

in segmented images. The selected edge definition naturally corresponds with the use of the chain code. However, this bears no consequences with regard to subsequent measurements and analysis, because the crack code can be derived from the chain code and vice versa. In Chapter 2, we also observed that synthetic test images designed for experimental evaluation of segmentation schemes should show transitions between adjacent regions with a width at least equal to the grid size. Surprisingly, this almost trivial rule is very often offended in the practice of image processing research.

In Chapter 3, we developed a sequential boundary detection method on the basis of dynamic programming. This optimization technique is used here to select the optimal edge with respect to some predefined merit function. Because the merit function is evaluated along the entire edge, or, in particular cases, along the complete closed contour, the exploitation of the spatial context is quite obvious. Generic a priori knowledge is incorporated in the connectivity constraints as well as in the structure of the cumulative merit function. Problem-dependent a priori knowledge guides the choice of the actual merit function and is also used to restrict the search to a predefined region of interest (ROI). We showed that the dynamic programming approach is greatly simplified by straightening the ROI by means of a geometric transformation. For blob-like regions with a simple closed contour, the polar transform is the method of choice. For more general boundaries, the ROI consists of a swath centered at the hypothesized boundary. In the continuous case, we established a relationship between the width B of the ROI and the minimum radius of curvature R_{min} along the curve, i.e., $B \leq 2.R_{min}$, as expressed in Eq. (3.29). Furthermore, we established a global constraint which a boundary is to satisfy in order for our dynamic programming method to be applicable to it. The constraint is that if the distance between two points, measured along the curve, exceeds a value of $\pi.R_{min}$, then their Eucledean distance should be at least equal to the width B of the ROI. These theoretical results merely establish some rules of thumb for the discrete case, in the sense that a highly curved boundary requires a rather narrow ROI, which is to be positioned quite accurately.

In other words, when the region boundary is very complex, the required amount of a priori knowledge grows tremendously.

In Chapter 4, we discussed a different method which allows the exploitation of the spatial context as well as the incorporation of generic a priori knowledge. In the parallel, though iterative, approach of continuous relaxation labeling, the compatibility coefficients between label pairs at related pixel positions can be selected on the basis of some notions of generic edge properties. The context is taken into account by considering iteratively expanding neighborhoods, in an attempt to prevent the occurrence of label inconsistencies. In particular, we developed a new implementation of the edge detector scheme proposed by Schachter et al. (1977). The new implementation is especially suited for modern digital image processing systems, in the sense that the computational problem is decomposed to a large extent into spatial convolutions and dyadic point operations, for which tasks special purpose processors are usually available.

In Chapter 5, both the dynamic programming method and the relaxation labeling scheme were evaluated experimentally by measuring their performances when applied to synthetic test images with various signal-to-noise ratios. The experiments were designed in accordance with the comparative study of traditional edge detector schemes as published by Abdou and Pratt (1979), in order to facilitate the comparison of results. The evaluation criteria included the estimated probabilities of correct and false edge pixel detection, similar to ROC-analysis in statistical detection theory, and Pratt's Figure of Merit (FOM). This measure is of particular importance because, contrary to the false detection probability, it distinguishes between the various types of errors. In principle, the FOM expresses the usefulness of the detected edge for further processing. The evaluation criteria were also used to tune the parameters of the investigated segmentation schemes. From these experiments we concluded that at lower values of the signal-to-noise ratio SNR, say $1 \leq SNR \leq 10$, the dynamic programming approach shows much better performance than the traditional method of gradient-magnitude thresholding, even if the latter method is used in combination with the continuous

relaxation labeling procedure. However encouraging this result, it is more confirmatory than surprising. The particular method of dynamic programming boundary detection developed here is forced to find one single boundary, containing a predefined number of edge pixels, within a predefined region of interest, on the basis of gradient values measured perpendicular to the assumed true boundary, by implicitly evaluating the merit functions along all hypothesized boundaries within the ROI. This last point forms the heart of the method, the other points merely originated from our attempts to find a feasible implementation. In other words, it is conjectured here that the overall evaluation of the merit function along the entire boundary is the crucial element. In view of the extensive use of a priori knowledge and spatial context, at least in comparison with the other methods evaluated here, it may not be surprising that the performance at low values of SNR is much better. On the other hand, our experiments prove that the proposed way to exploit a priori knowledge and spatial context within the segmentation stage yields good results indeed.

In Chapter 6, we discussed various ways to detect the region of interest automatically. It was mentioned there that the sequential region-oriented segmentation method of split-and-merge, in combination with cluster analysis and a continuous relaxation scheme applied to the region labels, may produce a segmentation result which is quite acceptable in itself. We also mentioned the extension to more general pyramidal structures. It would certainly be of great interest to include such methods in an extended evaluation study.

The applications discussed in Chapter 7 show the deep involvement of the staff of the Laboratory for Clinical and Experimental Image Processing of the Thoraxcenter of the Erasmus University in Rotterdam in many aspects of the work reported here. An exciting and intriguing new development, investigated with particular reference to tomographic cardiac scintigraphy, is the extension of the dynamic programming boundary detection method to the three-dimensional case. Early attempts in this direction are described by Blokland (1982) and Reijs et al. (1986). A more fundamental analysis, including simulation studies to compare various approaches,

forms the subject of a thesis by Starink (1987).

We conclude that the restricted-search dynamic programming boundary detection method constitutes a very robust and useful tool, especially suited for application in image measurement systems. In addition, it would be very attractive to have the algorithm available as one of the segmentation modules in a more general knowledge-based image analysis system.

REFERENCES

I.E. Abdou (1978).
Quantitative Methods of Edge Detection,
University of Southern California, USCIPI Report 830.

I.E. Abdou and W.K. Pratt (1979).
'Quantitative design and evaluation of enhancement/thresholding edge detectors', **Proceedings IEEE**, Vol. 67, pp. 753 - 763.

G.P. Ashkar and J.W. Modestino (1978).
'The contour extraction problem with biomedical applications',
Computer Graphics and Image Processing, Vol. 7, pp. 331 - 355.

E. Backer and J.J. Gerbrands (1988).
'A flexible and intelligent system for fast measurements in binary images for in-line robotic control',
In: **Real-time Object and Environment Measurement and Classification** (A.K. Jain, editor), NATO ASI Series, Vol. F42,
Berlin: Springer.

Z.M.H. Bedawi (1988).
Dynamic Programming Edge Detection in Poisson-distributed Images,
Internal report (in Dutch), Delft University of Technology.

R. Bellman and S. Dreyfus (1962).
Applied Dynamic Programming,
Princeton, N.J.: Princeton University Press.

J. Biemond and J.J. Gerbrands (1979).
'An edge-preserving recursive noise-smoothing algorithm for image data', **IEEE Transactions on Systems, Man, and Cybernetics**, Vol. SMC-9, pp. 622 - 627.

J.A.K. Blokland (1982).
3-D Contour Detection in Emission-tomographic Images of the Heart, M.Sc. thesis (in Dutch), Delft University of Technology.

K.R. Castleman (1979).
Digital Image Processing,
Englewood Cliffs, N.J.: Prentice-Hall.

X.S. Cheng, J.J. Gerbrands and E. Backer (1988).
'The design of a distributive and anomaly-driven system for general image understanding',
In: **Proceedings 9th IAPR International Conference on Pattern Recognition**, Beijing (accepted).

J.W. Cohen and O.J. Boxma (1983).
Boundary Value Problems in Queueing System Analysis,
Amsterdam: North-Holland, p. 72.

G.J.M. Conijn (1987).
The Use of Markov Properties in Minimum-cost Edge Detection,
Internal report (in Dutch), Delft University of Technology.

D.B. Cooper and H. Elliott (1978).
'A maximum likelihood framework for boundary estimation in noisy images',
In: **Proceedings IEEE Computer Society Conference on Pattern Recognition and Image Processing**, Chicago, pp. 25 - 31.

D.B. Cooper (1979).
'Maximum likelihood estimation of Markov-process blob boundaries in noisy images',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-1, pp. 372 - 384.

D.B. Cooper, F. Sung and P.S. Schenker (1980).
'Toward a theory of multiple-window algorithms for fast adaptive boundary finding in computer vision',
In: **Proceedings 5th IAPR International Conference on Pattern Recognition**, Miami, pp. 1278 - 1284.

D.B. Cooper and F. Sung (1983).
'Multiple-window parallel adaptive boundary finding in computer vision',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-5, pp. 299 - 316.

E.S. Deutsch and J.R. Fram (1978).
'A quantitative study of the orientation bias of some edge detector schemes', **IEEE Transactions on Computers**, Vol. C-27, pp. 205 - 213.

L. Dorst (1985).
'The accuracy of the digital representation of a straight line',
In: **Fundamental algorithms for computer graphics** (R.A. Earnshaw, editor), NATO ASI Series, Vol. F17,
Berlin: Springer.

R.O. Duda and P.E. Hart (1973).
Pattern Classification and Scene Analysis,
New York: John Wiley.

E.J. Eijlers (1987).
Pyramidal Structures for Image Understanding Systems,
M.Sc. thesis, Delft University of Technology.

H. Elliott and L. Srinivasan (1981).
'An application of dynamic programming to sequential boundary estimation',
Computer Graphics and Image Processing, Vol. 17, pp. 291 - 314.

O. Faugeras and M. Berthod (1981).
'Improving consistency and reducing ambiguity in stochastic labeling: an optimization approach',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-3, pp. 412 - 424.

G.D. Forney (1972).
'Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference',
IEEE Transactions on Information Theory, Vol. IT-18, pp. 363 - 378.

G.D. Forney (1973).
'The Viterbi algorithm', **Proceedings IEEE**, Vol. 61, pp. 268 - 278.

J.R. Fram and E.S. Deutsch (1975).
'On the quantitative evaluation of edge detection schemes and their comparison with human performance',
IEEE Transactions on Computers, Vol. C-24, pp. 616 - 627.

J.J. Gerbrands, C. Hoek, J.H.C. Reiber, S.P. Lie and M.L. Simoons (1981).
'Automated left ventricular boundary extraction from Technetium-99m gated blood pool scintigrams with fixed or moving regions of interest',
In: **Proceedings 2nd IEEE International Conference on Visual Psychophysics and Medical Imaging** (C.C. Jaffe, editor),
Brussels, pp. 155 - 159.

J.J. Gerbrands and E. Backer (1983).
'Segmentation of multitemporal side-looking airborne radar (SLAR) images',
In: **Proceedings SPIE Conference on Applications of Digital Image Processing**, Geneva, pp. 173 - 179.

J.J. Gerbrands and E. Backer (1984).
'Split-and-merge segmentation of SLAR-imagery: segmentation consistency',
In: **Proceedings 7th IAPR International Conference on Pattern Recognition**, Montreal, pp. 284 - 286.

J.J. Gerbrands, E. Backer and W.A.G. van der Hoeven (1985).
'Edge detection by dynamic programming',
In: **Proceedings 6th Symposium on Information Theory in the Benelux** (A.J. Vinck, editor), Mierlo, pp. 35 - 42.

J.J. Gerbrands, E. Backer and X.S. Cheng (1986).
'Multiresolutional cluster segmentation using spatial context',
In: **Proceedings 8th IAPR International Conference on Pattern Recognition**, Paris, pp. 1333 - 1335.

J.J. Gerbrands, E. Backer and W.A.G. van der Hoeven (1986a).
'Quantitative evaluation of edge detection by dynamic programming',
In: **Pattern Recognition in Practice II**
(E.S. Gelsema and L.N. Kanal, editors),
Amsterdam: North-Holland.

J.J. Gerbrands, J.H.C. Reiber, A.E.M. Reijs, B.M. Wiezer and E. Backer (1987).
'ESATS: an expert system for the quantitative analysis of thallium-201 scintigrams',
In: **Proceedings SPIE International Conference on Advances in Image Processing**, The Hague (to appear).

R.C. Gonzalez and P. Wintz (1977/1987).
Digital Image Processing,
Reading, Mass.: Addison-Wesley.

F.C.A. Groen, P.P. Jonker and R.P.W. Duin (1988).
'Hardware versus software implementations of fast image processing algorithms',
In: **Real-time Object and Environment Measurement and Classification**
(A.K. Jain, editor), NATO ASI Series, Vol. F42,
Berlin: Springer.

H.P.A. Haas (1985).
Convexity Analysis of Hexagonally Sampled Images,
Thesis Delft University of Technology.

A.R. Hanson and E.M. Riseman (editors) (1978).
Computer Vision Systems,
New York: Academic.

R.M. Haralick and L.G. Shapiro (1979).
'The consistent labeling problem: part I',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-1, pp. 173 - 184.

R.M. Haralick and L. Watson (1981).
'A facet model for image data',
Computer Graphics and Image Processing, Vol. 15, pp. 113 - 129.

R.M. Haralick (1983).
'Decision making in context',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-5, pp. 417 - 428.

R.M. Haralick (1984).
'Digital step edges from zero crossing of second directional derivatives',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-6, pp. 58 - 68.

C.S. Ho (1983).
'Precision of digital vision systems',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-5, pp. 593 - 601.

C. Hoek (1980).
Automated Processing of Technetium Bloodpool Scans,
M.Sc. thesis (in Dutch), Delft University of Technology.

W.A.G. van der Hoeven (1985).
Implementation and Evaluation of a Minimum- cost Method of Edge Detection, M.Sc. thesis (in Dutch), Delft University of Technology.

S.L. Horowitz and T. Pavlidis (1976).
'Picture segmentation by a direct split-and-merge procedure',
Journal of the ACM, Vol. 17, pp. 368 - 388.

M.F. Hueckel (1971).
'An operator which locates edges in digitized pictures',
Journal of the ACM, Vol. 18, pp. 113 - 125.

R.A. Hummel and S.W. Zucker (1983).
'On the foundations of relaxation labeling processes',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-5, pp. 267 - 287.

A.A.M. Jongeling (1987).
Segmentation of Tomographic Images of Sponge Cores,
M.Sc. thesis, Delft University of Technology.

M.D. Kelly (1971).
'Edge detection in pictures by computer using planning',
Machine Intelligence, Vol. 6, pp. 397 - 409.

L.B. de Kok (1987).
Texture Segmentation with Split-and-Merge Algorithms,
M.Sc. thesis (in Dutch), Delft University of Technology.

C.J. Kooijman, J.H.C. Reiber, J.J. Gerbrands, J.C.H. Schuurbiers, C.J. Slager, A. den Boer and P.W. Serruys (1982).
'Computer aided quantitation of the severity of coronary obstructions from single view cineangiograms',
In: **Proceedings 1st IEEE Computer Society International Symposium on Medical Imaging and Image Interpretation**, Berlin, pp. 59 - 64.

V.A. Kovalevsky (1967).
'An optimal algorithm for the recognition of some sequences',
Cybernetics, Vol. 3.

V.A. Kovalevsky (1980).
Image Pattern Recognition,
Berlin: Springer.

M.D. Levine and A.M. Nazif (1985).
'Rule-based image segmentation: a dynamic control strategy approach',
Computer Vision, Graphics and Image Processing,
Vol. 32, pp. 104 - 126.

S.P. Lie (1979).
Quantitative Analysis of Thallium Scintigrams,
M.Sc. thesis (in Dutch), Delft University of Technology.

S.P. Lie, J.H.C. Reiber, C. Hoek, J.J. Gerbrands and M.L. Simoons (1981).
'Automated boundary extraction from cardiac scintigrams',
In: **Proceedings VII International Conference on Information Processing in Medical Imaging**, Stanford, pp. 310 - 328.

A. Martelli (1972).
'Edge detection using heuristic search methods',
Computer Graphics and Image Processing, Vol. 1, pp. 169 - 182.

A. Martelli (1976).
'An application of heuristic search methods to edge and contour detection', **Communications of the ACM**, Vol. 19, pp. 73 - 83.

J.L. Mohammed, R.A. Hummel and S.W. Zucker (1983).
'A gradient projection algorithm for relaxation methods',
IEEE Transactions Pattern Analysis Machine Intelligence,
Vol. PAMI-5, pp. 330 - 332.

U. Montanari (1971).
'On the optimal detection of curves in noisy pictures',
Communications of the ACM, Vol. 14, pp. 335 - 345.

N.E. Nahi and M.H. Jahanshahi (1977).
'Image boundary estimation',
IEEE Transactions on Computers, Vol. C-26, pp. 772 - 781.

N.E. Nahi and S. Lopez-Mora (1978).
'Estimation-detection of object boundaries in noisy images',
IEEE Transactions on Automatic Control, Vol. AC-23, pp. 834 - 846.

W. Niblack (1986).
An Introduction to Digital Image Processing,
Englewood Cliffs, N.J.: Prentice Hall.

N.J. Nilsson (1971).
Problem-solving Methods in Artificial Intelligence,
New York: McGraw-Hill.

J. van Ommeren (1984).
Detection of the Coronary Tree,
M.Sc. thesis (in Dutch), Delft University of Technology.

J. van Ommeren, C.J. Kooijman, R.J. van Meenen, J.J. Gerbrands,
A.V.M.C.L. Schulte and J.H.C. Reiber (1986).
'Artery detection and analysis in cine-angiograms',
In: **Pattern Recognition in Practice II**
(E.S. Gelsema and L.N. Kanal, editors),
Amsterdam: North-Holland.

P.J. van Otterloo and J.J. Gerbrands (1978).
'A note on a sampling theorem for simply connected closed contours',
Information and Control, Vol. 39, pp. 87 - 91.

T. Pavlidis (1977).
Structural Pattern Recognition,
Berlin: Springer.

S. Peleg and A. Rosenfeld (1978).
'Determining compatibility coefficients for curve enhancement relaxation processes',
IEEE Transactions on Systems, Man, and Cybernetics,
Vol. SMC-8, pp. 548 - 555.

S. Peleg (1980).
'A new probabilistic relaxation scheme',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-2, pp. 362 - 369.

T. Peli and D. Malah (1982).
'A study of edge detection algorithms',
Computer Graphics and Image Processing, Vol. 20, pp. 1 - 21.

W.K. Pratt (1978).
Digital Image Processing,
New York: John Wiley.

J. Prewitt (1970).
'Object enhancement and detection',
In: **Picture Processing and Psychopictorics**
(B. Lipkin and A. Rosenfeld, editors),
New York: Academic.

J.H.C. Reiber, J.J. Gerbrands, F. Booman, G.J. Troost, A. den Boer, C.J. Slager and J.C.H. Schuurbiers (1982).
'Objective characterization of coronary obstructions from monoplane cineangiograms and three-dimensional reconstruction of an arterial segment from two orthogonal views',
In: **Applications of Computers in Medicine** (M.D. Schwartz, editor),
New York: IEEE Press.

J.H.C. Reiber, S.P. Lie, M.L. Simoons, C. Hoek, J.J. Gerbrands, W. Wijns, W.H. Bakker and P.M. Kooij (1983).
'Clinical validation of fully automated computation of ejection fraction from gated equilibrium blood-pool scintigrams',
Journal of Nuclear Medicine, Vol. 24, pp. 1099 - 1107.

J.H.C. Reiber, C.J. Slager, J.C.H. Schuurbiers, A. den Boer, J.J. Gerbrands, G.J. Troost, B. Scholts, C.J. Kooijman and P.W. Serruys (1983a).
'Transfer function of the X-ray-cine-video chain applied to digital processing of coronary cineangiograms',
In: **Digital Imaging in Cardiovascular Radiology**
(P.H. Heintzen and R. Brennecke, editors),
Stuttgart: Georg Thieme.

J.H.C. Reiber (1984).
'Review of methods for computer analysis of global and regional left ventricular function from equilibrium gated blood pool scintigrams',
In: **Nuclear Imaging in Clinical Cardiology**
(M.L. Simoons and J.H.C. Reiber, editors),
Dordrecht: Martinus Nijhoff.

J.H.C. Reiber, C.J. Kooijman, C.J. Slager, J.J. Gerbrands, J.C.H. Schuurbiers, A. den Boer, W. Wijns and P.W. Serruys (1984a). 'Computer assisted analysis of the severity of obstructions from coronary cineangiograms: a methodological review', **Automedica**, Vol. 5, pp. 219 - 238.

J.H.C. Reiber, C.J. Kooijman, C.J. Slager, J.J. Gerbrands, J.C.H. Schuurbiers, A. den Boer, W. Wijns, P.W. Serruys and P.G. Hugenholtz (1984b). 'Coronary artery dimensions from cineangiograms - methodology and validation of a computer-assisted analysis procedure', **IEEE Transactions on Medical Imaging**, Vol. MI-3, pp. 131 - 141.

J.H.C. Reiber, P.W. Serruys and C.J. Slager (1986). **Quantitative Coronary and Left Ventricular Cineangiography**, Dordrecht: Martinus Nijhoff.

J.H.C. Reiber and P.W. Serruys (editors)(1986a). **State of the Art in Quantitative Coronary Arteriography**, Dordrecht: Martinus Nijhoff.

J.H.C. Reiber, C.J. Kooijman, C.J. Slager, J.J. Gerbrands, A. den Boer, J. van Ommeren, F. Zijlstra and P.W. Serruys (1987). 'Quantitative digital angiographic techniques', In: **Coronary Circulation** (J.A.E. Spaan et al.(editors)), Dordrecht: Martinus Nijhoff.

A.E.M. Reijs, J.H.C. Reiber, P. Fioretti, J.J. Gerbrands, M.L. Simoons and P.P.M. Kooij (1986). 'Thallium-201 tomography: developments towards quantitative analysis', In: **Signal Processing III: Theories and Applications** (I.T. Young et al., editors), Amsterdam: North-Holland.

A. Rosenfeld (1969). **Picture Processing by Computer**, New York: Academic.

A. Rosenfeld, R.A. Hummel and S.W. Zucker (1976). 'Scene labeling by relaxation operations', **IEEE Transactions on Systems, Man, and Cybernetics**, Vol. SMC-6, pp. 420 - 433.

A. Rosenfeld and L.S. Davis (1979). 'Image segmentation and image models', **Proceedings IEEE**, Vol. 67, pp. 764 - 772.

A. Rosenfeld and A.C. Kak (1982). **Digital Picture Processing**, New York: Academic.

B.J. Schachter, A. Lev, S.W. Zucker and A. Rosenfeld (1977).
'An application of relaxation methods to edge reinforcement',
IEEE Transactions on Systems, Man, and Cybernetics,
Vol. SMC-7, pp. 813 - 816.

L.L. Scharf and H. Elliott (1981).
'Aspects of dynamic programming in signal and image processing',
IEEE Transactions on Automatic Control, Vol. AC-26, pp. 1018 - 1029.

W.F. Schreiber and D.E. Troxel (1985).
'Transformation between continuous and discrete representations of images: a perceptual approach',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-7, pp. 178 - 186.

G.B. Shaw (1979).
'Local and regional edge detectors: some comparisons',
Computer Graphics and Image Processing, Vol. 9, pp. 135 - 149.

P.C.W. Sommen (1979).
Relaxation Methods in Edge Detection,
Internal report (in Dutch), Delft University of Technology.

J.P.P. Starink (1986).
Experiments with Dynamic Programming Edge Detection,
Internal report (in Dutch), Delft University of Technology.

J.P.P. Starink (1987).
3-D Contour Detection by means of Dynamic Programming,
M.Sc. thesis (in Dutch), Delft University of Technology.

D.E. Troxel and C. Lynn (1978).
'Enhancement of news photos',
Computer Graphics and Image Processing, Vol. 7, pp. 266 - 281.

S. Ullman (1979).
'Relaxation and constrained optimization by local processes',
Computer Graphics and Image Processing, Vol. 10, pp. 115 - 125.

A.J. Viterbi (1967).
'Error bounds for convolutional codes and an asymptotically optimum decoding algorithm',
IEEE Transactions on Information Theory, Vol. IT-13, pp. 260 - 269.

D.L. Waltz (1975).
'Understanding line drawings of scenes with shadows',
In: **The Psychology of Computer Vision** (P.H. Winston, editor),
New York: McGraw-Hill.

S.W. Zucker (1976).
'Relaxation labeling and the reduction of local ambiguities',
In: **Proceedings 3rd International Joint Conference on Pattern Recognition,** San Diego, pp. 852 - 861.

S.W. Zucker, E.V. Krishnamurty and R.L. Haar (1978).
'Relaxation processes for scene labeling: convergence, speed and stability', **IEEE Transactions on Systems, Man, and Cybernetics,**
Vol. SMC-8, pp. 41 - 48.

S.W. Zucker and J.L. Mohammed (1978a).
'Analysis of probabilistic relaxation labeling procedures',
In: **Proceedings IEEE Computer Society Conference on Pattern Recognition and Image Processing,** Chicago, pp. 307 - 312.

S.W. Zucker, Y.G. Leclerc and J.L. Mohammed (1981).
'Continuous relaxation and local maxima selection: conditions for equivalence',
IEEE Transactions on Pattern Analysis and Machine Intelligence,
Vol. PAMI-3, pp. 117 - 127.

SAMENVATTING

Het multidisciplinaire vakgebied van de digitale beeldverwerking richt zich op de methodologie en de technologie van het bewerken van visuele beelden, hetzij met behulp van vrij-programmeerbare computers, of met behulp van processoren of elektronische apparaten die ontworpen zijn voor het uitvoeren van specifieke taken. Het doel van deze digitale verwerking kan velerlei zijn, maar we richten onze aandacht hier op het met algoritmen analyseren van de beeldinhoud, teneinde daar informatie aan te onttrekken omtrent de door het beeld gerepresenteerde verschijnselen. Beeldanalyse kan op die manier beschreven worden als een transformatie die het beeld omzet in aan het beeld ontleende gegevens, zoals een aantal meetwaarden, een verzameling herkende objecten, of zelfs een beschrijving van de afgebeelde verschijnselen.

Een van de belangrijkste stappen in dit analyse-proces is de segmentatie van het beeld, d.w.z. het teweegbrengen van een partitie van het beeldvlak in deelgebieden die volgens een te kiezen criterium homogeen genoemd kunnen worden. Het resultaat van de segmentatie-stap is dus een plattegrond van de onderscheiden gebieden. De bedoeling is natuurlijk dat deze plattegrond betekenisvol is ten opzichte van de afgebeelde verschijnselen, maar dat kan in het algemeen pas later in het analyse-

proces worden vastgesteld, omdat hiervoor een veel hoger niveau van abstractie vereist is.

Dit proefschrift is gewijd aan de segmentatie van beelden die ernstig verstoord zijn door ruis. We beperken ons daarbij tot twee-dimensionale tijd-invariante grijstoonbeelden die een afbeelding zijn van geïsoleerde objecten. Hoewel de hier besproken segmentatie-methoden ook toepasselijk zijn op andersoortige beelden, impliceert deze laatste restrictie dat men het hier bestudeerde segmentatie-probleem kan beschrijven als het automatisch intekenen van de contouren van de afgebeelde objecten in sterk door ruis verstoorde beelden. Eenvoudige traditionele segmentatie-methoden voldoen in het algemeen slecht als er veel ruis is, hetgeen vooral veroorzaakt wordt doordat de ruis de samenhang van de structuren in het beeld aantast. De invloed van de ruis kan natuurlijk worden verminderd, althans tot op zekere hoogte, door het toepassen van geschikte voor- of nabewerkingstechnieken. In deze studie concentreren we ons op pogingen om de prestaties van de segmentatie-stap zelf te verbeteren. Het ligt voor de hand dat de resultaten van de hier besproken methoden nog verder verbeterd kunnen worden door ze te combineren met geschikte voor- of nabewerkingstechnieken.

Doordat de ruis de samenhang van de beeldstructuren aantast, kunnen de beslissingen binnen de segmentatie-procedure niet onafhankelijk per discreet beeldelement (pixel) genomen worden; er moet gebruik gemaakt worden van de ruimtelijke of spatiële context in het beeld. Sequentiële segmentatie-methoden zijn in dit opzicht aantrekkelijk. Het gebruik van sequentiële methoden die primair gericht zijn op het detecteren van coherente gebieden resulteert vaak in een onnauwkeurige lokalisering van de gebiedsgrenzen. We geven hier dan ook de voorkeur aan sequentiële methoden die primair gericht zijn op detectie van de grenzen. Er wordt in detail een methode beschreven en geanalyseerd waarbij met behulp van de optimaliseringstechniek van het dynamisch programmeren en onder gebruikmaking van expliciet ingebrachte a priori-kennis, de volgens een te kiezen criterium optimale gebiedsgrens wordt gevonden. Daarnaast wordt aandacht gegeven aan zogenaamde "continuous relaxation labeling procedures". Dit zijn weliswaar geen sequentiële maar iteratieve

parallelle procedures die eveneens gebruik maken van a priori-kennis en van de context.

Hoofdstuk 1 van het proefschrift bevat een korte inleiding over digitale beeldsegmentatie. Een aantal ideeën en begrippen wordt besproken en er wordt een methodologisch overzicht gegeven van segmentatie-technieken.

In hoofdstuk 2 wordt de inleiding voortgezet maar nu gericht op modelvorming. De zogenaamde "replacement models" worden beschreven en een aantal aspecten van beeld-digitalisering wordt besproken. Vervolgens onderbouwen we onze voorkeur om een gebiedsgrens te definiëren als een aaneenschakeling van randpixels in plaats van als een aaneenschakeling van micro-randen tussen buurpixels behorend tot verschillende gebieden. De schattingstheoretische benadering van het grensdetectie-probleem wordt besproken aan de hand van enige belangrijke referenties.

Hoofdstuk 3 is gewijd aan de optimalisering met behulp van dynamisch programmeren. Na een literatuuroverzicht ontvouwen we onze methode, die er vooral op gericht is de oplossingsruimte zoveel mogelijk te beperken. Er wordt een verwachtingsvenster gedefinieerd en het daarbinnen gelegen deel van het beeld wordt geometrisch getransformeerd tot een rechthoekige matrix. Vervolgens wordt een overeenkomstige matrix van kostencoëfficiënten opgesteld. Met behulp van dynamisch programmeren wordt het optimale pad door de matrix bepaald. Dit pad wordt teruggetransformeerd naar het oorspronkelijke beelddomein en vormt daar de gezochte gebiedsgrens. De methode wordt in detail beschreven en geanalyseerd. Er wordt vooral aandacht gegeven aan de geometrische transformatie. Er worden eisen opgesteld met betrekking tot de curvatuur en de globale vorm van een gebiedsgrens, waaraan voldaan moet zijn om de methode toe te kunnen passen.

In hoofdstuk 4 wordt het principe van "continuous relaxation labeling" besproken. In het bijzonder wordt aandacht gegeven aan een speciale versie van een relaxatie-algoritme voor de detectie van grenzen die een praktisch uitvoerbare en aantrekkelijke implementatie mogelijk maakt. Het blijkt dat een groot gedeelte van de vereiste berekeningen kan worden geformuleerd in termen van spatiële convoluties en puntbewerkingen op paren van beelden. Dergelijke operaties kunnen zeer snel worden uit-

gevoerd op moderne digitale beeldverwerkings-systemen die hiervoor speciale processoren hebben. Het bepalen van de initiële label-toewijzing voor het iteratieve proces wordt in detail besproken.

In hoofdstuk 5 worden de experimenten beschreven waarmee de in de voorgaande hoofdstukken ontwikkelde methoden zijn geëvalueerd. Op basis van synthetische testbeelden met verschillende waarden van de signaal-ruisverhouding worden de segmentatie-resultaten kwantitatief beoordeeld in termen van de geschatte detectiekans, de geschatte loos-alarmkans en de zogenaamde "Figure of Merit" van Pratt. Er wordt aangetoond dat vooral bij zeer lage waarden van de signaal-ruisverhouding de methode van dynamisch programmeren veel beter voldoet dan de traditionele manieren van grensdetectie, ook als die worden gecombineerd met een relaxatie-procedure.

Onze methode van grensdetectie met behulp van dynamisch programmeren vereist het definiëren van een verwachtingsvenster. In hoodstuk 6 wordt een aantal manieren besproken om een dergelijk venster automatisch te bepalen. Een van de voorstellen behelst het uitvoeren van een ruwe pre-segmentatie van het beeld door middel van een gebieds-georiënteerde segmentatie-methode. Dergelijke methoden detecteren de aanwezigheid van gebieden dikwijls correct, maar leiden tot een onnauwkeurige lokalisering van de gebiedsgrenzen.

Hoofdstuk 7 is gewijd aan een tweetal medische toepassingen waarin de in dit proefschrift besproken methode van grensdetectie met behulp van dynamisch programmeren haar waarde reeds heeft bewezen. De eerste toepassing betreft de analyse van scintigrafische beelden van de linker hartkamer, de tweede de analyse van röntgenbeelden van de kransslagaderen.

De conclusies en een bespreking van de resultaten zijn opgenomen in hoofdstuk 8.

ACKNOWLEDGEMENTS

From this thesis, it is quite evident how much value I attach to context, and on its long awaited completion I would like to thank all those who have contributed in one way or another.

First of all, I wish to express my gratitude towards both the academic staff and the supporting staff of the Laboratory for Information Theory, headed until recently by professor IJsbrand Boxma, for their continuing support and for providing a very stimulating environment. Special thanks are due to professor Eric Backer, who put me on the image processing track in the first place and who has been an enthusiastic guide and companion ever since. I am particularly grateful to our students for their active participation and important contributions.

I deeply appreciate the close and amicable cooperation, both in research and in education, with professor Ian T. Young of the Pattern Recognition Section of the Department of Applied Physics, who has had a profound influence ever since he was a visiting professor in 1975. I would like to extend my thanks to his entire staff, as well as to the people at the Institute of Applied Physics TNO-TH, with whom we co-operate in the Delft Center for Image Processing CBD.

In a slightly larger context, and focussing on the subject of this thesis, I gratefully acknowledge the continuing cooperation with the Laboratory for Clinical and Experimental Image Processing of the Thorax-center of the Erasmus University in Rotterdam, and particularly with its director Hans Reiber. The restricted-search dynamic programming method of boundary detection discussed in this thesis has emerged from this cooperative project. The diploma work of Cees Hoek on the quantitative analysis of cardiac scintigrams, co-supervised by Lie Swan Peng, is really the cradle of the method. I gratefully acknowledge their crucial contributions. The extension of the method to coronary cineangiograms is particularly due to Cees Kooijman. The contributions by Fred Booman, Peter van Leeuwen and Jan van Ommeren are also gratefully acknowledged. The extension of the method to three-dimensional tomographic scintigraphy is indissolubly connected with the names of Koos Blokland, Ambroos Reijs and Pascual Starink. The support of the Dutch Heart Foundation in various aspects of this cooperative project has been indispensable.

The first implementation of the method on the computer facilities of our own laboratory was programmed by Gilbert Houtekamer. The extensive evaluation study described in Section 5.2 was performed by Wim van der Hoeven. I am particularly thankful for his stimulating enthusiasm and his great effort. Additional experiments have been performed by Gerard Conijn and Zouhair Bedawi. The fruitful discussions with André Jongeling and Inald Lagendijk, who used the method in some other applications, were very enjoyable. Our early experiments with continuous relaxation labeling procedures were carried out by Piet Sommen.

Without the contributions from all these people I could not have written this thesis. I would also like to thank Xiang Sheng Cheng and Ad de Ridder for their help in preparing the photographs and Mirella van Velzen and Annett Bosch for their great effort in typing the manuscript.

Finally, I would like to take the opportunity to acknowledge the fruitful cooperation on research topics outside the scope of this thesis with Jan Biemond, Peter van Otterloo, Fred Booman, Cees Slump, Peter Hoogeboom, Xiang Sheng Cheng and professor Jaap van den Herik.

CURRICULUM VITAE

Jan Jacob Gerbrands was born in The Hague, the Netherlands, in 1948. In 1965 he obtained the HBS-B diploma from the Rembrandt Lyceum in Leyden. He then spent a year in the U.S.A. as an exchange student in the program of AFS International Scholarships. Subsequently he enrolled in the Department of Electrical Engineering of the Delft University of Technology, from which he received the degree of Ingenieur in 1974.

From 1972 till 1974 he worked as a teaching assistant in the Laboratory for Information Theory of the above department. After his graduation in May 1974 he joined the faculty of the same group, then headed by professor IJ. Boxma. In 1981 he was promoted associate professor.

He is the (co-)author of more than 70 publications in international scientific journals and conference proceedings in the field of digital image processing, focussing on image analysis and pattern recognition, with particular emphasis on applications in medicine and remote sensing.

Since 1977 he teaches regular courses in digital image processing, in cooperation with professor E. Backer and with professor I.T. Young. He supervised the diploma work of more than 70 students. Since 1982 he is on the teaching staff of the Special Course on Image Processing for Industrial Applications, organized by the Delft Center for Image Proces-

sing CBD, and he is a member of the governing board of this center which coordinates the external activities of the image processing groups in Delft.

He is an active member of the European Association for Signal Processing EURASIP. He is the editor of the 1983 EURASIP Directory of European Signal Processing Research Institutions, published by Reidel. He was the secretary of the 3rd EURASIP signal processing conference EUSIPCO-86, and he is co-editor of the proceedings which were published in bookform under the title 'Signal Processing III: theories and applications' by North-Holland. He served on the organizing committee of the 5th IEEE-ASSP/EURASIP Workshop on Multidimensional Signal Processing. He is a member of the governing board of the Dutch Society for Pattern Recognition and Image Processing NVPHBV.