Hierarchical Image Segmentation by Stepwise Optimization

New edition of 1984 Thesis

Jean-Marie BEAULIEU

BeaulieuJM.ca

ISBN 978-1-7388812-0-8
Publisher Jean-Marie Beaulieu

Abstract

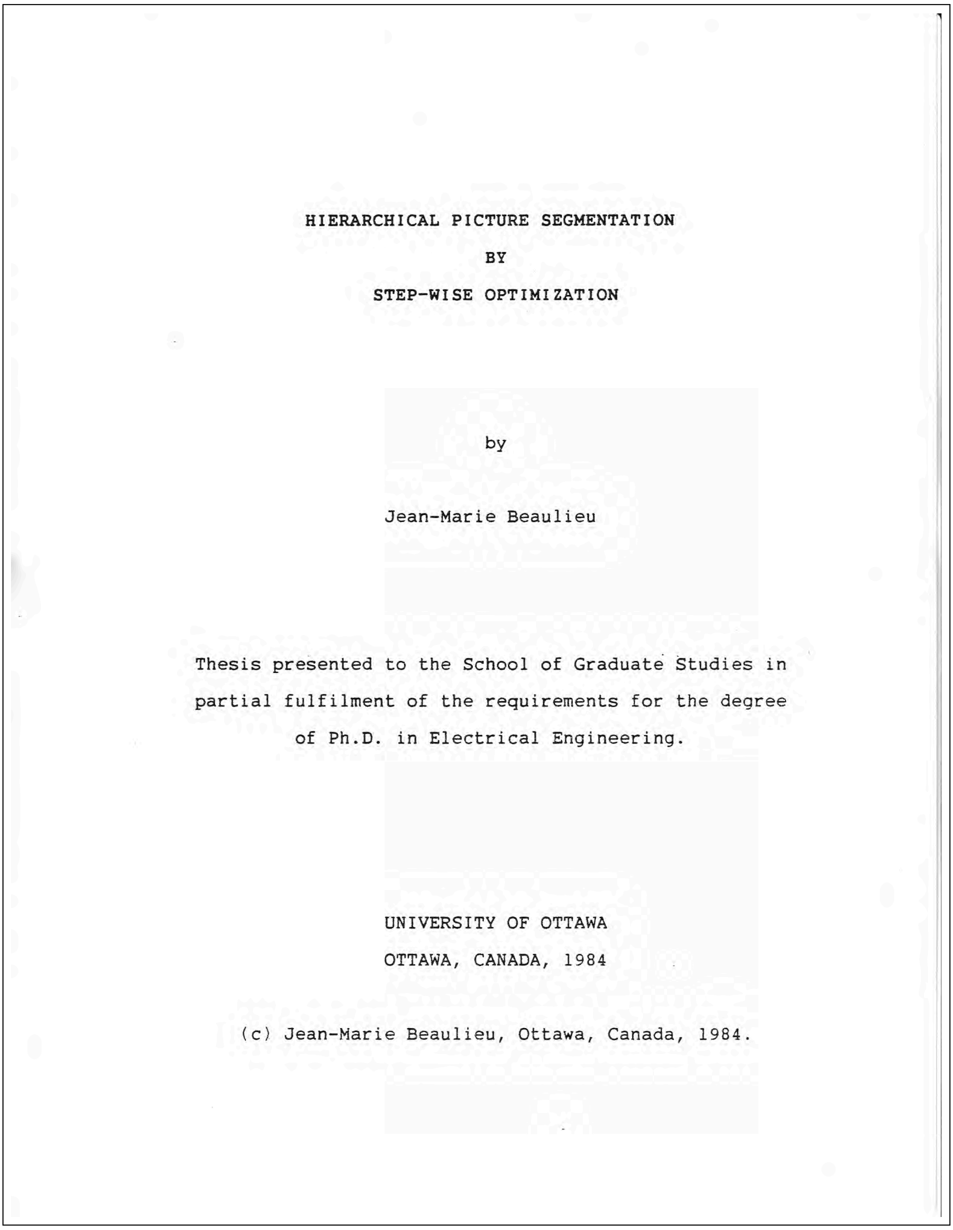
The survey of image segmentation considers four different approaches: pixel classification, pixel linking and region growing, hierarchical segmentation, and segmentation optimization. A new Hierarchical Stepwise Optimization (HSO) algorithm is proposed, which combines these last two approaches, The algorithm employs a sequence of optimization processes to produce a hierarchical segmentation. Starting with an initial image partition, two segments are then merged at each iteration by using an optimization process to select the segment pair that minimizes a "stepwise criterion." The algorithm is then employed for piecewise image approximation where the stepwise criterion is derived from the global criterion, the overall approximation error. The stepwise criterion is then related to statistical hypothesis testing, and it is shown how the probability of error can be minimized in a stepwise fashion. It is also shown experimentally how convenient stopping points in the hierarchy can be found from the criterion values. Different criteria are tested on Landsat and SAR imagery.

Preface

Even after more than 35 years, I think that the thesis will make a good reading. I did hope to have the text converted to a modern document file format instead of a simple binary scan of the thesis. In the new edition, a modern proportional spacing font is used instead of a mono space font. Latex is used to have better equation rendering. The figures are gray level scans from the original paper document. The word "picture" is replaced by "image," "step-wise" by "stepwise," "HSWO" by "HSO" and few other modifications. Otherwise, the original text is preserved. To have a book like format, B5 paper size is used, with a narrow margin.

Jean-Marie Beaulieu

Thesis Front Page



HIERARCHICAL PICTURE SEGMENTATION
BY
STEP-WISE OPTIMIZATION

by

Jean-Marie Beaulieu

Thesis presented to the School of Graduate Studies in partial fulfilment of the requirements for the degree of Ph.D. in Electrical Engineering.

UNIVERSITY OF OTTAWA
OTTAWA, CANADA, 1984

Contents

List of Figures

List of Tables

List of Symbols

B_i	List of neighbors of a segment
$C_{i,j}$	Stepwise criterion
C_{min} or $C_{min,k}$	Minimum value of the criterion
D_i	Segment descriptive parameters
$d_{i,j}$	Statistic, difference of segment means
$d^*_{i,j}$	Normalized statistic
$d^*_{min,k}$	Minimum value of the statistic $d^*_{i,j}$
d_{true} and d^*_{true}	True value of the statistics
$\mathrm{ERF}\,(\cdot)$	Error function
$\bar{f}(x,y)$	Image function
$\mathbb{G}(\mathbb{P})$	Image approximation error
$\mathbb{H}(S_i)$	Segment approximation error
$\mathcal{H}_0$, $\mathcal{H}_1$ and $\mathcal{H}_A$	Statistical hypotheses
HSO	Hierarchical Stepwise Optimization
$\mathfrak{m}_i$	True gray level value for the region R_k
Minimum{...}	Minimum value of the set {...}
MSS	Multi-Spectral Scanner
N_i	Size of the segment S_i
$\mathcal{N}(\mu,\sigma^2)$	Gaussian distribution
$\mathbb{P}$ or P_n	Image partition
$\mathrm{P}_{\mathcal{H}0}$ and $\mathrm{P}_{\mathcal{H}1}$	Probabilities functions of $d^*_{i,j}$ under $\mathcal{H}_0$ and $\mathcal{H}_1$ hypotheses
PBHM	Predicate Based Hierarchical Merging
$Q(.)$	Logical predicate equation
R_k	True image region
$r_i(x,y)$	Segment approximation function
S_i or S_i^k	Image segment
SAR	Synthetic Aperture Radar
SSE	Sum of Squared Errors
τ	Threshold value
w_λ	Weighting factor for an image channel
α	Probability of type I error
β	Probability of type II error
$\bar{\mu}_i$	Mean value of the segment S_i
σ^2	Noise variance

Acknowledgements

The author wishes to express his sincere gratitude and respect to his thesis advisor Dr. Morris Goldberg, for his patient guidance provided in the course of this work and for the stimulating discussions held throughout.

Thanks are also due to various members of the staff of the Electrical Engineering Department for their cooperation.

The help and cooperation of the graduate students of the department is greatly appreciated.

The author wishes also to acknowledge support from Dr. David Goodenough and the Canada Centre for Remote Sensing.

Chapter 1

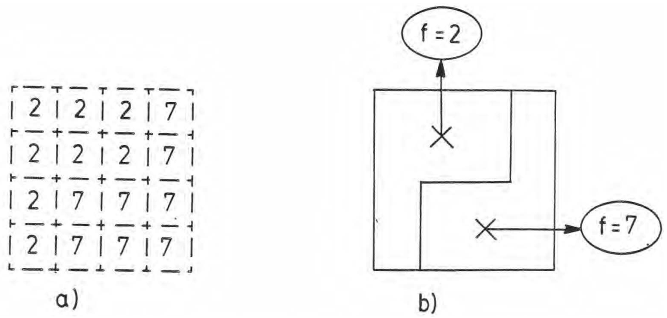
INTRODUCTION

An image can be considered as a 2-dimensional function, $\bar{f}(x,y)$, where the domain corresponds to the image plane and the range to the gray level intensity. In a digital image, the plane is divided into elementary regions, called pixels, identified by the spatial coordinates (x,y), $\mathbb{I} = \{(x,y)\}$. The word ***pixel*** is often used to designate not only the elementary region (x,y) but also the associated spectral value $\bar{f}(x,y)$. Note that for colour and multi-spectral images, $\bar{f}(x,y)$ is a vector.

Image segmentation is the division of the image into different regions, each having certain properties. For example, in Figure 1.1 the pixels with the same gray level are merged to form regions. A segmentation involves a partition $\mathbb{P}$ of the image plane $\mathbb{I}$ into disjoint regions or segments S, i.e. $\mathbb{P} = \{S_1, S_2 \ldots S_n\}$ such that $\bigcup S_i = \mathbb{I}$ and $S_i \cap S_j = \varnothing$ for $i \neq j$. It also implies a description of each segment. In the example, each segment could be described by its gray level value. In more complex cases, a segment could be represented by the mean value; the probability distribution of gray levels $p(\bar{f})$ inside a segment; the functional approximation parameters; the segment shape parameters; to give but a few examples. Once the pixel set that forms a segment is known, it is generally easy to calculate the descriptive parameters. However, the determination of the image partition that will yield segments with some predefined properties is a more complex and difficult problem.

Image segmentation employs many of the approaches and techniques of pattern recognition such as data classification and clustering. Hence, image segmentation can be regarded as the recognition of segment patterns. However, image segmentation possesses its own difficulties; namely, the large size of the image and the spatial relationship between pixels.

Pattern classification and clustering involves separating data or patterns, $\bar{v}_i$, into classes or clusters. A sample pattern $\bar{v}_i$ is considered as a point



2	2	2	7
2	2	2	7
2	7	7	7
2	7	7	7

Figure 1.1: Image segmentation: a) pixel values, b) image partition and segment descriptions.

in an n-dimensional sample, or feature space, $V = V^1 \times V^2 \times \ldots V^n$, $\bar{v}_i = (v_i^1, v_i^2 \ldots v_i^n)$ and $\bar{v}_i \in V$. Classification techniques assign a class to a sample according to its position in the feature space V, while clustering techniques attempt to separate and identify clumps of sample points.

In image segmentation, it is impossible to consider the whole image as a sample pattern because of the size of the image. Instead, each pixel is regarded as a sample pattern, $\bar{v}_i$, in an n-dimensional space, each dimension corresponding to a different colour or spectral band. The information contained in the set $\{\bar{v}_i\}$, without consideration of pixel position, is called the spectral information. Classification and clustering techniques can then easily be applied to these spectral values $\{\bar{v}_i\}$.

The spatial information of an image concerns the inter-relation between pixels, or the dependence of a pixel value on its position and on the values of its neighbours. The spatial information is an important aspect of an image, and must be taken into account if good image segmentation results are to be expected. It is shown that classification or clustering techniques can still be used for image segmentation if the spatial information is included as new features or in the distance measures used.

The first part of this thesis consists in a survey of image segmentation techniques, while the second part presents a new hierarchical segmentation algorithm. In the survey, four general approaches for image segmentation are distinguished according to the adopted definitions of segment :

1) Pixel classification (Chap. 2)

A region is assumed to be composed of pixels belonging to the same statistical population or class, each class having a specific pixel value range. Spatial aspects are included by the addition of feature values to the pixel gray level values. Statistical decisions are used to find the class membership of each pixel. Pixels with the same class label are grouped to form regions. Histograms can be used to estimate the class probabilities and thresholds can be employed for the assignment of class labels.

2) Pixel linking and region growing (Chap. 3)

A region is considered as a group or clump of pixels in a combined spatial-spectral space where both the spatial and spectral distances between pixels inside a region (cluster) must be small. This implies that the pixels must be adjacent and have similar gray levels. In pixel linking, a pixel-to-pixel measure is used to join a pixel with its closest neighbours. Whereas, a region growing approach relies upon a pixel-to-region measure for annexing the neighbouring pixels which are similar.

3) Hierarchical segmentation (Chap. 4)

The image regions are assumed to form a hierarchy. The hierarchy is produced by the merging or splitting of segments. A predicate is employed to determine if a segment must be divided into sub-parts, or if it must be merged with an adjacent one. The predicate evaluates the similarity of segments in considering, for example, the segment means and variances, or the weakness of segment boundaries.

4) Optimal image segmentation (Chap. 5)

It is often desired that the image regions satisfy a global or overall requirement. Such a requirement is usually defined by a cost function (global criterion) and an optimization process is then used to find the best solution. The segment approximation error, the image roughness, and the segment contour smoothness, are examples of criteria used in image segmentation. In practice, locating the global optimum partition is not feasible. In its place, two techniques have been proposed: the utilization of one-dimensional optimization techniques and the finding of a local optimum from an initial image partition by iterative processes.

Image segmentation algorithms have already been reviewed in many other papers [12], [25], [26], [31], [38], [60], [76], [77], [102]. Each emphasizes different aspects of the image segmentation problem and uses a different manner to classify algorithms. For example, Kanade [38] identifies three levels of knowledge, that segmentation algorithms can exploit: the signal, the physical and the semantic level. Fu and Mui [25] distinguish three categories of segmentation techniques: 1) characteristic feature thresholding or clustering, 2) edge detection, and 3) region extraction. Rosenfeld and Davis

[77] regard the assumptions or models that an image should satisfy in order to apply a particular technique. They examine statistical and spatial models. The present survey complements the preceding ones by comparing the segmentation algorithms with those for data classification and clustering. This allows the exploitation of knowledge about data clustering for characterizing the image segmentation algorithms. This can also be useful for suggesting a new exploratory area.

It should be noted that the chosen classification scheme involves only general low-level ways for describing regions or image segmentations. Knowledge driven, expert systems, ad-hoc constructs and heuristic programmings that adapt a program to a particular application are not discussed.

The second part of the thesis presents a new hierarchical segmentation algorithm based upon sequential optimization. The algorithm starts with an initial image partition, and at each iteration, merges two segments. An optimization process is used to select the segment pair that minimizes a ***stepwise criterion***, $C_{i,j}$ corresponding to the cost of merging the segment S_i with the segment S_j. The properties of this algorithm and its operation on real data are analysed in detail. It is shown that the algorithm is a valuable tool, and produces good segmentation results.

The contributions of each chapter of the second part are now outlined.

1) A hierarchical image segmentation algorithm (Chap. 6)

The Hierarchical Stepwise Optimization algorithm (HSO) is described in detail. The algorithm is designed so as to reduce the computing time. An essential reduction results from the fact that only adjacent segments can be merged. Moreover, recalculations are avoided by making explicit the information needed, and by updating only the values that are modified by a segment merger. The HSO algorithm is also compared with hierarchical segmentation algorithms based upon predicate equations.

2) Optimization and segment hierarchy (Chap. 7)

Image segmentation can advantageously be stated as a global optimization problem. The finding of the global optimum is generally not feasible. The stepwise optimization (HSO) algorithm is therefore presented as a suboptimal alternative. The stepwise criterion is then derived from the global criterion, and equated to the increase of the global criterion produced by the merging of two segments.

3) Probability of error in hierarchical segmentation (Ch. 8)

Image segmentation can be regarded as a hypothesis testing process which merges two segments only if they belong to the same region. For hierarchical segmentation, the advantages of minimizing the probability of dissimilar segment merges, at each step, are stressed. This is achieved by the proposed stepwise optimization (HSO) algorithm which finds and merges the most similar segment pair. The probability of stepwise error (i.e. the probability of merging dissimilar segments) is also calculated.

4) Algorithm operation and criterion selection (Chap. 9)

The operation of the segmentation algorithm on remote sensing images is analysed. In particular, the selection of a stopping point for the algorithm is examined. The problem of selecting the appropriate segment models and the corresponding stepwise criteria are also discussed and illustrated by experiments on a Landsat satellite image. The algorithm is shown to be capable of adaptation to different segmentation tasks.

Part I

Survey of Image Segmentation

Chapter 2

Pixel Classification

Classification techniques play an important role in pattern recognition and image analysis [17], [25], [77]. The classification approach assumes that an observed sample or pattern $\bar{v}_i$ has been generated by a unknown class or population from a set of n possible classes, $C_1, C_2, ...C_n$, with known characteristics. A classification process examines the value of the sample $\bar{v}_i$ to identify the class C_k.

A simple model for image segmentation can be derived from the classification approach. It assumes that a region is composed of connected pixels belonging to the same class C_k. Moreover, it supposes that the observed pixel values $\bar{v}_i = \bar{f}(x_i, y_i)$ are only dependent upon their class memberships. Therefore, the segmentation problem reduces to discovering (or estimating) the true class memberships of pixels from their observed values $\bar{v}_i$. Figure 2.1 represents a pixel classification process where the decision consists in assigning a class label to each pixel. The parameters required for the decisions are given by the user and/or calculated from the data.

For example, Figure 2.2 shows one line of a remote sensing image (the .60-.70 urn band of a Multi-Spectral Scanner image taken by the Landsat-I satellite in August 1972, frame E-1031-17265). Two classes can be considered: water and vegetative cover. Knowing that water produces lower values than vegetation, the central low value pixels (lower than 25) can be regarded as water, the other ones as vegetation. Regions are then formed by connecting pixels belonging to the same classes. Thus, the regions are strictly defined by the classes.

In the following sections, it is first shown that pixel classification can be regarded as a statistical decision process, where the classes are defined by their probability density functions. Classification techniques based upon the spectral histograms which are used to estimate the probability density functions are then considered. The exploitation of the spatial information

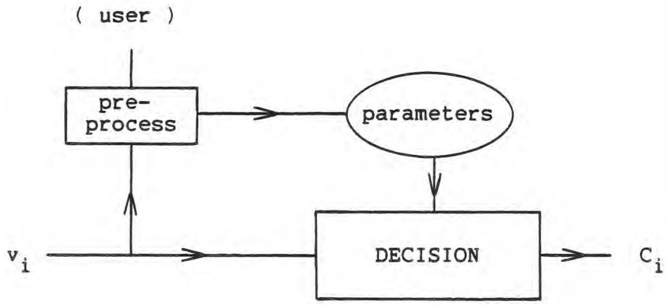


Figure 2.1: Block diagram of a pixel classification process. The decision parameters are first calculated from the data under user control. Then, the decision assigns a class label to each pixel.

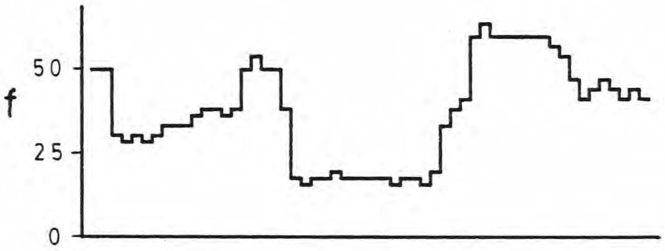


Figure 2.2: Intensity levels of one line of a remote sensing image.

by the addition of new features is examined. Finally, the limitations of this approach are discussed.

2.1 Statistical decision

Statistical decision can be used to estimate the class membership of an observed sample value $\bar{v}_i$ [17]. Classes are then viewed as statistical populations, defined by their probability density functions $P(\bar{v}_i|C_k)$. These density functions are used to determine the class membership for the pixel value $\bar{v}_i$. For example, a maximum likelihood classifier selects the class which maximizes $P(\bar{v}_i|C_k)$, but the expected class C_k will not always correspond to the true class. Figure 2.3 presents two density functions, P_1 and P_2, associated with two classes, C_1 and C_2. Let C_2 be the true state of nature, and suppose that the sample $\bar{v}$ has been drawn. $P_1(\bar{v})$ is higher than $P_2(\bar{v})$, and therefore the class C_1 is selected, producing a classification error. Such classification errors can be avoided only when the probability densities $P(\bar{v}|C_k)$ do not

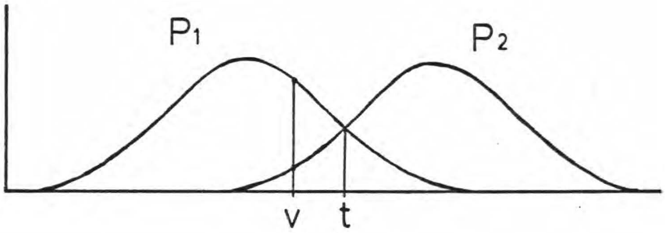


Figure 2.3: The probability density functions of two classes. Note the overlapping of density functions, v represents a sample point, and t is a threshold value.

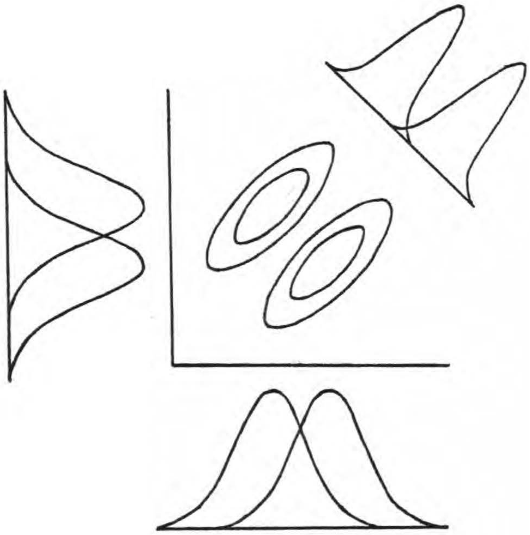


Figure 2.4: Overlapping of probability density functions in one and two dimensional spaces. There is more overlap for both the horizontal and vertical projections than for the oblique projection. However, the two-dimensional density functions show less overlap.

overlap.

Classification error reduction can result from the appropriate selection of variables or data features. For example, in Figure 2.4, the oblique projection produces two relatively distinct one dimensional density functions. Whereas, there is much overlap for both the horizontal and vertical projections. Addition of new dimensions to the sample vector $\bar{v}_i$ can also be used to reduce classification errors. For example, in Figure 2.4, the two-dimensional probability density functions show less overlap.

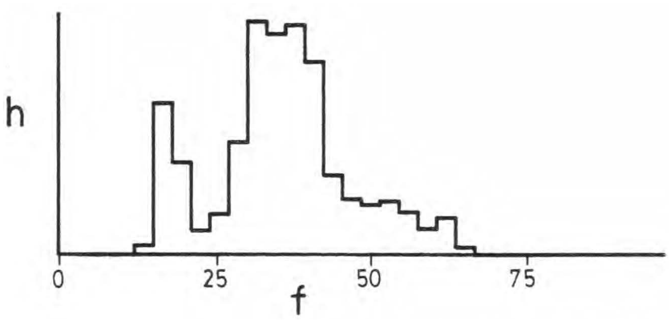


Figure 2.5: Histogram of 20 lines per 55 pixels of a remote sensing image. One line of the image is shown in Figure 2.2. The left mode corresponds to water area, and the right mode to vegetative cover.

2.2 Histogramming

In many applications, the class conditional probability density functions $P(\bar{v}|C_k)$ are unknown, and must be estimated from the data. The histogram $h(\bar{v})$ is generally used as an estimate of the composite probability density,

$$h(\bar{v}) = \sum_m p(\bar{v}|C_m)\ P(C_m) \tag{2.1}$$

It is usually assumed that the conditional densities $P(\bar{v}|C_m)$ are unimodal, and that the composite probability density is composed of many well distinct modes, one for each class. Hence, it is possible to identify these modes in the histogram $h(\bar{v})$, and use them to estimate $P(\bar{v}|C_k)$ [17].

A classification process can also be defined by a partitioning of the sample space [17], [91]. For example, in Figure 2.3, a maximum likelihood classifier assigns any sample v lower than t to class and those higher to class C_1 and those higher to class C_2. The histogram modes can therefore be employed to define such a partitioning of the sample space. In 1-D histograms, threshold values are used to divide the sample space into regions centered on these modes. As an example. Figure 2.5 shows the histogram of a remote sensing image (the .60-.70 um band of a Multi-Spectral Scanner image taken by the Landsat-I satellite in August 1972, frame E-1031-17265). The left-most mode corresponds to a water area. A threshold value located in the valley between modes, can be used to discriminate between water and vegetation. There are many well known techniques for threshold selection or spectral space partitioning [95], [94], [26], [42].

The quality of the histogram is important for class estimation and pixel classification. A large number of samples is needed to obtain a good estimate

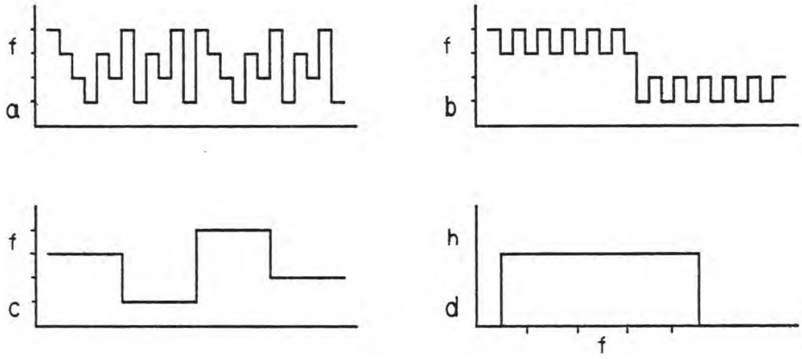


Figure 2.6: 1-D functions composed of one region (a), two regions (b), and four regions (c). The histograms (d) of these functions are identical.

of the composite probability density function, and the required number of samples grows exponentially with the number of dimensions of the sample space [43]. Therefore, in many cases, dimension reduction techniques must be employed [28], [13].

There are many applications where the utilization of the gray level value alone can be sufficient to discriminate between regions. In medical applications, for instance, the cytoplasm, the nucleus and the background of blood cells can be differentiated [4], [73], Color information can also be useful, e.g. for the recognition of red blood cells [11], [63], [80].

2.3 Spatial features

In many applications, regions cannot be found using only the spectral information; some spatial aspects must also be involved. This is illustrated by Figure 2.6 which represents three functions having the same unimodal histogram. Depending upon how the spectral values are distributed in the spatial space, these functions can be regarded as composed of 1, 2 or 4 distinct regions. Spatial information must, therefore, be added to classification techniques to take account of the pixel positions or of the gray level variations among adjacent pixels.

One way to treat the spatial information is to consider a group of pixels (e.g. 3x3) at a time. The pixel values are concatenated to form a unique sample vector $\bar{v}_c$. Good results have been reported when the segmentation process has been applied to these vectors [1], [87). However, the large number of dimensions in the vector $\bar{v}_c$ can reduce the accuracy of the histogram estimate and the reliability of the classification results. Therefore, a small number of parameters or features are often used to represent the cell and, thus, reduce the number of dimensions. Cell mean, variance, entropy, slope, and gradient are among the proposed spatial features. Techniques such as principal components and K-L transform to name but two, are available to select the features with the highest discrimination between classes or regions [13], [28], [81], [96], [86].

Edges are another important and widely used spatial feature. Edge point detection is closely related to segmentation, as edge points correspond to region boundaries. Edge detection searches for discontinuities in pixel values, i.e. important changes between adjacent pixels, while segmentation groups pixels with similar values into regions. Edge detection can be regarded as a statistical decision problem involving the partition of points into two classes: edge points or non-edge points. The decision is usually based upon a set of parameters calculated from a set of connected pixels; for example, gradient value, slope, variance, etc.. These parameters are then thresholded to yield edge points and non-edge points, Much work has been done on edge detection [15], [44], [47], [65], [75], [35).

Some heuristic or ad hoc processes have been proposed to improve the results obtained by the classification approach: pre-processing such as image filtering, post-processing such as image cleaning, multi-thresholding or variable thresholding [9], [13], [26], [32], [42], [57], [96], [105]. One particular approach proposes the combination of pixel classification and edge detection. For example, the edge value can be used for selecting the threshold for pixel classification, or the classification can be done in the spectral-edge space [9], [53], [56], [54]. Region contours found from pixel classification and edge detection can also be combined [50], [51].

2.4 Limitations

Three major limitations of the pixel classification approach are now outlined. The first one is related to the use of the histogram for estimating the probability densities. The second concerns the selection of features. The last limitation, which is not restricted to the pixel classification approach, regards the use of thresholding processes.

The histogram is inadequate for handling all the spatial information. It represents global information and involves a summation over the whole image. This does not seem to be appropriate for spatial information, which can be more local in nature. This lack of spatial information in the histogram is illustrated by the example of Figure 2.6 where the different cases are indistinguishable from the histogram. An additional difficulty is that a small region can have its histogram mode hidden by those of larger regions [55], [83].

The second limitation concerns the selection of feature values to represent the spatial information. Since the number of dimensions of the sample space should be limited, we must therefore select the features which minimize the loss of information. However, a feature may perform well only on a part of an image. For example, the mean value, calculated from an image window, is meaningful only if the pixels involved belong to the same region. Thus, an ideal feature must adapt itself to the context. This requires decisions that cannot be reliably made at this early stage of image analysis [46], [81], [18].

The last limitation regards the use of thresholds that must be supplied by the user. Often, the effect of these thresholds are not well understood. and therefore, their selection can be difficult. In many cases, the user must carry out many experiments, in order to select the best values.

Chapter 3

Pixel Linking and Region Growing

The preceding chapter has presented a pixel classification approach, where the classes and the corresponding image regions are associated with modes in the histogram $h(\bar{v})$. There are severe limitations on the spatial information that can be handled by this approach. A clustering approach based upon a combined spectral/spatial distance measure can alleviate some of these shortcomings.

The clustering approach attempts to identify groups or clumps of sample points in the feature space [17], [91], [84]. A group or cluster is defined as a dense set of points separated from other groups, and the identification of clusters is based upon the distance between sample points. Thus, in a cluster, a point is surrounded by other similar points located at a small distance, while the distances between points of different groups are large. An example is shown in Figure 3.1.

For image segmentation purposes, a region can be regarded as a cluster. This implies that the distance between pixels inside a region must be small.

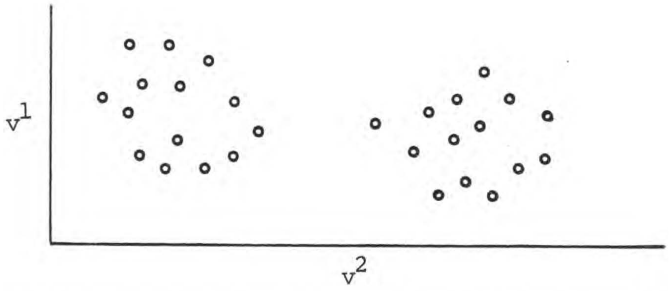


Figure 3.1: Data points forming two clusters.

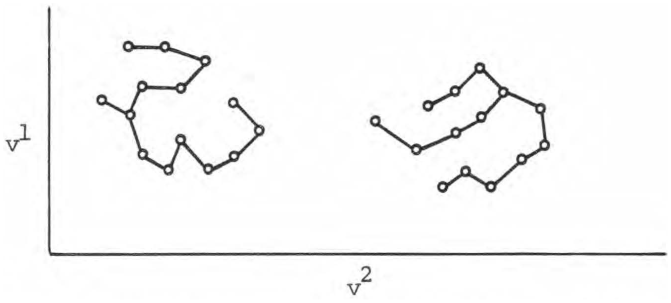


Figure 3.2: Data point linking for cluster detection. Note that each cluster forms a tree.

The pixels must be both spectrally and spatially close to each other. In other words, they must be adjacent and have similar gray level values.

A distance measure can be regarded as an evaluation of the similarity or "natural association" between pixels or group of pixels. The spatial distance $|(x_i, y_i) - (x_j, y_j)|$ can easily be combined with the spectral distance $|\bar{f}(x_i, y_i) - \bar{f}(x_j, y_j)|$ by treating these as two orthogonal vectors that have to be summed.

Two different kinds of segmentation algorithms are now examined: 1) pixel linking algorithms that are based upon a pixel-to-pixel measure, and 2) region growing algorithms which involve a pixel-to-region measure.

3.1 Pixel linking

Linking algorithms exploit the distance between sample points, $d(i, j) = |\bar{v}_i - \bar{v}_j|$, in order to identify clusters [100], [37]. A sample point $\bar{v}_i$ is attached to one or more of its neighbours $\bar{v}_j$ depending upon the distance $d(i, j)$. A group or cluster is thus formed of interrelated sample points, (see Figure 3.2).

A well known algorithm for data point linking is the minimum spanning tree [17], [100], [37]. In a tree, there is a unique path that connects one point to all others. A minimum spanning tree algorithm tries to join each point with its closest neighbours while preserving the tree structure.

The identification of clusters here depends upon point proximity which is a local information. Therefore, the results can be affected by local fluctuations. For example, the elimination of a junction can divide a cluster into two parts, while the addition of a new link can form a bridge joining two clusters.

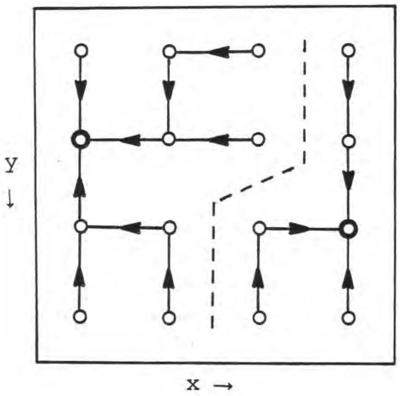


Figure 3.3: Two directed trees with pixel as nodes. Note that each tree defines an image region.

In image segmentation, the spatial distance between pixels, $|(x_i, y_i) - (x_j, y_j)|$, must be combined with their spectral distance, $|\bar{f}(x_i, y_i) - \bar{f}(x_j, y_j)|$. However, the spatial distance between pixels is fixed, and therefore, pixel linking algorithms generally consider only the 4 or 8 neighbours of a pixel as spatially adjacent.

A single linkage algorithm, presented by Narendra and Goldberg [58] is now described. A single linkage means that each pixel has a directed link to only one of its 8 neighbours. A directed tree with pixels as nodes, therefore, defines a region. Hence in Figure 3.3, there are two pixel trees corresponding to two regions.

A gradient image $g(x, y)$ is first calculated (see Figure 3.4). In the gradient image, the low value areas, where the pixel values remain constant (valleys), correspond to region interiors, whereas, high gradient values occur on region boundaries. Therefore, pixel trees are produced by linking pixels from highest values to lowest ones, the roots of the trees corresponding usually to the region centers. The pixel trees are constructed by connecting each pixel to the neighbour (8-neighbour) with the lowest gradient value. This process is represented by the block diagram of Figure 3.5. In the preprocessing step, the gradient image is calculated. Then, the decision process determines the pixel linkage.

Small perturbations in the gradient image resulting from noise effects can divide the image into many small pixel trees. Therefore, a smoothing process is introduced to merge connected regions without high gradient value boundaries.

A double-link approach called "relative similarity" is presented by Yokoya, Kitahashi and Tanaka, [99]. A pixel can have many directed links toward

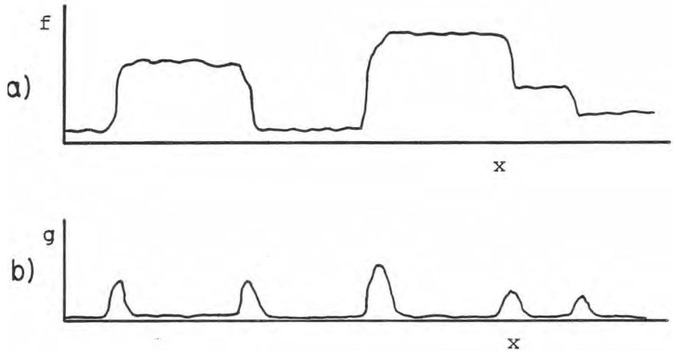


Figure 3.4: a) One-dimensional example of a image. b) The corresponding gradient image.

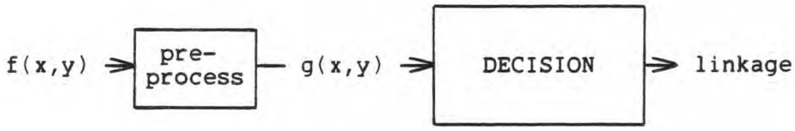


Figure 3.5: Block diagram of a pixel linking process. In the preprocessing step, the gradient image, $g(x,y)$, is first calculated. The decision process then determines the pixel linkage.

any of its 8 neighbours, if they are evaluated as similar. Similarity is a binary relation obtained by thresholding the gray level difference between adjacent pixels. The threshold varies from one pixel to the next so that similarity is not necessarily reflexive, that is pixel i can be similar to pixel j, while pixel j is not similar to i. The algorithm places two adjacent pixels i and j into the same region only if they are joined by two-way (double) links; i.e. i is similar to j and j is similar to i. Other modifications are also proposed and good results are reported [3], [7], [23].

3.2 Region growing

Algorithms based upon point-to-cluster or pixel-to-region distances are now examined. An important aspect to be considered is that region or cluster descriptive parameters (e.g. the mean value) can be sequentially learned from the data.

A well known and basic algorithm for spectral point clustering is the K-means algorithm [91]. It defines a cluster C_k by its center $\bar{u}_k$. The algorithm starts with some initial $\bar{u}_k$ values. These values are iteratively updated to augment their representativity of the data. The iterative process is composed of two phases: 1) each data point $\bar{v}_i$ is first joined to the closest cluster C_k, determined from the distance to the cluster mean $\bar{u}_k$, and 2) the data mean values for each cluster are then recalculated using the latest class assignment. This latter process is repeated until there are no more changes in the data partition and cluster mean values.

Region growing, a popular segmentation approach [30], [41], [45], is based upon a pixel-to-region distance measure, and also involves the sequential learning of the region descriptive parameters from image data. However, region growing does not iterate as does the K-means algorithm. The decision criteria used for pixel-region merging (phase 1) are now examined in more detail. The sequential updating of descriptive parameters (phase 2) is then studied.

Region growing algorithms involve a decision process which determines if a pixel or, more generally, a cell belongs to a region [41], [45], [30]. This decision can be based upon a cell-to-region distance measure where the cells and regions are assumed to be Gaussian statistical populations described by their means and variances. A likelihood ratio test can be used for deciding if a cell and a region belong to the same population. One proposed simplification consists in sequentially testing the similarity of variances and means for each feature dimension [41], [30]. Spatial information is included in the process by considering only adjacent cells as allowable candidates for fusion with a region.

It is now examined how the region description parameters (means and variançes) are calculated and updated. The algorithms begin by considering that all regions are initially empty. The first cell becomes the starting point of the first region. Then, the other cells are sequentially examined and assigned to regions. For each cell, the similarities between the cell and one of its adjacent regions is first tested. If the test succeeds, the cell is merged with the region; otherwise, the next adjacent region is considered. If the cell cannot be merged with any adjacent regions, then it becomes the starting point of a new region.

This implies that a region starts from a single cell and expands horizontally and vertically by absorbing more cells until it reaches its natural boundaries where the similarity test fails. Horeover, each time that a cell is added to a region, its descriptive parameters (means and variances) are updated. This implies that the next statistical test involving this region will take account of the new information, and hopefully, produce more reliable results [30], [41], [45). This process is represented by the block diagram of Figure 3.6, where a cell and a region are compared and merged if similar.

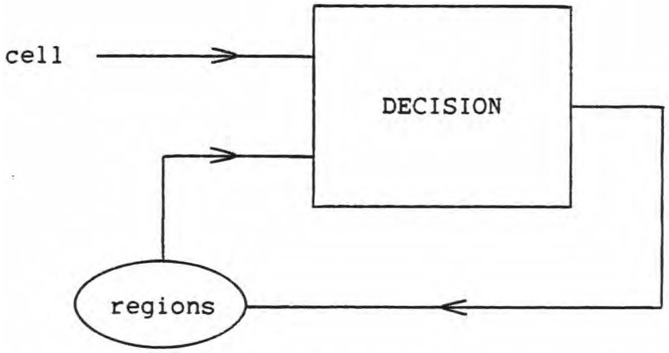


Figure 3.6: Block diagram of a region growing process. The decision process compares a cell and a region, and merges them if similar. The region descriptions are stored and used as an input to the decision. The descriptions are updated after each merge.

The region descriptions are stored and used as an input to the decision. The descriptions are updated after each merge.

Figure 3.7 illustrates the way that a region growing algorithm examines and groups cells. The six cells are examined in the following order; A, B, C, D, E, F, with cell A as the starting point of region 1. Cell B is tested for similarity with region 1. The test is successful, the region is then extended to include the ce1l B and the parameters of region 1 are updated. As cell C is found to be different from region 1, it becomes the starting point of a new region (#2). Ce1l D and E are subsequently added to region 1, the region parameters being recalculated at each step.

One of the principal limitations of these algorithms is the order dependancy [102]. The order in which the cells are processed affects the evolution of the region parameters, and, therefore, the outcome of the similarity test. Thus, different partitions can be obtained. For example, in Figure 3.7, the assignment of ce1l F can differ whether it is first compared with region 1 or region 2. Some modifications to the approach are suggested to reduce this prob1em [81], [72], [10], [40], [89]. For examp1e, region growing can start from ce1ls 1ocated in the most uniform areas.

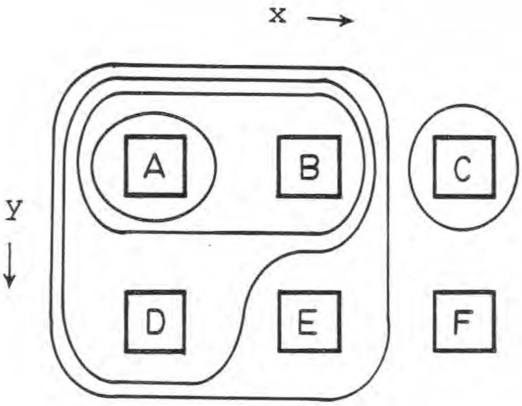


Figure 3.7: Sequential cell merging in a region growing algorithm.

Chapter 4

Hierarchical Segmentation

The preceding chapter has presented segmentation algorithms involving pixel-to-pixel or pixel-to-region distances. Hierarchical segmentation algorithms which involve region-to-region distances are now examined. Segment hierarchy and predicate equations are first defined. Then, a number of segmentation algorithms based upon predicate equations are studied, namely, region splitting by histogram analysis, region merging, and pyramid based segmentation. Two other algorithms producing a segment hierarchy without the utilization of predicate equations are also discussed, multi-thresholding and linked-pyramid algorithms.

4.1 Segment hierarchy and predicate equations:

A segment hierarchy can be represented by a tree [67], [102] (see Figure 4.1). In a tree, segments at lower levels are joined to form segments at higher levels. The nodes of the tree correspond to the segments S_i^ℓ, and the links between nodes indicate set inclusion. Thus, a link between a segment $S_k^{\ell+1}$ (ancestor or parent) and its disjoint sub-parts S_i^ℓ (descendents or sons) indicates that $S_i^\ell \in S_k^{\ell+1}$. The root of the tree corresponds to the whole image $\mathbb{I}$, and the leaves to pixels.

A image partition, $\mathbb{P}$, therefore corresponds to a node set $\{S_1, S_2 \dots S_n\}$, called a node cutset, which is the minimal set of nodes separating the root from all the leaves [34]. A node cutset divides the tree into many sub-trees: one with $\mathbb{I}$ as root and with S_1, S_2 S_n as leaves, and many other sub-trees (below $\mathbb{P}$) starting with the descendents of S_i ($\in \mathbb{P}$).

In hierarchical segmentation, the desired image partition $\mathbb{P} = \{S_1 \dots S_n\}$

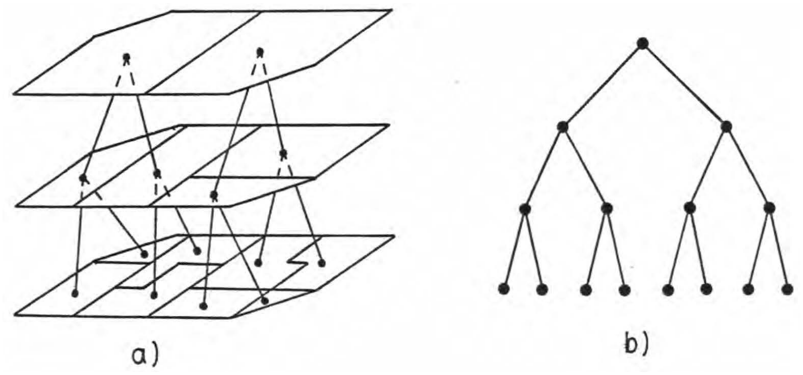


Figure 4.1: Segment hierarchy and segment tree.

is usually defined by predicate equations [102]:

$$\begin{array}{lrll} \mathrm{Prd}_1 : & Q(S_i) = true & \text{for all } i & \\ \mathrm{Prd}_2 : & Q(S_i \cup S_j) = false & \text{for all } i \neq j & (4.1) \\ & & \quad \text{and } S_i \text{ adjacent to } S_j & \end{array}$$

where S_i represents a segment or a region. The logical predicate $Q(.)$ is used to express the requirements that all segments S_i of a partition $\mathbb{P}$ must satisfy.

The predicate equations Prd_1 and Prd_2 can therefore be regarded as the definition of a node cutset. In practice, both merging and splitting schemes can be used. A merging scheme starts with small segments S_i (or pixels) which satisfy Prd_1, and proceeds to fulfill Prd_2 by region merging. It starts from the leaves of the tree, and climbs up the tree until it meets nodes $S_k^{\ell+l}$ $(=S_i^{\ell} \cup S_j^{\ell})$ for which the predicate values are false, $Q(S_i^{\ell} \cup S_j^{\ell})$ = false. Thus, S_i^{ℓ} and S_j^{ℓ} are in the node cutset, when Prd_2 is used as the stopping criterion. Here, $Q(S_i^{\ell} \cup S_j^{\ell})$ can be considered as an evaluation of the similarity of S_i^{ℓ} and S_j^{ℓ}, thus segment merging stops when there are no more similar segments. A splitting scheme starts with the root segment $\mathbb{I}$ which satisfies Prd_2, and proceeds to fulfill Prd_1. It descends the segment tree by dividing the segments $S_k^{\ell=1}$ into sub-parts S_i^{ℓ} and S_j^{ℓ}, unti1 Prd_1 is fulfilled, $Q(S_i^{\ell}) = true$ and $Q(S_j^{\ell}) = true$. The b1ock diagram of Figure 4.2 represents these processes. The region descriptions are first obtained from an initialization step. The regions are then split or merged by the decision process. The region descriptions are updated after each decision.

Another approach to hierarchical segmentation is based upon step-wise optimization. It is wide1y employed in data clustering where a stepwise optimization process selects the clusters that must be merged or split [16],

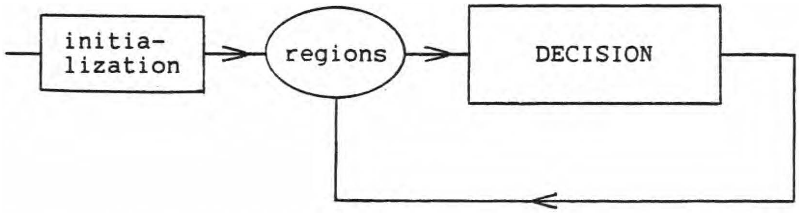


Figure 4.2: Block diagram of a predicate based hierarchical segmentation process. The region descriptions are first obtained from an initialization step. The regions are then split or merged by the decision process. The region descriptions are updated after each decision.

[17], [84], [85], [91]. For instance, in a merging scheme, the cluster pair, (C_i, C_j), that optimizes a similarity measure, $d(C_i, C_j)$ is found and the corresponding clusters merged. This sequential merging can continue until the required number of clusters is obtained. In Part II of this thesis, it will be shown that this approach can be advantageously used for image segmentation.

4.2 Region splitting by histogram analysis:

In Chapter 2, it was shown that the spectral histogram can be used for segmentation. For example, in simple images, the histogram can be analyzed to determine a threshold setting that separates an object from the background. Algorithms using the same idea, but applied recursively, have been proposed [52], [62], [83], [61], [92].

These algorithms typically start with one segment corresponding to the whole image plane $S^0 = \mathbb{I}$. The splitting of a segment S_i^ℓ is based upon its set of 1-D histograms, each one corresponding to a feature f^λ, $(\bar{f} = (f^1, f^2 \ldots f^\nu))$. A thresho1d is selected for one of the histograms, the thresholding operation then divides the segment into many sub-regions $S_k^{\ell+1}$. By repeating the same process, these sub-regions are then further segmented. This approach yields a segment tree, and the segment splitting stops when the segment histograms are unimodal, implying that the segments are homogeneous . This suggests that Prd_1 can be written as follows :

$$Q(S_i) = true \iff \text{histograms of } S_i \text{ are unimodal}$$

Using the local segment based histograms as opposed to image-wide histograms allows for a finer differentiation of image regions. However, many of the limitations of the histogram-based approaches remain; e.g. the feature selection problem, and the loss of spatial information. Horeover, the calculation of many histograms can involve a large processing time. A number of modifications have been proposed to improve the algorithm: better feature and threshold selection [62], [83], edge-contour correspondence [83], and planning from a lower resolution image [62].

4.3 Region merging:

A segment tree can also be produced by merging two or more segments, thus ascending the tree. In this approach, the algorithm employs a segment similarity measure to define the predicate equation Prd_2, which is the stopping criterion for a merging scheme.

$$Q(S_i \cup S_j) = false \iff S_i \text{ is not similar to } S_j$$

Moreover, an evident spatial constraint is that only adjacent segments can be joined.

Region merging can be regarded as an extension of region growing (see Chapter 3.2). In region growing, the cells and regions are distinguished. The image is first divided into cells, and then the regions are formed by merging of cells. A region begins with one cell and grows by annexing adjacent cells. A merger can only occur between a region and a cell. In region merging, there are only regions (or segments). The initial regions can be of any sizes and shapes, and merges can occur between any adjacent regions. Therefore, region growing and region merging possess sorne common characteristics: both involve sequential learning of segment description parameters, and the final result can be dependent upon the order of processing.

An example of a region merging algorithm is the one proposed by Brice and Fennema [6]. They use two heuristics, based upon information from the segment boundaries, to evaluate the similarity of two segments: the phagocyte and the weakness heuristics. Let L be the length of the common part of two region boundaries (see Figure 4.3). Let W be the length of the weak part of this common boundary, the weak part being those boundary points where the pixel difference between both sides is less than sorne threshold t_w. Let P_{min} be the length of the smaller perimeter, and let α and β be two parameters. Then, the phagocyte heuristic merges two regions if W exceeds a predefined portion, α, of P_{min}, i.e. if $W > \alpha\, P_{min}$. While, the weakness heuristic merges two regions if $W > \beta\, L$. The phagocyte heuristic guides the merging of regions in such a way as to smooth or shorten the resulting boundary. Two regions are merged if their common boundary is weak and

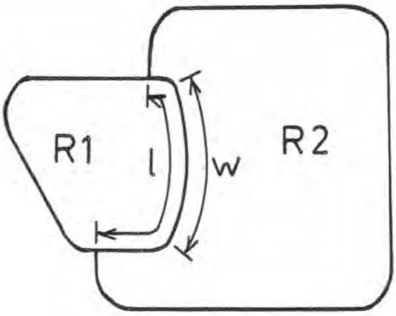


Figure 4.3: Two regions and their common boundary. L is the length of the common boundary and W is the length of the weak part of this boundary.

if the segment boundary length does not increase too quickly. The weakness heuristic merges two regions if a prescribed portion α of their common boundary is weak; i.e. if the weak part is at !east a certain percentage of L. The phagocyte heuristic is applied first, followed by the weakness one. The requirement of many threshold values is a limitation of the algorithm. Furthermore, the results can be affected by the order in which the regions are examined and merged.

Freuder [23] presents an algorithm where the similarity of two segments is a function of the surrounding segments. For each segment S_i, the adjacent segment, S_j, which is the most similar to S_i is selected. A directed link is drawn from S_i to S_j. The similarity is related to the difference between segment means values and the segment sizes:

$$|\,\mu_i - \mu_j\,| \times (\,A(S_i) + A(S_j)\,)$$

where μ_i is the mean value of segment S_i and $A(S_i)$ is the area or size of the segment. Thus, each segment points to one of its neighbours, the one with the closest mean value (weighted to take account of segment size). All segments related by a "double link" are now merged. A double link indicates a local minimum of the segment similarity measure, S_j being minimum among the neighbours of S_i, and S_i among the neighbours of S_j. Note that many segments can be merged at each iteration. An advantage of this algorithm is that no threshold is needed for the evaluation of segment similarity. Moreover, the utilization of local minimization avoids the problem of processing order dependency of the results.

4.4 Pyramid:

A pyramidal approach has been proposed for many image analysis tasks [78], [88], [98], [71], [8], [34]. A pyramid is a stack of successively lower resolution versions of the input image, with the original image $\bar{f}^0(x, y)$ $(=\bar{f}(x, y))$ at the bottom. Blocks of $m \times m$ points (image elements) at level ℓ are combined (e.g. averaged) to produce the single ancestor (parent) point at level $\ell + l$. In the following text, it is assumed that blocks are formed of 2×2 points. Thus, if the level zero $\bar{f}^0(x, y)$ contains $n \times n$ pixels, level l holds $n/2^\ell \times n/2^\ell$ points corresponding to blacks of $2^\ell \times 2^\ell$ pixels.

The main characteristic of a pyramid is its multi-resolution representation of an image. The higher levels of the pyramid correspond to lower resolution images. Pyramids have been found useful for many image analysis tasks because pattern element sizes can be very different and are not generally known a priori. Pyramids allow a process to operate at many different resolution levels. Moreover, these computations can be performed in parallel.

A pyramid can be regarded as a segment tree where each node corresponds to a block of $2^\ell \times 2^\ell$ pixels. However, it is a fixed tree, as a given node in the tree always corresponàs to the same block of pixels.

Horowitz and Pavlidis [34] propose a split-and-merge approach using the pyramidal data structure. The logical predicate $Q(S_i^\ell)$, which can be regarded as an evaluation of segment homogeneity, is defined from the segment approximation error, $Err(S_i^\ell)$:

$$Q(S_i^\ell) = true \iff Err(S_i^\ell) \leq thr$$

where *thr* is a threshold value. The process begins at an intermediate level ℓ of the tree, $S^\ell = \{S_i^\ell\}$. This level ℓ is regarded as the initial node cutset CS, $CS = S^{ell}$. This node cutset will be moved upward by segment merging or downward by splitting, until the segments of the node cutset satisfy Prd_1 (after splitting) and Prd_2 (after merging).

Block mergers are performed first. The four descendants of $S_i^{\ell+1}$, designated by, $S_{i,1}^\ell$, $S_{i,2}^\ell$, $S_{i,3}^\ell$, and $S_{i,4}^\ell$, are merged only if they are all in the current node cutset CS, they are all individually homogeneous, and finally, $S_i^{\ell+1}$ itself is homogeneous. This means that the node $S_i^{\ell+1}$ is added to the node cutset CS while $S_{i,1}^\ell$, $S_{i,2}^\ell$, $S_{i,3}^\ell$, and $S_{i,4}^\ell$ are removed. This process is repeated until no more mergers are possible (i.e. until Prd_2 is fulfilled).

Block splitting begins after all allowed mergers have been performed. Only blocks that have not been formed by merging are considered . If a block S_i^ℓ is not homogeneous ($Q(S_i^\ell) = false$) then it is divided into its four

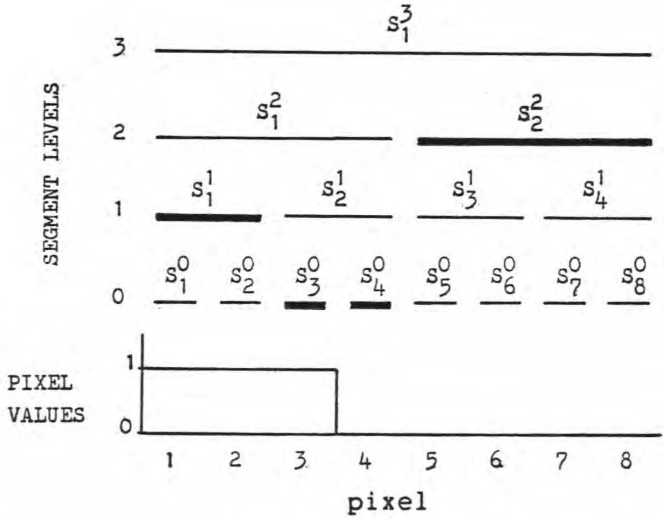


Figure 4.4: Image segmentation in a pyramid. The 8 pixels of one image line are grouped into segments of varying sizes to form a pyramid. The level 0 contains 8 segments, S_i^0 , of one pixel each. The level 1 contains 4 segments, S_i^1, of two pixels, etc.. The first three pixels have a value of one, and the five others, that of zero. The partition of this line is therefore composed of S_1^1, S_3^0, S_4^0 and S_2^2.

descendants. This means that $S_{i,1}^{\ell-1}$, $S_{i,2}^{\ell-1}$, $S_{i,3}^{\ell-1}$, and $S_{i,4}^{\ell-1}$ are added to the node cutset CS, while S_i^ℓ is removed. This is repeated until all blocks in the node cutset are homogeneous (i.e. until Prd_1 is fulfilled).

The block structure of the pyramid imposes arbitrary boundaries for the regions. This implies that often the final partition contains too many segments. For example, in Figure 4.4, the pyramid structure imposes an arbitrary boundary between pixel 4 and 5 up to level 2. They can only be joined if S_1^2 and S_2^2 are merged. The figure shows that the node cutset contains 4 segments, and S_1^1 , S_3^0 , S_4^0 and S_2^2 while only two distinct regions are present in the image.

To alleviate these anomalies, Horowitz and Pavlidis [34] propose a region merging procedure to merge similar adjacent blocks contained in the node cutset. In the example of Figure 4.4, it implies that S_1^1 and S_3^0 are merged as well as S_4^0 and S_2^2.

4.5 Multi-thresholding

Algorithms of the preceding chapters which involve a thresholding operation, can produce a segment hierarchy by varying the threshold values. However, no predicate equations required here since a unique threshold value is sufficient to define a partition or a node cutset. For example, a pixel linking algorithm can be employed to produced a segment hierarchy. Consider, for instance, that two pixels are linked only if their difference is lower than a threshold value. Thus, a modification of the threshold value can delete or add some links between pixels, resulting in the fusion or division of regions [37], [54]. Suppose that the use of a first threshold value t_1 yields segments $S_i^{t_1}$. Then, if a lower threshold t_2 is employed, some links are removed, which divide a region into two or more parts. The utilization of a sequence of threshold values $t_1 > t_2 > \ldots > t_n$ can, therefore, yield an hierarchy of segments or a segment tree. A second example is an edge detection algorithm that thresholds a gradient type value to identify potential edge points. Reducing the threshold value can produce new edge points which divide a region into two or more parts.

Multi-thresholding algorithms produce a distinct image partition for each threshold value t. This partition is independent of any previously calculated ones, and thus requires no sequential learning as in some segment merging or splitting algorithms. Furthermore, no segment measure are involved.

4.6 Linked-pyramid

A linked-pyramid scheme, where the junctions between nodes of different levels can be changed, has been presented to overcome the block structure imposed by pyramids [8], [78]. It involves an overlapped pyramid defined by 4x4 block where the blocks overlap by 50 percent in both the x and y directions. This implies that each block S_i^ℓ has four possible ancestors $S_a^{\ell+1}$ on the level above it, and has 16 possible descendants $S_d^{\ell-1}$ on the lower level.

In the linked-pyramid, the node cutset is predefined. Suppose that the level k of the pyramid which contains m nodes is selected as the node cutset, then the image can be divided into m regions corresponding to the descendants of these m nodes. Therefore, no predicate equations are required. For example, in Figure 4.5, there are two top nodes (the cutset) which involve a partition of the image into two regions. A linked-pyramid algorithm modifies the image partition by changing the links that relate the pixel nodes to the cutset nodes.

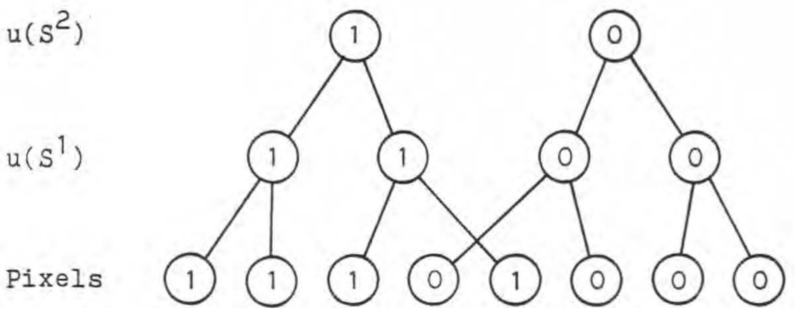


Figure 4.5: Pixel grouping in a linked-pyramid. Each of the two top nodes defines a image region composed of the pixels linked to this node. Note that the pixels of both regions are interleaved.

The segmentation begins by the initialization of node values. It is performed by a bottom-up process which calculates the value of a node $u(S_i^\ell)$ as the weighted average of its 4x4 possible descendants An iterative process composed of 2 phases then follows. In the first phase, each node is linked to the ancestor with the closest node value $u(S_a^{\ell+1})$, i.e. which minimizes the difference $\mid u(S_i^\ell) - u(S_a^{\ell+1}) \mid$. In the second phase, the node values are recalculated in a bottom-up fashion. The value of a node is found by averaging only over those descendants that are linked to it. These two phases are repeated until no further change occurs, typically about 10 iterations.

A linked-pyramid relaxes the spatial constraints. The iterative modifications of node junctions results in pixels subsets that may be of any arbitrary sizes and shapes. They can even possess pixels that are spatially unconnected [8]. This results from the possible interleaving of descendants. For example, in Figure 4.5, the two central pixels are surrounded by pixels of the other group.

By reducing the spatial constraints, the linked-pyramid becomes more similar to spectral clustering algorithms. The value of a node can be viewed as the center (mean value) of a cluster composed of the node descendants. Thus, the iterative recalculations of links and node values can be regarded as a particular implementation of the K-means clustering algorithm (or the ISODATA algorithm [39]).

However, the linked-pyramid approach is also affected by the same problems as the K-means algorithm: selection of the correct number of clusters, and loss of small regions (clusters) when surrounded by larger ones. Many modifications have been proposed for the initialization process, for the selection of the ancestor, and for the calculation of node values [8], [33], [2],

[71], [70]. The advantages of these different modifications seem to depend upon the particular application considered. Two other limitations of this approach are 1) the theoretical analysis of the algorithm results is difficult, and 2) the results are not always reliable. In particular, examples have been presented where the algorithm yields incorrect resu1ts [33], [2].

Chapter 5

Image Segmentation Optimization

The algorithms described up to now employ local or regional decision processes to segment a image. However, what may be really required is a satisfactory global result. Such a result is often defined as the optimum point of a cost function, $\mathbb{G}$. In image segmentation, the possible results are the set of all image partitions, $\mathbb{U} = \{\mathbb{P}\}$, where $\mathbb{P} = \{S_1, S_2 \ldots\}$, $\bigcup S_i = \mathbb{I}$ and $S_i \cap S_j = \varnothing$ for $i \neq j$, and the function (or the global criterion) $\mathbb{G}(\mathbb{P})$ is a measure of the cost or benefit of the partition $\mathbb{P}$. The optimal partition P_{opt} is, therefore, defined as:

$$\mathbb{G}(\,P_{opt}\,) = \underset{\mathbb{P}\,\in\,\mathbb{U}}{\text{Minimum}} \quad \mathbb{G}(\,\mathbb{P}\,) \tag{5.1}$$

Note that the size of $\mathbb{U}$ increases rapidly with the number of pixels.

An optimization process can, therefore, be used to segment an image. The function G is usually defined as follows:

$$\mathbb{G}(\,\mathbb{P}\,) = \sum_{S_i \in \mathbb{P}} \quad \mathbb{H}(\,S_i\,) \tag{5.2}$$

where $\mathbb{H}(S_i)$ is a segment characteristic measure. For example, the segment homogeneity can be evaluated by the segment variance.

Image segmentation can also be viewed as the piece-wise approximation of a two-dimensional function $f(x,y)$ by a set of polynomial functions, $r(x,y) = \sum a_{i,j}\, x^i y^j$ for $i,j = 0,1,2\ldots$, [34], [67]. Then, the segment cost $\mathbb{H}(S_i)$ is often defined as the sum of the squared approximation errors:

$$\mathbb{H}(S_k) = \sum_{(x.y)\in S_k} \quad (\,f(x,y) - r_k(x,y)\,)^2 \tag{5.3}$$

The spatial aspects of images, such as the roughness or smoothness, and the entropy, can also be considered [59], [69], [97].

Once a global criterion has been selected, segmentation becomes a well-defined problem in discrete optimization, and in theory, can always be solved by an exhaustive search. However, in practice, the large number of possible image partitions precludes an exhaustive search for all but the simplest problems. In its place, two alternative approaches are employed: the utilization of one-dimensional optimization techniques, and the finding of a local optimum from an initial partition by iterative processes.

5.1 Image segmentation by 1-D optimization

Two image segmentation algorithms based upon one-dimensional optimization techniques are now presented: line by line segmentation and contour detection. Pavlidis [67] presents an algorithm which slices the image into thin strips, each consisting of one image line. A one-dimensional optimization algorithm is then used to divide each strip into m segments. Each segment, S_i, is approximated by a linear function $r(x) = a_i + b_i\, x$ for $x \in S_i$. The strip partition is performed so as to minimize the overall approximation error.

These strip segments are then grouped by a segment merging algorithm to form image regions. The algorithm imposes a bound on the approximation error for each image region. Each segment is sequentially examined and merged with its most similar neighbour. The similarity between two segments, S_i and S_j, is measured by the difference between the slope coefficients, $|b_i - b_j|$, and the merger is performed only if the approximation error of the resulting segment is smaller than a prescribed threshold. The algorithm proceeds until no more mergers are possible.

Image segmentation can also be viewed as a problem of segment contour detection. The definition of a ***good*** contour usually involves local criteria. For example, the difference of gray levels between both sides of the contour must be large and the contour must be smooth. Using a contour tracking approach, contour detection can be presented as a sequential decision process. An edge element is defined as the common boundary between two adjacent pixels, and therefore, it can be followed by only three other edge elements. Thus, a contour tracking implies that a starting edge element must be given and the subsequent elements must be sequentially selected.

Martelli [49] proposes the utilization of a tree search algorithm, based upon the branch-and-bound algorithm, to find the best contour. Each branch leaving a tree node corresponds to the selection of one of the three possible following edge elements. Only the most promising branches are explored. Cooper [14] uses a likelihood measure to define the goodness of a contour and tries to find the optimum one.

Good results have been reported for problems involving a few contours, such as object-background differentiation problem [49], [20]. It seem particularly useful for very noisy image and where the general contour shape is known. Its extension to more complex image, with many contours intersecting one another, seems difficult [49].

5.2 2-D local optimization

An interesting alternative, when the global optimum cannot be found, is the location of a local optimum. Hence, an iterative process can be employed to improve an initial image segmentation by seeking a local optimum. However, as the cost function (or the global criterion) $\mathbb{G}$ can possess a multitude of local optimum points, the result depends upon the initial partition [59]. In this section, iterative algorithms based upon "steepest descent" and relaxation approaches are examined.

A "steepest descent" like algorithm starts with an initial image partition P^0 and generates a sequence of partitions $P^1 \ldots P^k$ such that $\mathbb{G}(P^0) > \mathbb{G}(P^1) > \ldots > \mathbb{G}(P^k)$. Each partition P^k is located in the neighbourhood of the preceding one, P^{k-l}. Peleg [69] presents an algorithm where each pixel receives a label indicating the region membership. The algorithm examines sequentially each pixel, and finds the best label for this pixel; i.e. the label that produces the lowest criterion $\mathbb{G}$. The process is repeated until no more changes occurs. Peleg [69] uses a local criterion to speed up the process, while Narayanan, O'Leary and Rosenfeld [59] employ the derivative of the global criterion to find the direction and the amplitude of the pixel value change ("steepest descent").

Good results are reported for cases where only a few segment classes are considered, for example, for object-background separation problems [69]. The criterion $\mathbb{G}$ is obtained by a combination of the segment approximation error with a roughness measure (i.e. measure of gray level change between adjacent pixels) and with a measure of the goodness of fit between image edge values and segment contours.

Relaxation and stochastic labeling algorithms can also be regarded as local optimization processes [21], [22], [36]. Stochastic labeling means that a probability vector $p_i(\lambda)$ is associated with each object i, corresponding to the probability that the object i receives the label λ. In image segmentation, each pixel can be regarded as an object and the labels can correspond to classes.

Such stochastic labeling can be characterized by its ambiguity and its consistency. An object labeling is non-ambiguous if $p_i(\lambda)$ equals one for one label and is zero for all others. An ambiguity measure indicates if the probability $p_i(\lambda)$ is concentrated in one label, or if it is distributed over the set of labels. An example of such a measure is

$$A_i = \sum_{\lambda} \quad p_i(\lambda)\,(\,1 - p_i(\lambda)\,) \tag{5.4}$$

The consistency is a measure of the compatibility of an object labeling p_i with those of its neighbours p_j. It involves the utilization of object interrelations. Let $q_i(\lambda)$ be a measure of the support from the neighbours of i to the label λ:

$$q_i(\lambda) = \sum_{j \in N(i)} \sum_{\lambda'} \quad r(i, \lambda, j, \lambda')\, p_j(\lambda') \tag{5.5}$$

where $r(i, \lambda, j, \lambda')$ expresses the support from object j with label λ' to the assignation of the label λ to object i, and where $N(i)$ designates the neighbours of i. q_i can be normalized such that its components sum to one:

$$q_i^*(\lambda) = q_i(\lambda) \Big/ \sum_{\lambda'} q_i(\lambda') \tag{5.6}$$

A consistency measure can then be defined as:

$$C_i = \|\, \bar{p}_i - \bar{q}_i \,\| \tag{5.7}$$

where $\| \cdot \|$ can be any norm (e.g. the Euclidean distance). These measures, A_i and C_i, can be combined to yield a local criterion, with a global criterion resulting from averaging over the set of objects.

An iterative process can then be used to find a local optimum labeling. However, there are many ways to define the global criterion and the iterative optimization process [76], [36], [22], and it can be difficult to select the appropriate definitions. Eklundh, Yamamoto and Rosenfeld [19] present an example of a relaxation algorithm used to improve pixel classification. At each iteration k, they calculate $q_i^k(\lambda)$, as defined before, from the current $p_i^{k-1}(\lambda)$, the indices i corresponding to pixel number. Then the new value for $p_i^k(\lambda)$ are calculated by

$$p_i^k(\lambda) = \frac{p_i^{k-1}(\lambda)\ (1 + q_i^k(\lambda))}{\sum_\lambda p_i^{k-1}(\lambda)\ (1 + q_i^k(\lambda))} \tag{5.8}$$

Here the probability $p_i(\lambda)$ is updated by multiplication with $(1+q_i(\lambda))$. The result, p_i, is then normalized so that its components sum to one.

Part II

Hierarchical Image Segmentation by Stepwise Optimization

Chapter 6

A Hierarchical Image Segmentation Algorithm

In the preceding chapters, image segmentation algorithms have been reviewed. It has been shown that:

1) Pixel classification techniques regard regions as composed of pixels belonging to the same statistical population or class. Good results are obtained if the pixel properties (e.g. probability distribution of pixel values) are uniform over a region, and are easily distinguishable from those of others regions. Pixel classification and edge detection are found useful for many applications, but they are limited by the way they treat the spatial information, i.e. as local spatial features.

2) Pixel linking and region growing algorithms consider regions as formed of similar pixels. Versatile algorithms are obtained by the utilization of spatial-spectral distances to evaluate the similarity of pixels. Hence, the distance between pixels inside a regio must be small. Distance measures can easily take account of the spatial aspect, particularly of the neighbour relation.

3) Hierarchical segmentation exploits the hierarchical structure of images. It is based upon segment measures which can evaluate complex properties of a image area such as the contour shape, the boundary weakness or the spectral histogram. Segment measures consider a larger image area than pixel distances, and therefore yield more reliable results.

4) A global criterion states the overall requirement that image regions must satisfy. A segmentation process can thus consist in finding the partition that optimizes the criterion. However, the appropriate definition of the criterion and the derivation of an algorithm that finds the optimum image

partition can be difficult tasks. The global criterion can also be useful to compare and evaluate image partitions produced by different algorithms.

5) An important aspect, found in many segmentation algorithms such as region growing and segment merging, is that segment descriptive parameters (e.g. the mean value) are sequentially learned from the data. Hence, the latter iterations of a segmentation process benefit from more accurate parameter values.

In this chapter, a new hierarchical segmentation algorithm based upon step-wise optimization is presented. The approach is inspired from hierarchical data clustering. For instance, in a merging scheme, a hierarchical clustering starts with N clusters corresponding to each of the N data points, and sequentially reduces the number of clusters by merging. At each iteration, the similarity measures $d(C_i, C_j)$, are calculated for all clusters pairs (C_i, C_j), and the clusters that minimize the measure are merged. This merging is repeated sequentially until the required number of clusters is obtained.

An important limitation of the hierarchical clustering approach is its excessive computing time for large data sets. If there are N clusters, then the similarity measure for $N \times (N-1)$ possible clusters pairs must be calculated. In image segmentation, however, only adjacent segments can be merged, reducing the number of considered segment pairs per iteration to $N \times M$, where N is the number of segments, and M the average number of neighbours per segment. M is usually small $(4 \leq M \leq 8)$ and is quite independent of N. Furthermore, a segment merge affects only the surrounding segments, and only the pairs involving those segments need to be modified or updated. Thus, only a limited number of new segment pairs must be considered at each iteration. Note that this gain of computing efficiency is only obtained for agglomerative and not divisive hierarchical segmentation.

The next section describes the proposed algorithm. It is then briefly compared with other hierarchical segmentation algorithms based upon a logical predicate.

6.1 The Hierarchical Stepwise Optimization (HSO) algorithm

An algorithm employing a sequence of optimization processes to produce a hierarchical segmentation is now presented, and called the Hierarchical Stepwise Optimization algorithm (HSO). It starts with an initial image partition, $P^0 = \{S_1, S_2, \ldots S_n\}$, and at each iteration, merges two segments to yield a segment hierarchy. An optimization process is used to select the segment pair that minimizes a "stepwise criterion" $C_{i,j}$.corresponding to the cost of merging S_i with S_j. The variables involved in the algorithm are:

1) B_i, the list of the segments S_j adjacent to S_i,
2) D_i, the parameters that describe the segment S_i,
e.g. the segment mean and size, and
3) $C_{i,j} = C(D_i, D_j)$, the cost of merging segment S_i with S_j,
wheres S_j is contained in B_i.

In the following presentation, the stepwise criterion and the stopping condition are not defined, and these must be specified according to the special characteristics of each application.

The algorithm consists of the following steps:

step 1 - Initialization step:
Define an initial image partition $P^0 = \{S_1, S_2, \ldots S_n\}$ and
for each segment S_i calculate

i) the segment descriptive parameters, Di

ii) the neighbour list $B_i = \{\, S_j \mid S_j$ is adjacent to $S_i \,\}$

iii) the stepwise criteria $\{C_{i,j} \mid S_j \in B_i\}$
$C_{i,j} =$ cost of merging S_i with S_j

step 2 - Find the criterion $C_{m,n}$ which has the lowest value.

$$C_{m,n} = \underset{i,j}{\text{Minimum}} \{\, C_{i,j} \,\}$$

Merge S_m and $S_n$into S_v and calculate

i) D_v from D_n and D_m

ii) $B_v = B_m \bigcup B_n \bigcap \overline{\{S_m, S_n\}}$

step 3 - Update the neighbour lists B_j and criteria $\{C_{i,j}\}$

$\forall\, S_j \in B_v$ (i.e. for all j such that S_j is a neighbour of S_v)

i) $B_j = B_j \bigcup \{S_v\} \bigcap \overline{\{S_m, S_n\}}$

ii) delete $C_{j,m}$, $C_{m,j}$, $C_{j,n}$ and $C_{n,j}$.

iii) calculate $C_{v,j}$ and $C_{j,v}$

step 4 - Stopping condition:
Stop if no more mergers are required.
Otherwise, go to step 2.

The algorithm is composed of an initialization part (step 1) and of an iterative part (steps 2 through 4). An initial image partition, $P^0 = \{S_1, S_2, \ldots S_n\}$, is first defined with strictly homogeneous segments (step 1). For example, each initial segment could contain only one pixel. Then, for each pair of adjacent segments, S_i and S_j, a stepwise criterion is calculated. The criterion corresponds to the cost of merging the two segments. For example, the increase of the sum of the squared errors around the segment means could be used.

At each iteration, the segment pair, S_m and S_n, which minimizes the step-wise criterion is first found and merged to produce S_v (step 2). The criterion values and neighbour lists are then updated in step 3. The step 4 terminates the algorithm if no more mergers are required. Otherwise, a new iteration is executed.

The algorithm is designed so as to reduce the computing time. Recalculations are avoided by 1) making explicit the information needed, and 2) updating the only values that are modified by a segment merger. The required information for the calculation of $C_{i,j}$ is contained in B_i and D_i. The neighbour lists B_i keep track of all segments adjacent to S_i. $C_{i,j}$ is calculated only for segment S_i and S_j such that $S_j \in B_i$. The evaluation of $C_{i,j}$ is then performed in terms of the segment descriptive parameters D_i and D_j, $C_{i,j} = C(D_i, D_j)$. Therefore, from B_i and D_i, the criterion values can rapidly be calculated. Moreover, the number of evaluated criterion values can be reduced by one half if the criterion is symmetric, $C_{i,j} = C_{j,i}$.

In the initialization step, B_i and D_i are first calculated directly from the image $\bar{f}(x, y)$ and the pixel membership function that indicate to which segment a pixel belongs. In the subsequent steps, the B_v and D_v values for a new segment S_v are obtained in a recursive manner as shown in step 2.

In a similar fashion, the merging of two segments affects only the neighbours. Hence, the step 3 needs to update only the neighbour lists B_j of segments adjacent to S_v and to evaluate only the related criteria $C_{v,j}$ and $C_{j,v}$. Thus, the number of operations is proportional to the number of neighbours of S_v.

The small number of modified criteria between two iterations can also be exploited to speed up the finding of the minimum criterion $C_{m,n}$ (step 2).

To sum up, the computing time of the initialization step is a function of the image size, the number of initial segments and the number of neighbours per segment. The computing time of the iterative steps, on the other hand, is mainly a function of the number of neighbours of S_v. The number of iterations depends upon the number of initial and final segments, each iteration reducing by one the number of segments. However, the algorithm requires substantial temporary memory space to store the current descriptive parameters, neighbour lists and criterion values.

6.2 Stepwise optimization vs logical predicates

This section compares the HSO algorithm with hierarchical segmentation algorithms based upon logical predicates. A typical predicate based hierarchical merging algorithm (PBHM) can be defined as follows (see Chapter 4):

PBHM algorithm:

i) Define an initial partition, $\{S_i\}$.

ii) Select randomly two adjacent segments and
merge them if $Q(S_i \bigcup S_j) =$ true.

iii) Stop if no more merges are possible,
i.e if $Q(S_i \bigcup S_j) =$ false for all segment pairs.
Otherwise, go to ii).

The hierarchical step-wise optimization algorithm (HSO) can be summarized as follows, in order to compare it with the PBHM algorithm:

HSO algorithm:

i) Define an initial partition, $\{S_i\}$.

ii) Select the two segments that minimize $C_{i,j}$ and merge them.
If there are many equal minimum $C_{i,j}$ values,
select one of them randomly.

iii) Stop if no more merges are required.
Note that a stopping condition must be defined.
Otherwise, go to ii).

The main difference between both algorithms is in the step ii) where the HSO algorithm examines all segment pairs, (S_i, S_j) in order to find the minimum $C_{i,j}$, while the PBHM algorithm considers only two segments at the same time. The stepwise optimization rule implies that the HSO algorithm considers the whole image context before merging two segments. This is indicated in the block diagram of Figure 6.1 by the multiple arrows entering the decision box. The region descriptions are first calculated by

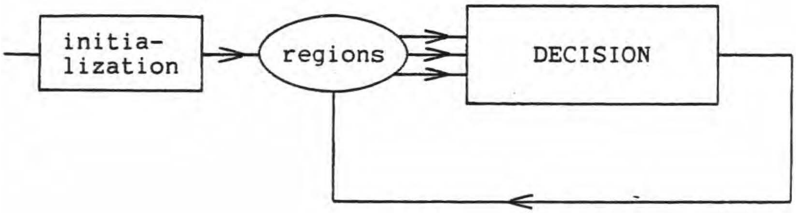


Figure 6.1: Block diagram of a hierarchical stepwise optimization process. The region descriptions are first calculated by an initialization step, then they are updated after each decision or segment merger. The decision examines all region pairs and selects one for the merger.

an initialization step, then they are updated after each decision or segment merger.

The stepwise optimization rule also implies that the most similar segments are merged first. The HSO algorithm gradually merges the segments, starting with the ones having the smallest $C_{i,j}$ values. This gradual aspect is not possible in the PBHM algorithm where only two states are considered: the true state for similar segments and the false state for segments not similar. Both the global and the gradual aspect confirm the advantage of the HSO algorithm over the PBHM algorith .

Moreover, a PBHM algorithm can be rewritten as a HSO algorithm, or in other words, the PBHM algorithms form a sub-set of HSO algorithms. The rewritting of a PBHM algorithm can be done in the following manner:

1) Define $C_{i,j}$ as:

$$C_{i,j} = \begin{cases} 0 & \text{if } Q(S_i \bigcup S_j) = true \\ 1 & \text{if } Q(S_i \bigcup S_j) = false \end{cases} \tag{6.1}$$

2) Define the step iii) of the HSO algorithm as:

iii) Stop if there are no more $C_{i,j} = 0$.
Otherwise, go to ii).

The resulting HSO algorithm is equivalent to the PBHM algorithm because the algorithm randomly merges segment pairs having $Q(S_i \bigcup S_j) =$ true, and stops when no such pairs remain.

6.3 Content of the following chapters

The following chapters stress the properties and advantages of the step-wise optimization algorithm. It is first shown that the algorithm combines the advantages of both the hierarchical and the optimization based segmentation. It is also demonstrated that, in hierarchical segmentation, the step-wise optimization rule reduces the probability of error. The probability of step-wise error is derived for a simple statistical image model. Finally, the selection of an appropriate step-wise criterion for a particular application is considered, and image segmentation examples are examined.

The contributions of each chapter are now described in more detail:

1) Optimization and segment hierarchy (Chap. 7)

Image segmentation can advantageously be stated as a global optimization problem. Hence, a piece-wise polynomial approximation is often used to represent a image, and the segmentation is then regarded as an optimization process which finds the partition having the minimum approximation error. The finding of the global optimum is generally unfeasible. The step-wise optimzation (HSO) algorithm is, therefore, presented as a sub-optimal alternative. The algorithm benefits from the utilization of a hierarchical structure and of segment based measures. Moreover, the global criterion is employed to derive the step-wise criterion, which corresponds to the increase of the global criterion produced by the merging of two segments. The operation of the algorithm is illustrated by an example.

2) Probability of error in hierarchical segmentation(Ch. 8)

Image segmentation can be regarded as an hypothesis testing process which merges two segments only if they belong to the same region. Two types of error can then occur: type I error when two similar segments are kept disjoint, and type II error when dissimilar segments are merged. The classical hypothesis testing approach can be employed and is reviewed. It is stressed that, at each step of a hierarchical segmentation process, the type II error is the most serious and, therefor it is advantageous to minimize its probability. This is acheived by the proposed stepwise optimization (HSO) algorithm which finds and merges the most similar segment pair. The probability of step-wise error (i. e. the probability of merging dissimilar segments) is then calculated, and the effects of segment sizes and criterion value are examined. Finally, the progression of the criterion minimum values from step to step is analysed and used to determine when the algorithm begins to merge very dissimilar segments.

3) Algorithm operation and criterion selection (Chap. 9)

The operation of the segmentation algorithm on real images is examined, and the problem of criterion selection is considered in more detail. The relation between the global optimization and the statistical testing approach is first outlined. A simple image segmentation example is then analysed. It is shown that the image possesses a hierarchical structure which allows many possible stopping points. User input is, therefore, needed to specify at which level of the segment hierarchy the algorithm must be stopped. The problem of selecting the appropriate segment models and the corresponding step-wise criteria are then examined. It is illustrated by experiments with a number of different criteria on a remote sensing image. The algorithm is shown to be capable of adaptation to different segmentation tasks.

Chapter 7

Optimization and Segment Hierarchy

A central problem in image analysis is that of segmentation; i.e. partitioning an image into disjoint regions that are homogeneous in some sense. In this chapter, image segmentation is presented as an optimization problem. A piece-wise polynomial approximation is often used to represent a image. The approximation error can then be employed as a global criterion $\mathbb{G}(\mathbb{P})$, and an optimization process can be used to find the partition that minimizes this criterion.

The piece-wise image approximation problem is presented and it is shown that the HSO algorithm constitues an interesting sub-optimal approach to the global optimization problem. This algorithm features both segment hierarchy and stepwise optimization. The segment hierarchy assumption reduces the search space, while the stepwise optimization assures that each iteration optimizes the global criterion . A detailed description of the algorithm for the constant piecewise approximation case is given, and its operation is illustrated by a simple example. The case of planar approximation is also examined.

7.1 Piecewise image approximation

An image can be regarded as a two dimensional function $f(x, y)$, where $(x, y) \in \mathbb{I}$, $\mathbb{I}$ being the image plane. An image partition $\mathbb{P}$ divides the image plane $\mathbb{I}$ into n regions, $S_1, S_2, \ldots S_n$. Let $f_i(x, y)$ designate the pixel values for the region S_i, $f_i(x, y) = f(x, y)$ for $(x, y) \in S_i$. Then, each region S_i can be approximated by a polynomial function, $r_i(x, y)$,

$$f_i(x, y) \simeq r_i(x, y) = \sum_{(\mathfrak{p},\mathfrak{q})\in\Gamma} a^i_{\mathfrak{p},\mathfrak{q}} \ (x)^{\mathfrak{p}} \ (y)^{\mathfrak{q}} \tag{7.1}$$

where Γ is the set of $(\mathfrak{p}, \mathfrak{q})$ pairs employed to define the terms of the polynomial function, $a_{\mathfrak{p},\mathfrak{q}}$. The approximation error for each segment can then be calculated by the sum of the squared deviations:

$$\mathbb{H}(S_i) = \sum_{(x.y)\in S_i} (\, f(x,y) - r_i(x,y)\,)^2. \tag{7.2}$$

Once the segment S_i is given, the coefficients $a^i_{\mathfrak{p},\mathfrak{q}}$ that minimize $\mathbb{H}(S_i)$ can be calculated. These must yield the best polynomial approximation for S_i. The minimization of $\mathbb{H}(S_i)$ implies:

$$\frac{\partial \mathbb{H}}{\partial a_{\mathfrak{p},\mathfrak{q}}} = 0 \quad | \quad \forall\ (\mathfrak{p}, \mathfrak{q}) \in \Gamma \tag{7.3}$$

This can be rewritten as follows:

$$\sum_{\mathfrak{p}',\mathfrak{q}'} a_{\mathfrak{p}',\mathfrak{q}'} \Big(\sum_{(x,y)} (x)^{\mathfrak{p}+\mathfrak{p}'} (y)^{\mathfrak{q}+\mathfrak{q}'} \Big) = \sum_{(x,y)} f(x,y)\, (x)^{\mathfrak{p}} (y)^{\mathfrak{q}} \quad | \ \forall\ (\mathfrak{p}, \mathfrak{q}) \in \Gamma \tag{7.4}$$

This is a linear system with m equations and m unknowns, m being the number of allowed pairs $(\mathfrak{p}, \mathfrak{q})$, (e.g. if $\Gamma = (0,0)$, $(0,1)$ and $(1,0)$ then m=3). The polynomial coefficients $a^i_{\mathfrak{p},\mathfrak{q}}$ that minimize $\mathbb{H}(S_i)$ can be obtained by solving this linear system. A unique solution may not result. This is particularly the case when the number of pixels in S_i is smaller than m, the number of coefficients.

Having defined the segment approximation problem, the problem of image approximation is now considered. Piecewise image approximation will then be a tie-in with image segmentation. Once a image is divided into segments $S_1, S_2, \ldots S_n$, each of them can be approximated, and a image approximation, $r(x,y)$, results from the concatenation of each piecewise approximation $r_i(x,y)$:

$$r(x,y) = \begin{cases} r_1(x,y), & \text{if } (x,y) \in S_1 \\ \vdots & \vdots \\ r_n(x,y), & \text{if } (x,y) \in S_n \end{cases} \tag{7.5}$$

The approximation error for the whole image is, consequently:

$$\mathbb{G}(\mathbb{P}) = \sum_{S_i \in \mathbb{P}} \mathbb{H}(S_i) \tag{7.6}$$

where $\mathbb{P} = \{S_1, S_2, \ldots\}$, $\bigcup S_i = \mathbb{I}$ and $S_i \cap S_j = \varnothing$ for $i \neq j$.The minimum value for $\mathbb{G}$ results necessarily from the sum of the minimum values for $\mathbb{H}(S_i)$, $\mathbb{H}(S_i) = \mathbb{H}_{min}(S_i)$. The image approximation consists then in finding the partition $\mathbb{P}_{min}$ that minimizes the global criterion $\mathbb{G}$.

The importance of the number of segments n in the minimization of $\mathbb{G}(\mathbb{P})$ must be stressed. The minimum value $\mathbb{G}_{min}(\mathbb{P})$ will monotonically decrease with the increase of the allowed number of segments for $\mathbb{P}$. For example, splitting a segment into two sub-parts can only reduce the approximation error, no increase is possible. Therefore, a image approximation problem consists in finding the partition $\mathbb{P}_n^*$ such that

$$\mathbb{G}(\mathbb{P}_n^*) \;=\; \underset{\{P_n'\}}{\text{Minimum}} \;\{\; \mathbb{G}(P_n') \;\} \tag{7.7}$$

where $\mathbb{P}_n^*$ and P_n' are image partitions with n segments.

7.2 Stepwise optimization for image segmentation

The identification of the partition $\mathbb{P}_{min}$ that minimizes a global criterion or cost function $\mathbb{G}$ is now discussed. It requires a search over the entire space of all possible image partitions, $\{\mathbb{P}\}$. However, the implementation of this search is prohibitive because of the large size of the $\{\mathbb{P}\}$ space. Therefore, the search space $\mathbb{U}$ must be constrained to a subset of $\{\mathbb{P}\}$, $\mathbb{U} \subset \{\mathbb{P}\}$. Hence, only a sub-optimum is obtained, which can be very close to the global optimum if the subset $\mathbb{U}$ is properly selected.

Two kinds of subsets $\mathbb{U}$ often used are 1) the neighbourhood of an initial image partition, and 2) the subset yielded by a hierarchical data structure. Thus, if it is known that the optimum is close to an initial image partition $\mathcal{P}^0$, the search can surely be constrained to the neighbours of $\mathcal{P}^0$. Furthermore, a ***gradient descent*** like procedure can be used (see section 5.2). This consists in moving a pixel from one segment to another if such a move improves the global criterion or cost function $\mathbb{G}$. This iterative process is terminated when a local optimum is found. However, in general, it is difficult to specify a sufficiently good initial partition $\mathcal{P}^0$.

A hierarchical data structure can also be employed to define a useful subset of image partitions (see Chapter 4). A hierarchy of segments can be represented by a segment tree in which nodes correspond to segments. Each segment S_i^ℓ is linked to the segments of a lower level $S_{i,1}^{\ell-1}, S_{i,2}^{\ell-1}, \ldots$ which are disjoint subsets of S_i^ℓ, and which are called ***sons*** of S_i^ℓ. Therefore, a image partition corresponds to a subset of these tree nodes.

A image segmentation algorithm which involves the construction of a segment tree as the result of a sequence of stepwise optimizations is now introduced. The presentation is similar to the one proposed by Ward [93] for hierarchical clustering. It requires a global criterion or cost function

$\mathbb{G}(\mathbb{P})$ which reflects the cost or loss of information resulting by representing the image with the partition $\mathbb{P}$.

An initial image partition $\mathcal{P}^0 = \{S_1^0, S_2^0, \ldots S_n^0\}$ with n segments is first defined. At the k^{th} iteration, the algorithm merges two segments from the $\mathcal{P}^{k-l}$partition to produce a new partition $\mathcal{P}^k = \{S_1^k, S_2^k \ldots S_{n-k}^k\}$. As the number of segments is decreased by one at each iteration, $\mathcal{P}^k$ must contain $n-k$ segments. $\mathbb{G}(\mathcal{P}^k)$ tends generally to increase from step to step and can be written as:

$$\mathbb{G}(\mathcal{P}^k) = \mathbb{G}(\mathcal{P}^0) + \sum_{\ell=1}^{k} \left(\mathbb{G}(\mathcal{P}^\ell) - \mathbb{G}(\mathcal{P}^{\ell-1})\right) \tag{7.8}$$

The minimization of $\mathbb{G}(\mathcal{P}^k)$ is, therefore, associated with the minimization of each term of the summation which corresponds to the increase of $\mathbb{G}$ at each iteration. Thus, the global optimization problem is reduced to a sequence of stepwise optimizations. However, the minimization of each term, $\mathbb{G}(\mathcal{P}^\ell) - \mathbb{G}(\mathcal{P}^{\ell-l})$, yields the global optimum for $\mathbb{G}(\mathcal{P}^k)$ only if the terms are independant, which is not necessarily the case. Nevertheless, it can constitute an interesting sub-optimal approach.

The goal of the stepwise optimization is, therefore, to find the two segments whose merger produces the smallest increase of $\mathbb{G}$. For the image approximation problem, $\mathbb{G}$ increases monotonically with the number of iterations, k.

$$\mathbb{G}(\mathcal{P}^0) \leq \mathbb{G}(\mathcal{P}^1) \ldots \leq \mathbb{G}(\mathcal{P}^\ell) \ldots \leq \mathbb{G}(\mathcal{P}^k) \tag{7.9}$$

such that the increase $\mathbb{G}(\mathcal{P}^k) - \mathbb{G}(\mathcal{P}^{k-1})$ is always nul or positive. This increase results from the merging of two segments S_i and S_j, and can easily be calculated from equation 7.6:

$$C_{i,j} = \mathbb{H}(S_i \cup S_j) - \mathbb{H}(S_i) - \mathbb{H}(S_j) \tag{7.10}$$

The only terms of $\mathbb{G}(\mathbb{P})$ that are affected by the merging are $\mathbb{H}(S_i)$ and $\mathbb{H}(S_j)$ which are replaced by $\mathbb{H}(S_i \cup S_j)$. Thus, $C_{i,j}$ is the stepwise criterion to be optimized. So, each iteration k involves:

1) the identification of all pairs of connected segments (S_i, S_j),

2) the calculation of $C_{i,j}$,

3) the selection of the lowest $C_{i,j}$, and

4) the merging of the two corresponding segments.

It must be noted that the algorithm does not guarantee that $\mathcal{P}^k$ will optimize $\mathbb{G}(\mathbb{P})$ amongst all the partitions with $n-k$ segments. Nevertheless, it yields good results as will be seen, and the implied hierarchical data structure can constitute an advantage for many applications.

The main characteristic of the algorithm is the stepwise optimization process for the selection of the two segments to merge. This is different from the predicate based segmentation (see Chapter 4) where the decision to merge or split segments is local (based upon S_i or $S_i \cup S_j$) and involves a threshold-like decision process. The utilization of a predicate only guarantees that the segment S_i will satisfy the predicate, but not that the partition will satisfy some overall criterion. Another important point is that the proposed algorithm only merges two segments at each iteration.

7.3 Image approximation by constant value regions

The previous section has presented a segmentation algorithm using a global criterion, with stepwise optimization. The stepwise optimization algorithm (HSO) described in section 6.1, can be adapted to the global optimization case by an appropriate definition of the stepwise criterion. As a particular example, the concepts are illustrated by the piecewise approximation of a multi-channel image $\bar{f}(x,y) = (\ f_1(x,y),\ f_2(x,y),\ \ldots f_\kappa(x,y)\ )$ by constant value regions.

In constant piecewise approximation (or zero order approximation), an image is divided into segments S_i which are approximated by their mean values, $\bar{\mu}_i = (\ \mu_{1,i},\ \mu_{2,i}\ \ldots\ \mu_{\kappa,i}\ )$:

$$r_i(x,y) = \bar{\mu}_i \tag{7.11}$$

This corresponds to using only one term of the polynomial function, the $a^i_{0,0}$ term, the optimal value being the segment mean, $a^i_{0,0} = \bar{\mu}_i$ (see section 7.1). The segment approximation error or the segment cost is then the weighted sum of the squared differences between the pixel values and the segment mean $\bar{\mu}_i$:

$$\mathbb{H}(S_i) = \sum_{\lambda=1}^{\kappa} w_\lambda \Big(\sum_{(x,y)\in S_i} (\ f_\lambda(x,y) - \mu_{\lambda,i}\ )^2 \Big) \tag{7.12}$$

where κ is the number of channels and w_λ is a weighting factor which takes into account the different dynamic ranges. Simultaneously, the minimization of $\mathbb{G}(\mathbb{P})$ $(= \sum \mathbb{H}(S_i)\ )$ can then be regarded as the minimization of the intra-cluster (segment) variance or as the maximization of the between cluster (segment) variance ([17] p. 222).

In order to use the HSO algorithm (see section 6.1), the stepwise criterion and the segment descriptive parameters D_i must be defined. The step-wise criterion is as given before:

$$C_{i,j} = \mathbb{H}(S_i \cup S_j) - \mathbb{H}(S_i) - \mathbb{H}(S_j) \tag{7.13}$$

which can now be rewritten as:

$$C_{i,j} \;=\; \frac{N_i \cdot N_j}{N_i + N_j} \; \sum_{\lambda=1}^{\kappa} \; w_\lambda \; (\, \mu_{\lambda,i} - \mu_{\lambda,j} \,)^2 \tag{7.14}$$

where N_i and N_j are the number of pixels in S_i and S_j. The stepwise minimization of $C_{i,j}$, therefore, results in the merger that minimizes the increase in the overall pixel variance around the segment means. The segment descriptive parameters, D_i, needed to calculate the criterion are the segment size, N_i, and the mean, $\bar{\mu}_i$.

In order to complete the definition of the HSO algorithm, the stopping condition (step 4) must also be specified. It is assumed that the algorithm stops when the number of segments reaches a predefined value. The stopping condition is discussed in more detail in Chapter 9.

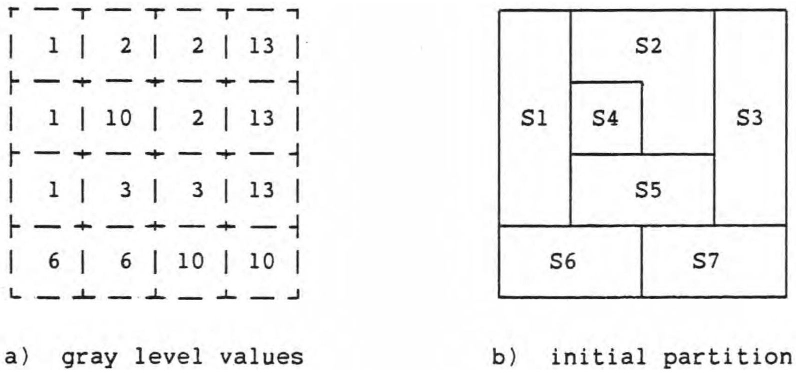
7.4 An illustrative example

The operation of the algorithm is now illustated by means of a simple example. Figure 7.1 shows a small image (4×4 pixels) with 7 initial constant level segments. This is a one channel image ($\kappa = 1$). Therefore the indices, λ, are omitted for simplicity, and the channel weighting factor is set to one ($w = 1$).

The algorithm starts with an initial partition $\mathcal{P}^0$ of 7 segments, $S_1, S_2 \ldots S_7$. The first iteration merges two segments and yields a new one labeled S_8. At the following iterations, segments S_9, S_{10}, $\ldots$ are sequentially created. Figure 7.2 shows a segment tree which represents the sequence of segment mergings.

Table 7.1 contains the corresponding segment description parameters, i.e. the size N_i, the mean μ_i and the neighbour set B_i of each segment S_i. There is one line per segment. These values are computed and stored by the algorithm. The upper part of Table 7.1 is calculated at the initialization step (step 1), while the last lines are calculated one by one at each iteration (steps 2 to 4).

Table 7.2 shows the lists of criterion values $C_{i,j}$ which are used to find which segments to merge. These lists must be updated after each merge.



1	2	2	13
1	10	2	13
1	3	3	13
6	6	10	10

a) gray level values

b) initial partition

Figure 7.1: A small image with its initial partition.

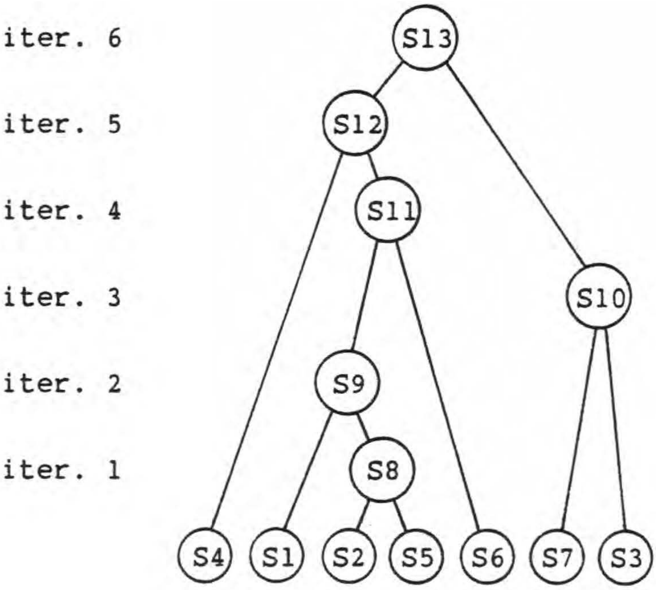
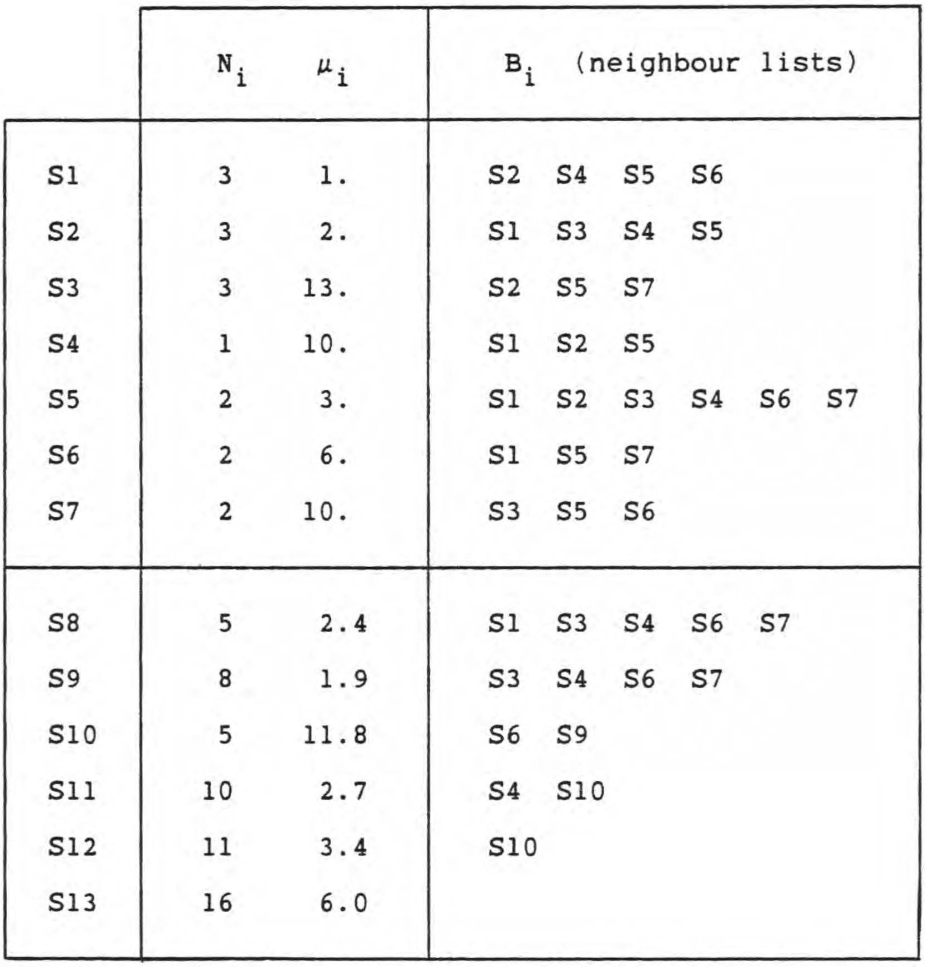


Figure 7.2: Sequence of segment merges.

Table 7.1: Segment description parameters and neighbour lists.



	N_i	μ_i	B_i (neighbour lists)
S1	3	1.	S2 S4 S5 S6
S2	3	2.	S1 S3 S4 S5
S3	3	13.	S2 S5 S7
S4	1	10.	S1 S2 S5
S5	2	3.	S1 S2 S3 S4 S6 S7
S6	2	6.	S1 S5 S7
S7	2	10.	S3 S5 S6
S8	5	2.4	S1 S3 S4 S6 S7
S9	8	1.9	S3 S4 S6 S7
S10	5	11.8	S6 S9
S11	10	2.7	S4 S10
S12	11	3.4	S10
S13	16	6.0	

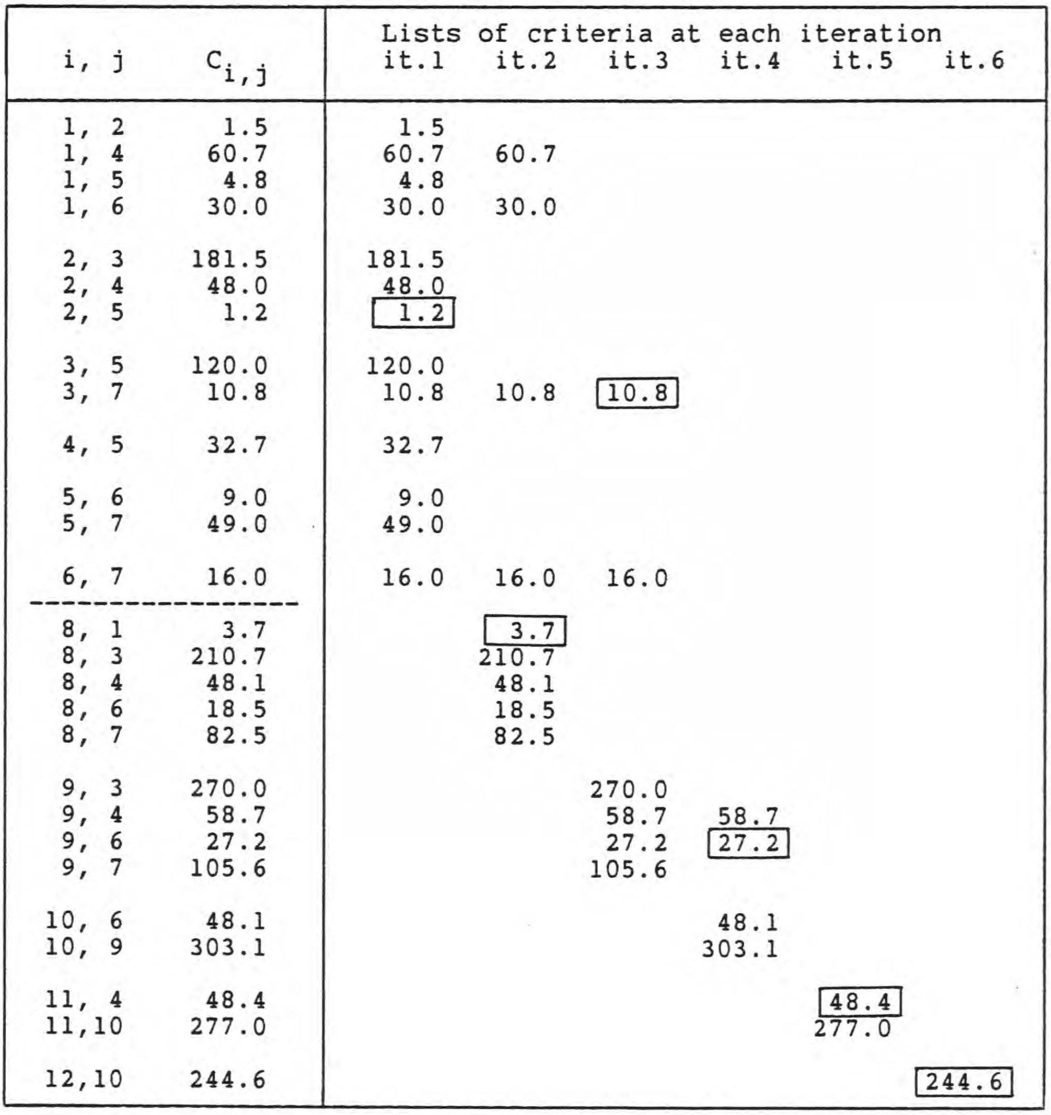
The different columns of Table 7.2 show the content of these criterion lists, with the minimum enclosed by a rectangle.

The operation of the algorithm is now detailed step-by-step. The initialization step is the only one which requires the image gray level matrix (Figure 7.1-a) and the initial image partition (Figure 7.1-b). For each initial segment S_i, the number of pixels N_i, the mean μ_i and the segment neighbours B_i are calculated. These values are shown in the upper lines of Table 7.1. In addition, the criterion values $C_{i,j}$ for all pairs of initial segments S_i are evaluated and shown in the upper part of Table 7.2.

The iterative part of the algorithm requires the following operations:

step 2) Select the minimum value in the criterion list (the minimum is enclosed by a rectangle), and create a new segment which is added

Table 7.2: Lists of criterion values $C_{i,j}$.



i, j	$C_{i,j}$	Lists of criteria at each iteration					
		it.1	it.2	it.3	it.4	it.5	it.6
1, 2	1.5	1.5					
1, 4	60.7	60.7	60.7				
1, 5	4.8	4.8					
1, 6	30.0	30.0	30.0				
2, 3	181.5	181.5					
2, 4	48.0	48.0					
2, 5	1.2	$\boxed{1.2}$					
3, 5	120.0	120.0					
3, 7	10.8	10.8	10.8	$\boxed{10.8}$			
4, 5	32.7	32.7					
5, 6	9.0	9.0					
5, 7	49.0	49.0					
6, 7	16.0	16.0	16.0	16.0			
8, 1	3.7		$\boxed{3.7}$				
8, 3	210.7		210.7				
8, 4	48.1		48.1				
8, 6	18.5		18.5				
8, 7	82.5		82.5				
9, 3	270.0			270.0			
9, 4	58.7			58.7	58.7		
9, 6	27.2			27.2	$\boxed{27.2}$		
9, 7	105.6			105.6			
10, 6	48.1				48.1		
10, 9	303.1				303.1		
11, 4	48.4					$\boxed{48.4}$	
11,10	277.0					277.0	
12,10	244.6						$\boxed{244.6}$

to Table 7.1.

step 3) Update the criterion values (this is shown by the addition of a new column in Table 7.2), and update the neighbour lists.

step 4) Decide to continue or to stop.

As an example, these operations are described in more detail for the first iteration:

step 2) Find the minimum value in column "iteration l" of Table 7.2, which is 1.2 for the criterion $C_{2,5}$.
Produce a new segment labeled S_8 by merging S_2 and S_5. The descriptive parameters and neighbour list of S_8 are calculated from those of S_2 and S_5 and are noted in a new line of Table 7.1.

step 3) Update the criterion values by removing any $C_{i,j}$ involving S_2 or S_5, and by calculating the new criteria involving S_8 and its neighbours. The updated criterion values are shown in the column "iteration 2" of Table 7.2.
Update the neighbour lists by replacing any appearances of S_2 or S_5 by S_8 in the B_i lists.

step 4) Decide to continue.

This completes the first iteration.

This process is repeated until the number of segments has been reduced to the preset value. In the present example, the image can be considered as composed of two regions, and,therefor the algorithm is stopped at this time. The final partition is thus composed of S_{10} and S_{12}.

7.5 Planar approximation

In many cases, constant value approximations are inappropriate to represent the image regions, and more complex approximation functions must be employed. For instance, the approximation of the one-dimensional example of Figure 7.3 requires first order polynomial functions. Higher order polynomials need more calculations but can be applied to a larger class of image regions. For demonstration purposes, the planar approximation case is now examined. Approximation with higher order polynomials can be developed in a similar manner.

A segment S_i is approximated by a plane:

$$r_i(x,y) = a^i_{0,0} + a^i_{1,0}(x) + a^i_{0,1}(y) \tag{7.15}$$

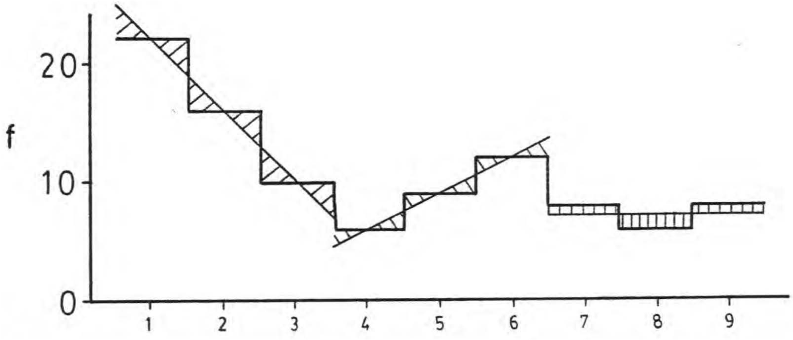


Figure 7.3: A one-dimensional example of planar approximation.

The sum of the squared errors is used as the segment cost $\mathbb{H}(S_i)$. The calculation of the coefficients $a^i_{0,0}$, $a^i_{1,0}$ and $a^i_{0,1}$ that minimize $\mathbb{H}(S_i)$ is now developed. All the needed information about the segment values $f(x,y)$, for $(x,y) \in S_i$, is contained in the following moments:

$$\begin{aligned}
N &= \text{number of pixels in } S_i \\
M_z &= \textstyle\sum_{S_i} \ f(x,y) \\
M_{zz} &= \textstyle\sum_{S_i} \ f(x,y)^2 \\
M_{zx} &= \textstyle\sum_{S_i} \ x \cdot f(x,y) \\
M_{zy} &= \textstyle\sum_{S_i} \ y \cdot f(x,y) \\
M_x &= \textstyle\sum_{S_i} \ x \\
M_y &= \textstyle\sum_{S_i} \ y \\
M_{xx} &= \textstyle\sum_{S_i} \ x^2 \\
M_{xy} &= \textstyle\sum_{S_i} \ x \cdot y \\
M_{yy} &= \textstyle\sum_{S_i} \ y^2
\end{aligned} \tag{7.16}$$

where all the summations are over $(x,y) \in S_i$. From these moments, the following covariance values are calculated:

$$\begin{aligned}
V_{zz} &= M_{zz} - (M_z)^2 \Big/ N \\
V_{zx} &= M_{zx} - M_z\, M_x \Big/ N \\
V_{zy} &= M_{zy} - M_z\, M_y \Big/ N \\
V_{xx} &= M_{xx} - (M_x)^2 \Big/ N \;+\; N \Big/ 12 \\
V_{xy} &= M_{xy} - M_x\, M_y \Big/ N \\
V_{yy} &= M_{yy} - (M_y)^2 \Big/ N \;+\; N \Big/ 12
\end{aligned} \tag{7.17}$$

The term $N/12$ in V_{xx} and V_{yy} results from the consideration of a pixel as an elementary area of constant value (see Figure 7.3). Regarding a pixel as an elementary area instead of a dimensionless point resolves the problems of small region approximations.

The coefficients $a^i_{0,0}$, $a^i_{1,0}$ and $a^i_{0,1}$ that minimize $\mathbb{H}(S_i)$ can now be calculated:

$$\begin{aligned}
&\text{If } V_{xy} = 0 \text{ then } \begin{cases} a_{1,0} = V_{zx} \Big/ V_{xx} \\ a_{0,1} = V_{zy} \Big/ V_{yy} \end{cases} \\
&\text{If } V_{xy} \neq 0 \text{ then } \begin{cases} a_{1,0} = \dfrac{V_{zx}\, V_{yy} - V_{zy}\, V_{xy}}{V_{xx}\, V_{yy} - (V_{xy})^2} \\[2ex] a_{0,1} = \dfrac{V_{zy}\, V_{xx} - V_{zx}\, V_{xy}}{V_{xx}\, V_{yy} - (V_{xy})^2} \end{cases} \\
&\text{and in both cases} \quad a_{0,0} = (\, M_z - a_{1,0}\, M_x - a_{0,1}\, M_y \,) \Big/ N
\end{aligned} \tag{7.18}$$

The sum of the squared errors $\mathbb{H}(S_i)$ of this optimal approximation is:

$$\mathbb{H}(S_i) = V^i_{zz} - a^i_{1,0}\, V^i_{zx} - a^i_{0,1}\, V^i_{zy} \tag{7.19}$$

As for the constant approximation case, the stepwise optimization algorithm (HSO) can easily be adapted to the planar approximation case. The stepwise criterion is as defined in equation (7.10)

$$C_{i,j} = \mathbb{H}(S_i \cup S_j) - \mathbb{H}(S_i) - \mathbb{H}(S_j) \tag{7.20}$$

where the segment approximation error $\mathbb{H}(S_i)$ is calculated from the preceding equations. The segment descriptive parameters D_i now include all the predefined moments; i.e. N, M_z, M_{zz}, M_{zx}, M_{zy}, M_x, M_y, M_{xx}, M_{xy} and M_{yy}. Note that the moments of new segments can be obtained in a recursive manner.

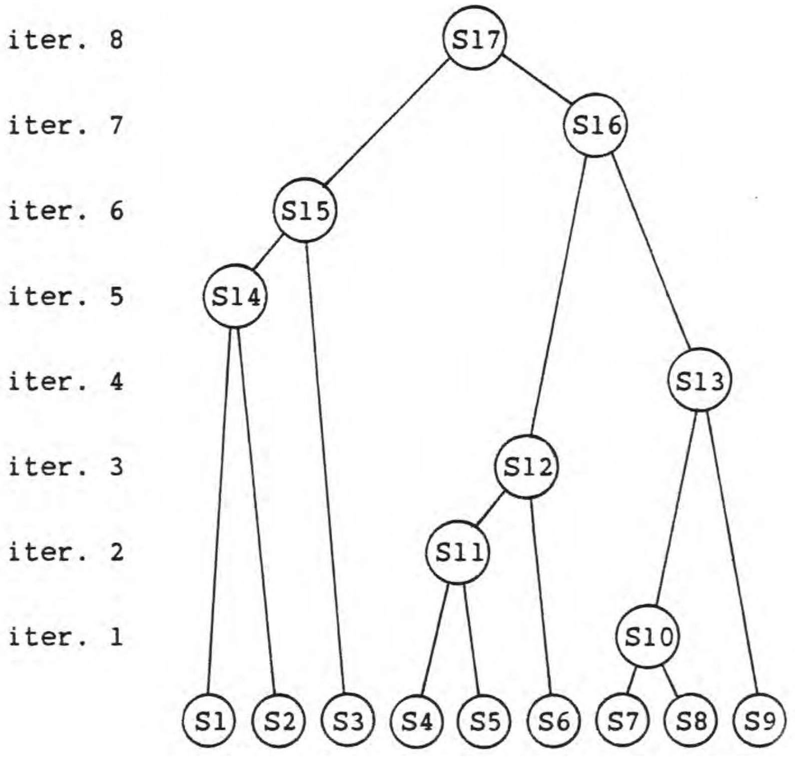


Figure 7.4: Sequence of segment merges for planar approximation.

As an illustration, the planar approximation algorithm is applied to the one-dimensional example of Figure 7.3, where the pixel values are 22, 16, 10, 6, 9, 12, 8, 6 and 8. The algorithm begins with a partition of 9 segments containing one pixel each, $\{S_1, S_2 \ldots S_9\}$. Figure 7.4 shows the sequence of segment mergings. A three segment partition, $\{S_{15}, S_{12}, S_{13}\}$, is produced by stopping the algorithm after 6 iterations. In Figure 7.3, these three segments are approximated by straight lines.

In Chapter 9, both constant value and planar approximation will be employed to segment remote sensing images. The advantages and limitations of both will also be discussed in the more general context of criterion selection.

Chapter 8

Probability of Error in Hierarchical Segmentation

Pattern recognition and image analysis are often regarded as statistical decision processes. In this chapter, it is first shown that statistical testing can be employed for image segmentation. An image is regarded as composed of constant value regions corrupted by Gaussian white noise. An image segmentation can then be produced by testing and merging two segments if they belong to the same region. Two types of error can then occur: type I error when two similar segments are kept disjoint, and type II error when dissimilar segments are merged. The classical hypothesis testing approach can be employed and is reviewed. It is stressed that, in hierarchical segmentation, the type II errors are the most important and it is, therefore, advantageous to minimize this probability of error. This is achieved by a stepwise optimization process which finds and merges the most similar segment pair.

The probability of error in the stepwise optimization approach is then examined. A normalized statistic is used as a stepwise criterion and its probability functions are derived. The probability of stepwise error (i.e. the probability of merging dissimilar segments) is then calculated, and the effects of segment sizes and criterion value are examined. Finally, the progression of the criterion minimum values from step to step is analysed, and is used to determine when the algorithm begins to merge really dissimilar segments and, therefor must be stopped.

8.1 A statistical model for image segmentation

A simple image model that can be employed for image segmentation is first defined. It is assumed that an ideal image $f_{true}(x,y)$ is composed of constant value regions $\{R_k\}$, where $\mathfrak{m}_k$ designates the true value for region R_k,

$$f_{true}(x,y) = \mathfrak{m}_k, \quad \text{for } \forall\,(x,y) \in R_k \tag{8.1}$$

The observed image values, $f(x,y)$, result from the addition of Gaussian white noise e(x,y) to the ideal image:

$$f(x,y) = f_{true}(x,y) + e(x,y) \tag{8.2}$$

where the variance of the noise is σ^2.

The goal of a image segmentation process could be then to find the true image partition $\{R_k\}$. Let S_i designate any arbitrary subpart of a true region R_k, $S_i \subset R_k$. Thus, $f_{true}(x,y) = \mathfrak{m}_k$ for $(x,y) \in S_i$, and if we consider $\mathfrak{m}_i$ as the true value for the pixels of S_i, then we have $\mathfrak{m}_i = \mathfrak{m}_k$. Hence, image segmentation will consist in merging together the segments S_i that belong to the same true region R_k.

The merging of segments can be based upon hypothesis testing. For example, considering two arbitrary adjacent segments, S_i and S_j, a statistical test can be used to determine if they belong to the same true region R_k, $S_i \subset R_k$ and $S_j \subset R_k$. However, as the characteristics of R_k are unknown, the statistical decision must instead consider whether the true values $\mathfrak{m}_i$ and $\mathfrak{m}_j$ of segments S_i and S_j are the same.

8.2 Hypothesis testing

Classical hypothesis testing is now reviewed and applied to image segmentation. Each pixel value $f(x,y)$ inside a segment S_i, $(x,y) \in S_i$, is regarded as a random variable, which is Gaussian distributed with mean $\mathfrak{m}_i$ and variance σ^2, $\mathcal{N}(\mathfrak{m}_i, \sigma^2)$. A segmentation process must therefore decide whether two segments, S_i and S_j, have the same true mean value, $\mathfrak{m}_i = \mathfrak{m}_j$, or not.

A statistical decision process can be used to determine which one of the following two hypotheses is true [48].

$$\begin{aligned} \mathcal{H}_0 :&\quad \mathfrak{m}_i - \mathfrak{m}_j = 0 \qquad (\text{ or } \ \mathfrak{m}_i = \mathfrak{m}_j\,) \\ \mathcal{H}_A :&\quad \mathfrak{m}_i - \mathfrak{m}_j \neq 0 \end{aligned} \tag{8.3}$$

In order to simplify the analysis, the multiple alternative hypothesis $\mathcal{H}_A$ is replaced by a single alternative one, $\mathcal{H}_1$, which is defined by the parameter d_{true}, $d_{true} > 0$.

$$\mathcal{H}_1 \;:\quad |\mathfrak{m}_i - \mathfrak{m}_j| = d_{true} \tag{8.4}$$

Therefore, it is assumed that the difference between the true segment means are either equal to zero, or to d_{true}. The difference d, between segment means is a sufficient statistic for this test:

$$\begin{aligned} d &= \mu_i - \mu_j \\ d &= \frac{1}{N_i}\sum_{S_i} f(x,y) - \frac{1}{N_j}\sum_{S_j} f(x,y) \end{aligned} \tag{8.5}$$

where μ_i, μ_j and N_i, N_j are, respectively, the mean values and sizes of segment S_i and S_j. Therefore, the statistical decision consists in accepting $\mathcal{H}_0$ if d is small, more precisely, if $-\tau \leq d \leq \tau$, where τ is a selected threshold.

The performance of a test is judged according to its tendency to lead to wrong decisions. Two types of error can be considered:

$$\begin{aligned} &\text{Type I} &: \text{ rejecting } \mathcal{H}_0 \text{ when } \mathcal{H}_0 \text{ is true} \\ &\text{Type II} &: \text{ accepting } \mathcal{H}_0 \text{ when } \mathcal{H}_1 \text{ is true} \end{aligned} \tag{8.6}$$

The probability of these two types of error can be calculated [48].

Probability of type I error (α)

If $\mathcal{H}_0$ is true, the statistic d has a Gaussian distribution

$$d \approx \mathcal{N}(0, \sigma_d^2) \quad \text{with} \;\; \sigma_d^2 = (1/N_i + 1/N_j)\,\sigma^2 \tag{8.7}$$

and where N_i and N_j are, respectively, the sizes of segments S_i and S_j. Therefore, the probability of type I error is:

$$\alpha = 1 - \int_{-\tau}^{\tau} \frac{1}{\sqrt{2\pi}\,\sigma_d}\; \exp\left(\frac{-x^2}{2\sigma_d^2}\right) dx \tag{8.8}$$

Probability of type II error (β)

If $\mathcal{H}_1$ is true, the statistic d is also Gaussian distributed but with mean d_{true}:

$$d \approx \mathcal{N}(d_{true}, \sigma_d^2) \tag{8.9}$$

Table 8.1: Probabilities of errors for different segment sizes with $d_{true} = 3\sigma$ and $\tau = 1.5\sigma$.

Segment sizes	α	β
1 pixel	.289	.144
2 pixels	.134	.067
4 pixels	.034	.017

The probability of type II error is then:

$$\beta = \int_{-\tau}^{\tau} \frac{1}{\sqrt{2\pi}\,\sigma_d} \exp\left(\frac{-(x - d_{true})^2}{2\sigma_d^2}\right) dx \tag{8.10}$$

These probabilities of errors are functions of τ, σ_d^2 and d_{true}. They must both be low for a good decision process. For a given σ_d^2 and d_{true}, we can modify τ such as to reduced α or β, but both cannot be reduced at the same time. Therefore, some compromise must be reached, for example, select τ such that $\alpha = \beta$.

Small values for α and β simultaneously can be achieved only if $d_{true}\big/\sigma_d$ is large. As σ_d decreases with the segment sizes, the probabilities of errors are smaller for decisions involving larger segments. This is illustrated by Table 8.1 which gives the α and β values for different segment sizes, with $d_{true} = 3\sigma$ and $\tau = 1.5\sigma$.

8.3 Sequential testing in hierarchical segmentation

Hierarchical segmentation begins with many small segments which are sequentially merged to produce larger ones. Statistical decision can be employed to determine whether, or not, two adjacent segments must be merged. However, the sequential aspect of hierarchical segmentation must be considered in the design of the decision process. Thus, the type II error results from the merging of two different segments, and therefore, cannot be recovered by an agglomerative process. Whereas, a type I error keeps separated two similar segments which can be corrected in a following step. Therefore, it seems preferable to keep β at a low level to avoid type II errors. To develop

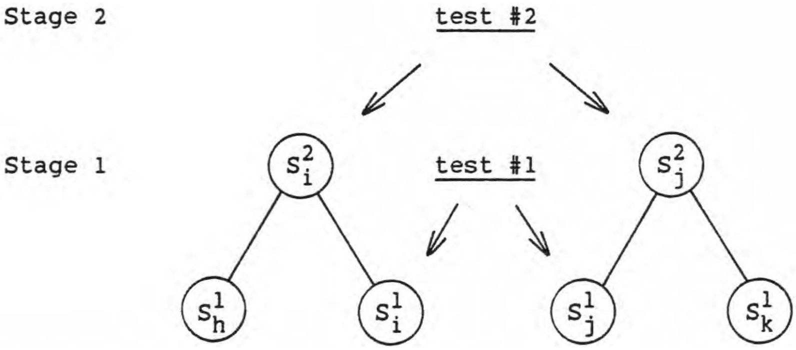


Figure 8.1: Sequence of segment testings in a hierarchy.

this point, the hierarchical segmentation is now regarded as a sequential testing process and the associated probabilities of errors are derived.

A two stage test for merging is first examined. In stage 1 (see Figure 8.1), the segments S_i^1 and S_j^1 are compared by a first test, test #1. If the segments are not merged after this test, they will, sooner or later, be involved in a second test. Before this second comparison, S_i^1 and/or S_j^1 are merged with some adjacent segments belonging to the same regions in order to produce S_i^2 and S_j^2, $S_i^1 \subset S_i^2 \subset R_i$ and $S_j^1 \subset S_j^2 \subset R_j$. The second stage test, test #2, considers, therefore, the segments S_i^2 and S_j^2 in which S_i^1 and S_j^1 are still disjoint. The same hypotheses, $\mathcal{H}_0$ vs $\mathcal{H}_1$, are employed at both stages. Let α_1, β_1 be the probabilities of errors for test #1, and α_2, β_2 for test #2. The probabilities of errors for the combined test are designated by α_{1+2} and β_{1+2}. If $\mathcal{H}_0$ is accepted in test #l, then as the segments are merged, test #2 is not needed.

If $\mathcal{H}_0$ is true, we have:

$$\begin{aligned} Prob(\text{ accept } \mathcal{H}_0 \text{ at test } \#1\,) &= 1-\alpha_1 \\ Prob(\text{ accept } \mathcal{H}_0 \text{ at test } \#2\,) &= \alpha_1\,(1-\alpha_2^*) \end{aligned} \tag{8.11}$$

where α_2^* is the probability that test #2 rejects $\mathcal{H}_0$ when $\mathcal{H}_0$ has been rejected by test #l, $\alpha_2 \leq \alpha_2^* \leq 1$. Then, we obtain:

$$\begin{aligned} \alpha_{1+2} = 1 \;&-\; Prob(\text{ accept } \mathcal{H}_0 \text{ at test } \#1\,) \\ &-\; Prob(\text{ accept } \mathcal{H}_0 \text{ at test } \#2\,) \\ \alpha_{1+2} = \alpha_1\,\alpha_2^* & \end{aligned} \tag{8.12}$$

If $\mathcal{H}_1$ is true, we have

$$\begin{aligned} Prob(\text{ accept } \mathcal{H}_0 \text{ at test } \#1\,) &= \beta_1 \\ Prob(\text{ accept } \mathcal{H}_0 \text{ at test } \#2\,) &= (1-\beta_1)\,\beta_2^* \end{aligned} \tag{8.13}$$

where β_2^* is the probability that test #2 accepts $\mathcal{H}_0$ when $\mathcal{H}_0$ has been rejected by test #1, $0 \le \beta_2^* \le \beta_2$. Thus, we obtain:

$$\begin{aligned} \beta_{1+2} &= Prob(\text{ accept } \mathcal{H}_0 \text{ at test } \#1\,) \\ &\qquad + Prob(\text{ accept } \mathcal{H}_0 \text{ at test } \#2\,) \\ \beta_{1+2} &= \beta_1 + (1-\beta_1)\,\beta_2^* \;\simeq\; \beta_1 + \beta_2^* \end{aligned} \tag{8.14}$$

where the term $\beta_1\beta_2^*$ is usually small and thus can be ignored.

It can be noted that if the two tests are identical (i.e. if they always give the same results), then $\alpha_2^* = 1$ and $\beta_2^* = 0$. If the tests are independant (i.e. if the results of tests #1 do not affect the results of test #2) than $\alpha_2^* = \alpha_2$ and $\beta_2^* = \beta_2$.

If a third step is then added to the process, we obtain:

$$\begin{aligned} \alpha_{1+2+3} &= (\alpha_{1+2})\,\alpha_3^* &&= \alpha_1\,\alpha_2^*\,\alpha_3^* \\ \beta_{1+2+3} &\simeq (\beta_{1+2}) + \beta_3^* &&\simeq \beta_1 + \beta_2^* + \beta_3^* \end{aligned} \tag{8.15}$$

and, more generally, for a m step process, we have:

$$\begin{aligned} \alpha_{1+\cdots+m} &= \alpha_1\,\alpha_2^* \cdots \alpha_m^* \\ \beta_{1+\cdots+m} &\simeq \beta_1 + \cdots + \beta_m^* \end{aligned} \tag{8.16}$$

The probability of type I error is, therefore, reduced from one stage to the next. Hence, a high value can be assigned to α_1, as the following tests will reduce the overall α. It is even shown that:

$$\alpha_{1\cdots+m} \le \text{Minimum }\,(\,\alpha_1,\,\alpha_2,\,\ldots,\,\alpha_m\,) \tag{8.17}$$

On the other hand, the probability of type II error increases from stage to stage. A large β at the first stage cannot be subsequently reduced.

$$\beta_{1\cdots+m} \ge \text{Maximum }\,(\,\beta_1,\,\beta_2,\,\ldots,\,\beta_m\,) \tag{8.18}$$

An upper bound for $\beta_{1\cdots+m}$ is given by

$$\beta_{1\cdots+m} \le \beta_1 + \beta_2 + \cdots + \beta_m \tag{8.19}$$

As an example, Table 8.2 shows the probabilities of errors for a three stage process using the same threshold value, $\tau = 1\,\frac{1}{2}\sigma$. The segment sizes are respectively of 1, 2 and 4 pixels for stage 1, 2 and 3. As noted before, the probabilities of errors decrease with the segment sizes. The progression of the lower and upper bound of β_{1+2+3} is also reported. The bound for β_{1+2+3} will usually be determined by test #1 where the value of β_1 (= 0.144) is high,

Table 8.2: Probabilities of error for sequential testing with the same threshold ($d_{true} = 3\,\sigma$).

Test	Threshold	α_k	β_k	$\beta_{1\,\cdots+k}$ bounds	
				lower	upper
# 1	$1.5\,\sigma$	.289	.144	.144	.144
# 2	$1.5\,\sigma$	.134	.067	.144	.211
# 3	$1.5\,\sigma$	.034	.017	.144	.228

Table 8.3: Probabilities of error for sequential testing with different thresholds ($d_{true} = 3\,\sigma$).

Test	Threshold	α_k	β_k	$\beta_{1\,\cdots+k}$ bounds	
				lower	upper
# 1	$\frac{1}{4}\,\sigma$	.859	.015	.015	.015
# 2	$\frac{3}{4}\,\sigma$	.453	.012	.015	.027
# 3	$1\,\frac{1}{2}\,\sigma$	.034	.017	.017	.044

$0.144 \leq \beta_{1+2+3} \leq 0.228$. The upper bound for α_{1+2+3}, which is the minimum of α_k values, will instead be determined by test #3, $\alpha_{1+2+3} \leq \alpha_3 = 0.034$.

By reducing the threshold values of test #l and test #2, the β_{1+2+3} bounds can be reduced without changing the upper bound of α_{1+2+3}. In Table 8.3, the threshold values are chosen such that β_k values are small and rather equal for the three stages. This results in smaller bounds for β_{1+2+3}, $0.017 \leq \beta_{1+2+3} \leq 0.044$. On the other hand, the corresponding increases in α_1 and α_2 have not changed the upper bound of α_{1+2+3}, which is still determined by test #3. $\alpha_{1+2+3} \leq \alpha_3 = 0.034$.

In hierarchical segmentation, it is, therefore, advantageous to increase the α values in the first stages and then subsequently reduce the α, in order to keep β at an appropriately low level for each stage. This concept is exploited in the next section.

8.4 Stepwise optimization

In hierarchical segmentation, it is preferable for each stage to keep β_k, the probability of type II error, as low as possible. Unfortunately, β_k cannot usually be evaluated because d_{true} is unknown. Also, the probability of type I error, α_k must be employed instead to select the appropriate threshold value τ. Thus, the evaluation of the maximum value allowed for α_k at stage k is now examined, the maximization of α_k being associated with the minimization of β_k.

At each stage or segment level, there are many possible segment mergers, which can be represented by segment pairs (S_i, S_j). The mean difference statistics, $d_{i,j} = \mu_i - \mu_j$, can be calculated for each pair. Under the $\mathcal{H}_0$ hypothesis, this statistic can be associated with a confidence level, $v_{i,j}$, which is just the probability of obtaining a value d such that, $|d| \leq |d_{i,j}|$:

$$\begin{aligned} v_{i,j} &= Prob(\ |d| \leq |d_{i,j}| \ \Big| \ \mathcal{H}_0\ ) \\ v_{i,j} &= \int_{-|d_{i,j}|}^{|d_{i,j}|} \frac{1}{\sqrt{2\pi}\,\sigma_d}\ \exp\left(\frac{-x^2}{2\sigma_d}\right)\, dx \\ v_{i,j} &= 2\ \mathrm{ERF}\left(d_{i,j} \Big/ \sigma_d\right) \end{aligned} \tag{8.20}$$

where

$$\begin{aligned} \sigma_d^2 \ &= \ (1 \big/ N_i + 1 \big/ N_j)\, \sigma^2 \\ \mathrm{ERF}\,(y) &= \int_0^y \frac{1}{\sqrt{2\pi}}\ \exp\left(-\frac{1}{2}\,x^2\right)\, dx \end{aligned}$$

Using $v_{i,j}$ instead of $d_{i,j}$, a statistical decision process accepts the hypothesis $\mathcal{H}_0$ and merges segments only if:

$$v_{i,j} \geq 1 - \alpha \quad \text{or} \quad \alpha \leq 1 - v_{i,j} \tag{8.21}$$

Defining v_{min} as the minimum over $v_{i,j}$, $v_{min} = Min(v_{i,j})$, then the utilization of an α greater than $1 - v_{min}$ implies that no segments are merged, which renders the stage redundant. Therefore, the maximum allowed value for α is $\alpha_{max} = 1 - v_{min}$, which results at least in one merger. Hence, a hierarchical segmentation algorithm can employ a stepwise process which finds the segment pair with the minimum confidence level $v_{i,j}$ and merges the corresponding segments. This is equivalent to using the maximum allowed α value for each stage.

Stepwise optimization, by maximizing α_k, assures that, at each step, the probability of type II error β_k is kept to it lowest value. This should also keep $\beta_{1+\dots+m}$ at a low value.

8.5 A stepwise criterion and its probability functions

It has been shown that the stepwise minimization of $v_{i,j}$ reduces the probabi1ity of error. In the following four sections, the probability of error produced by this stepwise optimization process is calculated and analysed. A statistic used as a stepwise criterion is first introduced and the probability functions are defined. The statistic employed results from the normalization of $d_{i,j}$:

$$d^*_{i,j} = \frac{|d_{i,j}|}{\sigma_d} = \sqrt{\frac{N_i\,N_j}{N_i + N_j}}\;\frac{|\mu_i - \mu_j|}{\sigma} \tag{8.22}$$

where σ^2 is the variance of noise. The segment pair, S_i and S_j, that minimizes $d^*_{i,j}$ will also minimize $v_{i,j}$ because:

$$v_{i,j} = 2\ \mathrm{ERF}\left(d^*_{i,j}\right) \tag{8.23}$$

Therefore, $d^*_{i,j}$ can be used as the stepwise criterion.

The probability functions of $d^*_{i,j}$ can easily be evaluated. Hence, under the $\mathcal{H}_0$ hypothesis it follows that (see Figure 8.2):

$$\begin{aligned} \mathrm{P}_{\mathcal{H}0}(d^*_{i,j}) &= \ Prob(\,d^* \le d^*_{i,j} \;\Big|\; \mathcal{H}_0\,) \\ &= \ 2\ \mathrm{ERF}\left(\,d^*_{i,j}\,\right) \end{aligned} \tag{8.24}$$

In using $d^*_{i,j}$ (or $v_{i,j}$) in a statistical decision, the alternative hypothesis $\mathcal{H}_1$ must be rewritten as:

$$\mathcal{H}_1 \ : \quad \sqrt{\frac{N_i\,N_j}{N_i + N_j}}\;\frac{|\mathfrak{m}_i - \mathfrak{m}_j|}{\sigma} \ = \ d^*_{true} \tag{8.25}$$

The probabi1ity function of $d^*_{i,j}$ under $\mathcal{H}_1$ is (see Figure 8.2):

$$\begin{aligned} \mathrm{P}_{\mathcal{H}1}(d^*_{i,j}) &= \ Prob(\,d^* \le d^*_{i,j} \;\Big|\; \mathcal{H}_1\,) \\ &= \int_{-d^*_{i,j}}^{d^*_{i,j}} \frac{1}{\sqrt{2\pi}}\ \exp\left(-\frac{1}{2}\,(x - d^*_{true})^2\right)\,dx \\ &= \begin{cases} \mathrm{ERF}\left(d^*_{true} + d^*_{i,j}\right) \\ \qquad -\mathrm{ERF}\left(d^*_{true} - d^*_{i,j}\right) & \text{if } d^*_{i,j} \le d^*_{true} \\ \\ \mathrm{ERF}\left(d^*_{i,j} + d^*_{true}\right) \\ \qquad +\mathrm{ERF}\left(d^*_{i,j} - d^*_{true}\right) & \text{if } d^*_{i,j} \ge d^*_{true} \end{cases} \end{aligned} \tag{8.26}$$

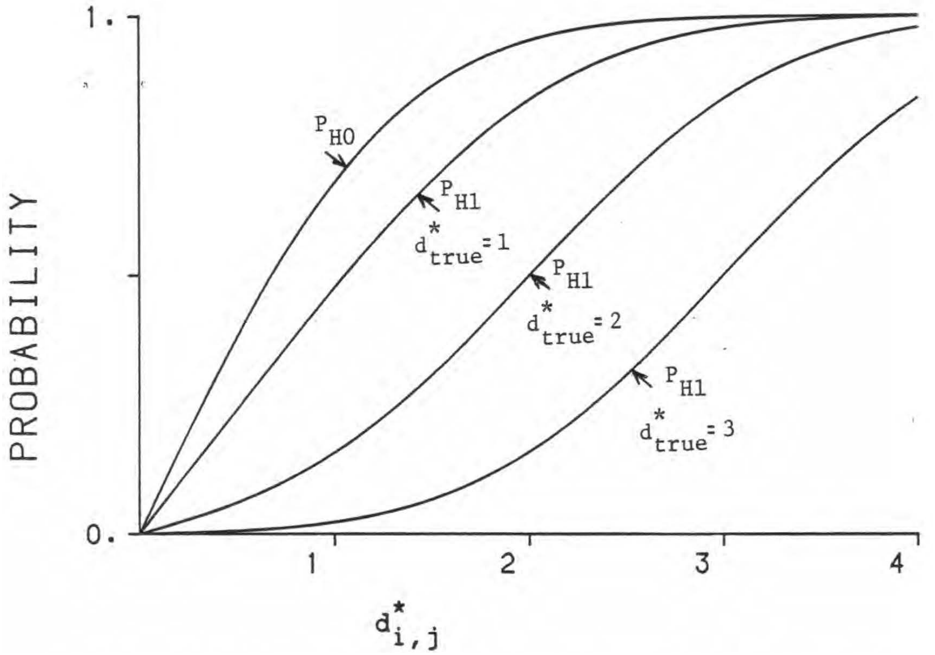


Figure 8.2: Probability functions of $d^*_{i,j}$ under $\mathcal{H}_0$ ($\mathrm{P}_{\mathcal{H}0}$) and $\mathcal{H}_1$ ($\mathrm{P}_{\mathcal{H}1}$) hypotheses.

Thus, the probability of type II error under $\mathcal{H}_1$ is now a function of d^*_{true} and is equal to:

$$\beta = \mathrm{P}_{\mathcal{H}1}(\tau^*) \tag{8.27}$$

where τ^* is the decision threshold, ($\mathcal{H}_0$ is accepted if $d^*_{i,j} \leq \tau^*$).

It must be noted that, under $\mathcal{H}_0$, the probability of $d^*_{i,j}$ is no longer a function of the segment sizes N_i and N_j. Under $\mathcal{H}_1$, the probability of $d^*_{i,j}$ and also of type II error is only a function of d^*_{true}, which integrates the effect of the difference between the true segment means, $\mathfrak{m}_i - \mathfrak{m}_j$, and the effect of segment sizes.

The probability function for the minimum value of a set of $d^*_{i,j}$ is now derived. Consider a set of K independant random variables, x_i, with identical probability functions $\mathrm{P}(x)$, and let x_{min} be the minimum of this set. For $K = 2$, it can be shown that the probability function of x_{min} is [66]:

$$\mathrm{P}_{min}(x_{min}) = 2\ \mathrm{P}(x_{min}) - \mathrm{P}(x_{min})^2 \tag{8.28}$$

and more generally, for $K \geq 2$, (see Appendix A)

$$\mathrm{P}_{min,K}(x_{min}) = \sum_{j=1}^{K} (-1)^{j+1} \frac{K!}{j!\,(K-j)!}\ \mathrm{P}(x_{min})^j \tag{8.29}$$

However, if the minimum is calculated from a set of $d^*_{i,j}$ values with distribution given by $\mathrm{P}_{\mathcal{H}0}$ or $\mathrm{P}_{\mathcal{H}1}$, then the probability function $\mathrm{P}(x)$ in the preceding formula must be replaced by either $\mathrm{P}_{\mathcal{H}0}$ or $\mathrm{P}_{\mathcal{H}1}$.

8.6 Stepwise error probability

The stepwise optimization approach considers, at each step, all possible segment mergers, represented by segment pairs (S_i, S_j). The statistic $d^*_{i,j}$ is calculated for each pair, the minimum among these values is found, and the corresponding segments are merged. A type II error occurs if the minimum statistic d^*_{min} comes from segments belonging to the $\mathcal{H}_1$ hypothesis. This would mean that two dissimilar segments are merged.

The probability of error at each step is now evaluated. Although some assumptions are made in order to simplify the derivation, the deduction from this simpler model can be extended to more realistic ones. Hence, we assume that the random variables, $d^*_{i,j}$, are independant, and are divided into two sets, $D_{\mathcal{H}0}$ and $D_{\mathcal{H}1}$. $D_{\mathcal{H}0}$ contains the values produced by segments belonging to $\mathcal{H}_0$, while $D_{\mathcal{H}1}$ corresponds to those of the $\mathcal{H}_1$ hypothesis.

An error occurs if the minimum over $D_{\mathcal{H}1}$ is smaller than the minimum over $D_{\mathcal{H}0}$:

$$\underset{D_{\mathcal{H}1}}{\mathrm{Minimum}} \left\{ d^*_{i,j} \right\} < \underset{D_{\mathcal{H}0}}{\mathrm{Minimum}} \left\{ d^*_{i,j} \right\} \tag{8.30}$$

Let $\mathrm{P}_{min,K_0}(d^*_{min} \mid \mathcal{H}_0)$ designate the probability function for the minimum value over $D_{\mathcal{H}0}$, and $\mathrm{P}_{min,K_1}(d^*_{min} \mid \mathcal{H}_1)$ for the minimum over $D_{\mathcal{H}1}$, where K_0 and K_1 are the numbers of elements in $D_{\mathcal{H}0}$ and $D_{\mathcal{H}1}$. Thus, the probability of error is:

$$Prob(\,error\,) = \int_0^\infty \mathrm{P}_{min,K_1}(\,x \mid \mathcal{H}_1)\; p_{\mathcal{H}0}(x)\; dx \tag{8.31}$$

where $p_{\mathcal{H}0}$ is the probability density:

$$p_{\mathcal{H}0}(z) = \frac{d}{dx}\,\mathrm{P}_{min,K_0}(\,x \mid \mathcal{H}_0)\Big|_{x=z}$$

The probability of error can be regarded as the average of the probability that the minimum over $D_{\mathcal{H}1}$ is lower than x, weighted by the probability (density) that the minimum over $D_{\mathcal{H}0}$ equals x, $p_{\mathcal{H}0}(x)$.

The evaluation of P_{min,K_0} and P_{min,K_1} can be done by using equation (8.29) where P is replaced by $\mathrm{P}_{\mathcal{H}0}$ or $\mathrm{P}_{\mathcal{H}1}$. The probability of error can then be calculated by numerical integration. Figure 8.3 presents the results for some parameter values. The probability of error is a function of d^*_{true}, K_0 and

K_1. However, it can be observed that the probability is mainly affected by d^*_{true} and the ratio K_0/K_1. The importance of the ratio K_0/K_1 is illustrated by the fact that, for $d^*_{true} = 0$, the probability is equal to $K_1/(K_0 + K_1)$ or $(1 + K_0/K_1)^{-1}$. Thus, the probability of error is reduced if d^*_{true} or the K_0/K_1 ratio is increased.

8.7 Error and segment sizes

Hierarchical segmentation begins with many small segments which are sequentially merged to produce larger ones. However, the probability of error is a function of the segment sizes. Hence, the parameter d^*_{true} decreases with the segment sizes, thus small segments tend to possess small d^*_{true} values which make it difficult to decide if d^*_{true} come from $\mathcal{H}_0$ or $\mathcal{H}_1$.

In stepwise optimization, the probability of error is not only a function of d^*_{true}, but also of K_0 and K_1. While small segments tend to involve small d^*_{true} values, they are also associated with large K_0/K_1 ratios. The large value of K_0/K_1 compensates for the small value of d^*_{true} and keeps the probability of error at a low level.

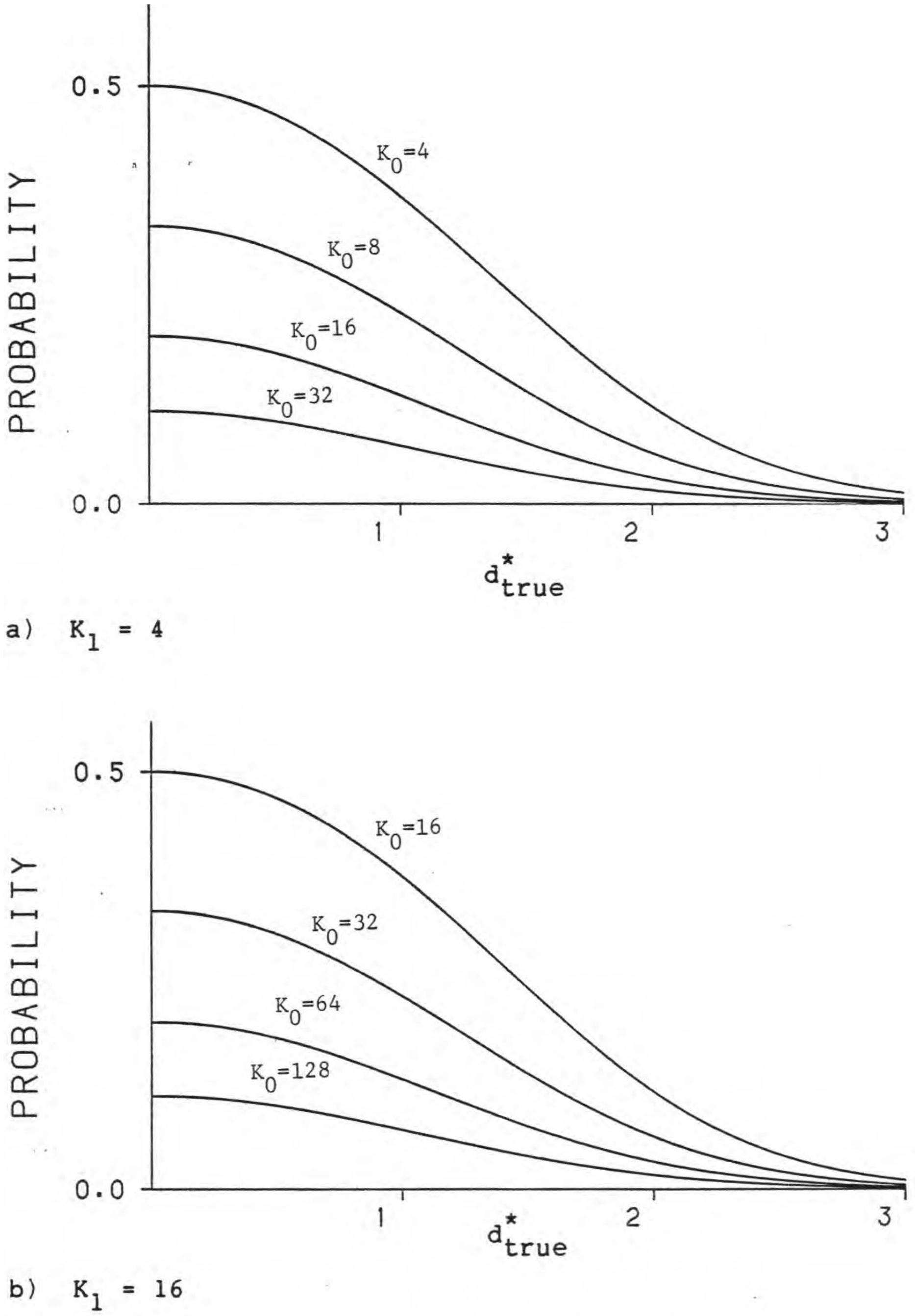
An example is employed to show the relation between K_0/K_1 and the segment sizes. Figure 8.4 shows a image containing two regions which are divided into segments. In the example A, the segments contain only one pixel each, while in B they are composed of four pixels. The number of possible segment merges corresponds to the number of 4-adjacent segment pairs. In example A, there are 82 different segment pairs, and for each of them, we can calculate the statistics $d^*_{i,j}$. Among these pairs, 76 involve segments that belong to the same region, $K_0 = 76$, while 6 contain segments from both regions, $K_1 = 6$. Thus, the ratio K_0/K_1 is equal to $12\frac{2}{3}$. For the example B, there are 17 different segment pairs, and $K_0 = 14$, $K_1 = 3$ and $K_0/K_1 = 4\frac{2}{3}$. These examples show that the ratio K_0/K_1 is higher for the image with the smaller segments. Generally, a large K_0/K_1 ratio can be expected if there are much more segments inside regions than on the boundaries.

The probabilities of error for these two examples are now calculated. Let $\mathfrak{m}_1$ and $\mathfrak{m}_2$ be the true values for region 1 and 2, and let σ be the noise level. For example A, the value of d^*_{true} is

$$d^*_{true,A} = \sqrt{\frac{1 \cdot 1}{1+1}}\,\frac{|\mathfrak{m}_1 - \mathfrak{m}_2|}{\sigma} = \frac{1}{\sqrt{2}\,\sigma}\,|\mathfrak{m}_1 - \mathfrak{m}_2|$$

while, for example B,

$$d^*_{true,B} = \sqrt{\frac{4 \cdot 4}{4+4}}\,\frac{|\mathfrak{m}_1 - \mathfrak{m}_2|}{\sigma} = \frac{2}{\sqrt{2}\,\sigma}\,|\mathfrak{m}_1 - \mathfrak{m}_2|$$



a) $K_1 = 4$

b) $K_1 = 16$

Figure 8.3: Stepwise probability of error in function of d^*_{true} for different values of K_0 and K_1.

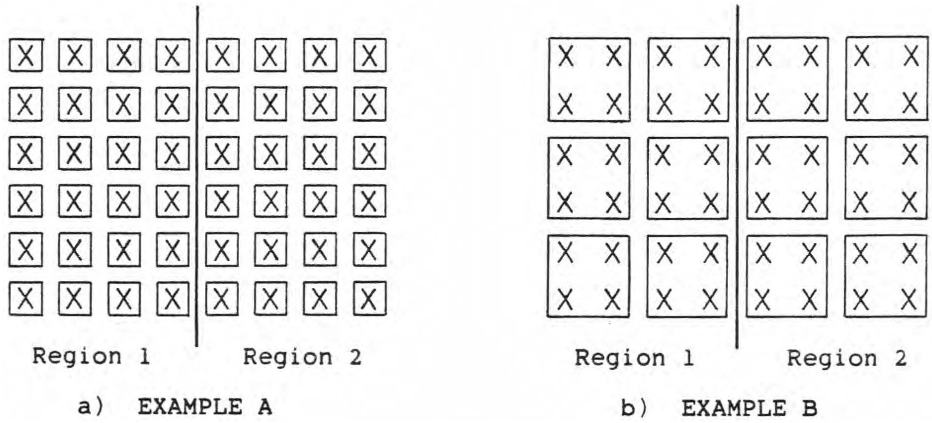


Figure 8.4: Division of a two region image into segments a) segments of one pixel, b) segments of four pixels.

The probability of stepwise optimization error is the probability that the minimum of the K_1 statistics values is lower than the corresponding minimum from the K_0 values. These statistics are assumed to be independent and the probabilities of error are calculated with equation (8.31). The results for different values of $|\mathfrak{m}_1 - \mathfrak{m}_2|$ are reported in Table 8.4.

It can be noted that the probabilities of error are higher for example A than for B, when $|\mathfrak{m}_i - \mathfrak{m}_j|$ is greater than σ, the standard deviation of noise. This suggests that, in hierarchical segmentation, the probabilities of error are higher in the first steps which involves small segments. The rapid reduction of the probability of error with the increase of $|\mathfrak{m}_i - \mathfrak{m}_j|$ must also be noted. These probability values can be advantageously compared with those of classical hypothesis testing. For example, consider the probabilities of type II error for a test based upon the difference of segment means, $|\mu_i - \mu_j|$, where the $\mathcal{H}_0$ hypothesis is accepted if the difference is lower than a threshold, τ, equated to $|\mathfrak{m}_i - \mathfrak{m}_j|/2$. The probability values are shown in Table 8.5, and are all larger than those of Table 8.4.

8.8 Error probability vs minimum criterion value

The interrelation between steps has not been considered in the evaluation of the stepwise error probability in the preceding sections. The minimum criterion value for each step k, d^*_{min}, forms a sequence which tends to increase as segment merging eliminates small values. The error probability is thus

Table 8.4: Probability of step-wise error.

	Probability of error	
$\lvert\mathfrak{m}_i - \mathfrak{m}_j\rvert$	example A	example B
1 σ	.0578	.07100
2 σ	.0280	.00383
3 σ	.0081	.00003
4 σ	.0014	.00000

Table 8.5: Probability of type II error for a threshold of $\lvert\mathfrak{m}_i - \mathfrak{m}_j\rvert/2$.

	Probability of error	
$\lvert\mathfrak{m}_i - \mathfrak{m}_j\rvert$	example A	example B
1 σ	.2174	.2228
2 σ	.2228	.0786
3 σ	.1437	.0169
4 σ	.0786	.0023

re-examined in order to take into account the observed minimum criterion value for the step.

Let $\mathrm{P}_{min,K_0}(d^*_{min} \mid \mathcal{H}0)$ designate the probability function for the minimum value over $D_{\mathcal{H}0}$ and $\mathrm{P}_{min,K_1}(d^*_{min} \mid \mathcal{H}_1)$ over $D_{\mathcal{H}1}$. Thus, if the observed minimum value is d^*_{min}, the conditional probability of error is:

$$Prob(\,error \mid d^*_{min}) \quad = \quad \frac{A}{A+B} \tag{8.32}$$

where

$$A = (\,1 - \mathrm{P}_{min,K_0}(d^*_{min} \mid \mathcal{H}_0)\,)\;\; p_{\mathcal{H}1}(d^*_{min})$$

$$B = (\,1 - \mathrm{P}_{min,K_1}(d^*_{min} \mid \mathcal{H}_1)\,)\;\; p_{\mathcal{H}0}(d^*_{min})$$

and where $p_{\mathcal{H}0}$ and$p_{\mathcal{H}1}$ are the probability densities:

$$p_{\mathcal{H}0}(z) = \frac{d}{dx}\,\mathrm{P}_{min,K_0}(\,x \mid \mathcal{H}_0)\Big|_{x=z}$$

$$p_{\mathcal{H}1}(z) = \frac{d}{dx}\,\mathrm{P}_{min,K_1}(\,x \mid \mathcal{H}_1)\Big|_{x=z}$$

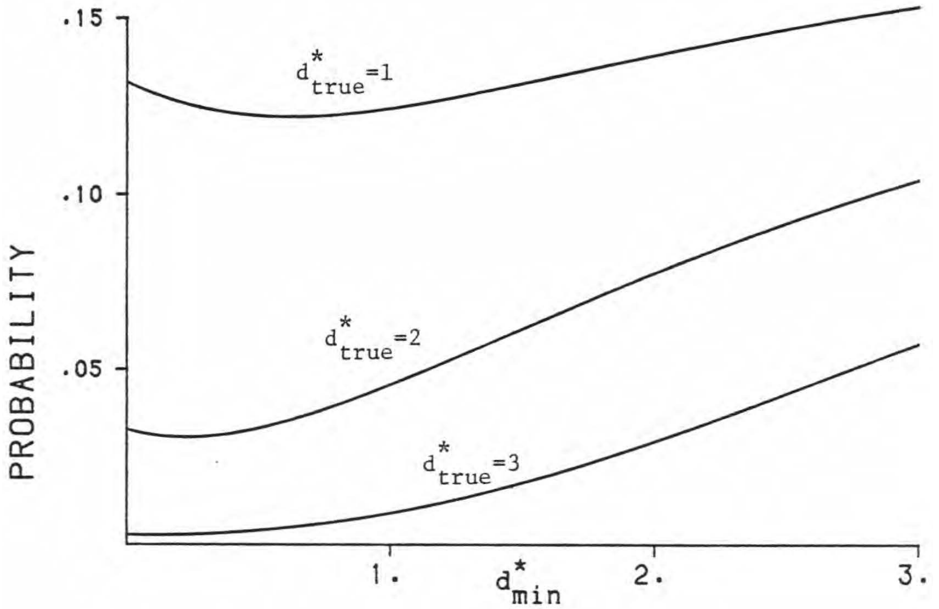


Figure 8.5: Stepwise probability of error for $K_0 = 16$ and $K_1 = 4$.

The probability values, as shown in Figure 8.5, increase with the observed d^*_{min}. This increase is less important when d^*_{true} is small, meaning that $\mathcal{H}_0$ and $\mathcal{H}_1$ are difficult to distinguish. Thus, it is useful to consider the observed d^*_{min} in addition to the other parameters, d^*_{true}, K_0 and K_1, in the evaluation of the stepwise probability of error, a low d^*_{min} value being associated with a small probability of error.

8.9 Minimum criterion value sequence

In the following two sections, the problem of discriminating between the signal and the noise components of a image is examined. In hierarchical segmentation, this corresponds to defining when the segment merging must be stopped. The probability of stepwise error cannot unfortunately be employed in real applications because the evaluation is based upon the d^*_{true}, K_0 and K_1 parameters which are generally unknown. Instead, the sequence of minimum values for the criterion at each step k, $d^*_{min,k}$, is used to characterize the image structure and to distinguish between signal and noise components. The case of a white noise image with uniform background is first examined. The results of this analysis are used in the next section to show how the signal components of a image are distinguished from the noise.

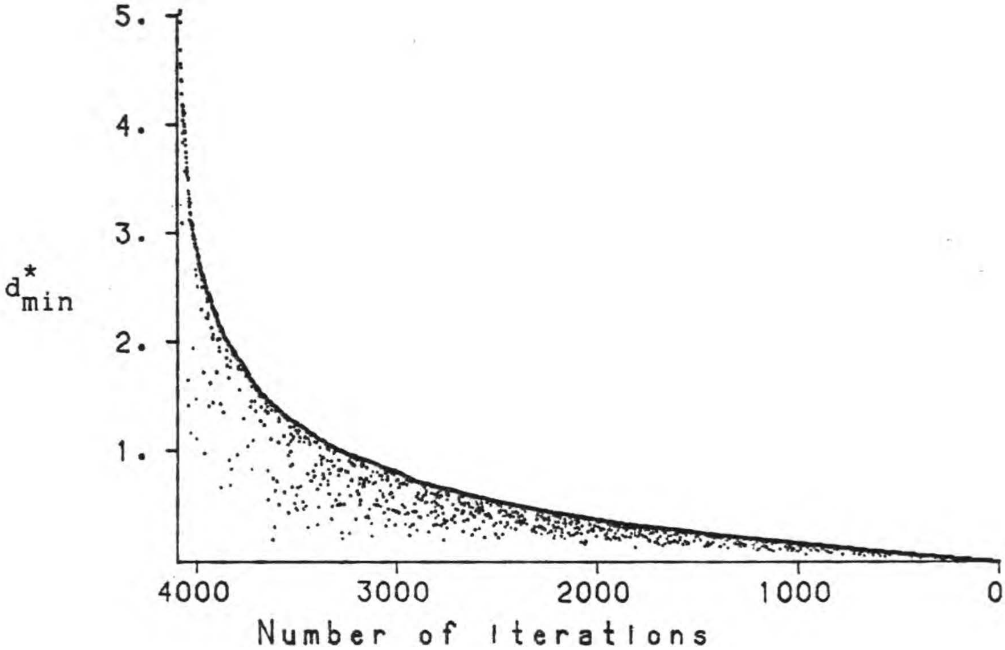


Figure 8.6: Sequence of minimum criterion values for a white noise image.

The stepwise optimization algorithm is applied upon a Gaussian white noise image with a zero mean and a standard deviation of 50, (64×64 pixels). All segment pairs found in the image belong therefore to the $\mathcal{H}_0$ hypothesis. The image is initially divided into 4096 (= 64×64) segments of one pixel, and 4095 merging steps are performed, the number of segments being reduced by one at each iteration. The resultant sequence of minimum criterion values, $d^*_{min,k}$, is shown in Figure 8.6. k is the iteration or step number. This figure is composed of 4095 dots that mostly concentrate around the compact line. The first steps of the algorithm yield $d^*_{min,k}$ values close to zero (right-most points of the curve), then increasing in the following steps as bigger segments are merged (moving leftward).

The relation between the $d^*_{min,k}$ curve and the probability function of the criterion is now examined. Consider first the simpler problem where a set of n criterion values, $D = \{d^*_{i,j}\}$, are ordered, and where $d^*_{min,k}$ corresponds to the k^{th} lowest value of the set D. Then, it can be shown ([48] Chap. 11) that

$$\mathrm{P}_{\mathcal{H}0}(\ \mathcal{E}\{d^*_{min,k}\}\ ) = k \Big/ n + 1 \tag{8.33}$$

where $\mathrm{P}_{\mathcal{H}0}$ is the probability function of the criteria $d^*_{i,j}$, $\mathcal{E}\{\cdot\}$ indicates the mathematical expectation or the mean value, and n is the number of elements in the criterion set. This implies that the mean value of $d^*_{min,k}$ is a

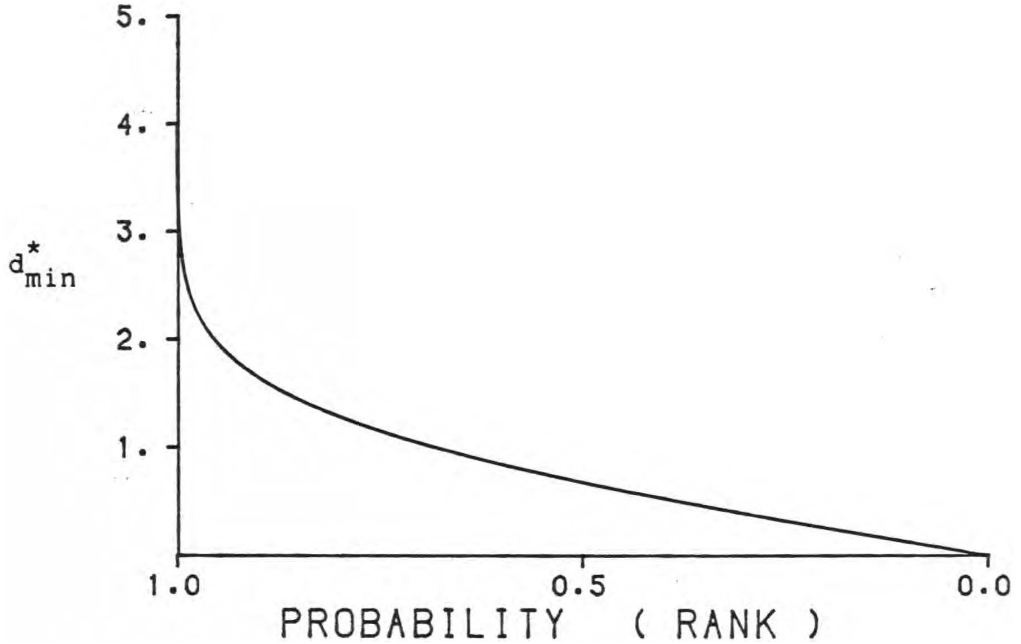


Figure 8.7: The inverse function of $\mathrm{P}_{\mathcal{H}0}(d^*_{min})$.

function of the rank k in the ordered list. The probability function $\mathrm{P}_{\mathcal{H}0}$ is drawn in Figure 8.7. Using the equation (8.33), the probability axis of the curve can be associated with the criterion rank, k. This result, obtained by ordering the set D, can also be produced by a stepwise optimization process where, at each iteration, the minimum $d^*_{min,k}$ is removed from the set D. Thus, at step k, the minimum, $d^*_{min,k}$, is extracted from a set where the preceding minimums, $d^*_{min,k-1}, \cdots d^*_{min,1}$, have been removed, $d^*_{min,k}$ corresponding, therefor to the k^{th} lowest value of the initial set D.

In the hierarchical segmentation algorithm, at each step, there are a number of criterion values that are removed from the set D and new ones are added. Accounting for this in the derivation of the equation for the $d^*_{min,k}$ curve is a difficult, if not impossible, task. Two aspects which distinguish the Figure 8.6 from the Figure 8.7 should be mentioned. On one hand, the addition of new criterion values implies that the minimum of the next step, $d^*_{min,k+1}$, can be lower than the current minimum. Hence, in Figure 8.6, there are many points located below the more compact curve. The occurrence of these lower points is rather random and unpredictable, while the more compact curve can be analysed and used to characterize the image structure.

On the other hand, the number of criterion values is reduced on average by more than one at each iteration. In equation (8.33), n corresponds simultaneously to the number of steps and to the initial number of elements in the set D. In the hierarchical segmentation, the number of steps is equal to the initial number of segments minus one, $n_{init} - 1$, while the initial number of elements in the set D is approximately equal to $2 \times n_{init}$. Using equation (8.33), the Figure 8.6 and the Figure 8.7 can be compared only if n is equated to the number of steps, and in this case, the curve of Figure 8.6 increases slower than expected for the first steps, while the values become rapidly higher in the last steps. We have found empirically a more appropriate equation for the Figure 8.6:

$$\mathrm{P}_{\mathcal{H}0}(\ \mathcal{E}\{d^*_{min,k}\}\ ) = k \Big/ (\,2\,n_{init} - k\,) \tag{8.34}$$

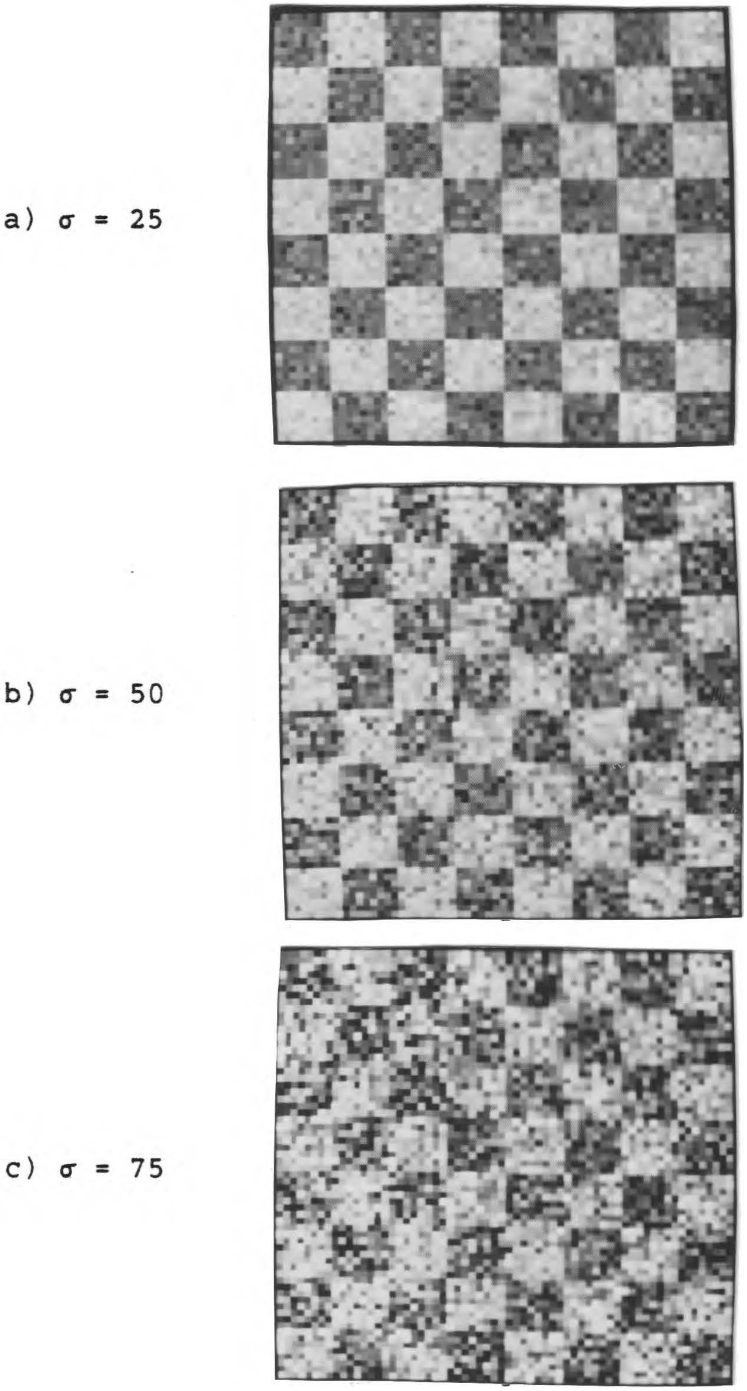
where n_{init} is the inital number of segments.

This discussion gives some indication on the progression of the $d^*_{min,k}$ values under the $\mathcal{H}_0$ hypothesis. In particular, the role of the probability function $\mathrm{P}_{\mathcal{H}0}$ is shown.

8.10 Signal vs noise

The progression of the $d^*_{min,k}$ values is now employed to discriminate between noise and signal components. The presence of segment pairs belonging to both hypotheses, $\mathcal{H}_0$ and $\mathcal{H}_1$, means that the probability function used in equations (8.33) and (8.34) must now combine both $\mathrm{P}_{\mathcal{H}0}$ and $\mathrm{P}_{\mathcal{H}1}$. The utilization of the composite probability function is associated with $d^*_{min,k}$ curves having higher values than those obtained from the pure noise case.

Some examples are now employed to show the effect of the signal components on $d^*_{min,k}$ values. They are produced from a checkerboard where the two tones are designated by $\mathfrak{m}_1$ (=0) and $\mathfrak{m}_2$ (=100), and on which Gaussian noise with different variances is added; namely, σ = 25, 50 and 75 (see Figure 8.8). Except for the final steps, these examples produce $d^*_{min,k}$ values that follow a progression similar to the pure noise case, shown in Figure 8.7. This similarity reflects the fact that, until the final steps, the algorithm merges segments belonging predominately to the same region. In the last steps, however, the algorithm is forced to merge dissimilar segments. Figure 8.9 shows the $d^*_{min,k}$ values for the last few steps. The points where the curve shapes deviate from the pure noise pattern can easily be identified, and are indicated by arrows. Figure 8.10 presents the image segmentations produced when the segment merging is stopped at these points. In Figure 8.9-b, where the noise variance (σ = 25) is small compared to the signal ($|\mathfrak{m}_1 - \mathfrak{m}_2|$ = 100), there is an important jump of the $d^*_{min,k}$ values when the merging of dissimilar segments begins. Whereas, it is difficult to identify the true regions when the noise variance is large (see Figure 8.8-c where $\sigma = 75$). This produces a $d^*_{min,k}$ curve (Figure 8.9-d) more similar to the pure noise curve (Figure 8.9-a). There is no jump in the $d^*_{min,k}$ values, but instead, only a change of the curve slope.

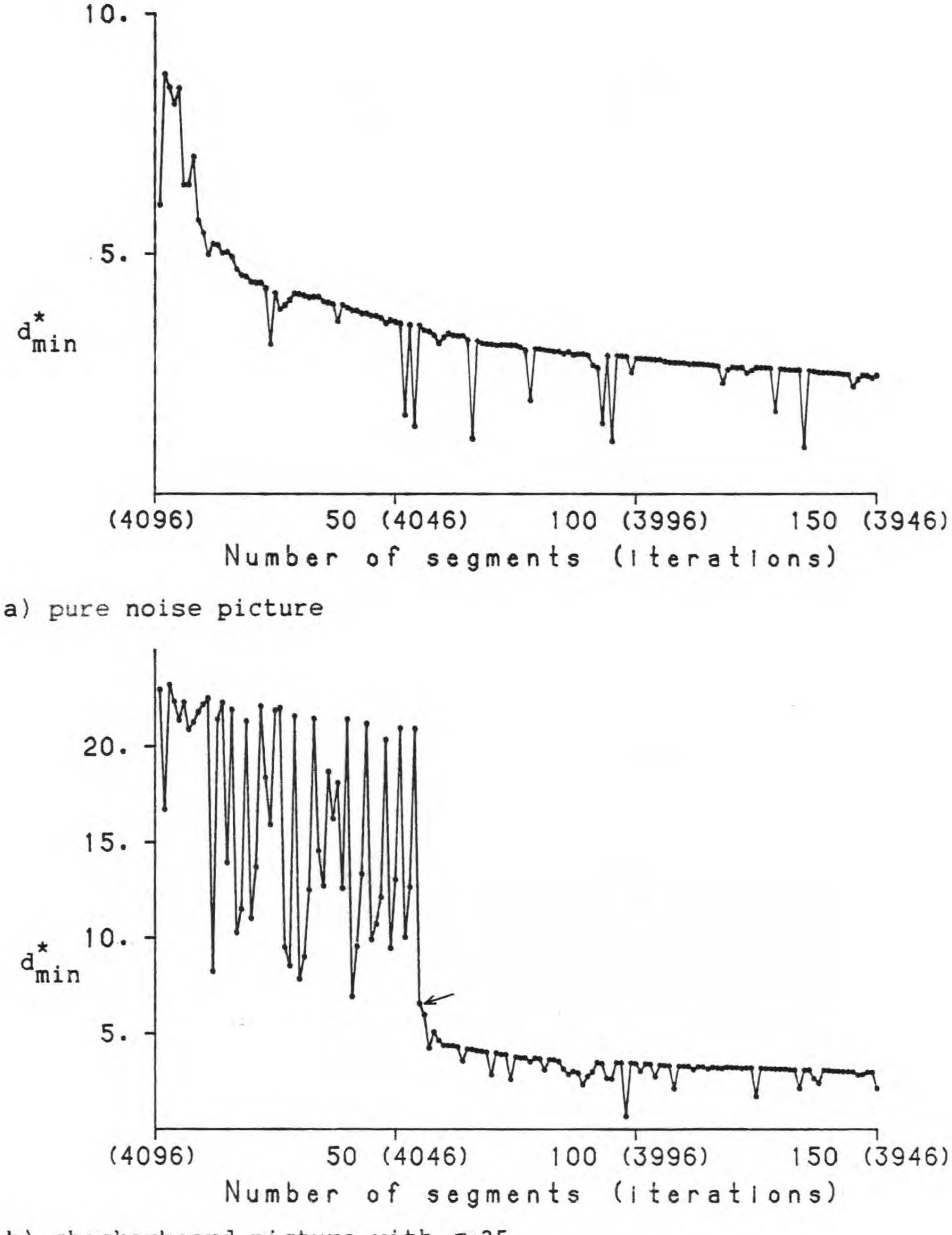


a) $\sigma = 25$

b) $\sigma = 50$

c) $\sigma = 75$

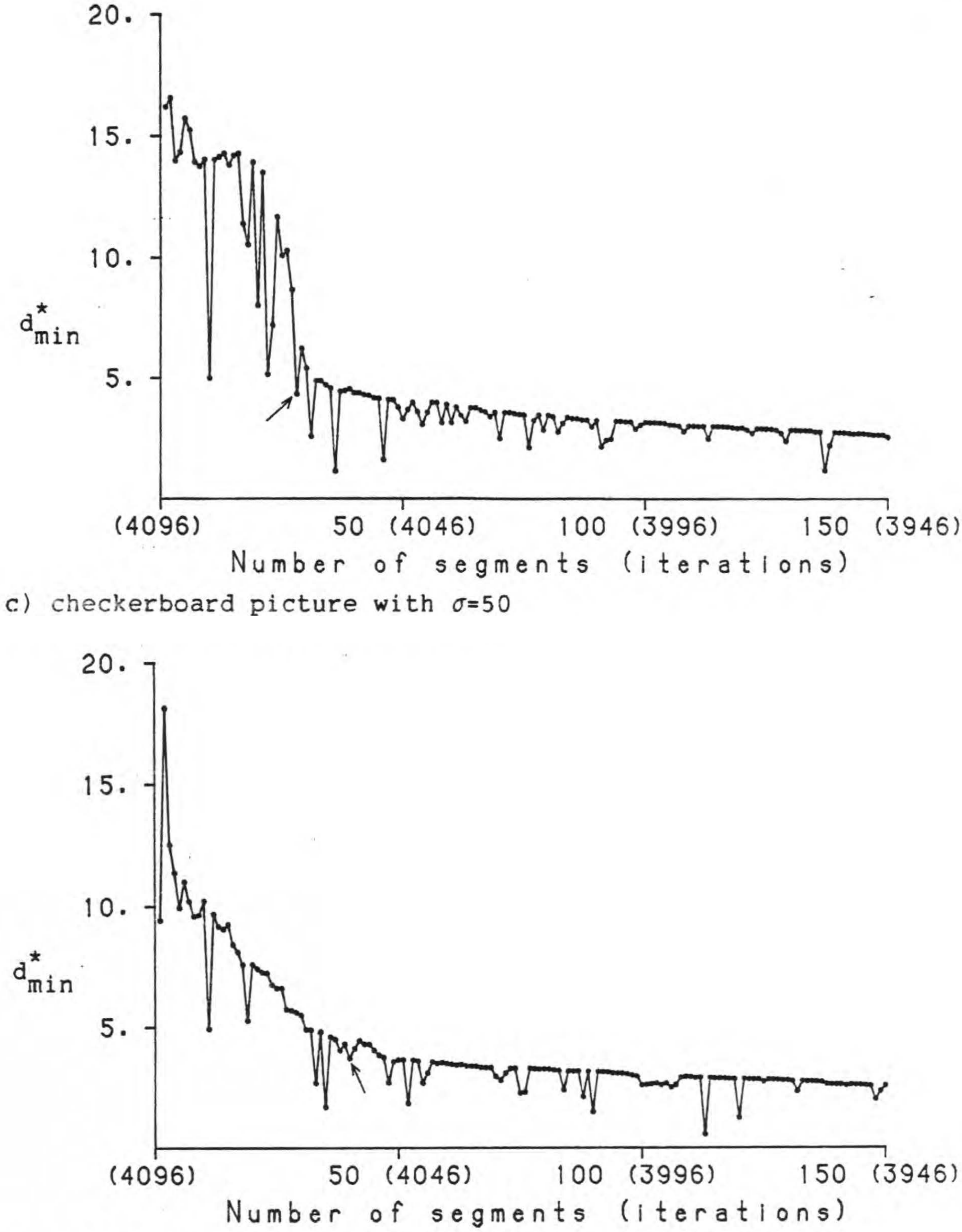
Figure 8.8: Checkerboard images with noise.



a) pure noise picture

b) checkerboard picture with σ=25

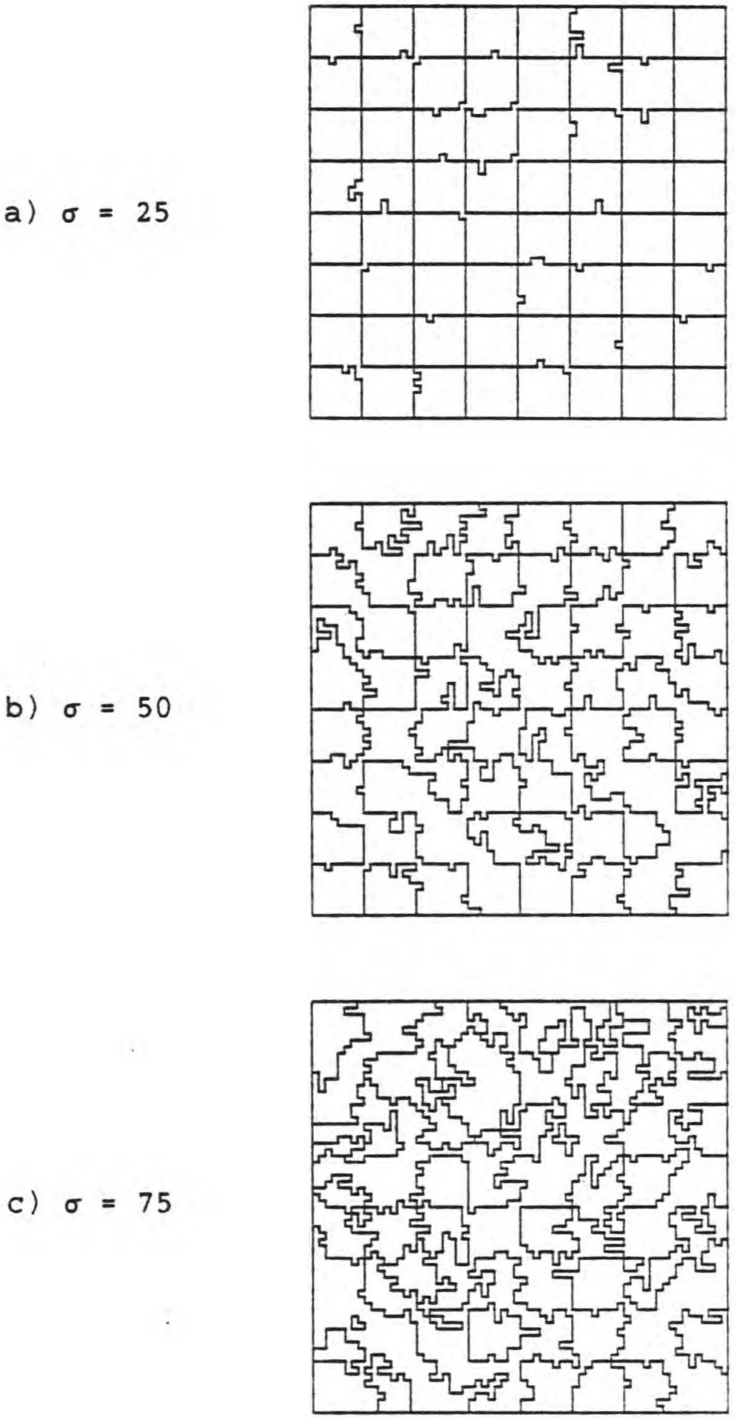
Figure 8.9: Minimum criterion value curves.



c) checkerboard picture with σ=50

d) checkerboard picture with σ=75

Figure 8.9: Continued


a) σ = 25

b) σ = 50

c) σ = 75

Figure 8.10: Segmentations of the checkerboard images.

Chapter 9

Algorithm Operation and Criterion Selection

A hierarchical segmentation algorithm based upon stepwise optimization has been described and analysed in the preceding chapters. This chapter examines the operation of the segmentation algorithm on real images, and considers the problems of stopping points and criterion selection. The relationship between the global optimization and the statistical testing approaches is first outlined, and illustrated by using a simple image segmentation example. The selection of appropriate stopping points in the segment hierarchy is examined in detail. It is shown that the image under study possesses a complex structure with a number of possible stopping points. Good results are reported for the segmentation of a Landsat satellite (MSS) Image.

The HSO algorithm presented is shown to be a valuable tool, but it does not answer the basic question: what kind of segments must be detected or what image model must be employed for a given segmentation task? Selection of different segment models corresponds to the utilization of different stepwise criteria in the algorithm. This problem is discussed and illustrated by the utilization of a number of different criteria on a remote sensing image. Good results are also reported for the segmentation of a SAR image, showing the adaptability of the algorithm to a different class of images. The combination of different criteria is shown to be particularly advantegeous. The problem of comparing different image partitions is also examined.

9.1 Analysis of the Image segmentation results

The HSO algorithm presented in section 6.1 is now applied to a remote sensing image and the results are analysed in detail. The selection of appropriate stopping points in the hierarchical segmentation is examined in particular. It is first shown that the image approximation and the statistical testing approach can be simultaneously used to analyse the algorithm results.

9.1.1 Global optimization and statistical testing

In Chapter 7, image segmentation is regarded as a image approximation problem which involves the optimization of a global criterion: the approximation error. The stepwise criterion of the segmentation algorithm is then derived from the global criterion and defined as the increase of the approximation error. For the case of image approximation by constant value regions, the stepwise criterion is

$$C_{i,j} \;=\; \frac{N_i \cdot N_j}{N_i + N_j} \;\; (\, \mu_i - \mu_j \,)^2 \tag{9.1}$$

where N_i is the size of segment S_i and μ_i is the corresponding mean value.

In Chapter 8, on the other hand, a statistical testing approach is employed for segment merging. An image is regarded as composed of constant value regions corrupted by Gaussian white noise with known variance σ^2. The stepwise optimization is associated with the minimization of the probability of error, and the derived stepwise criterion is:

$$d^*_{i,j} = \sqrt{\frac{N_i\, N_j}{N_i + N_j}} \;\; \frac{|\mu_i - \mu_j|}{\sigma} \tag{9.2}$$

It can be noted that $d^*_{i,j} = \sqrt{C_{i,j}}/\sigma$, and therefore the minimization of either of these criteria will produce the same results. Therefore, both approches can be simultaneously used to analyse the results of the HSO segmentation algorithm.

It can be noted that the best estimate of an image partition, which is defined as the maximum of a likelihood function, corresponds to the partition that minimizes the approximation error (see Appendix B).

9.1.2 Analysis of a simple example

Figure 9.1 shows a 32×32 Landsat satellite image of an agricultural area near Melfort in Saskatchewan. This is the 0.8-1.1 μm band of a Multi-Spectral Scanner image taken by the Landsat-I satellite in August 1972, frame E-1031-17265. The image is initially divided into 1024 regions of one pixel each, and is segmented by the HSO algorithm using $C_{i,j}$, the increase of the constant approximation error, as stepwise criterion.

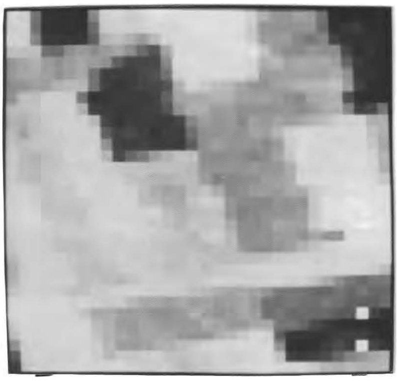


Figure 9.1: A Landsat satellite image (32×32 pixels, 0.8-1.1 μm band).

Using a image approximation approach, an image partition can be characterized by the number of segments and the approximation error. After n steps, the image partition contains $1024 - n$ segments, and the sum of the approximation errors is

$$SSE = \sum_{k=1}^{n} C_{min,k} \tag{9.3}$$

where $C_{min,k}$ is the minimum criterion at step k. The standard deviation of the approximation error $(=\sqrt{SSE/1024}\ )$ is shown in Figure 9.2. The approximation error increases as the number of segments is reduced by merging. Each step tries to minimize this increase, yielding small increases for the first steps. Hence, the approximation error is null for the first 326 steps, where adjacent equal value pixels are merged. However, the increases become more important at the latter steps where large different segments are merged. In Figure 9.2, the change in the slope is gradual. Going from right to left, the magnitude of the curve slope grows slowly up to the point marked where the increase becomes more pronounced. The segmentation algorithm must, therefore, be stopped before or around this point in order to obtain an acceptable approximation error for the image partition.

A statistical approach can also be employed to analyse these results. It has been shown that the stepwise minimization of $d^*_{i,j}$ or $C_{i,j}$ reduces the probability of error (see section 8.4) , and that higher $d^*_{min,k}$ values correspond to higher probabilities of merging dissimilar segments (see section 8.8). The examination of the minimum criterion values, $d^*_{min,k}$ or $C_{min,k}$, can then indicate when dissimilar segments begin to be merged. For instance, in the checkerboard examples of section 8.10, the merging of distinct regions is

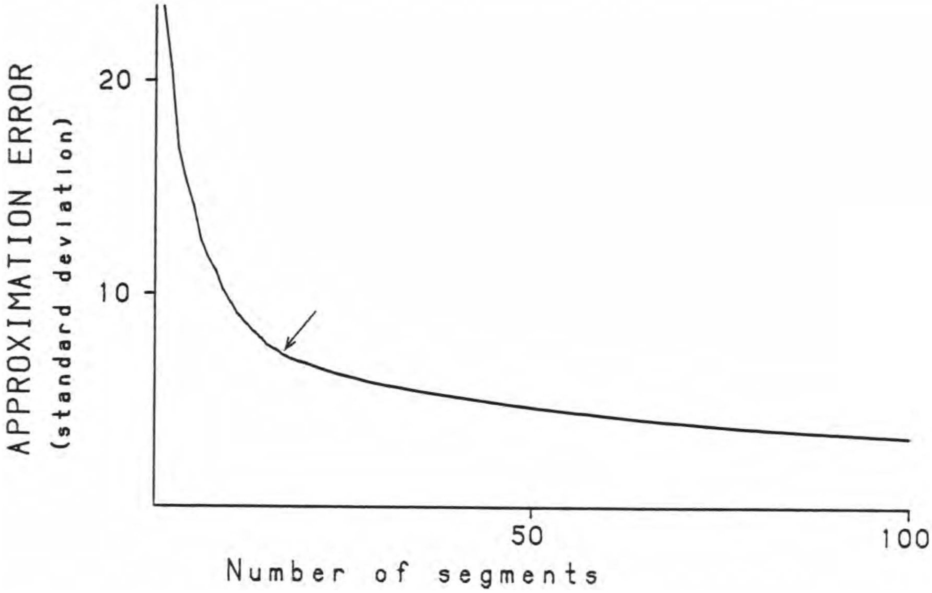


Figure 9.2: Approximation error of image segmentations.

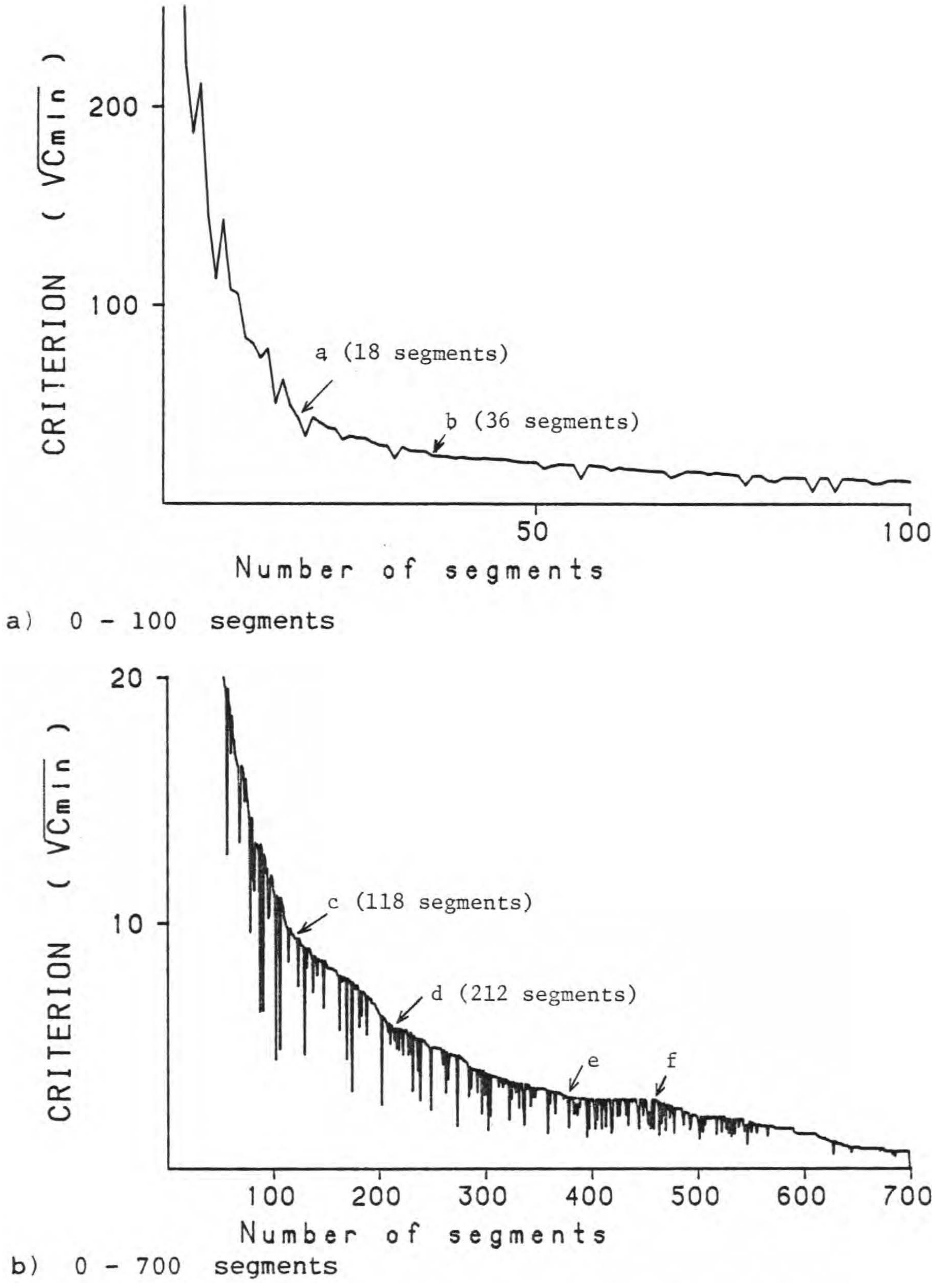
associated with a jump in the $d^*_{min,k}$ values or a change in the slope of the curve.

In Figure 9.3, the $C_{min,k}$ values for the remote sensing image are drawn as a function of the number of segments contained in the image partition at step k. The presence of negative impulses in the curve complicates the analysis, and as noted in section 8.9, the occurrence of these lower points is rather random and not really meaningful. Instead, an upper bound curve, $UB(C_{min,k})$, defined as the maximum $C_{min,k}$ over the preceding steps, is introduced:

$$UB(C_{min,k}) = \operatorname*{Max}_{i=1...k} \{\ C_{min,i}\ \} \tag{9.4}$$

This upper bound curve, presented in Figure 9.4, is more smooth and facilitates the analysis.

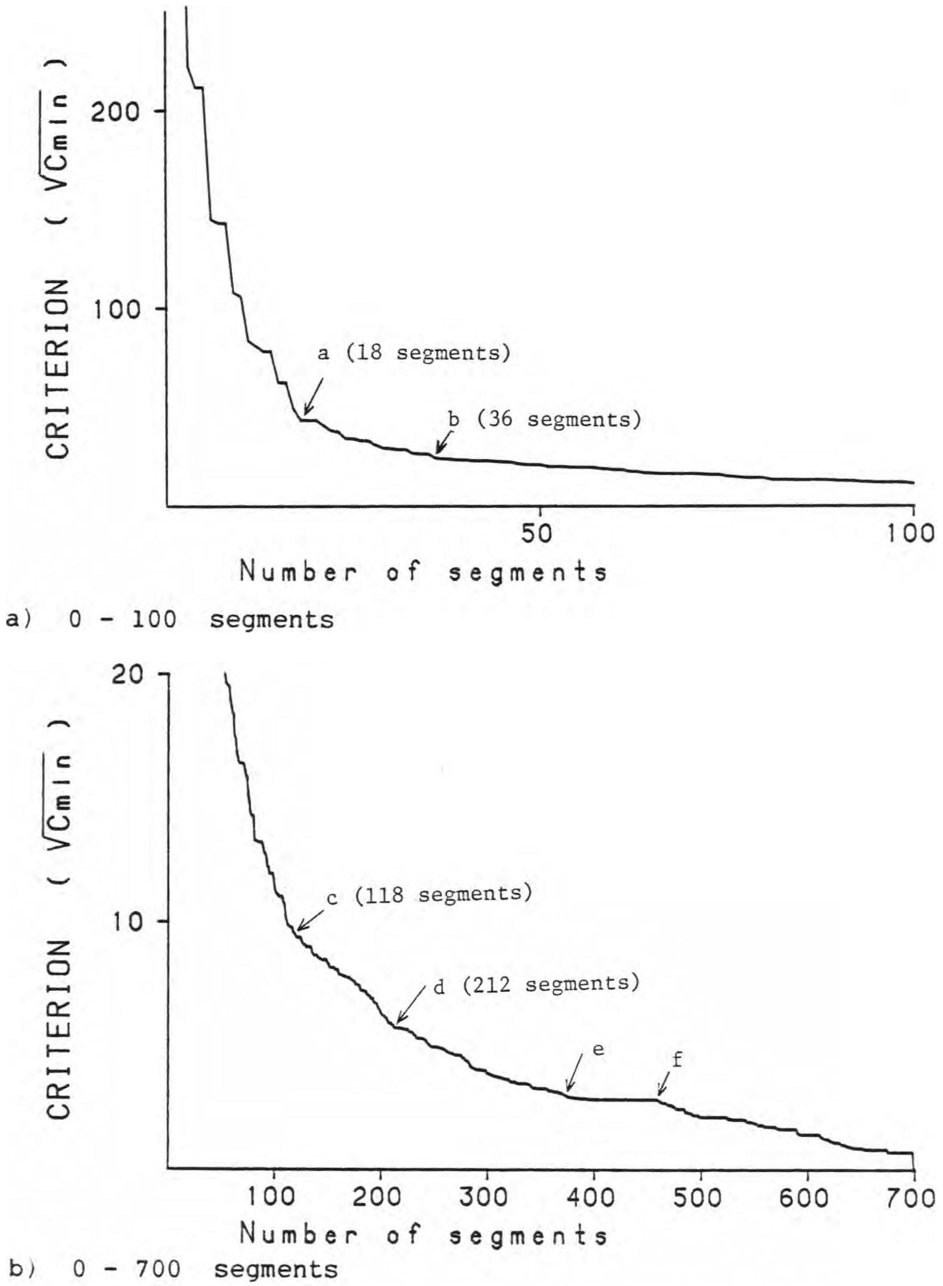
The C_{min} curve of Figure 9.4-a can be divided into two distinct regions: a region with low C_{min} values and the other with rapidly increasing C_{min} values. The point "a" can be selected to delimit these two regions $C_{min} = (43.4)^2$. Stopping the merging at this point produces a partition with 18 segments. All previous mergers have yielded C_{min} values lower than $(43.4)^2$, and small increases from step to step. Whereas, for the mergers following the point "a" the C_{min} values increase rapidly, suggesting that really dissimilar segments are then merged. The Figure 9.5-a shows the corresponding image partition. The partition divides the image into what seems to be its most basic parts.



a) 0 - 100 segments

b) 0 - 700 segments

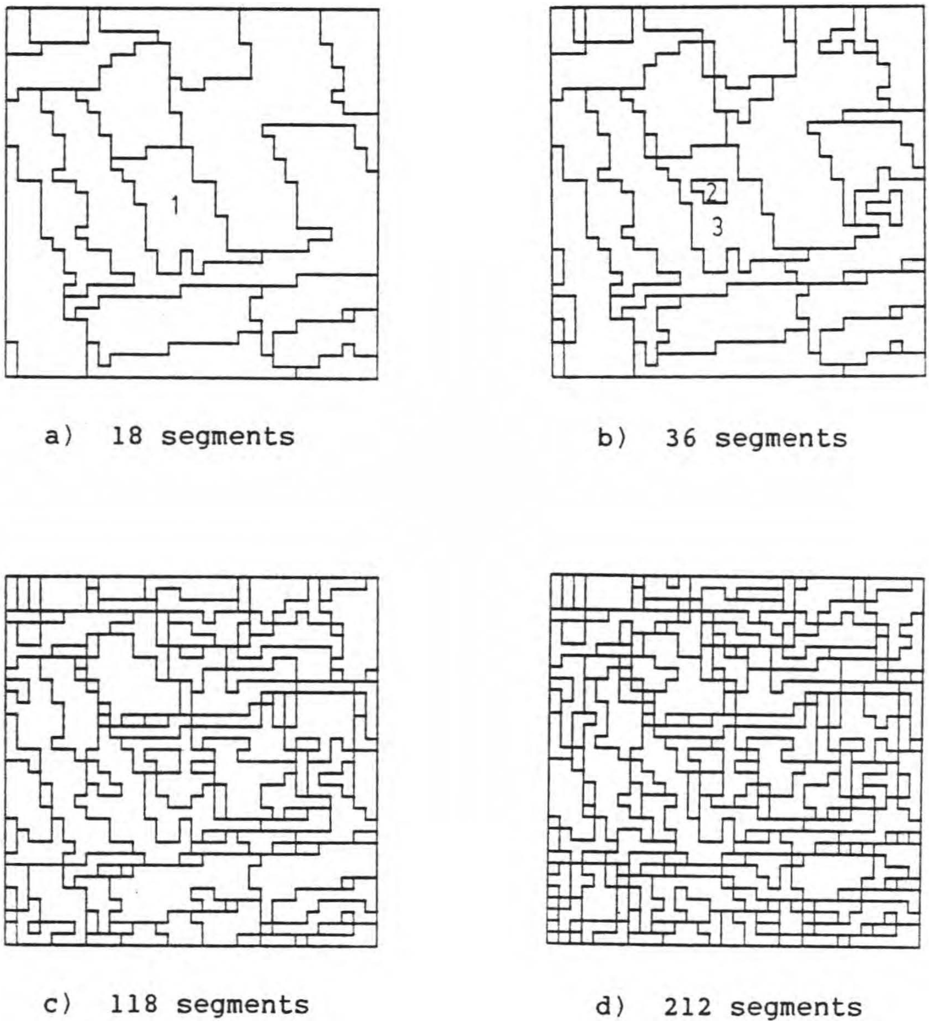
Figure 9.3: Minimum step-wise criterion curve for different axis scales.



a) 0 - 100 segments

b) 0 - 700 segments

Figure 9.4: Upper bound curve of the minimum stepwise criterion for different axis scales.



a) 18 segments

b) 36 segments

c) 118 segments

d) 212 segments

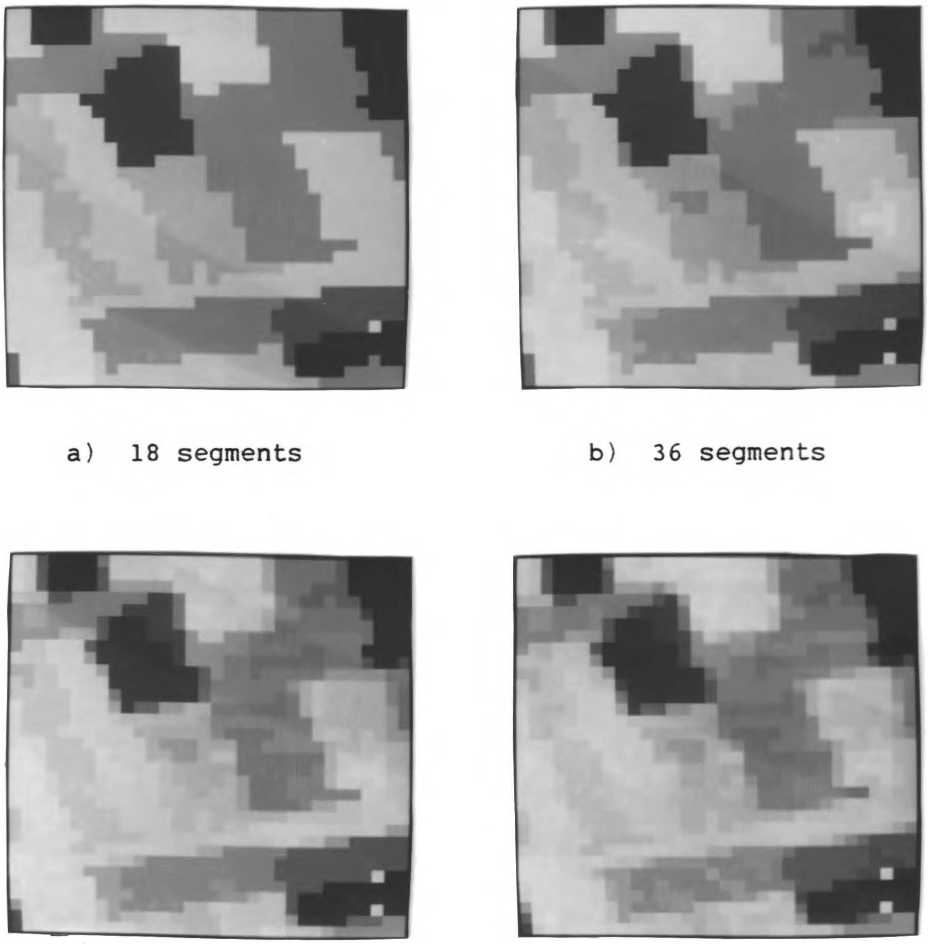
Figure 9.5: Segmentations of the Landsat image (32×32 pixels).

The remote sensing image possesses a rather complex structure with regions having varying sizes and mean value differences. The basic regions of Figure 9.5-a can moreover be considered as composed of finer elements. These finer image components are obtained by reducing the number of segment mergers. The selection of corresponding stopping points can be done randomly, but more appropriate points can be obtained by examining the C_{min} curve. For example, the point "b" of Figure 9.4-a can be employed. This point corresponds to a change in the slope of the curve. Figure 9.5-b shows the image partition with 36 segments that is produced by this stopping point. This partition corresponds to dividing the regions of Figure 9.5-a into finer elements. For example, segment 1 of Figure 9.5-a is divided into segments 2 and 3 of Figure 9.5-b. There is a small gray level difference between region 2 and 3, thus, the region 2 represents a finer image component than region 1.

Changing the scale of the axis in Figure 9.4-b, other special points can be selected, such as the points "c" and "d" which again correspond to changes in the slope of the C_{min} curve. Stopping the segment merging at those points produce partitions with 118 and 212 segments. In these partitions, many small details of the image are preserved. For example, there are a number of segments of one or two pixels. However, 212 segments seems an excessive number to represent this small image (32×32 pixels). Note that, in the first steps, there are many consecutive $C_{min,k}$ having the same value which produce special points like "e" and "f". This results from the original gray level quantization as, in the initial steps, the $C_{min,k}$ values correspond to differences of only one gray level. In the next section, the meaning and importance of the different slope change points are examined.

9.1.3 Hierarchical image structure

The presence of a hierarchical structure in the image can explain the appearance of many slope change points in the C_{min} curve, each level of the hierarchy being associated with a particular slope value. A hierarchical structure means that the image contains components at different resolution levels. The hierarchical segmentation algorithm takes account of this image component hierarchy. Hence, in the remote sensing image, the partition with 18 segments can be regarded as the highest level where only the most important components of the image are preserved. This is illustrated by Figure 9.6-a where each image segment has been replaced by its mean value. These segments encode the gross information of the image. They indicate the most prominent areas. The other partitions of Figure 9.5 correspond to splitting these segments into sub-units. The corresponding approximation



a) 18 segments

b) 36 segments

c) 118 segments

d) 212 segments

Figure 9.6: Approximation of the Landsat image (32×32 pixels).

images are shown in Figure 9.6 and indicate that finer image components are retained. These image partitions can be regarded as different levels of the hierarchy which correspond to different image component resolutions.

The problem of finding a well defined stopping point in the segment hierarchy is now examined. This is related to the existence of distinct layers in the segment hierarchy. The layers can be produced by large gaps between segment levels. The segment merging can then be stopped between two layers. This is illustrated by Figure 9.7 where the hierarchy "a" contains no distinct layers, while two layers can be recognized in "b". The layers can be distinguished only if they possess different characteristics. For instance, in the checkerboard examples of section 8.10, the position of a node on

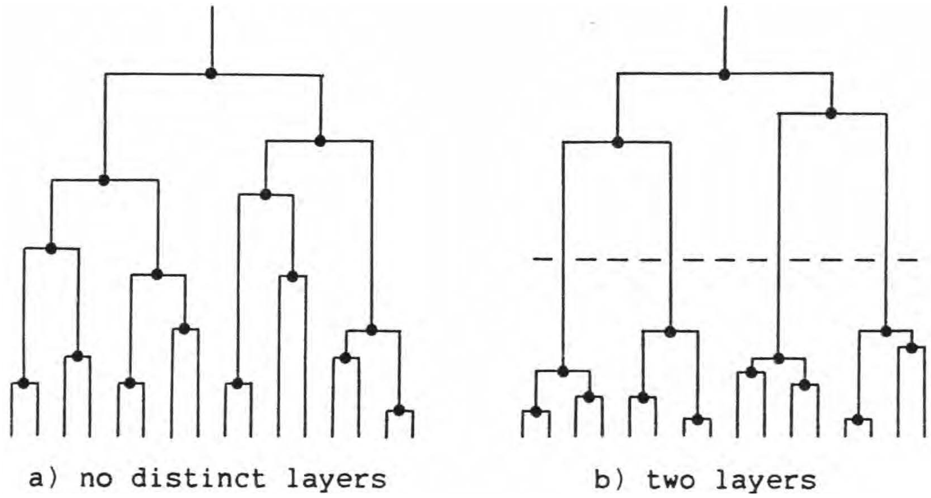
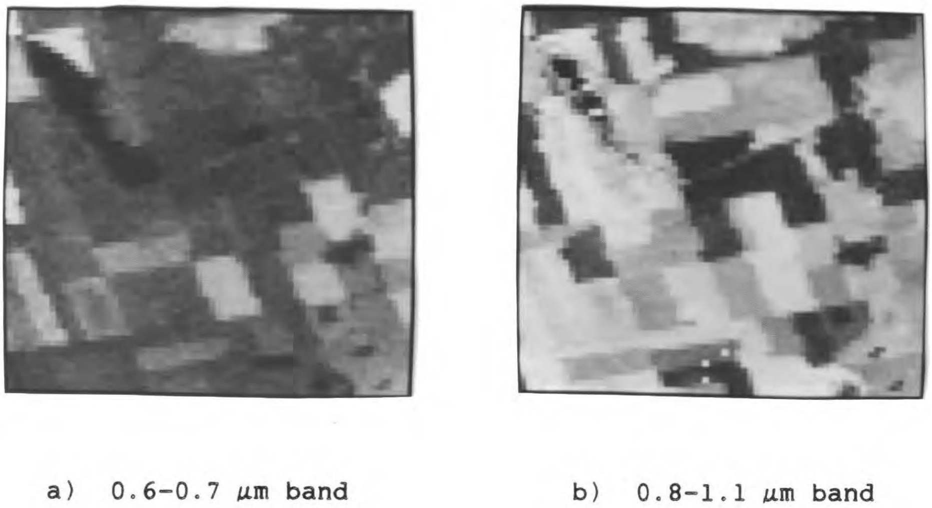


Figure 9.7: Hierarchy with no distinct layers (a) and with two layers (b).

the vertical axis can correspond to its $d^*_{min,k}$ (or $C_{min,k}$) value. The first mergers, which are associated with the elimination of noise, produce small $d^*_{min,k}$ values. While, the last mergers involve the fusions of the checker areas which give large $d^*_{min,k}$ values. This results in a large gap between these two segment levels. In real applications, however, it is not clear that distinct segment layers exist and can be distinguished.

Considering the minimum criterion curve, a layer corresponds to an interval of the curve, and can be characterized by the slope value. Thus different layers can be distinguished only if they possess different slope values. In the remote sensing example, many curve intervals with different slope values have been identified, and the boundary points of these intervals have been used to stop the algorithm. However, some slope changes (points "a" and "b") are more important and clear than others, reflecting more distinct layer transitions. For the first steps of the algorithm, when there is still a large number of segments in the image partition, the slope changes tend to be smaller and more difficult to recognize, the progession of the $C_{min,k}$ values being more regular and smooth.

One important consequence of this hierarchical structure for the stepwise optimization algorithm is that the user must specify at which level to stop the segment merging. The segment level can be defined by the approximation error, by the C_{min} value, or by the number of segments in the partition, each of these parameters being interrelated. The examination of the approximation error and C_{min} curves can then complement the context knowledge in order to select a stopping point.



a) 0.6-0.7 μm band

b) 0.8-1.1 μm band

Figure 9.8: A Landsat image (64×64 pixels).

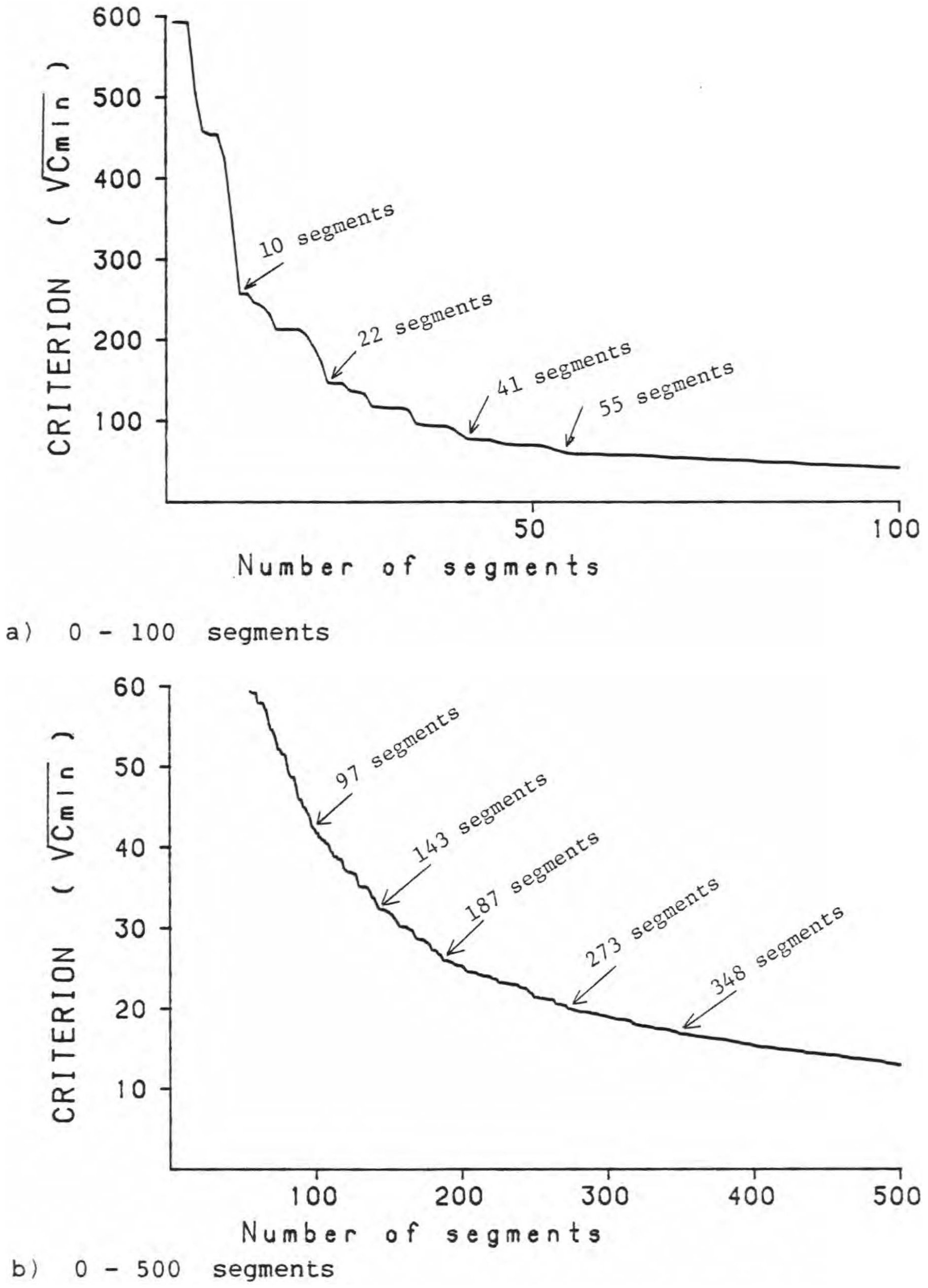
9.1.4 Segmentation of a remote sensing image

The HSO algorithm is now applied to a more complex image: a 64×64 pixels, two channel image. This is a MSS Landsat satellite image of an agricultural area near Melfort in Saskatchewan, imaged in August 1972 (frame E-1031-17265). The two channels of the image are presented in Figure 9.8, and correspond to the 0.6-0.7 μm band and the 0.8-1.1 μm band. The image is initially partitioned into 4096 segments where each segment corresponds to only one pixel. The stepwise criterion, $C_{i,j}$, corresponds as before to the increase of the constant approximation error,

$$C_{i,j} \;=\; \frac{N_i \cdot N_j}{N_i + N_j} \sum_{\lambda=1,2} w_\lambda \left(\mu_{\lambda,i} - \mu_{\lambda,j} \right)^2 \tag{9.5}$$

where $\mu_{\lambda,i}$ is the mean value for channel λ of segment S_i, N_i is the size of S_i, and w_λ is a weighting factor used to combine the different channels (see section 7.4). Here, w_1 and w_2 are equated to one.

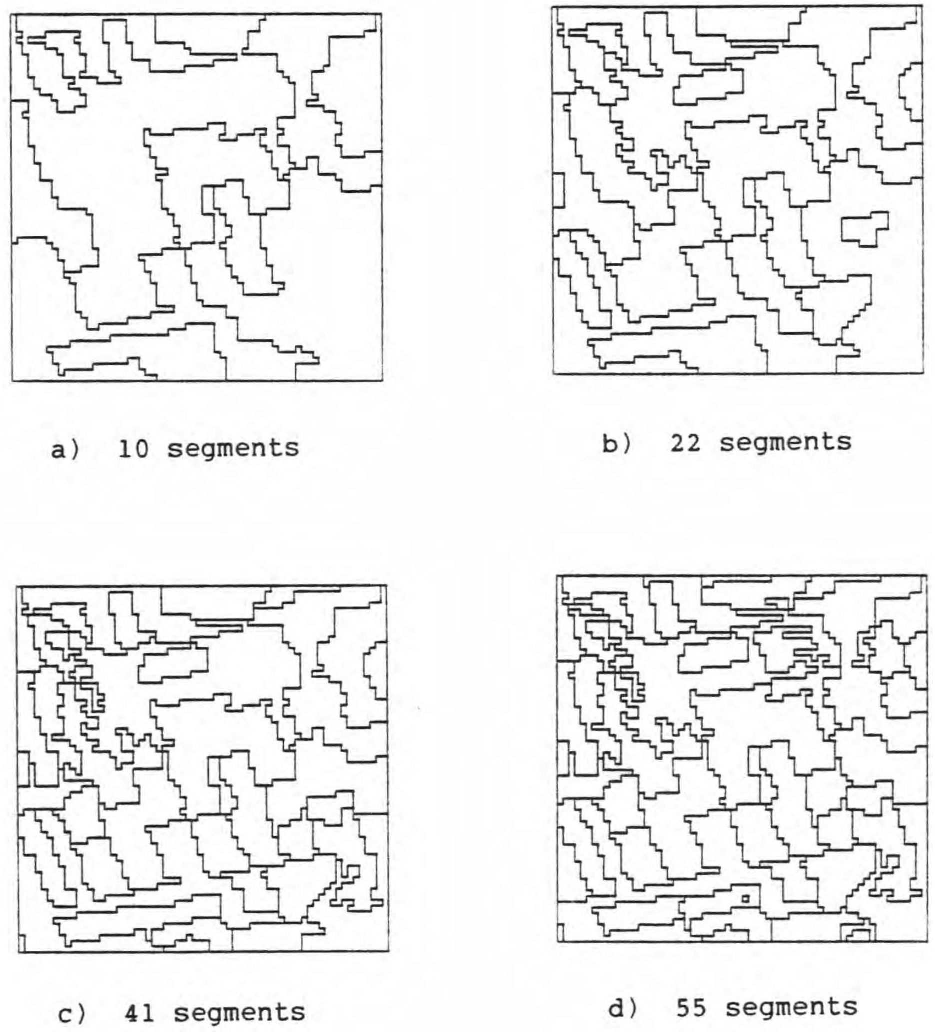
Figure 9.9 shows the criterion upper bound curve. As in the preceding example, a number of slope change points are selected. The changes in the slope are more gradual here than in the preceding 32×32 image. The image partitions associated with each of these points are shown in Figure 9.10. The user must choose between these different stopping points. The image approximation error is presented in Figure 9.11.



a) 0 - 100 segments

b) 0 - 500 segments

Figure 9.9: Upper bound curve of the minimum stepwise criterion.



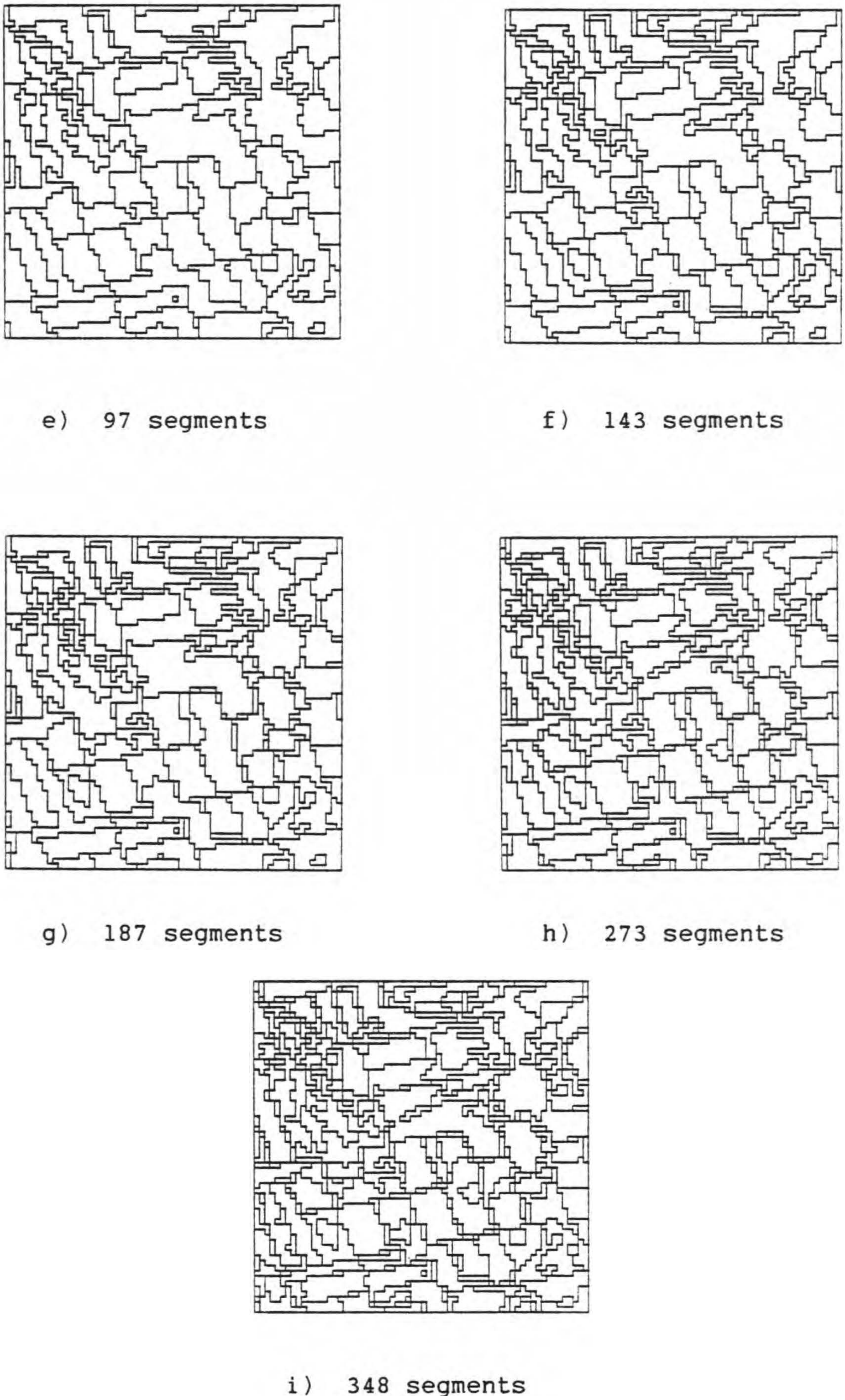
a) 10 segments

b) 22 segments

c) 41 segments

d) 55 segments

Figure 9.10: Segmentations of the Landsat image (64×64 pixels).



e) 97 segments

f) 143 segments

g) 187 segments

h) 273 segments

i) 348 segments

Figure 9.10: Continued

9.1.5 Computing time

The segmentation algorithm is coded in a Fortran program and executed on a VAX-11/750 computer. The preceding segmentation of the 64×64 pixel, 2 channel image takes 15.7 sec. of CPU time for the initialization step and approximately 0.012 sec. per segment merging iteration (see section 6.1). For the partition of the image into 97 segments, there are 3999 iterations requiring 50.6 sec. of CPU time, giving a total computing time of 66.3 sec. Each iteration reduces the number of segments by one. The number of iterations is thus equal to the difference between the initial and final number of segments. It can, therefore, be advantegeous to use a simple segmentation process to perform a first reduction of the number of segments in order to reduce the total computing time.

Parallel computation can also be employed to reduce the CPU time [90], [79]. For example, each iteration can perform, in parallel, m merges instead of only one, these m mergers corresponding to the m lowest criterion values. The image could be divided into m distinct blocks in another possible approach, and the first merging iterations could be performed independently on each block by different processors, while the latter iterations use the entire image.

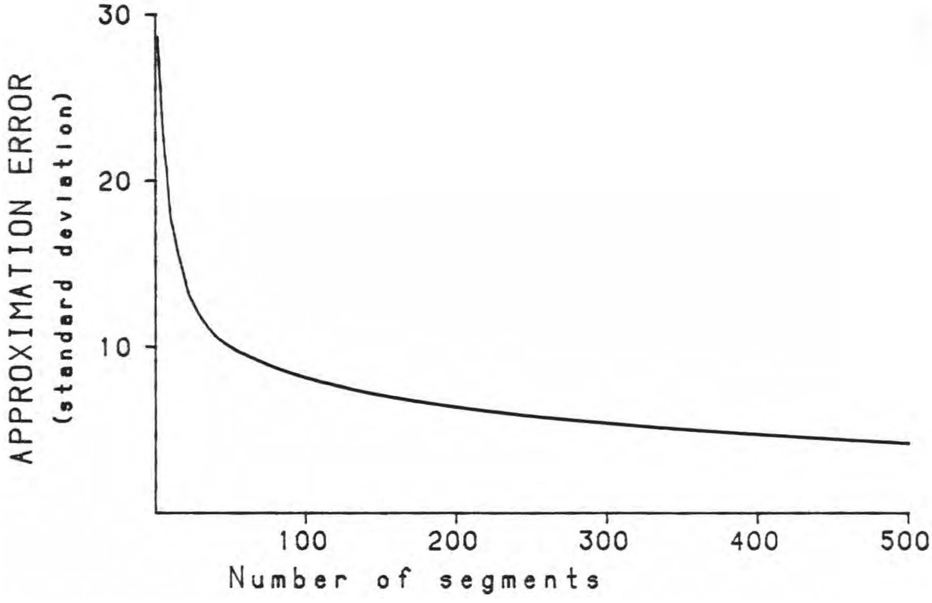


Figure 9.11: Approximation error of image segmentations.

9.2 Criterion selection

In the preceding section, it has been assumed that the constant value region is a good model for the representation of an image. However, it is not clear that it is the best one. The selection of an appropriate model for a image segmentation task is a major issue. In the present stepwise optimization approach, this is related to the selection of the appropriate stepwise criterion. In this section, new models are proposed for the Landsat image. The stepwise criteria are then derived and the results analysed.

The Landsat image employed is presented in Figure 9.12, with an enlargement of the two sub-areas used in the following discussion. The image contains 64×64 pixels, while the sizes of areas A and B are, respectively, 24×24 and 22×22 pixels. The results of the constant approximation criterion are shown in Figure 9.13 for comparison purposes. The image is divided into 100 segments, and each segment is replaced by its mean value to produce an approximation image.

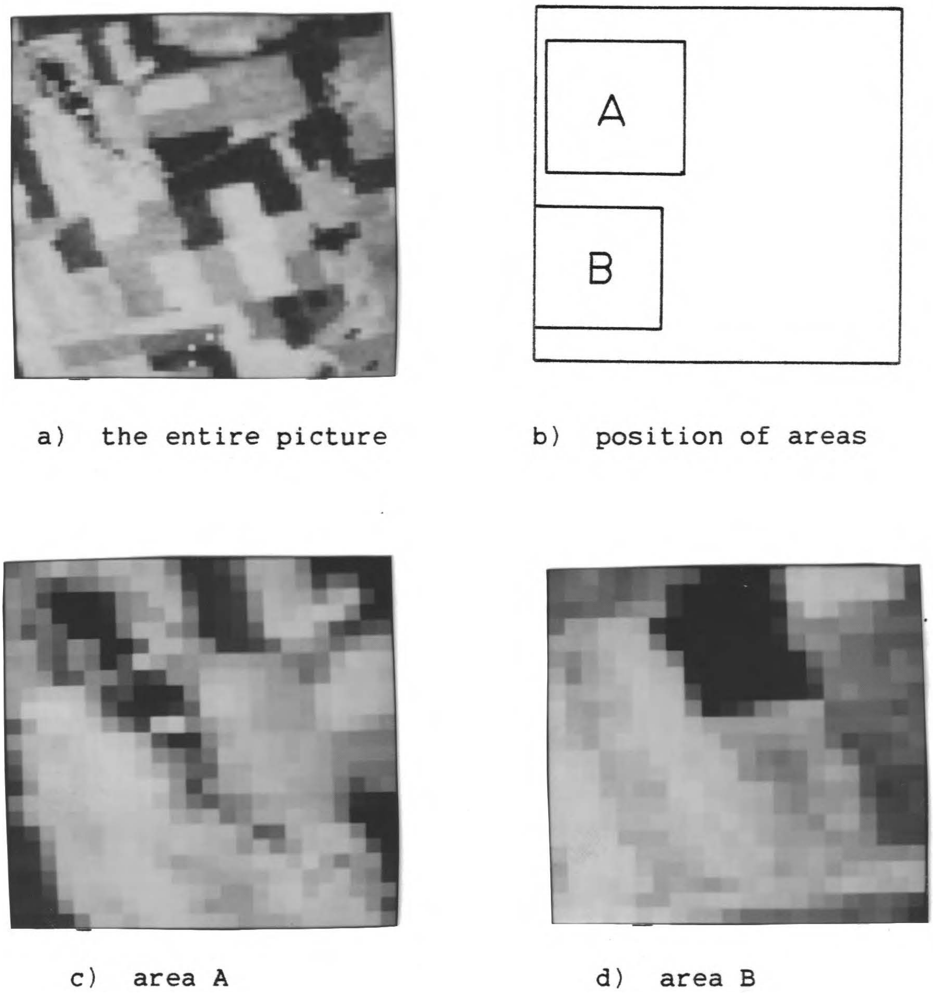
9.2.1 Planar approximation

The criterion used in the preceding sections considers the regions as constant value areas. While this model seems generally appropriate for remote sensing images, there are some cases where it is clearly deficient. For example, Figure 9.14 shows a 1-D example where a constant value region is appropriate for regions 1 and 3, while it is inappropriate for region 2. The planar approximation model of section 7.5 can be employed:

$$r_i(x, y) = a^i_{0,0} + a^i_{1,0}(x) + a^i_{0,1}(y) \tag{9.6}$$

with the corresponding stepwise criterion given by equation (7.20). This criterion is used to segment the Landsat image of Figure 9.12, and the results are shown in Figure 9.15. It will be shown that the planar approximation, unfortunately, does not produce a better image partition than the constant value region model (Figure 9.13). The characteristics of the planar approximation is now discussed in order to explain its deficiency for the Landsat image.

The utilization of a higher degree polynomial means that more spurious results can arise as illustrated in Figure 9.16. A signal composed of two constant value regions (a) is corrupted by noise (b). In this case, the true regions are correctly detected by a constant approximation (c) while a first order approximation is misleading (d). A higher degree polynomial, having more degrees of freedom, can match more closely the noise deformation, yielding spurious results. A similar situation is observed by comparing the



a) the entire picture

b) position of areas

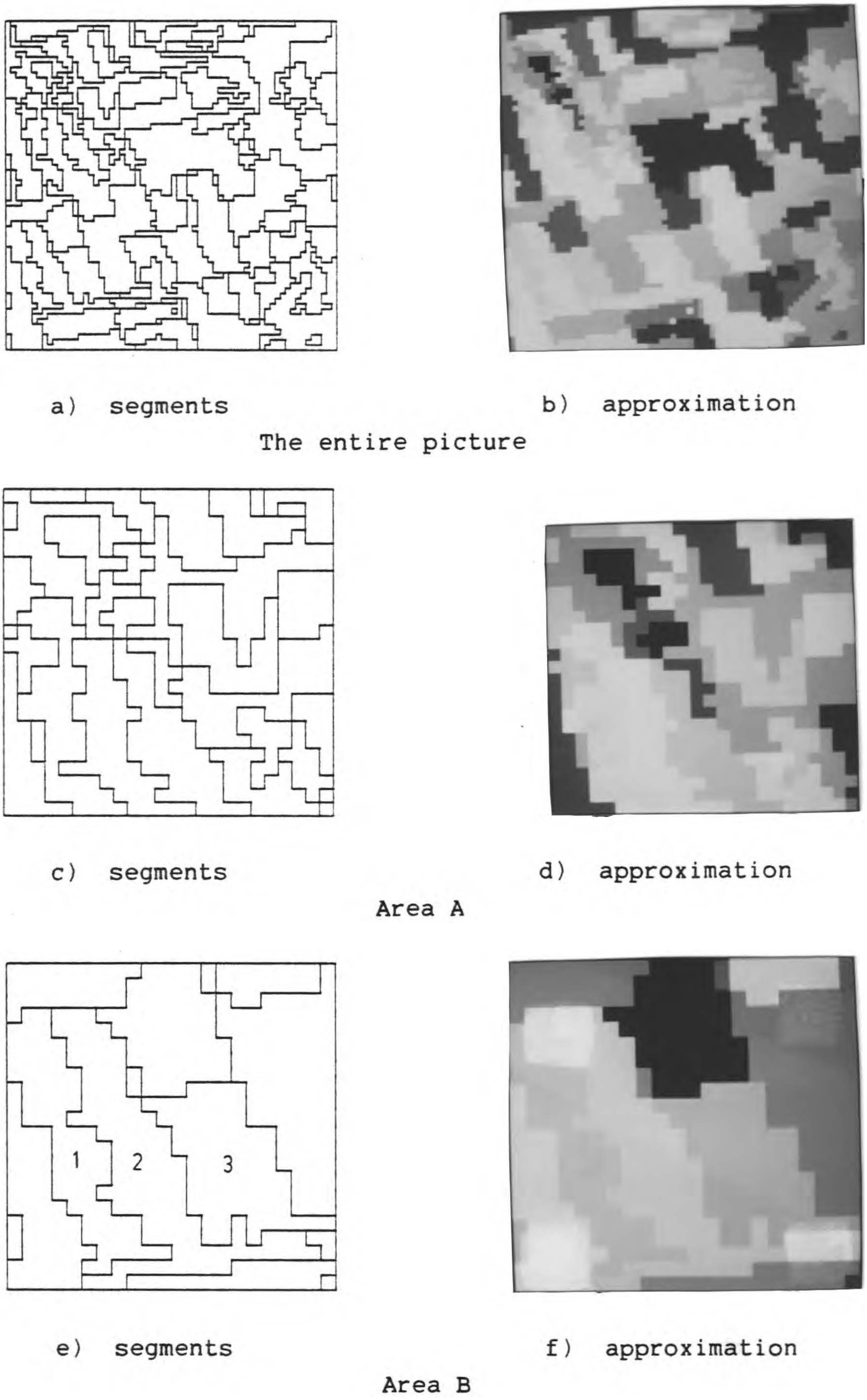
c) area A

d) area B

Figure 9.12: Landsat image and two sub-areas.

regions 1, 2 and 3 of Figure 9.13-e, and the regions 5, 6 and 7 of Figure 9.15-e. The region 2 in constant approximation is divided between regions 5 and 6 of planar approximation. The inclined plane of region 6 represents both the light values of region 2 and the darker values of region 3. The pixels of region 7 do not fit this plane and therefore form a distinct region. The constant approximation results are, therefore, more appropriate for these regions.

The difference between planar approximation and constant approximation is highlighted by the treatment of stair-like regions. If such regions occur between two large regions, as in Figure 9.17-a, the stair-like region can be regarded as a transition area, and it is advantageous to represent it by an oblique line. Region 8 of Figure 9.15-e is an example where a planar



a) segments

b) approximation

The entire picture

c) segments

d) approximation

Area A

e) segments

f) approximation

Area B

Figure 9.13: Segmentation results for constant approximation.

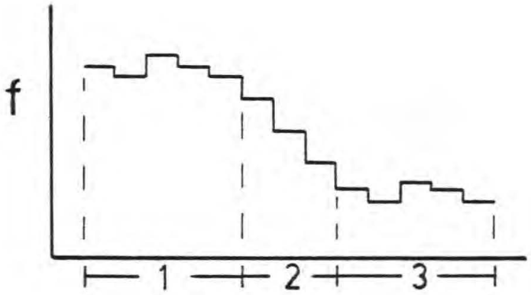


Figure 9.14: An example with constant value and inclined line regions.

approximation is needed. On the other hand, if each stair corresponds to a large constant region, as in Figure 9.17-b, it would be preferable to keep each region distinct. A first order approximation tends to merge these regions; the criterion value for two adjacent regions is three times smaller for the first order than for the zero order approximation. The region 4 of Figure 9.15-c constitutes an example of merging such adjacent large regions.

9.2.2 Local variance

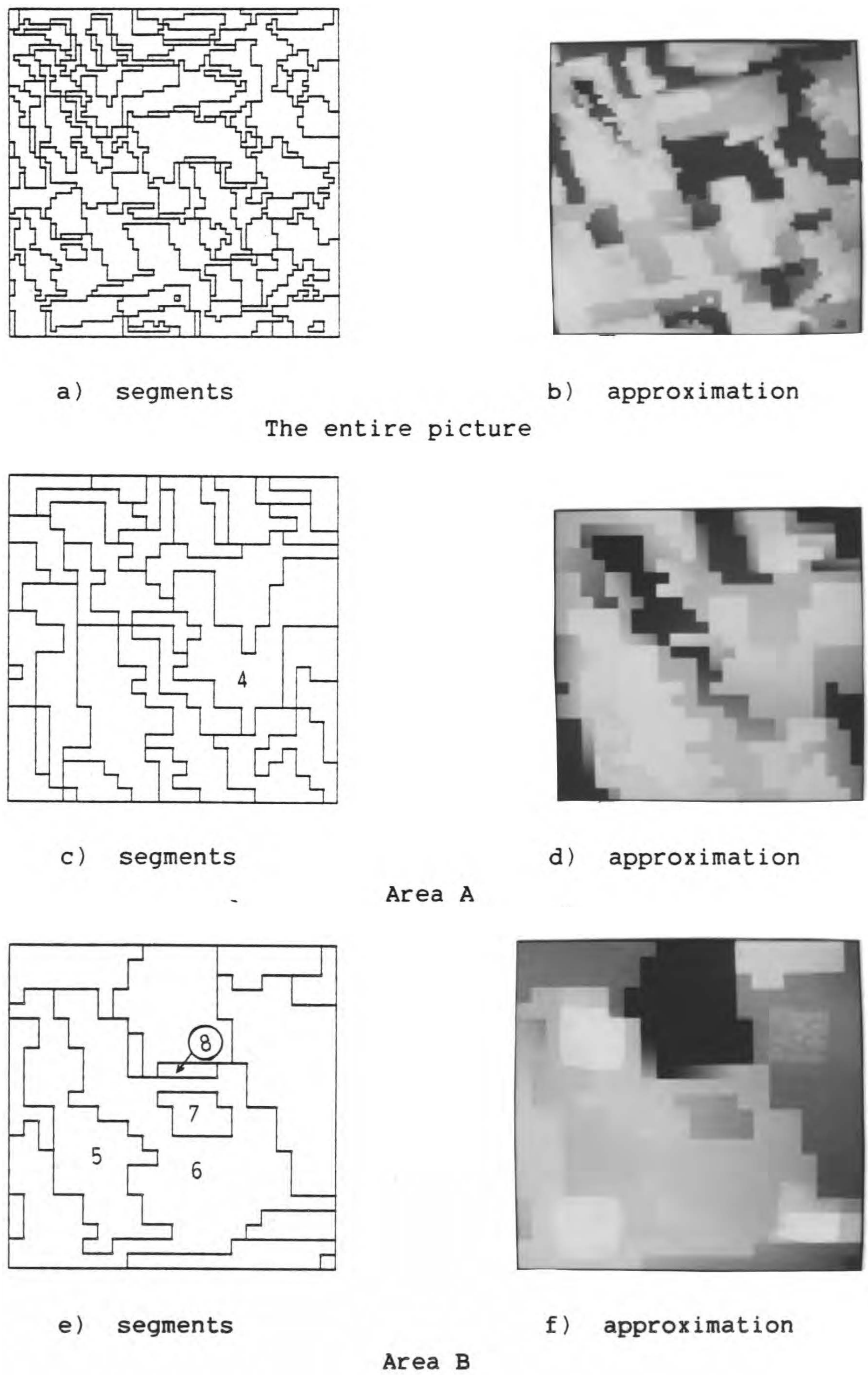
The HSO algorithms based upon constant or planar approximation try to minimize the approximation error. The evaluation of the error for a given pixel does not consider the importance of the gray level variance in the surrounding area. Hence, in Figure 9.18, both examples, a and b, have the same criterion value with respect to the regions 1 and 2. It can be advantageous to make the criterion value depend upon the segment variance, and define a new criterion such as:

$$C^{\circ}_{i,j} = C_{i,j} \Big/ \left(1 + \sigma_{i,j} \right) \tag{9.7}$$

where $\sigma^2_{i,j}$ is the mean value of the squared approximation error:

$$\sigma^2_{i,j} = \frac{\mathbb{H}(S_i) + \mathbb{H}(S_j)}{N_i + N_j}$$

Here, $\mathbb{H}(S_i)$ is as defined in section 7.1, the sum of the squared approximation error for segment S_i, and N_i is its size. For constant approximation, $\sigma^2_{i,j}$ corresponds to the combined variance of both segments. Thus $C^{\circ}_{i,j}$ is equal to $C_{i,j}$ when $\sigma_{i,j}$ is zero, and decreases for large values of the variance. The results given by this new criterion are shown in Figure 9.19 and can be compared with the corresponding Figure 9.13, for the constant approximation criterion $C_{i,j}$. For instance, the regions 9, 10 and 11 of Figure



a) segments

b) approximation

The entire picture

c) segments

d) approximation

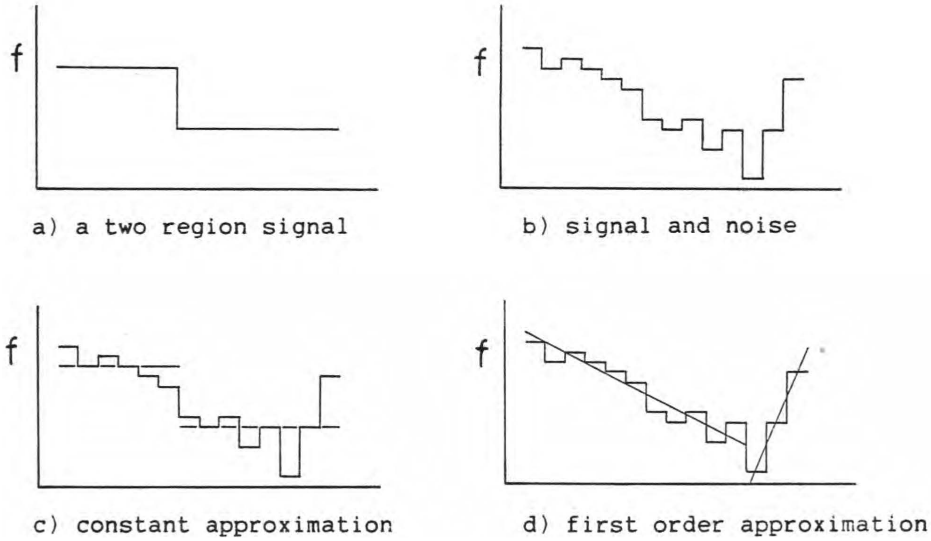
Area A

e) segments

f) approximation

Area B

Figure 9.15: Segmentation results for planar approximation.



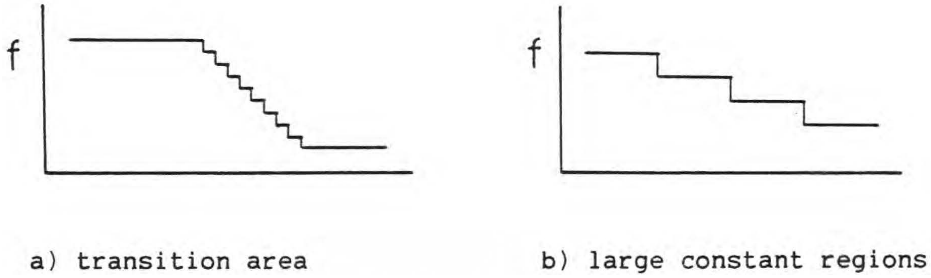
a) a two region signal

b) signal and noise

c) constant approximation

d) first order approximation

Figure 9.16: A 1-D function composed of two regions (a), with added .noise (b), and approximated by constant values (c) and inclined lines (d).



a) transition area

b) large constant regions

Figure 9.17: Examples of stair-like regions.

9.19 correspond to a zone of large gray level variation. Using $C^{\circ}_{i,j}$ produces smaller criterion values because of the large segment variances, and therefore forces more segment merging in this area. Thus, this new criterion seems preferable because it adjust itself to local image variance.

9.2.3 Criterion combination

The stepwise optimization algorithm can employ different criteria which correspond to different segment description models. The previously used criteria involve very simple models. However, more complex models can be required by segmentation tasks. Complex models can be obtained from combinations of simpler ones.

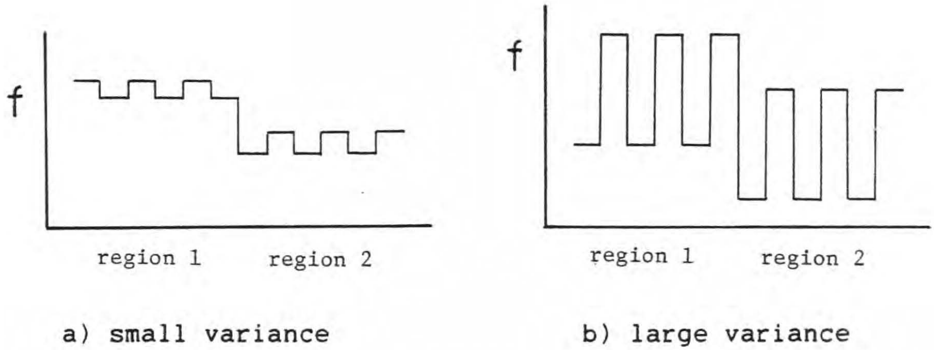


Figure 9.18: Examples of regions with the same criterion values but different gray level variances.

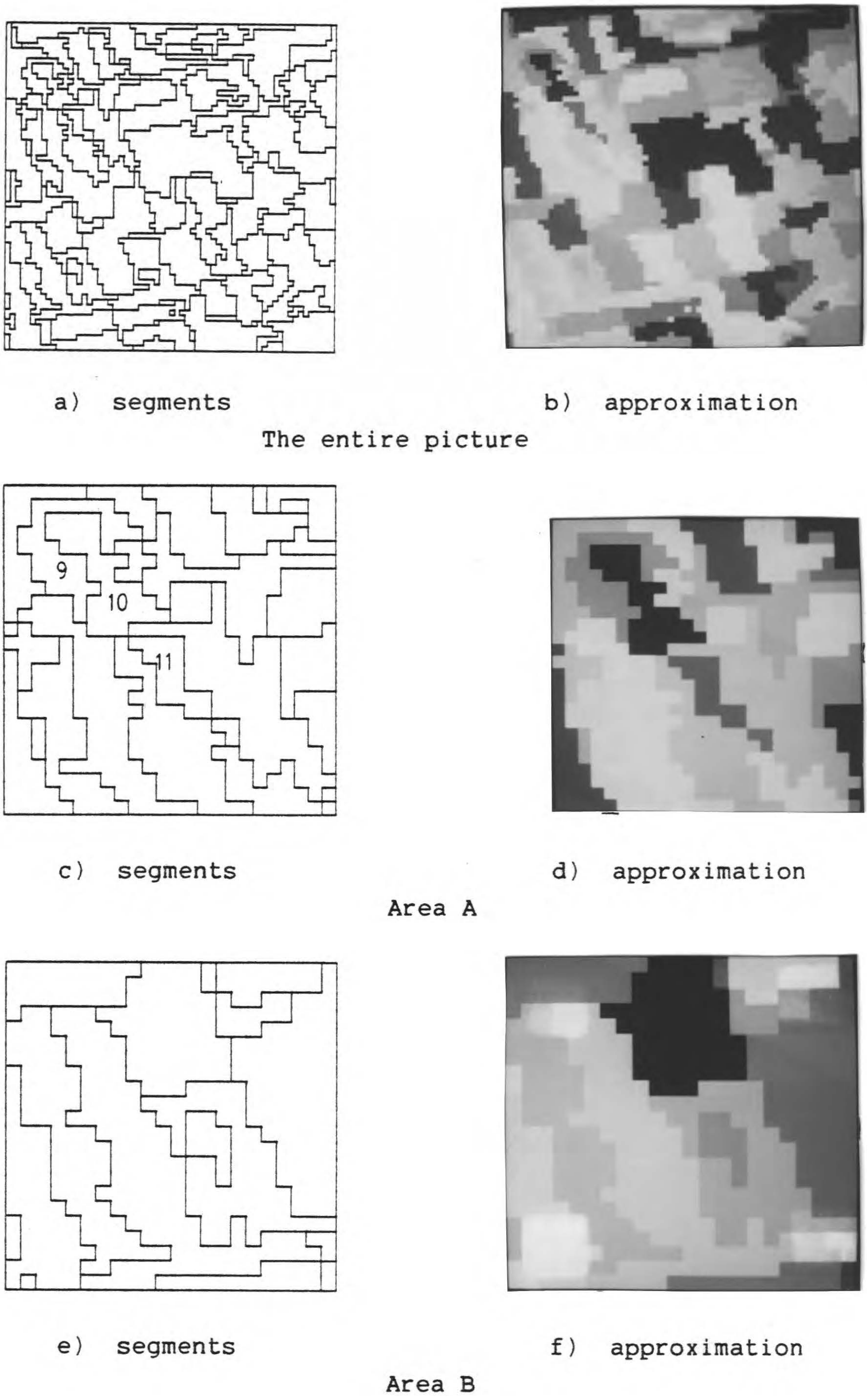
Zobrist and Thompson [101] point out that human vision employs many cues such as brightness, contour, color, texture and stereopsis to perform perceptual grouping. They stress the limitations of using only one cue at the time for computer grouping, and show the importance of studying mechanisms that combine many cues. For computer simulation of human perception, they derive from each cue a distance function that measures the similarity of two scene parts. Then, they perform a weighted sum of these distances to obtain a global perceptual distance.

Applying this approach to image segmentation, it can be noted that different image areas can require different segment models (cues) and that these models must be combined in order to obtain good overall results. Hence, the constant approximation can be appropriate for some parts of a image while the planar approximtion can be preferable for some other parts. Thus, it can be advantageous to combine the stepwise criteria associated with both models. For example, a composite criterion can be obtained as follows:

$$C_{(composite)} = C^{\circ}_{(constant)} \cdot C^{\circ}_{(planar)} \tag{9.8}$$

This corresponds to using the geometric mean of two criteria to form the composite one. $C^{\circ}_{(\cdot)}$ indicates a local variance adaptable criterion as defined in the preceding section.

In image segmentation, an ordering of segment descriptions can also be considered [68]. For example, the pixel gray level can be employed to form small homogeneous regions, then more complex descriptors, such as segment contour shape, can be considered for forming larger regions. Many segment descriptors, such as contour shape, or higher order approximation coefficients, are meaningless for small regions and only become useful at a latter stage. In the hierarchical segmentation scheme, this corresponds



a) segments

b) approximation

The entire picture

c) segments

d) approximation

Area A

e) segments

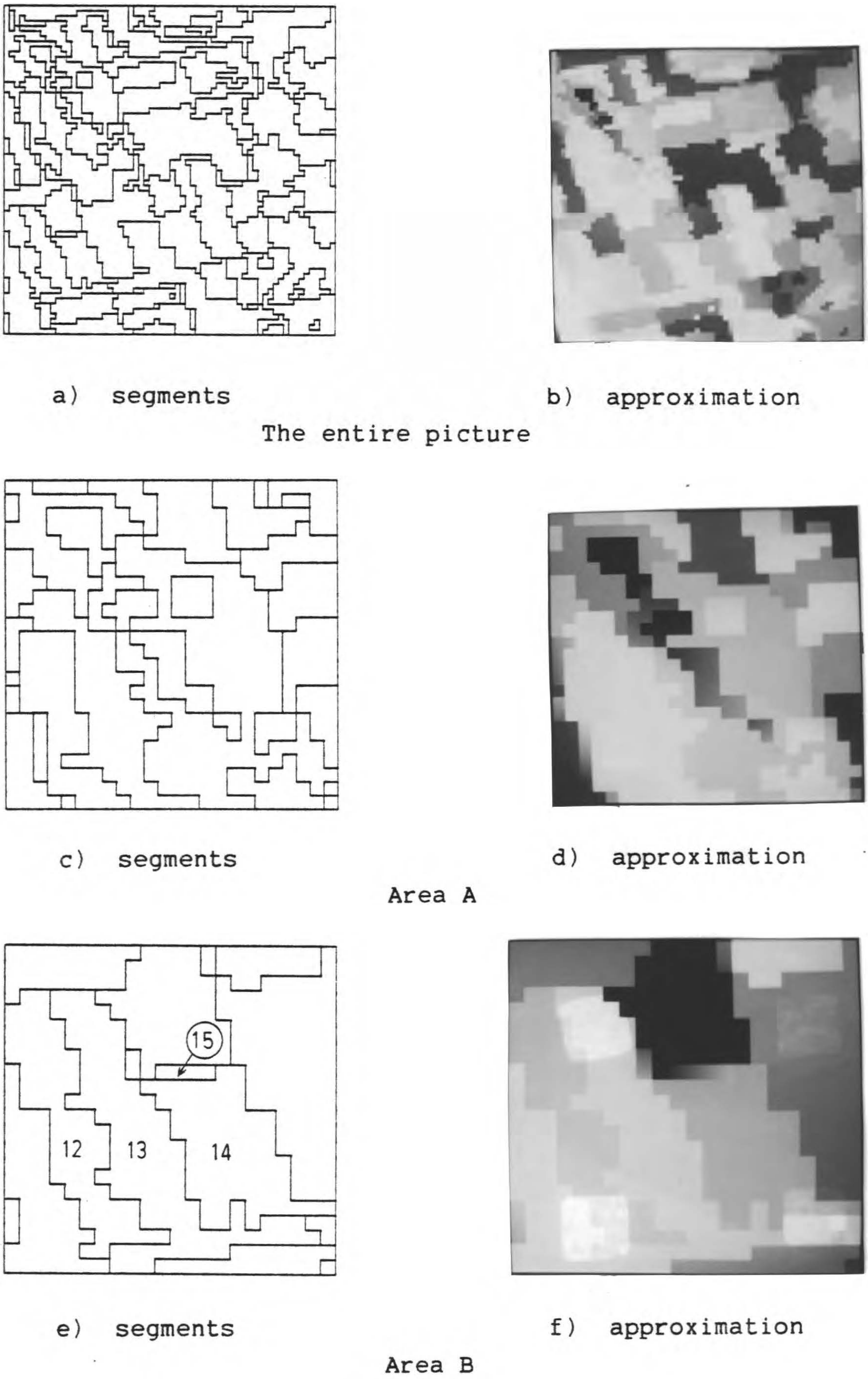
f) approximation

Area B

Figure 9.19: Segmentation results for the local variance adaptable approximation.

to using a simple criterion for the first merging steps, then, as we get to a higher level in the segment hierarchy, more complex criteria are used, involving more complex segment descriptors.

The ordering of segment descriptions and composite criteria are now employed to segment the Landsat image. The constant approximation criterion, $C_{i,j}$, is first used to obtain a partition with 1000 segments. Then the previously defined composite criterion is employed to continue the segment merging. The results which combine the characteristics of the preceding criteria are shown in Figure 9.20. For example, in Figure 9.20-e, the region 15 is represented by an inclined plane as is shown in Figure 9.15-e for the planar approximation. While the regions 12, 13 and 14 correspond to those obtained by constant value approximation in Figure 9.13. Thus, the advantages of planar approximation are exploited, while the previously noted artefacts are avoided. The constant value approximation is still predominant for large constant areas.



a) segments

b) approximation

The entire picture

c) segments

d) approximation

Area A

e) segments

f) approximation

Area B

Figure 9.20: Segmentation results from criterion combination.

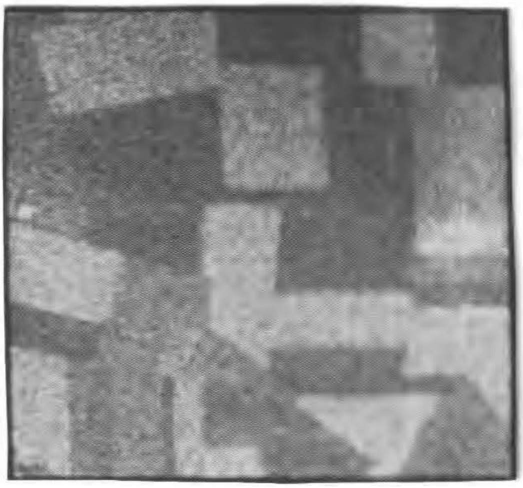


Figure 9.21: The SAR image (256×256 pixels).

9.3 Segmentation of a SAR image

The HSO algorithm is now used for a quite different type of image: a SAR (Synthetic Aperture Radar) image where the presence of speckle produces an important texture component. Good results are obtained, which demonstrate the versatility of the algorithm.

The one channel SAR image used in this section is presented in Figure 9.21, [29], [74]. This is an airborne X-band radar image with vertical-vertical polarization, 256×256 pixels, and a 5 meter resolution. The image covers a 1.28 km × 1.28 km area near Makofen, in the Federal Republic of Germany. It is an agricultural site composed of sugar beet, wheat, winter barley, potato, mixed hay and summer wheat, and corn fields.

The presence of coherent speckle makes the image noisy and greatly complicates the segmentation task. The derivation of the best image model or stepwise criterion for this segmentation task seems difficult [5], [24], [82]. An ad-hoc approach is employed instead, where the image characteristics are used in a more or less formal way to define stepwise criteria. The segmentation task is divided into two phases. A simple criterion is employed for an initial partition of the image, then a composite criterion is used for the subsequent merging steps.

The first phase consists in the partition of the image into 3000 segments using a simple criterion. The previously defined constant approximation

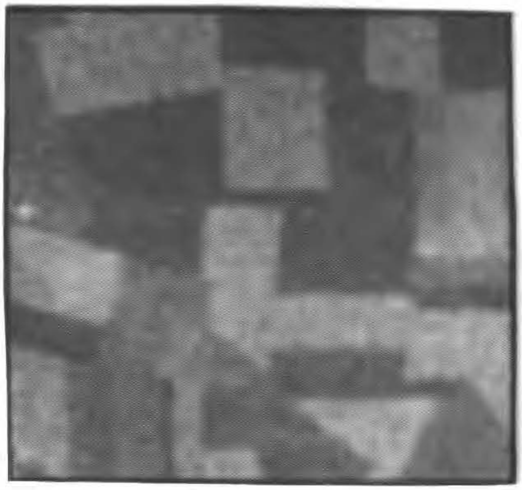


Figure 9.22: The average image calculated with a 5×5 window.

criterion is employed:

$$C_{(constant)} \;=\; \frac{N_i \cdot N_j}{N_i + N_j}\;(\,\mu_i - \mu_j\,)^2 \tag{9.9}$$

where N_i is the size of the segment S_i and μ_i is its mean value. The criterion is not applied to the original SAR image, but instead to an averaged version of this image. The average image is formed by assigning to each pixel the mean value of a 5×5 centered window (see Figure 9.22). The utilization of the average image, by reducing the effect of noise (speckle), results in the division of the image regions into more similar segments. It avoids, for example, the division of a homogeneous area into some segments which contain only the lighter pixels while the other segments are composed of the darker pixels, these two kinds of segments being interleaved. Note that, because of memory limitations, the image is divided into four independant blocks of 128×l28 pixels for the first segment mergers.

The second phase employs a composite criterion applied to the original SAR image to continue the merging of the initial 3000 segments. The segments can now be characterized by their means, μ_i, and their variances, σ_i^2, which can be exploited in the derivation of a segment similarity measure (criterion). Moreover, the utilization of a segment shape parameter can be useful to reduce the formation of random contours, an artifacts produced by the important noise component. Therefore, the employed composite criterion is composed of three parts:

$$C_{(composite)} = C_{(constant)} \cdot C_{(variance)} \cdot C_{(shape)} \tag{9.10}$$

where $C_{(constant)}$ is the previously defined constant approximation criterion which takes account of the difference between segment means and of the segment sizes.

$C_{(variance)}$ is defined as:

$$C_{(variance)} \;=\; 1 + |\sigma_i - \sigma_j| \tag{9.11}$$

where σ_i^2 is the gray level variance for segment S_i. The variances of the two segments are employed here in the evaluation of segment similarity. If two segments possess the same variance, then $C_{(variance)}$ is equal to one, which does not affect the composite result. If $|\sigma_i - \sigma_j|$ is equa1 to one or more, then the composite result is multiplied by 2 or more.

Finally, $C_{(shape)}$ measures the compactness of the segment, S_k, produced by the merging of S_i and S_j, $S_k = S_i \cup S_j$, [27], [17], [76]. The following definition is used:

$$C_{(shape)} = 1 \;+\; (1+\sigma_x)(1+\sigma_y)\Big/N_k \tag{9.12}$$

where,

$$\sigma_x^2 = \Big(\frac{1}{N_k}\sum_{(x,y)\in S_k} x^2\Big) - \Big(\frac{1}{N_k}\sum_{(x,y)\in S_k} x\Big)^2$$
$$\sigma_y^2 = \Big(\frac{1}{N_k}\sum_{(x,y)\in S_k} y^2\Big) - \Big(\frac{1}{N_k}\sum_{(x,y)\in S_k} y\Big)^2$$

and where N_k ($=N_i + N_j$) is the size of S_k ($=S_i \cup S_j$). σ_x and σ_y measure the pixel dispersion along the x and y axes respectively. These values tend to be small when a segment is compact. Their product is divided by N_k to compensate for the segment size. A bias of one is added to σ_x and σ_y in order to secure the effect of any one even if the other is null.

This composite criterion is applied to the SAR image in order to continue the merging of segments. The resulting minimum criterion values, $C_{min,k}$, are presented in Figure 9.23, while Figure 9.24 shows the corresponding image partitions for 25, 37 and 86 segments. For 25 segments, the most prominent areas of the image are correctly distinguished, but there remain a number of segments that are sub-parts of larger homogeneous regions. Some of these segments are marked by dots. They result from variations inside the homogeneous regions. These variations can be regarded as noise effects and are smaller than the variations between the main regions.

In the 37 segment partition of Figure 9.24-b, finer image components are considered. A number of the additional segments are distinct regions,

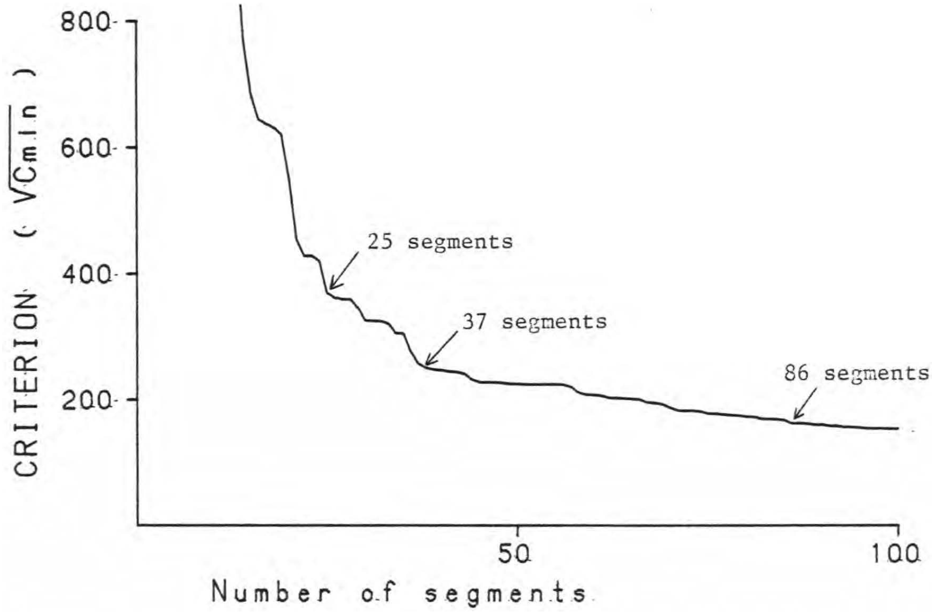


Figure 9.23: Upper bound of the minimum criterion values.

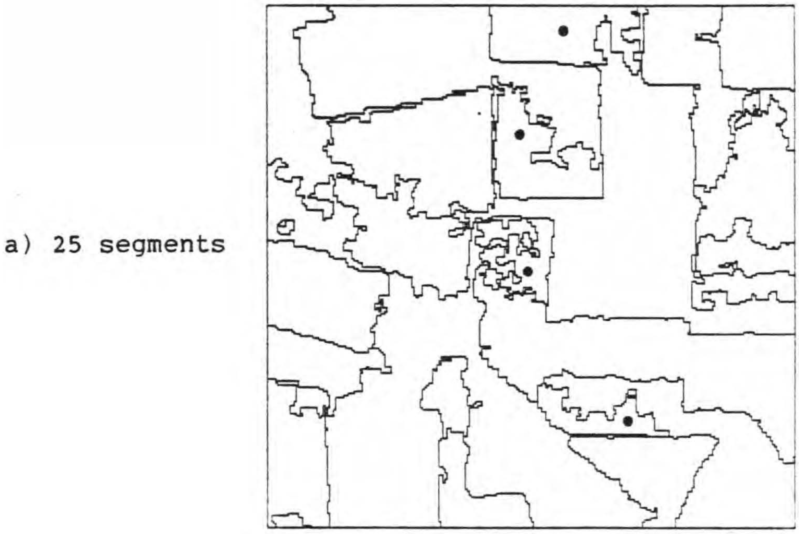
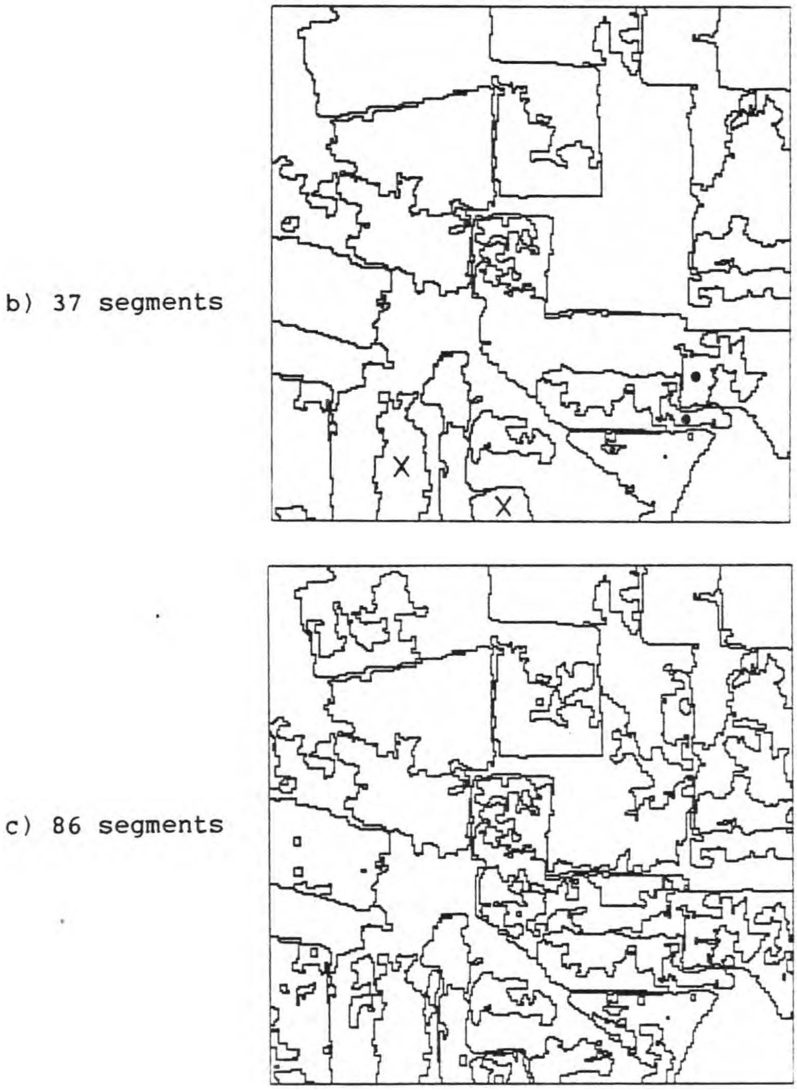


Figure 9.24: Segmentations of the SAR image with the composite criterion.



b) 37 segments

c) 86 segments

Figure 9.24: Continued

and are indicated by a cross "X". The other additional segments result from variations inside homogeneous regions, some of which are marked by dots. In the 86 segment partition, most of the additional segments can be regarded as due to noise effects.

Using the composite criterion in a second phase improves the image segmentation results. Figure 9.25 shows the results obtained by the utilization of the first phase only. The segment merging is performed with the constant approximation criterion of equation (9.9), until partitions of 25 and 37 segments are obtained. One evident difference is the occurrence of segments along the region boundaries. For example, in Figure 9.25, many region boundaries, indicated by arrows, are defined by double contour lines. These double lines delimit an area which must contain the true boundaries. However, the previous results with the composite criterion show better definition of the region boundaries.

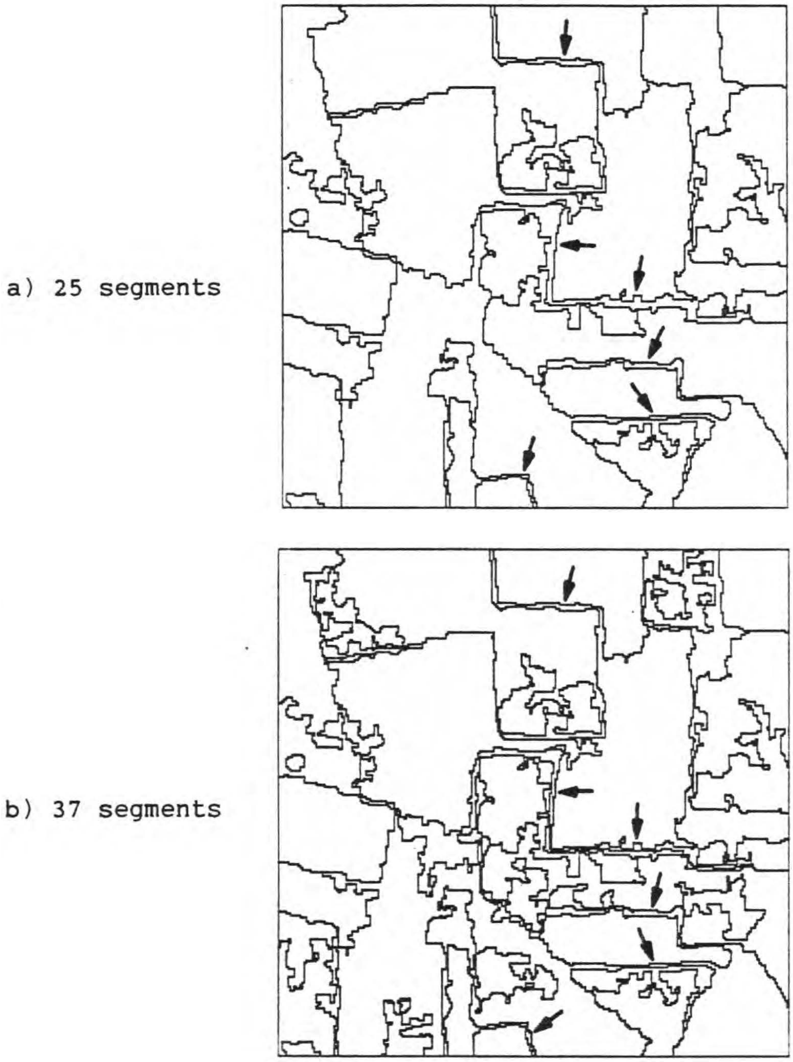


Figure 9.25: Segmentations of the SAR image with the constant approximation criterion only.

9.4 Comparing image segmentations

The problem of comparing the image partitions produced by different segmentation algorithms is now examined. A cost function, $\mathbb{G}(\mathbb{P})$, can be used to evaluate a image partition, $\mathbb{P}$. The algorithm producing the partition with the lowest cost is considered as the best one. However, the appropriate definition of the evaluation function, $\mathbb{G}(\mathbb{P})$, is a difficult problem. Previous works on this topic are first reviewed, then, the results of the HSO algorithm are compared with the results of other algorithms.

The evaluation of sample point partitions in classification and clustering techniques is first examined. The classification approach [17] (supervised learning) assumes that the sample points, $\bar{v}_i$, come from different classes, C_k. Training samples from each class are employed to calculate the probability functions of the classes, and to define the classification process. This classifier is then used to find the class memberships of unknown sample points. The probability of classification error constitutes a measure of the classifier performance. Testing samples with known class memberships are employed to evaluate this probability of error. Thus, the classification approach is a well formalized problem, and possesses correctly defined evaluation functions. However, the need of training samples is a serious limitation .

The clustering approach [16] (unsupervised learning) does not have this limitation, but the definition of a "cluster" is more ambiguous; e.g. a cluster is often defined as a group of similar samples. This affects the evaluation of the sample partitions. As an example, an often used partition evaluation measure is the within cluster variance. However, many variations around this measure are also used. The selection of an evaluation function is often ad-hoc, which limits the value of the conclusions that can result.

Image segmentation can be considered as a clustering process where spatial information is taken into account. The difficulties are to correctly define the goal of the image segmentation, and derive performance parameters that can be effectively calculated. It is often expected that the image segmentation processes match the human vision characteristics, which results in ill-defined goals.

There are few papers that compare the results of segmentation algorithms. Fram and Deutsch [103] compare different edge detection algorithms. The tests are performed on synthetic images containing vertical edges only. Two performance parameters are used: 1) the number of detected edge points belonging to the edge zone (signal) divided by the total number of edges (signal+noise), and 2) the number of rows containing at least one edge point .

Goehring and Ledford [26] evaluate segmentation algorithms for target detection. Different thresholding algorithms are employed to separate the target areas from the background. The parameters used are 1) the probability of segmentation (the probability of detecting a true or false target), and 2) the nuisance rate (the number of false detections).

Levine and Nazif [104] define a segmentation as a partition of the image into regions that are uniform among themselves and bear contrast to their adjacent neighbors. They define different performance parameters based upon region uniformity, region contrast, line contrast, line connectivity and texture measures.

It must be noted that, in the examined cases, the selected preformance parameters have a degree of arbitrariness, and that no firm conclusions can be drawn from the evaluations, although, they can be useful to confirm subjective statements.

In this thesis, a hierarchical stepwise optimization algorithm (HSO) is described, and its advantages are analysed for well defined image segmentation goals: the low error approximation of image and the minimization of the probability of merging error. It is shown that the HSO algorithm can be used to optimize a global criterion, $\mathbb{G}(\mathbb{P})$. The goal of the image segmentation is then expressed through this global criterion, and the evaluation of the resulting partitions must, therefore, be also based upon this global criterion. For example, in image approximation, the global criterion is the approximation error, and the result evaluations must be based upon this approximation error.

In section 9.2, the HSO algorithm is employed for constant value approximation. The algorithm is applied to a satellite image, shown in Figure 9.12-a, and a 100 segment partition is obtained (Figure 9.13-a). This produces an approximation error (standard deviation) of 6.71. This result is now compared with the one produced by another segmentation algorithm: an improved version of the algorithm of Narendra and Goldberg [58] presented in section 3.1. The gradient operator used is the variance inside a 3×3 window. The algorithm is applied to the same image (Figure 9.12-a), and the smoothing parameter is selected such as to produce approximately 100 segments. The result is a 101 segment partition with an approximation error (standard deviation) of 10.05, which is higher than the HSO algorithm result.

In section 9.3, the HSO algorithm is employed to segment a SAR image (Figure 9.21). A composite stepwise criterion is used, defined in an ad-hoc manner. Segmentation results are presented in Figure 9.24. Goodenough et al. [29] have used the Narendra and Goldberg algorithm [58] to segment this same image, and have evaluated the different partitions obtained. An adaptive filter is first applied to the image to reduce the multiplicative noise while preserving the edges. Different image partitions result from the utilization of different window sizes for the adaptive filter, and different gradient operators and smoothing parameters for the segmentation algorithm.

Two criteria are used to evaluate the resulting image partitions. First, it is determined if the manually defined boundaries are present in the segmentation results. From a manually draw edge image, a mask is created by thickening the edges by ±2 pixels. The mask is applied to each image segmentation in order to retain only the segment contours inside the edge mask. This is employed to determine the number of manually defined edges that are also present in the segmentation results. Only the continuous boundaries are counted for a maximum of 41 edges.

The second criterion evaluates the segmentation performance by the total number of segments created within known homogeneous fields. Segments inside eleven fields are counted for each image segmentations. Partitions where these fields are broken into the fewest number of segments are considered to be the best.

The best image partition obtained by Goodenough et al. possesses 32 correctly identified edges, and has 404 segments inside the 11 homogeneous fields. This partition contains a total of 703 segments, and is produced by using a 11×11 window for the filter and a variance operator for the gradient image.

The same evaluation procedure is now applied to the results of the HSO algorithm which are presented in the section 9.3. A 703 segment partition is first used in order to facilitate the comparison. This partition possesses 33 correctly identified edges and has 361 segments inside the 11 homogeneous fields (see Table 9.1). The partition presented in Figure 9.24-c is also evaluated. In this case, 29 edges are correctly identified and the 11 homogeneous fields are split into 52 segments. This partition contains a total of 86 segments.

For an initial trial, these last results compare favorably with those of Goodenough et al. [29]. They can be improved by using more appropriate stepwise criteria. The HSO algorithm has the advantages that a partition with the required number of segments is easily produced, and that good results are obtained even for partitions with a small number of segments.

Table 9.1: Image partition evaluation.

	identified edges	segments inside homogeneous fields
Goodenough et al. [29] (703 segments)	32	404
HSO algorithm (703 segments)	33	361
(86 segments)	29	52

Chapter 10

Summary and Suggestions for Further Research

The main contributions of the thesis are as follows:

1) The survey of image segmentation techniques:

   The segmentation algorithms are divided into categories according to the definitions of segments adopted. Image segmentation techniques are regarded as data classification and clustering processes where the spatial information is included as new features or in the distance measures used.

2) The presentation of a new hierarchical segmentation algorithm:

   A hierarchical segmentation algorithm based upon stepwise optimization is described. The advantages of the HSO algorithm over hierarchical algorithms using logical predicates are shown. Considerations for an efficient implementation of the HSO algorithm are also discussed.

3) The combination of segment optimization and hierarchy:

   The algorithm is advantageously employed in a global optimization problem: the piece-wise approximation of images. When the stepwise criterion is derived from the global one, the stepwise optimization algorithm becomes a sub-optimal process which exploits the advantage of segment hierarchy to reduce the search space.

4) The analysis of error probability in hierarchical segmentation:

   Image segmentation is regarded as an hypothesis testing process which merges two segments only if they belong to the same region. It is shown that, at each step of a hierarchical segmentation process, the merging of dissimilar segments is the most serious error, and therefore its probability must be minimized. This is acheived by the HSO algorithm. The probability of stepwise error is derived, and the effects of segment sizes are analysed. The increase of the minimum criterion values is also employed to distinguish the noise from the signal components.

5) The illustration of the capability of the HSO algorithm:

   The algorithm is shown to be adaptable to different segmentation tasks by an appropriate selection of the stepwise criterion. Segmentation examples illustrate the operation of the algorithm, and its ability to exploit the hierarchical structure of the image in order to produce a partition with the required amount of details. Good segmentation results are reported for remote sensing images.

Several areas of future research in image segmentation which can complement this thesis are now presented.

1) Evaluation of the segmentation results:

   The HSO algorithm can be regarded as a sub-optimal process. The evaluation of the difference between the algorithm result and the global optimum can be useful, in particular, to measure the performance of the algorithm. However, this evaluation is difficult because, in general, the global optimum can be calculated only for simple cases. More generally, there is a need to compare and evaluate the results of image segmentation algorithms, and any contribution to this topic will be worthwhile.

2) Combination of stepwise optimization and iterative local optimization:

   An iterative process, using a "steepest descent" like approach, can be employed to improve an initial image partition (see section 5.2). A "steepest descent" algorithm and a HSO algorithm can be derived from the same global criterion. These two versatile sub-optimum processes could then be combined to produce results that are closer to the desired global optimum. For example, the "steepest descent" algorithm could be applied to improve the result of the HSO algorithm.

3) Analysis of the sequence of minimum criterion values:

   The analysis of the minimum criterion values yielded by the HSO algorithm can provide useful information on the image structure and on the appropriate stopping points. Although some aspects have been studied in the section 8.9, 8.10 and 9.1, much work remains.

4) Exploration of new segment models and segmentation criteria:

   An important issue in image segmentation is the selection of the segment model and segmentation criterion that is the most appropriate for each particular application. This issue is related to the problem of image partition evaluation, and concerns basically the appropriate definition of the goal of a image segmentation task. Some aspects of this problem have been discussed in the thesis. However, this is a complex problem that still requires considerable research.

5) Developing faster versions of the algorithm:

   For many applications, the large computing time required by the HSO algorithm constitues a severe limitation. The characteristics of specific applications can be exploited in order to develop faster versions of the algorithm.

Appendix A

Probability Function of the Minimum Value

The following proposition is proved by a recursive demonstration.

> **Proposition**: Let $x_{(K)}$ be the minimum value of a set of K independant random variables, $\{x_i\}$, with identical probability functions $\mathrm{P}(x)$, then, for any $K \geq 2$, the probability function of $x_{(K)}$ is:

$$\mathrm{P}_{(K)}(x_{(K)}) \;=\; \sum_{j=1}^{K} (-1)^{j+1} \frac{K!}{j!\,(K-j)!} \, \mathrm{P}(x_{(K)})^j \tag{A.1}$$

Proof:

a) The proposition is proved for $K = 2$ as follows (see [66], p. 192):

$$\begin{aligned} \mathrm{P}_{(2)}(x_{(2)}) \;&=\; 2\ \mathrm{P}(x_{(2)}) - \mathrm{P}(x_{(2)})^2 \\ &= \sum_{j=1}^{2} (-1)^{j+1} \frac{2!}{j!\,(2-j)!} \, \mathrm{P}(x_{(2)})^j \end{aligned} \tag{A.2}$$

b) It is now proved that, if the proposition is true for $K = n$, then the proposition is also true for $K = n + 1$.

If the probability function for $x_{(n)}$ (=min$\{x_1, x_2 \ldots, x_{(n)}\}$) is:

$$\mathrm{P}_{(n)}(x_{(n)}) \;=\; \sum_{j=1}^{n} (-1)^{j+1} \frac{n!}{j!\,(n-j)!} \, \mathrm{P}(x_{(n)})^j \tag{A.3}$$

then the probability function for

$$\begin{aligned} y = x_{(n+1)} &= \min\{x_1, x_2 \ldots, x_{n+1}\} \\ &= \min\{\, x_{(n)},\, x_{n+1} \,\} \end{aligned} \tag{A.4}$$

is given by (see [66], p. 192)

$$
\begin{aligned}
\mathrm{P}_{(n+1)}(y) \;&= \mathrm{P}(y) + \mathrm{P}_{(n)}(y) - \mathrm{P}(y)\,\mathrm{P}_{(n)}(y) \\
\mathrm{P}_{(n+1)}(y) \;&= \mathrm{P}(y) + \sum_{j=1}^{n} (-1)^{j+1} \frac{n!}{j!\,(n-j)!}\; \mathrm{P}(y)^{j} \\
&\quad + \sum_{i=1}^{n} (-1)^{i+2} \frac{n!}{i!\,(n-i)!}\; \mathrm{P}(y)^{i+1} \\
\mathrm{P}_{(n+1)}(y) \;&= \mathrm{P}(y) + \sum_{j=1}^{n} (-1)^{j+1} \frac{n!}{j!\,(n-j)!}\; \mathrm{P}(y)^{j} \\
&\quad + \sum_{j=2}^{n+1} (-1)^{j+1} \frac{n!}{(j-1)!\,(n+1-j)!}\; \mathrm{P}(y)^{j} \\
\mathrm{P}_{(n+1)}(y) \;&= \mathrm{P}(y) + \;\frac{n!}{(n-1)!}\; \mathrm{P}(y) \\
&\quad + \sum_{j=2}^{n} (-1)^{j+1} \frac{n!}{(j\text{-}1)!\,(n\text{+}1\text{-}j)!} + \frac{n!}{j!\,(n\text{-}j)!}\; \mathrm{P}(y)^{j} \\
&\quad + \;(-1)^{n+2}\, \frac{n!}{n!}\; \mathrm{P}(y)^{n+1} \\
\mathrm{P}_{(n+1)}(y) \;&= \quad (n+1)\; \mathrm{P}(y) \\
&\quad + \sum_{j=2}^{n} (-1)^{j+1} \frac{(n+1)!}{j!\,(n+1-j)!}\; \mathrm{P}(y)^{j} \\
&\quad + \;(-1)^{n+2}\, \frac{(n+1)!}{(n+1)!}\; \mathrm{P}(y)^{n+1} \\
\mathrm{P}_{(n+1)}(y) \;&= \quad \sum_{j=1}^{n+1} (-1)^{j+1} \frac{(n+1)!}{j!\,(n+1-j)!}\; \mathrm{P}(y)^{j}
\end{aligned}
\tag{A.5}
$$

c) From a) and b), it can be concluded that the proposition is true for any $K \geq 2$.

Appendix B

The Best Estimate of an Image Partition

The best estimate of the true image partition is derived and shown to correspond to the image partition that minimizes the approximation error. Let $R = \{R_i\}$ be the true image partition, $\mathfrak{m}_i$ be the constant value for region R_i, and $f(x, y)$ be the observed image value

$$f(x, y) = \mathfrak{m}_i + e(x, y) \quad \text{for } \forall\, (x, y) \in R_i. \tag{B.1}$$

where $e(x, y)$ are Gaussian independant random variables with zero mean and variance of σ^2. Then, the best estimate $\hat{R} = \{\hat{R}_i\}$ of the image partition maximizes the likelihood function

$$\begin{aligned} L(\hat{R}|f) &= Prob(\, f \,|\, \hat{R}\,) \\ &= \prod_{(x,y)} \frac{1}{\sqrt{2\pi}\,\sigma} \; \exp\left(\frac{-(f(x,y) - \hat{m}_{(x,y)})^2}{2\,\sigma^2} \right) \\ &= \left(\frac{1}{\sqrt{2\pi}\,\sigma} \right)^n \; \exp\left(- \sum_{(x,y)} \frac{(f(x,y) - \hat{m}_{(x,y)})^2}{2\,\sigma^2} \right) \end{aligned} \tag{B.2}$$

where $\hat{m}_{(x,y)} = \hat{m}_i$ for $(x, y) \in \hat{R}_i$, $\hat{m}_i$ being the constant value for region $\hat{R}_i$, and where n is the number of pixels in the image. Thus, maximizing $L(\hat{R}|f)$ corresponds to minimizing

$$\sum_{(x,y)} \; (f(x, y) - \hat{m}_{(x,y)})^2 \tag{B.3}$$

which can be rewritten as

$$\sum_{\hat{R}_i} \; \sum_{(x,y) \in \hat{R}_i} \; (f(x, y) - \hat{m}_i)^2. \tag{B.4}$$

The best estimate thus corresponds to the partition with the lowest approximation error, and it can be shown that the best value for $\hat{m}_i$ is the mean value of the region $\hat{R}_i$.

REFERENCES

[1] N. Ahuja and A. Rosenfeld, "Neighbor Gray Level as Feature in Pixel Classification," *Pattern Recognition*, Vol. 12 (4), 1980, 251-260.

[2] H.J. Antonisse, "Image Segmentation in Pyramids," *Comp. Grap. & Image Proc.*, Vol. 19, 1982, 367-383.

[3] T. Asano and N. Yokoya, "Image Segmentation Schema for Low-Level Computer Vision," *Pattern Recognition*, Vol. 14, 1981, 267-273.

[4] J.W. Bacus, W.A. Yasnoff and M.G. Belanger, "Computer Techniques for Cell Analysis in Hematology," in *Proceeding of the First Annual Symposium on Computer Application in Medical Care*, Oct. 1977, 24-35.

[5] D.H. Berger, "Texture as a Discriminant of Crops on Radar Imagery," *IEEE Trans. on Geoscience Electronics*, Vol. GE-8 (4), 1970, 344-348.

[6] C. Brice and C. Fennema, "Scene Analysis Using Regions," *Artificial Intelligence*," vol. 1, 1970, 205-226.

[7] D.J. Burr and R.T. Chien, "The Minimal Spanning Tree in Visual Data Segmentation," *Proc. of 3 Int. Join Conf. on Pattern Rec.*, 1976; 519-523.

[8] P.J. Burt, T.H. Hong and A. Rosenfeld, "Segmentation and Estimation of Image Region Properties through Cooperative Hierarchical Computation," *IEEE Trans. on Sys., Man and Cyb.*, Vol. SMC-11, 1981, 802-809.

[9] S.G. Carlton and O.R. Mitchell, "Object/Background Segmentation in FLIR Imagery," *Conf. on Pattern Recognition and Image Processing*, 1978, 360-362.

[10] J.M. Chassery and C. Garbay, "Iterative Process for Contour Image Segmentation Using a Convexity Criterion," *Proc. of SPIE*, Vol. 397, 1983, 165-172.

[11] G.C. Cheng, "Color Information in Blood Cells," *The Journal of Histochemistry and Cytochemistry*, Vol. 22, 1974, 517-521.

[12] C.H. Chen, "A Comparison of Image Segmentation Techniques," *Proc. of Int. Conf. on Cybernetics and Society*, Atlanta, GA, USA, 1981, 364-372.

[13] G.B. Coleman and H.C. Andrews, "Image Segmentation by Clustering," *Proceeding of IEEE*, Vol. 67 (5), 1979, 773-385.

[14] D.B. Cooper, "Maximum Likelihood Estimation of Markov-Process Blob Boundaries in Noisy Images," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-1 (4), 1979, 372-384.

[15] L.S. Davis, "A Survey of Edge Detection Techniques," *Comp. Grap. & Image Proc.*, Vol. 4, 1975, 248-270.

[16] R. Dubes and A.K. Jain, "Clustering Methodologies in Exploratory Data Analysis," in Advances in Computer, Vol. 19, ed. by M.C. Yovits, Academic Press, New-York, 1980, 113-228.

[17] R.O. Duda and P.E. Hart, PATTERN CLASSIFICATION AND SCENE ANALYSIS, John Wiley and Sons, New-York, 1973.

[18] R.W. Ehrich and J.P. Foith, "Topology and Semantics of Intensity Arrays," in COMPUTER VISION SYSTEMS, A.R. Hanson and E.M. Riseman, ed., Academic Press, New-York, 1978, 111-127.

[19] J.O. Eklundh, H. Yamamoto and A. Rosenfeld, "A Relaxation Method for Multispectral Pixel Classification," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-2 (1), 1980, 72-75.

[20] H. Elliott, D.B. Cooper, F.S. Cohen and P.F. Symosek, "Implementation, Interpretation, and Analysis of a Suboptimal Boundary Finding Algorithm," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-4 (2), 1982, 167-182.

[21] O.D. Faugeras, "An Overview of Probabilistic Relaxation Theory and Applications," in *Digital Image Processing and Analysis*, NATO, Advanced Study Institute, Published by INRA, Rocquencourt, France, 1980.

[22] O.D. Faugeras and M. Berthed, "Improving Consistency and Reducing Ambiguity in Stochastic Labeling: an Optimization Approach," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-3 (4), 1981, 412-424.

[23] E.C. Freuder, "Affinity: A Relative Approach to Region Finding," *Comp. Graph. & Image Proc.*, vol. 5, 1976, 254-264.

[24] V.S. Frost, J.A. Stiles, K.S. Shanmugan and J.C. Holtzman, "A Model for Radar Images and Its Application to Adaptive Digital Filtering of

Multiplicative Noise," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-4 (2), 1982, 157-166.

[25] K.S. Fu and J.K. Mui, "A Survey on Image Segmentation," *Pattern Recognition*, Vol. 13, 1981, 3-16

[26] G.A. Goehring and L.G. Ledford, "Analysis of Image Segmentation Approaches with Emphasis on Performance Evaluation Criteria," *SPIE*, Vol. 252 (Smart Sensors II), 1980, 124-129.

[27] R.C. Gonzalez and P. Wintz, DIGITAL IMAGE PROCESSING, Addison-Wesley Publishing C., London, 1977.

[28] D.G. Goodenough, P.M. Narendra and K. O'Neill, "Feature Subset Selection in Remote Sensing," *Canadian Journal of Remote Sensing*, Vol. 4 (2), 1978, 143-148.

[29] D.G. Goodenough, B. Guindon, J.F. Meunier and N.A. Swanberg, "Adaptive Filtering and Image Segmentation for SAR Analysis," *Proc. of Machine Processing of Remotely Sensed Data Symposium*, June 1984, 315-324.

[30] J.N. Gupta and P.A. Wintz, "A Boundary Finding Algorithm and its Applications," *IEEE Trans. Circuits& Sys.*, vol. CAS-22(4), 1975, 351-362.

[31] E.L. Hall, COMPUTER IMAGE PROCESSING AND RECOGNITION, Academic Press, New-York, 1979.

[32] R.M. Haralick and I. Dinstein, "A Spatial Clustering Procedure for Multi Image Data," *IEEE Trans. on Circuits and Sys.*, Vol. CAS-22 (5), 1975, 440-450.

[33] T.H. Hong, K.A. Narayanan, S. Peleg, A. Rosenfeld and T. Silberberg, "Image Smoothing and Segmentation by Multiresolution Pixel Linking: Further Experiments and Extensions," *IEEE Trans. on Sys., Man and Cyb.*, Vol. SMC-12 (5), 1982, 611-622.

[34] S.L. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," *Journal of the Ass. for Computing Machinery*, vol. 23, 1976, 368-388.

[35] M.H. Hueckel, "An Operator which Locate Edges in Digital Picture," *Journal of ACM*, Vol. 18 (1), 1971, 113-125.

[36] R.A. Hummel and S.W. Zucker, "On the Foundations of Relaxation Labeling Processes," Computer Vision and Graphics Lab., Report TR-80-7, McGill University, Montreal, Canada, 1980.

[37] R.A. Jarvis and E.A. Patrick, "Clustering Using a Similarity Measure Based on Shared Near Neighbors," *IEEE Trans. on Computer*, Vol. C-22 (11), 1973, 1025-1034.

[38] T. Kanade, "Region Segmentation: Signal vs Semantics," *4th Int. Joint Conf. on Pattern Recognition*, 1978, 95-105.

[39] S. Kasif and A. Rosenfeld, "Pyramid Linking is a Special Case of Isodata," Technical Report TR-1096, MCS-79-23422, Computer Science Center, Univ. of Maryland, sept. 1981, (see also IEEE Trans. on Sys., Man and Cyb., Vol. SMC-13, 1983).

[40] W.Kestner, M. Bohner, R. Scharf and M.Sties, "Object Guided Segmentation of Aerial Images," *5th Int. Joint Conf. on Pattern Recognition*, 1980, 529-531.

[41] R.L. Kettig and D.A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects," *IEEE Trans. Geosci. Electron.*, vol. GE-14 (1), 1976, 19-26.

[42] R. Kohler, "A Segmentation System Based on Thresholding," *Comp. Grap. & Image Proc.*, Vol. 15, 1981, 319-338.

[43] V. A. Kovalevsky, IMAGE PATTERN RECOGNITION, Springer-Verlag, New-York, 1980.

[44] Murat Kunt, "Edge Detection: A Tutorial Review," *ICASSP-82*, 1982, 1172-1175.

[45] D.A. Landgrebe, "The Development of a Spectral-Spatial Classifier for Earth Observational Data," *Pattern Recognition*, vol. 12, 1980, 165-175.

[46] C.H. Lee, "Iterative Region Segmentation," *82 Int. Joint Conf. on Pattern Rec. and Image Proc.*, 1982, 557-559.

[47] S. Levialdi, "Finding the Edge," in DIGITAL IMAGE PROCESSING, J.C. Simon and R.M. Haralick, ed., D. Reidel Publishing Comp., 1981.

[48] B.W. Lindgren, STATISTICAL THEORY, Macmillan Publishing Co. Inc., New-York, third edition, 1976.

[49] A. Martelli, "An Application of Heuristic Search Methods to Edge and Contour Detection," *Commun. of the ACM*, Vol. 19 (2), 1976, 73-83.

[50] C.A. McNary, D.K. Conti and W.O. Eckhardt, "Segmentation-based Boundary Modeling for Natural Terrain Scenes," *SPIE*, Vol. 205, Image Understanding Systems II, 1979,108-116.

[51] D.L. Milgram, "Region Extraction Using Convergent Evidence," *Comp. Grap. & Image Proc.*, Vol. 11, 1979, 1-12.

[52] D.L. Milgram and D.J. Kahl, "Recursive Region Extraction," *Comp. Graph. & Image Proc.*, vol. 9, 1979, 82-88.

[53] D.L. Milgram and M. Herman, "Clustering Edge Values for Threshold Selection," *Comp. Grap. & Image Proc.*, Vol. 10, 1979, 272-280.

[54] R. Mizoguchi and O. Kakusho, "Hierarchical Clustering Algorithm Based on K-Nearest Neighbors," *Proc. 4 Int. Joint Conf. on Pattern Rec.*, 1978, 314-316.

[55] P.A. Nagin, A.R. Hanson and E.M. Riseman, "Studies in Global and Local Histogram-Guided Relaxation Algorithms," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-4 (3), 1982, 263-277.

[56] Y. Nakagawa and A. Rosenfeld, "Edge/Border Coincidence as an Aid in Edge Extraction," *IEEE Trans. on sys., Man and cyb.*, Vol. SMC-8, 1978, 899-903.

[57] Y. Nakagawa and A. Rosenfeld, "Some Experiments on Variable Thresholding," *Pattern Recognition*, Vol. 11 (3), 1979, 191-204.

[58] P.M. Narendra and M. Goldberg, "Image Segmentation with Directed Trees," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-2 (2), 1980, 185-191.

[59] K.A. Narayanan, D.P. O'Leary and A. Rosenfeld, "Image Smoothing and Segmentation by Cost Minimization," *IEEE Trans. on Sys., Man and Cyb.*, Vol. SMC-12 (1), 1982, 91-96.

[60] R. Nevatia and K. Price, "A Comparison of some Segmentation Techniques," *Proc. of Image Understanding Workshopp*, april 1977, 55-57.

[61] R. Ohlander, ANALYSIS OF NATURAL SCENES, Ph.D. Thesis, Carnegie-Mellon University, Computer Science Departement, Pittsburgh, Pa., June 1975.

[62] R. Ohlander, K. Price and D.R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," *Comp. Graph.& Image Proc.*, vol. 8, 1978, 313-333.

[63] Y. Ohta, T. Kanade and T. Sakai, "Color Information for Region Segmentation," *Comp. Grap. & Image Proc.*, Vol. 13 (3), 1980, 222-241.

[64] D.P. Panda and A. Rosenfeld, "Image Segmentation by Pixel Classification in (Gray Level, Edge value) Space," *IEEE Trans. on Computer*, Vol. C-27 (9), 1978, 875-880.

[65] D.P. Panda and T. Dubitzki, "Statistical Analysis of some Edge Operators," *Comp. Grap.& Image Proc.*, Vol. 11, 1979, 313-348.

[66] A. Papoulis, PROBABILITY, RANDOM VARIABLES, AND STOCHASTIC PROCESSES, McGraw-Hill Book Co., New-York, 1965.

[67] T. Pavlidis, "Segmentation of Pictures and Maps through Functional Approximation," *Comp.Graphic & Image Proc.*, Vol. l (4), 1972, 360-372.

[68] T. Pavlidis, "Hierarchies in Structural Pattern Recognition," *Proceeding of IEEE*, vol. 67 (5), 1979, 737-744.

[69] S. Peleg, "Classification by Discrete Optimization," *Comp. Vision, Graph. and Image Proc.*, Vol. 25 (1), 1984, 122-130.

[70] M. Pietikainen and A. Rosenfeld, "Image Segmentation by Texture Using Pyramid Node Linking," *IEEE Trans. on Sys., Man and Cyb.*, Vol. SM C-11, 1981, 822-825.

[71] M. Pietikainen, A. Rosenfeld and I. Walter, "Split-and-Link Algorithms for Image Segmentation," *Pattern Recognition*, Vol. 15 (4), 1982, 287-298.

[72] T.-c. Pong, L.G. Shapiro, L.T. Watson and R.M. Haralick, "Experiments in Segmentation Using a Facet Model Region Grower," *Comp. Vision, Graph. and Image Proc.*, Vol. 25 (1), 1984, 1-23.

[73] J.M.S. Prewitt and M.L. Mendelsohn, "The Analysis of Cell Images," *Annals of the New-York Academy of Sciences*, Vol. 128 (3), 1966, 1035-1053.

[74] J. Princz, "CCRS Digitally Processed Image Products," *Proceeding of the Eighth Canadian Symposium on Remote Sensing*, 1983.

[75] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," *IEEE Trans. on Compute*r, Vol. C-20 (5), 1971, 562-569.

[76] A. Rosenfeld and A.C. Kak, DIGITAL PICTURE PROCESSING, Academic Press, New-York, 1976.

[77] A. Rosenfeld and L.S. Davis, "Image Segmentation and Image Models," *Proceeding of IEEE*, vol. 67, 1979, 764-772.

[78] A. Rosenfeld, "Quadtrees and Pyramids for Pattern Recognition and Image Processing," *5th Int. Joint Conf. on Pattern Recognition*, 1980, 802-811.

[79] A. Rosenfeld and A.Y. Wu, "Parallel Computers for Region-level Image Processing," *Pattern Recognition*, Vol. 15 (1), 1982, 41-50.

[80] A. Sarabi and J.K. Aggarwal, "Segmentation of Chromatic Images," *Pattern Recognition*, Vol. 13 (6), 1981, 417-427.

[81] B.J. Schachter, L.S. Davis and A. Rosenfeld, "Some Experiments in Image Segmentation by Clustering of Local Feature Values," *Pattern Recognition*, Vol. 11, 1979, 19-28.

[82] K.S. Shanmugan, V. Narayanan, V.S. Frost, J.A. Stiles and J.C. Holtzman, "Textural Features for Radar Image Analysis," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. GE-19 (3), 1981, 153-156.

[83] S. Shafer and T. Kanade, "Recursive Region Segmentation by Analysis of Histograms," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Paris (France), 1982, 1166-1171.

[84] J.C. Simon, "Some Current Topics in Clustering in Relation with Pattern Recognition," *4th Int. Joint Conf. on Pattern Recognition*, 1978, 19-29.

[85] P.H.A. Sneath and R.R. Sokal, NUMERICAL TAXONOMY, Freeman, San Francisco, 1973.

[86] S.D. Stearns, "On Selecting Features for Pattern Classifiers," *3rd Inter. Joint Conf. on Pattern Recognition*, 1976, 71-75.

[87] P.H. Swain, S.B. Vardeman and J.C. Tilton, "Contextual Classification of Multispectral Image Data," *Pattern Recognition*, Vol. 13 (6), 1981, 429-441.

[88] S.L. Tanimoto, "Regular Hierarchical Image and Processing Structures in Machine Vision," in Computer Vision System, Academic Press, New-York, 1979, 165-174.

[89] J.M. Tenenbaum and H.G. Barrow, "Experiments in Interpretation Guided Segmentation," *Artificial Intelligence*, Vol. 8, 1976, 241-274.

[90] J.C. Tilton and S.C. Cox, "Segmentation of Remotely Sensed Data Using Parallel Region Growing," *Proc. of 1983 Inter. Geoscience and Remote Sensing Symposium (IGARSS'83)*, San Francisco, California, 1983, 9.1-9.6.

[91] J.T. Tou and R.C. Gonzalez, PATTERN RECOGNITION PRINCIPLES, Addison-Wesley Publishing Comp., London, 1974.

[92] S. Tsuji and F. Tomota, "A Structural Analyzer for a Class of Textures," *Comp. Grap. & Image Proc.*, Vol. 2, 1973, 216- 231.

[93] J.H. Ward, "Hierarchical Grouping to Optimize an Objective Function," *J. Amer. Stat. Ass.*, vol. 58, 1963, 236-245.

[94] J.S. Weszka, R.N. Nager and A. Rosenfeld , "A Threshold Selection Technique," *IEEE Trans. on Computer*, Vol. C-, Dec. 1974, 1322-1326.

[95] J.S. Weszka, "A Survey of Threshold Selection Techniques, " *Comp. Grap. & Image Proc.*, Vol. 7 (2), 1978, 259-265.

[96] S.W. Wharton, "A Contextual Classification Method for Recognizing Land Use in High Resolution Remotely Sensed Data," *Pattern Recognition*, Vol. 15 (4), 1982, 317-324.

[97] G. Winkler and K. Vattrodt, "Measures for Conspicuousness," *Comp. Grap. & Image Proc.*, Vol. 8, 1978, 355-368.

[98] A.Y. Wu, T.H. Hong and A. Rosenfeld, "Threshold Selection Using Quadtrees," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-4 (1), 1982, 90-94.

[99] N. Yokoya, T. Kitahashi, K. Tanaka and T. Asano, "Image Segmentation Scheme Based on a Concept of Relative Similarity," *4th Int. Joint Conf. on Pattern Recognition*, 1978, 645-647.

[100] C.T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. on Computer*, Vol. C-20 (1), 1971, 68-86.

[101] A.L. Zobrist and W.B. Thompson, "Building a Distance Function for Gestalt Grouping," *IEEE Trans. on Computer*, Vol. C-4 (7), 1975, 718-728.

[102] S.W. Zucker, "Region Growing: Childhood and Adolescence," *Comp. Grap. & Image Proc.*, Vo1. 5, 1976, 382-399.

[103] J.R. Fram and E.S. Deutsch, "On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance," *IEEE Trans. on Computer*, Vol. C-24 (5), 1975, 616-628.

[104] M.D. Levine and A.M. Nazif, "Dynamic Measurement of Computer Generated Image Segmentations," *IEEE Trans. on Pattern Analysis and Machine Int.*, Vol. PAMI-7 (2), 1985, 155-164.

[105] M. Goldberg and D.G. Goodenough, "Analysis of a Spatial Filter for Landsat Imagery," *Journal of Applied Photographic Engineering*, Vol. 4 (1), 1978, 25-27.

ISBN 978-1-7388812-0-8
Publisher Jean-Marie Beaulieu

BeaulieuJM.ca/publi/Beau23HS