USC–SIPI REPORT #199

Hierarchical and Graph Theoretic Approaches to Image Segmentation and Pattern Classification

by

Zhenyu Wu

May 1992

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

To the memory of my grandmother

Ming-Juan Liu

Acknowledgements

First of all I would like to express my gratitude to my advisor and chairman of my Dissertation Committee, Professor Richard Leahy, for his invaluable advice, guidance and encouragement during the course of my Ph.D. study. I am grateful too for the financial support which he makes available during my stay at SIPI. I'd also like to thank the other members of the committee, Professors Alexander Sawchuk, C.-C. Jay Kuo and William Navidi, for taking time to read, critique and evaluate my work.

The opportunity of pursuing a Ph.D. degree did not come easily for me. I am very thankful to two very special people, Dr. Roberto Canales, from the *Instituto de Investigaciones Eléctricas* of Mexico, and Dr. Nitza Lladó, former Basic Language Director in the Spanish and Portuguese Department at USC, who made it possible for me to come to and stay at USC.

The years of working at SIPI have been a very positive experience for me. I am very grateful for the help and support from the SIPI faculty, staff members and students. I'd like to thank Professor Rama Chellappa for having served as a member of my Guidance Committee and for the helpful suggestions to my research work. I appreciate the assistance of the SIPI staff members, Allen Weber, Toy Mayeda, Linda Varilla, Gloria Bullock, Delsa Tan and Mayitta Penoliar. Special thanks go to Dr. Allen Weber, who has rescued me in many occasions from problems of computer systems. The students at SIPI have been a wonderful group to share the day-to-day work experience with. I am especially thankful to Tom Hebert, Rae Lee, Xiao-Hong Yan, Sam Song, John Mosher, Ching-Yih Tseng, Anand Rangarajan, Qin-Fen Zheng, and Bangalore Manjunath for having made my life a little easier in a number of different ways.

This research work is supported by the Kaprielian Technology Innovation Fund and the Whittaker Foundation.

Contents

List of Figures

List of Tables

Abstract

This dissertation addresses the problems of image segmentation and pattern classification. Two new image segmentation schemes are proposed: hierarchical and graph theoretic. The later scheme is based on a novel graph theoretic approach to data clustering, also developed in this research work. These segmentation methods are applied to the problem of tissue classification of magnetic resonance (MR) images of the human brain.

The hierarchical segmentation scheme uses an unsupervised, region based approach. The images under study are modeled as a mosaic of homogeneous subimages where the pixel intensities in each subimage are characterized by a homogeneous Gauss-Markov random field (GMRF) with unknown parameters. The segmentation seeks to group the pixels into regions which, under a suitable hypothesis, are homogeneous. An analysis of homogeneity testing for GMRFs is presented and two tests are proposed: the hierarchical likelihood ratio test and the dispersion test. Based on the homogeneity tests a hierarchical segmentation approach is suggested. The image is represented by a quadtree, and a split and merge procedure is applied to find large homogeneous regions within the image. This is followed by a segmentation refinement step involving connected component labeling and region growing, in which most of the unclassified pixels will be attached to the neighboring regions to which they are most similar. The idea here is to allow the algorithm to learn about the image by first clustering the 'easy' pixels while delaying decisions on the 'hard' pixels until more information has been gathered. Finally, a clustering procedure, either a constrained maximum likelihood agglomerative approach or a graph theoretic approach, is used to reduce the number of segmented regions. One difficulty encountered in implementing the algorithm is evaluating the likelihood for irregularly shaped regions when the GMRF model is used. A new highly accurate approximation for the

determinant of the covariance matrix based on eigenanalysis is proposed to overcome this problem. This hierarchical scheme may be applied to segment either textured or non-textured images.

A novel graph theoretic approach for data clustering is developed using network flow theory. The data to be clustered are represented by an undirected adjacency graph $\mathcal{G}$ with arc capacities assigned to reflect the similarity between the linked vertices. Clustering is achieved by removing arcs of selected minimum cuts in $\mathcal{G}$ to form mutually exclusive subgraphs such that the largest inter-subgraph maximum flow is minimized. For graphs of moderate size ($\sim$ 1000 vertices), the optimal solution is obtained through partitioning a flow and cut equivalent tree of $\mathcal{G}$, which can be efficiently constructed using the Gomory-Hu algorithm. However for larger graphs this approach is impractical. New theorems for subgraph condensation are derived and are then used to develop a fast algorithm which hierarchically constructs and partitions a partially equivalent tree of much reduced size. This algorithm results in an optimal solution equivalent to that obtained by partitioning the complete equivalent tree and is able to handle very large graphs with several hundred thousand vertices. Using this clustering algorithm, an edge based image segmentation is developed. Each pixel of the image corresponds to a vertex of an undirected graph $\mathcal{G}$ and the segmentation is achieved by effectively searching for closed contours of edge elements (equivalent to minimum cuts in $\mathcal{G}$), which consist mostly of strong edges, while rejecting contours containing isolated strong edges. This method is able to accurately locate region boundaries and at the same time guarantees the formation of closed edge contours.

Chapter 1

Introduction

This dissertation addresses the problems of image segmentation and pattern classification. Two new image segmentation schemes are proposed: hierarchical and graph theoretic. The hierarchical scheme is an unsupervised, region based approach. It seeks to group the image pixels into regions which, under a suitable hypothesis, are homogeneous. Depending on the image model and the homogeneity criterion, this scheme may be applied to segment either fine-textured or non-textured images. The latter scheme is based on a novel graph theoretic approach to data clustering, also developed in this research work. The images are represented by an undirected graph and the segmentation is achieved by effectively searching for closed contours of edge elements. These segmentation methods are motivated by and then applied to the problem of tissue classification of magnetic resonance (MR) images of the human brain.

1.1 Motivation

MR imaging is a powerful non-invasive diagnostic tool which produces anatomical images of the human body based on the nuclear magnetic resonance properties of hydrogen nuclei [1], [2]. The properties may vary from tissue to tissue depending on a variety of factors such as water content, fat content, macromolecules and flow. Specifically, the parameters of interest for characterizing tissues are the proton density N(H), the spin-lattice relaxation time T1 and the spin-spin relaxation time

T2 after being excited by a radio-frequency (RF) magnetic field. By selecting a specific data acquisition mode, which is mainly controlled by the echo delay time TE and the pulse repetition interval TR, images with a wide range of contrast between different tissues may be produced. Commonly used imaging protocols are T2-weighted, proton density, T1-weighted and a combination of these using a multi spin-echo pulse sequence.

In order to cope with the enormous amounts of information contained in a 3-D image, there is growing interest in the utilization of computerized image analysis tools which are appropriate for the extraction of complex information from the data. For instance, a typical 3-D MR head scan produces an image sequence of 60 to 100 2-D cross-sectional slices which contain the 3-D information. In order to visualize 3-D anatomical structures, the physician must construct mentally the 3-D shape of the structures to be explored from the sequence of 2-D images. While the human visualization system is very effective at the interpretation of 2-D images, visualizing 3-D structures is a highly specialized task and requires intensive training. In complex situations, and if quantitative as well as qualitative results are desired, a pure interactive process by far exceeds practical time limits and must be supported by automated procedures. To exploit the full potential of 3-D medical imaging modalities, the generation of perspective 3-D surface display of anatomical structures has been a subject of great interest [3], [4], [5]. Volumetric display provides a direct and concise form of assisting the physician to visualize the internal structures of the body (in our case the brain). Used in conjunction with computer graphics tools, volume visualization can also be used for the planning of radiation therapy and the simulation of surgical procedures. In order to generate this kind of volumetric display, it is necessary to first segment the image into its constituent anatomical structures. The problem of MR image segmentation is the subject of this thesis. In addition to volume visualization, segmentation algorithms may also be involved in many other analysis applications such as morphological evaluation (*e.g.* volume of a brain tumor), registration of data from different imaging modalities, and computerized image interpretation [6], [7], [8].

There are several undesirable characteristics of MR images which make the problem of segmentation and tissue classification in MR images a difficult one.

A successful segmentation method has to take into account a large number of factors. Contrast, resolution, signal-to-noise ratio (SNR), scene complexity, slice thickness, and the homogeneity of the RF-coil are just a few of the important ones. One of the advantages of MR imaging is that the image contrast may be adjusted over a wide range by varying the data acquisition parameters TE and TR. But at the same time, this also represents a problem in terms of characterizing the soft tissue types using standardized parameters intrinsic to the tissues, because the pixel intensities, evaluated by a nonlinear function of the characteristic parameters such as T1, T2 and N(H) [9], are adjusted to have good contrast between tissues of interest. When multi-spectral MR data are used, *i.e.* several images are acquired for one anatomical slice using a multi-echo pulse sequence with different TE and TR, this problem still exists, though it is much less severe. Even if the intrinsic parameters T1, T2 and N(H) were extracted, their values for a specific tissue type may still change from patient to patient (*e.g.* fat content within the tissue may vary). Particularly for abnormal tissue types these parameters may vary over a wide range. In short, the parameters in MR images lack a common reference framework to which data sets from different patients and/or different acquisition modes can be normalized. As a result, if a supervised technique is used for segmenting MR images, it has to be trained for each individual data set in order to achieve satisfactory results. An even more disturbing problem associated with segmenting MR images is the radiometric variability [10]. Due to the fact that the RF-coil in a MR scanner is unable to generate a homogeneous RF magnetic field across the imaging volume, the image intensities may also vary as a function of the tissue's location within the image. Consequently, two different types of tissue may be indistinguishable in terms of the pixel intensity if they are located far apart from each other. While this is not a problem in visual analysis, provided the local contrast is sufficiently large, the variability of tissue intensity values with respect to pixel location can severely affect the segmentation result if the applied technique is based purely on global feature characterizations for different tissue types.

The partial volume effect in MR imaging also influences the selection of the segmentation approach. Each pixel of an MR images has a finite volume, and

pixels near tissue boundaries represent mixtures of different tissues. Since the pixel size is relatively large, the boundaries between tissues may not be clearly defined within the image. As a result, segmentation algorithms based on detecting edges using local differential operators can have difficulty locating edge contours in order to define tissue regions. On the other hand, edge detectors based on zero crossings [11], which guarantee the formation of closed edge contours, are tuned to a narrow range of spatial resolution. Consequently they are unable to locate the true tissue boundaries accurately, since they often have complex shapes due to the complex anatomy of the brain.

Accordingly, we have developed segmentation algorithms in this research which are compatible with these characteristics of MR images. Specifically the prospective algorithms should meet the following requirements: (a) the segmentation is based mostly on locally displayed feature differences rather than global properties governing the entire image; (b) it is ideally unsupervised and in practice must involve only minimum supervision; (c) it is able to accurately describe the complex anatomical structures of the human body. The first requirement constitutes a strong motivation for segmenting the image into connected regions(patches) so that local contrast displayed between adjacent tissue regions (neighboring patches) can be preserved.

1.2 Literature Review

Numerous techniques for image segmentation, some of these targeted specifically at the problem of segmentation and tissue classification in MR images, have been proposed in the literature. They generally fall into one of these two basic categories: region based and edge based.

Region based techniques seek to segment the image based on the statistics of regions corresponding to different tissue types. Algorithms based mainly on clustering pixels with similar features into regions can work well only for MR images with exceptional quality: high SNR and negligible radiometric variability effect [12], [13], [10]. Unfortunately most of the clinically available MR images, like those used in our experiments, do not meet this quality standard. To improve

the segmentation results, some researchers incorporate some spatial relationships between neighboring pixels into their clustering algorithms [14], [15]. A different but related approach is to model the image as a doubly stochastic process: a prior Gibbs distribution for the region formation, and a second random field for regions corresponding to each tissue type. The segmentation is then achieved by maximizing the *aposteriori* probability (MAP) [16], [17]. The drawbacks for this MAP formulation include the difficulty of finding a solution (in many cases only approximate locally optimal solution are obtained) and the possibility of excessive smoothing of tissue boundaries. Despite much improved performances, the above mentioned techniques share the same implicit assumption as the global intensity thresholding techniques: the pixels corresponding to the same type of soft tissue have identical feature characterization across the entire image. Their success depends very much on the validity of such a belief. Unfortunately for the MR images under our study, this assumption is difficult to justify due to radiometric variation. In addition, most of these techniques are supervised. Unsupervised methods are preferred for MR image segmentation for reasons discussed above. Another approach is to group similar neighboring pixels to form homogeneous regions. This approach is attractive for MR segmentation because it usually leads to unsupervised segmentation algorithms that can explore local contrast displayed between neighboring tissue types. Well known examples include the region growing and split-and-merge techniques [18], [19]. Others use a multi-resolution data structure such as a pyramid to extract the spatial relationships between adjacent pixels [20]. The main problems associated with existing algorithms are the loss of spatial resolution and noisy segmentation results.

On the other hand, edge based segmentation techniques do not rely heavily on the global properties of the image. Instead these explore locally exhibited intensity contrast, namely edges. They appear to be suitable for segmenting images that encode the most relevant information using local intensity contrast between neighboring pixels. Despite accurate edge localization, edge detectors based on local intensity differences (first order derivatives) have limited appeal for the purposes of MR image segmentation, due to the difficulty of linking edges to form closed contours. The class of edge detectors used in most of the MR

image segmentation algorithms are those based on the second order derivatives, namely detecting edges at zero crossings. Ehricke [21] and Bomans *et al.* [22] have developed MR segmentation algorithms based on region finding through the Marr-Hildreth operator. Their segmentation results for real MR image are among the most promising. However these techniques may still suffer from the drawbacks associated with the Marr-Hildreth operator, or all edge detectors based on zero crossings for that matter. While this operator guarantees the formation of regions with closed boundaries at zero crossings, these boundaries are often inaccurate. This is due to the fact that the Marr-Hildreth operator is tuned to a prespecified spatial resolution, and boundaries between objects with other spatial resolution may either be missed or inaccurately located. Recall that the MR images usually contain rather convolved boundaries, *e.g.* the cortical boundaries. Another very noticeable problem is that one must break connections between collections of pixels with distinct features for accurate segmentation and labeling. This problem is mainly due to an intrinsic drawback of techniques based on zero crossings. All closed contours have to be shared by exactly two connected regions and thus more complex structures, where three or more regions are connected, will be forced into at most 2 regions. Consequently, complex region formations in an image may be poorly encoded.

1.3 Overview

The new hierarchical segmentation scheme described in Chapters 2 and 3 uses an unsupervised, region based approach. The images under study are modeled as a mosaic of homogeneous subimages where the pixel intensities in each subimage are characterized as homogeneous Gauss-Markov random fields (GMRF) with unknown parameters. The GMRF image model used in the hierarchical segmentation is presented in Chapter 2. As in many of the segmentation techniques which use the MRF model, the problem of computing the partition function of the MRF for regions of irregular shape is encountered in our hierarchical segmentation approach. For the particular case of GMRFs, the challenge lies in evaluating the determinant of a large matrix. The computation of this partition function

is limited to very small irregular regions and a few regular shaped regions. A new highly accurate approximation is proposed in Chapter 2 for computing the partition function of a first order GMRF for arbitrarily shaped regions. The approximation is based on eigenanalysis of the inverse of the covariance matrix. Its evaluation is carried out as a summation over the pixel sites of the region, where each term depends only on the pixel site and its neighborhood. This form allows efficient computation when merging regions. Using this approximation, problems related to evaluating the likelihood function, estimating GMRF model parameters and other related computations are studied.

Using the image model of a mosaic of GMRFs, an unsupervised hierarchical segmentation scheme is developed in Chapter 3. The basis of the hierarchical segmentation is homogeneity testing for GMRF's. The algorithm seeks to group the pixels into regions which, under a suitable hypothesis, are homogeneous. There are two cases to consider: labeled data and unlabeled data. The labeled data case refers to the situation where the algorithm must decide whether to merge regions where the data are already grouped into homogeneous subsets. The likelihood ratio test is chosen here as the homogeneity criterion. Unfortunately when segmenting an image, the data are unlabeled. In other words, the data may be mixtures of samples from different populations, and a decision needs to be made about the occurrence of population mixtures. A statistical analysis on the performance of two commonly used approaches, testing homogeneity of unlabeled data based on the region/subregion similarity and the data dispersion, is presented. We also propose and evaluate a new homogeneity testing strategy named the hierarchical likelihood ratio test, which can be naturally implemented within a hierarchical data structure such as a quadtree. In our analysis we have found that tests based on region/subregion similarity usually have a low power of detecting inhomogeneity, regardless of the choice of the similarity criterion, due to the implicit but often invalid assumption that these subregions were homogeneous themselves. Consequently we propose to use the dispersion test and the hierarchical likelihood ratio test in our region based segmentation. Based on the analysis of homogeneity testing, a hierarchical segmentation approach is suggested in Chapter 3. The image is represented by a quadtree, and a split and merge procedure

is applied to find large homogeneous regions within the image. This is followed by a segmentation refinement step involving connected component labeling and region growing, in which most of the unclassified pixels will be attached to the neighboring regions to which they are most similar. The idea here is to allow the algorithm to learn about the image by first clustering the "easy" pixels while delaying decisions on the "hard" pixels until more information has been gathered. Finally, a clustering procedure, either a constrained maximum likelihood agglomerative approach or a graph theoretic approach, is used to reduce the number of segmented regions. This hierarchical scheme may be applied to segment either textured or non-textured images.

In Chapter 4, a novel graph theoretic approach for data clustering is developed using network flow theory. The data to be clustered are represented by an undirected adjacency graph $\mathcal{G}$: each vertex of $\mathcal{G}$ corresponds to a data point, and an arc links two vertices in $\mathcal{G}$ if the corresponding data points are neighbors according to a given neighborhood system. A flow capacity, assigned to each arc in $\mathcal{G}$, is chosen to reflect the feature similarity between the linked vertices. Clustering is then achieved by removing arcs of selected minimum cuts in $\mathcal{G}$ to form mutually exclusive subgraphs such that the largest inter-subgraph maximum flow is minimized, hence the similarity between the subgraphs (clusters) is also minimized. For graphs of moderate size ($\sim$ 1000 vertices), the optimal solution can be obtained through partitioning a flow and cut equivalent tree of $\mathcal{G}$, which can be efficiently constructed using the Gomory-Hu algorithm. However for larger graphs this approach is impractical. In order to overcome this problem, new theorems providing sufficient conditions for subgraph condensation are derived and are then used to develop a fast algorithm which hierarchically constructs and partitions a partially equivalent tree of much reduced size. The algorithm is based on the observation that most of the minimum cuts found in $\mathcal{G}$ are never used since their associated values (the value of a cut is defined as the capacity sum of its arcs) are sufficiently large that the arcs in those cuts will not be removed to form clusters. We show that many of the minimum cuts with large values can be identified using small local subgraphs, hence the vertices linked by them can be condensed *before* constructing the equivalent tree. This algorithm results in an optimal solution

equivalent to that obtained by partitioning the complete equivalent tree of $\mathcal{G}$ and is able to handle very large graphs with several hundred thousand vertices.

In Chapter 5 this new data clustering algorithm is applied to develop a graph theoretic image segmentation scheme, which achieves its segmentation goal by effectively seeking closed contours of strong edges. An adjacency graph $\mathcal{G}$ is constructed with each vertex in $\mathcal{G}$ corresponding to a pixel of the image, and an arc is placed between two vertices if their associated pixels are longitudinal or vertical neighbors of each other. The arc capacity is computed from an edge element which is defined as a separating element between the pixel pair linked by that arc: strong edges are mapped to small capacities and weak edges are mapped to large capacities. Here the strength of an edge element is computed using a local derivative operator. Our clustering algorithm is then applied to find minimum cuts in $\mathcal{G}$ with small value, or equivalently closed contours consisting mostly of strong edge elements of the image. This segmentation approach is able to accurately locate region boundaries, and at the same time guarantees the formation of closed edge contours. Both unsupervised and supervised versions of the algorithm are presented. In Chapter 5 we also show how the graph theoretic clustering algorithm can be used for patch clustering in the hierarchical segmentation scheme.

Experimental results, in which the two proposed image segmentation schemes are applied to tissue classification of magnetic resonance (MR) images of the human brain as well as to computer generated images, aerial photographs and nuclear medicine images are presented in Chapter 6. Chapter 7 draws conclusions on this research work and offers suggestions for further research.

1.4 Original Contributions

In this section, the original contributions of this research work are summarized as the following:

1. An unsupervised hierarchical segmentation scheme is developed for segmenting textured images. The images are modeled as a mosaic of "homogeneous" first order GMRFs, and the segmentation seeks to group the image pixels

into homogeneous regions. Based on homogeneity testing a three step segmentation scheme is developed : a split-and-merge procedure for finding large homogeneous regions, a patch forming for classifying boundary pixels and a final patch clustering to reduce the number of resulting patches.

2. A statistical analysis on testing homogeneity for GMRFs is presented. For the case of labeled data the likelihood ratio test is used; and for the unlabeled data case we propose to use the dispersion test and the new hierarchical likelihood ratio test. In particular, we have computed the power functions for the above two tests as well as the likelihood ratio test based on the similarity between the pixel region and arbitrarily partitioned subregions for the case of white Gaussian random fields. We have found that the commonly used region/subregion similarity approach usually has a low power of detecting inhomogeneity, while the other two tests have much better performance.

3. Accurate approximations have been proposed for computing the joint likelihood for first order GMRFs defined on irregularly shaped regions. This problem is equivalent to computing the partition function for Gibbs distributions for the case where the random field is Gaussian.

4. We present a novel graph theoretic approach for data clustering based on network flow theory. By representing the data by a properly constructed undirected adjacency graph $\mathcal{G}$, clustering is achieved by removing arcs corresponding to selected minimum cuts between vertex pairs in $\mathcal{G}$ to form mutually exclusive subgraphs. For an unconstrained optimal K-subgraph partition, the arcs selected for removal are those corresponding to the $K-1$ minimum cuts with the smallest $K-1$ cut values among all minimum cuts between every pair of vertices in $\mathcal{G}$. The resulting K-partition minimizes the largest inter-subgraph maximum flow among all possible K-partitions of $\mathcal{G}$. At the same time, this procedure also results in a nested sequence of partitions which are optimal for cluster numbers ranging from 2 to K, respectively. This is a desirable characteristic of a clustering technique, especially when the cluster number has to be determined from the data.

5. The minimum cuts of $\mathcal{G}$ are computed from a flow and cut equivalent tree $\mathcal{T}^*$ of $\mathcal{G}$, which can be constructed using the Gomory-Hu algorithm. Given $\mathcal{T}^*$, the optimal K-partition can be equivalently obtained by disconnecting the K$-$1 arcs in $\mathcal{T}^*$ with the smallest K$-$1 arc capacities. Based on the observation that most of the minimum cuts found in $\mathcal{T}^*$ are never used, since their values are sufficiently large that the arcs in those cuts will not be removed to form subgraphs, a fast hierarchical algorithm has been developed which constructs only a partially equivalent tree $\mathcal{T}_c^*$ of greatly reduced size. New theorems for subgraph condensation have been derived. Using these theorems, many of the minimum cuts with large values can be identified using local subgraphs, hence the vertices linked by them can be condensed *before* constructing the equivalent tree. Consequently the Gomory-Hu algorithm is applied to graphs of much smaller size without compromising the overall optimality of the clustering algorithm.

6. This graph theoretic clustering algorithm has been applied to the problem of image segmentation. Segmentation is achieved through seeking closed contours of edge elements of the image. By properly assigning arc capacities as a function of the computed edge strength, a minimum cut with small cut value, found in the adjacency graph formed by the image pixels, should contain mostly strong edges. We have observed that the proposed segmentation approach is able to accurately locate region boundaries while guaranteeing the formation of closed edge contours.

Chapter 2

GMRF Image Model and Related Computations

In this chapter, we describe the Gauss-Markov random field (GMRF) image model, which will be used in the next chapter for developing a hierarchical image segmentation algorithm. Markov random field models have been widely studied as a mathematical model for images [23], [24], [25], [26]. They have been used to solve a variety of problems in image processing such as image restoration and smoothing [27], [28], [29], [30], texture synthesis and classification [31], [32], [33], and image segmentation [34], [35], [36], [37].

As in most of the segmentation techniques which use the Markov random field model, the difficulty of computing the partition function, hence the likelihood function, for regions of irregular shape will be encountered in our hierarchical segmentation approach. For the particular case of GMRFs, the challenge lies in evaluating the determinant of a large matrix. Exact evaluation of the partition function is limited only to very small irregular regions and a few regular shaped regions such as a rectangular lattice. Here a new highly accurate approximation is proposed to compute the partition function for a first order GMRF. The approximation is based on eigenanalysis of the inverse of the covariance matrix. Its evaluation is carried out as a summation over the pixel sites of the region, where each term depends only on the pixel site and its neighborhood. This form allows efficient computation when merging regions. The approximation is also used for

GMRF parameter estimation.

2.1 Image Modeling Using GMRF

Let $\mathbf{X}$ denotes a real valued image defined on the image domain Z, usually a rectangular lattice. Let Z be composed by K exhaustive and mutually exclusive subsets $\{Z_k\}_{k=1}^K$, and X_k, the subimage defined on Z_k, be modeled as a Gauss-Markov random field (GMRF). The image $\mathbf{X}$ can then be described by its joint probability density function given by :

$$f(\mathbf{X}) = \prod_{k=1}^{K} f(X_k),$$

provided the X_k and X_j are independent for $k \neq j$. Based on this composite image model, the segmentation becomes a problem of fitting GMRF parameters and estimating the partition $\{Z_k\}$.

The GMRF model for X_k on Z_k is defined by the following joint probability density function [24] :

$$f(X_k) = \frac{|B_k|^{\frac{1}{2}}}{(2\pi\sigma_k^2)^{\frac{N_k}{2}}} e^{-\frac{1}{2\sigma_k^2}(X_k-\mu_k\cdot\underline{1})^T B_k (X_k-\mu_k\cdot\underline{1})} \tag{2.1}$$

where N_k = number of pixels on Z_k, $\underline{1}$ denotes an $N_k \times 1$ vector consisting of all 1's, and for a first order neighborhood

$$B_k(i,j) = \begin{cases} 1 & \text{if } j = i \\ -\beta_{1k} & \text{if } j \in N_H(i) \\ -\beta_{2k} & \text{if } j \in N_V(i) \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where $N_H(i)$ and $N_V(i)$ denote the sets of horizontal and vertical nearest neighbors of pixel i, respectively. Here β_{1k} and β_{2k} should be properly chosen so that B_k is positive definite. A sufficient condition for B to be positive definite is $|\beta_1| + |\beta_2| < 0.5$.

The GMRF X_k defined in (2.1) is introduced by Besag [24] as an approximation of a stationary spatial process observed on a finite window Z_k. Some alternative approximations are based on the spectral analysis of the underlying stationary process [38], [39]. However these are not easily tractable models for image segmentation. As an approximation X_k itself is only asymptotically stationary. Therefore when X_k is said to be "homogeneous" it should only be interpreted as a random field defined by the homogeneous parametric model given in (2.1) and (2.2). X_k, as defined in (2.1), has a free boundary. In other words, if X_k were viewed as a portion of an image on an infinite lattice, then $f(X_k)$ is identical to the conditional density given that all the pixels outside Z_k have been set to the mean, μ_k. Such a boundary assumption is preferable to others such as toroidal wrapping due to the likely shape irregularity of the image partition $\{Z_k\}$.

In this dissertation only a first order neighborhood and constant mean are considered. This choice is motivated in part by the need for computational tractability. But more importantly we are particularly interested in modeling MR images of the human brain where the variations within a region, corresponding to a single tissue type, generally exhibit very local correlation rather than large scale textural characteristics. Consequently this first order model should be appropriate in most cases. However we note that extensions to parameterized mean and higher order neighborhood are possible.

2.2 Likelihood Related Computations

For notational convenience the subscript k used in (2.1) and (2.2) will be dropped. Given the GMRF image model, the following operations are frequently performed when segmenting an image :

- Given an observation X on Z, compute the likelihood $f(X)$ for fixed parameter set $\theta = (\mu, \sigma^2, \beta_1, \beta_2)$.
- Find the maximum likelihood estimate (MLE) of θ given X.
- Efficiently update $f(X)$ and/or the MLE of θ when modifying Z.

Note that the evaluation of $|B|$, or equivalently $\ln|B|$, is involved in all of those operations. Unfortunately computing $\ln|B|$ is a costly operation, which becomes prohibitive when Z is a large irregularly shaped region. Evaluating $\ln|B|$ is equivalent to computing the partition function for Gibbs distributions [24], [26] for the case where the random field is Gaussian.

Exact evaluation of $\ln|B|$ is limited to small or certain regularly shaped regions. Ord [40] gives an analytic formula to compute $\ln|B|$ for a rectangular lattice with the free boundary assumption. Similar expression for the toroidal boundary condition is given by Besag and Moran [41]. Numerical approximations of the likelihood function for GMRFs have also been proposed in the literature, most within the context of parameter estimation. Whittle [38] computes $\ln|B|$ from the spectral density of the underlying stationary spatial process, which is observed only through a window Z. Due to the use of an asymptotic argument, only the number of pixels on Z is directly reflected in the approximation, which makes it insensitive to the shape of Z. Similar arguments are used in [39], [42]. Besag [24] also compares the parameter estimation results using his coding scheme against Whittle's method. His coding scheme is later generalized to the pseudo-likelihood function [43], [44] for more efficient parameter estimation. Although the pseudo-likelihood is not originally intended for likelihood approximation, it has been widely used for that purpose [34], [45]. The pseudo-likelihood is defined as the product of the conditional likelihoods for all pixels within the region, which can be interpreted as an approximation to the likelihood of the residual errors resulting from GMRF by assuming that the residuals are independent. However, the true likelihoods of the residuals is numerically different from the likelihood of the data even though both contain equivalent statistical information. Ripley [26] also mentions a simple approximation of $\ln|B|$ using Taylor expansion when discussing the fact that the likelihood function may not be a concave function of the GMRF parameters.

2.2.1 Approximate evaluation of the partition function

Grouping all pixels on Z into "$*$" and "$+$" classes, as shown in Fig 2.1, and using the property that the "$*$" pixels are independent of each other conditioned

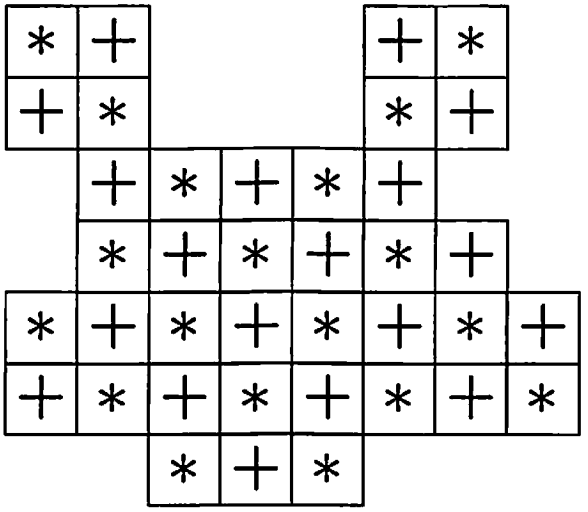


Figure 2.1: Indexing pattern for a first order GMRF on an irregular lattice

on the "+" pixels, and *vice versa* [24], B can be decomposed after reordering X as

$$B = \begin{bmatrix} I_* & B_{*+} \\ B_{+*} & I_+ \end{bmatrix} = \begin{bmatrix} I_* & B_{*+} \\ 0 & I_+ \end{bmatrix} \begin{bmatrix} I_* - B_{*+}B_{+*} & 0 \\ B_{*+} & I_+ \end{bmatrix}$$

Therefore defining $Q = B_{*+}B_{+*}$ one can show :

$$\ln|B| = \ln|I_* - Q| = \sum_{k=1}^{N_*} \ln(1 - \lambda_k) \tag{2.3}$$

where $0 \leq \lambda_k < 1$ are the eigenvalues of Q. It is impractical to compute the $\{\lambda\}$ for a large region Z, unless Z is a regular shape such as a rectangular lattice and the $\{\lambda_k\}$ can be evaluated analytically [40]. The proposed approximations for $\ln|B|$ are based on the sum and the square sum of the $\{\lambda_k\}$, whose computation is provided by the following theorem.

Theorem 2.1 *From the definitions of B and Q,*

$$\sum_{k=1}^{N_*} \lambda_k = tr(Q) = A_1\beta_1^2 + A_2\beta_2^2 \tag{2.4}$$

$$\sum_{k=1}^{N_*} \lambda_k^2 = tr(Q^2) = A_{11}\beta_1^4 + A_{22}\beta_2^4 + A_{12}\beta_1^2\beta_2^2 \tag{2.5}$$

Denote by $n_H(i)$, $n_V(i)$, $n_{H3}(i)$ and $n_{V3}(i)$ the number of first order horizontal, vertical and third order horizontal and vertical neighbors of site i in region Z, respectively; $N_2(i)$ the set of second order neighbors of i connected through Z and $n_C(i,j)$ the number of common (first order) neighbors between i and j. The constants in (2.4) and (2.5) are then computed as :

$$\begin{aligned} A_1 &= \sum_{i\in Z_*} n_H(i) \\ A_2 &= \sum_{i\in Z_*} n_V(i) \\ A_{11} &= \sum_{i\in Z_*} \{n_H^2(i) + n_{H3}(i)\} \\ A_{22} &= \sum_{i\in Z_*} \{n_V^2(i) + n_{V3}(i)\} \\ A_{12} &= \sum_{i\in Z_*} \{2\cdot n_H(i)\cdot n_V(i) + [\sum_{j\in N_2(i)} n_C(i,j)]^2\} \end{aligned} \tag{2.6}$$

Proof Let denote $N_H(i)$, $N_V(i)$, $N_2(i)$, N_{H3} and N_{V3} the (first order) horizontal and vertical neighbor sets, second order neighbor set, third order horizontal and vertical neighbor sets of a site i, respectively. Only neighboring pixels connected to i through Z are included in these neighbor sets. By definition, $B(i,k) = 0$ when i and k are not (first order) neighbors of each other. Hence

$$\begin{aligned} Q(i,j) &= \sum_{k\in Z_+} B(i,k)\cdot B(k,j) \\ &= \sum_{k\in N_H(i)} B(i,k)\cdot B(k,j) + \sum_{k\in N_V(i)} B(i,k)\cdot B(k,j) \\ &= -\beta_1 \sum_{k\in N_H(i)} B(k,j) - \beta_2 \sum_{k\in N_V(i)} B(k,j), \end{aligned}$$

By substituting $Q(i,j)$, we have

$$\begin{aligned} \sum_{i\in Z_*} \lambda_i &= \mathrm{tr}(Q) \\ &= \sum_{i\in Z_*} Q(i,i) \\ &= \sum_{i\in Z_*} \{\beta_1^2 \sum_{k\in N_H(i)} 1 + \beta_2^2 \sum_{k\in N_V(i)} 1\} \end{aligned}$$

$$
\begin{aligned}
&= \beta_1^2 \sum_{i \in Z_*} n_H(i) + \beta_2^2 \sum_{i \in Z_*} n_V(i) \\
&= A_1 \beta_1^2 + A_2 \beta_2^2 \\
\sum_{i \in Z_*} \lambda_i^2 &= \mathrm{tr}(Q^2) \\
&= \sum_{i \in Z_*} \sum_{j \in Z_*} Q^2(i,j) \\
&= \sum_{i \in Z_*} \{Q^2(i,i) + \sum_{j \in N_{H3}(i)} Q^2(i,j) + \sum_{j \in N_{V3}(i)} Q^2(i,j) + \sum_{j \in N_2(i)} Q^2(i,j)\} \\
&= \sum_{i \in Z_*} \{[\beta_1^2\, n_H(i) + \beta_2^2\, n_V(i)]^2 + \beta_1^4\, n_{H3}(i) + \beta_2^4\, n_{V3}(i) \\
&\qquad + \beta_1^2 \beta_2^2 \cdot [\sum_{j \in N_2(i)} n_C(i,j)]^2\} \\
&= \sum_{i \in Z_*} \{\beta_1^4 \cdot [n_H^2(i) + n_{H3}(i)] + \beta_2^4 \cdot [n_V^2(i) + n_{V3}(i)] \\
&\qquad + \beta_1^2 \beta_2^2 \cdot \{2\, n_H(i) n_V(i) + [\sum_{j \in N_2(i)} n_C(i,j)]^2\}\} \\
&= A_{11} \beta_1^4 + A_{22} \beta_2^4 + A_{12} \beta_1^2 \beta_2^2 \qquad \square
\end{aligned}
$$

Note that the constants (A_1, A_2, A_{11}, A_{22}, A_{12}) are shape descriptors of Z. In particular when Z is square, which is often encountered in a quadtree based hierarchical segmentation, the theorem simplifies to:

Corollary 2.1 *If Z is a $n \times n$ square array, then*

$$
\begin{aligned}
A_1 &= A_2 = n(n-1) \\
A_{11} &= A_{22} = n(3n-5) \qquad (n > 1) \\
A_{12} &= 12(n-1)
\end{aligned}
$$

The first approximation for $\ln |B|$ consists of finding a sequence $\{\hat{\lambda}_k\}$ "close" to $\{\lambda_k\}$ from which $\ln |B|$ is computed. Specifically, the following empirical model for the sequence $\{\hat{\lambda}_k\}$ was used :

$$
\hat{\lambda}_k = m_\lambda + C_k \cdot S_\lambda \qquad k = 1, \cdots, N_* \tag{2.7}
$$

where m_λ and S_λ^2 are the sample mean and variance of $\{\lambda_k\}$ computed from (2.4)

and (2.5), and

$$C_k = C_1 + (k-1)^p D,$$

where the constants C_1 and D are determined using $\sum_{k=1}^{N_*} C_k = 0$ and $\sum_{k=1}^{N_*} C_k^2 = N_*$, so that $\{\hat{\lambda}_k\}$ will have the same sum and square sum as $\{\lambda_k\}$. In our segmentation experiments $p = 3$ was chosen. The sequence $\{\hat{\lambda}_k\}$ is based on a heuristic model, the advantage of this model is that $\{C_k\}$ does not depend on θ, hence the estimation of θ becomes simpler when the approximate likelihood is used. In cases where any of the resulting $\hat{\lambda}_k$ fall outside the valid interval $[0, 1)$, a reduced length sequence $\hat{\lambda}_k$ should be recomputed by assigning some of the $\hat{\lambda}$'s to 0.

Repeated use of the above approximation may still be too time consuming. In this case a much faster but less accurate approximation may be used instead. From the Taylor expansion of $\ln|B|$ as given below,

$$\ln|B| = -\sum_{n=1}^{\infty} [\frac{1}{n} \sum_{k=1}^{N_*} \lambda_k^n]$$

a simple approximation for $\ln|B|$ is proposed

$$\begin{aligned} \ln|B| &\approx -\sum_{k=1}^{N_*} \lambda_k - \frac{1}{2}\sum_{k=1}^{N_*} \lambda_k^2 - \sum_{n=3}^{\infty} [\frac{N_*}{n} \cdot \rho^n] \\ &= -\sum_{k=1}^{N_*} \lambda_k - \frac{1}{2}\sum_{k=1}^{N_*} \lambda_k^2 + N_* \cdot [\ln(1-\rho) + \rho + \frac{1}{2}\rho^2] \end{aligned} \tag{2.8}$$

where

$$\rho = \frac{1}{N_*}(\sum_{k=1}^{N_*} \lambda_k + \frac{1}{2}\sum_{k=1}^{N_*} \lambda_k^2).$$

Note that the approximation consists of replacing the $\{\lambda_k\}$ by a constant ρ from the third term on in the Taylor expansion of $\ln|B|$. The two approximations described above are compared with the exact value of $\ln|B|$ for several regions and combinations of β_1 and β_2 in Appendix 2.A.

2.2.2 Sufficient statistics of θ

When segmenting images, the regions on which the likelihood needs to be evaluated often change progressively as the algorithm proceeds. In order to efficiently compute and update the likelihood $f(X)$ defined in (2.1), we keep track of a set of minimal sufficient statistics of θ. Expanding the exponent term of $f(X)$ gives

$$(X-\mu\cdot\underline{1})^T B(X-\mu\cdot\underline{1}) = X^T BX - 2\mu\cdot X^T B\underline{1} + \mu^2\cdot\underline{1}^T B\underline{1}.$$

Substituting B from (2.2) :

$$\begin{aligned} X^T BX &= \sum_{i\in Z} x_i^2 - (\sum_{i\in Z}\sum_{j\in N_H(i)} x_i x_j)\beta_1 - (\sum_{i\in Z}\sum_{j\in N_V(i)} x_i x_j)\beta_2 \\ X^T B\underline{1} &= \sum_{i\in Z} x_i - (\sum_{i\in Z} n_H(i)\cdot x_i)\beta_1 - (\sum_{i\in Z} n_V(i)\cdot x_i)\beta_2 \\ \underline{1}^T B\underline{1} &= N - (\sum_{i\in Z} n_H(i))\beta_1 - (\sum_{i\in Z} n_V(i))\beta_2 \\ &= N - 2A_1\beta_1 - 2A_2\beta_2 \end{aligned} \tag{2.9}$$

Hence the vector

$$(\sum_{i\in Z} x_i, \sum_{i\in Z} x_i^2, \sum_{i\in Z}\sum_{j\in N_H(i)} x_i x_j, \sum_{i\in Z}\sum_{j\in N_V(i)} x_i x_j, \sum_{i\in Z} n_H(i)\cdot x_i, \sum_{i\in Z} n_V(i)\cdot x_i)$$

is a sufficient statistics for θ. The first four terms are commonly expected statistics, the last two are closely related to the boundary shape of Z.

Observe that the sufficient statistics, as well as the shape descriptors defined earlier, are summations of local properties which depend on each individual pixel and its neighbors exclusively. Therefore their computation can be carried out in either recursive or parallel fashion, and consequently they are ideal for region split-and-merge or relaxation type of algorithms. This form of computing sufficient statistics should be more efficient than the hierarchical form proposed in [36].

2.2.3 MLE of parameter set θ

In this subsection, we will briefly look at the problem of MLE of parameter set θ for GMRFs, despite that our main objective of developing an accurate partition function approximation is the likelihood evaluation rather than the model parameter estimation.

For any given β_1 and β_2, the MLE of μ and σ^2 are easily computed [26] :

$$\hat{\mu} = \frac{X^T B \underline{1}}{\underline{1}^T B \underline{1}} \tag{2.10}$$

$$\hat{\sigma}^2 = \frac{(X - \hat{\mu} \cdot \underline{1})^T B (X - \hat{\mu} \cdot \underline{1})}{N} \tag{2.11}$$

It can be shown that $\hat{\mu}$ and $\frac{N\hat{\sigma}^2}{\sigma^2}$ are independent random variables with distributions $\mathcal{N}(\mu, \frac{\sigma^2}{\underline{1}^T B \underline{1}})$ and χ^2 with $N-1$ degree of freedom, respectively.

When β_1 and β_2 are unknown it is only practical to compute approximate MLEs of β_1 and β_2 for regions of irregular shape and then compute $\hat{\mu}$ and $\hat{\sigma}^2$ according to (2.10) and (2.11). The coding method [24] and the pseudo-likelihood method [43], [44] are the popular estimation techniques in the literature. These methods provide consistent estimates for β_1 and β_2 like the ML estimator, but with a larger variance than the MLE. Besag [46] has computed the asymptotic efficiency of the pseudolikelihood estimation for the first order isotropical GMRFs. For example, when $\beta_1 = \beta_2 = 0.2$ the asymptotic efficiency of the pseudolikelihood estimation is 0.709, which translates into a variance roughly 40% larger than the MLE. Alternatively, we may find an approximate MLE of β_1 and β_2 by using the proposed approximation of $|B|$ in place of the true value of $|B|$ for the joint likelihood given in (2.1). As our experimental results suggest, these estimates are numerically very close to the true ML estimates. Therefore we can expect that the approximate MLE also has a smaller variance, hence is more accurate, than the estimate resulting from the pseudo-likelihood method. A variance reduction is particularly desirable when the parameters have to be estimated from image regions of small to moderate size.

When the true β's approach to the boundary of their valid domain, in the asymptotic case $|\beta_1| + |\beta_2| \rightarrow 0.5$, maximizing either the approximate likelihood

or the pseudo-likelihood will likely result in invalid estimates $\hat{\beta}_1$ and $\hat{\beta}_2$ which lead to a non-positive definite matrix $\hat{B}$. To assure a positive definite $\hat{B}$, the maximization is constrained to the asymptotic boundary of $|\beta_1| + |\beta_2| < 0.5$ (this constraint may be too conservative for small regions). In our experiment the barrier method [47] was used to solve this constrained optimization problem, where a penalty term of $\alpha \ln(0.5 - |\beta_1| - |\beta_2|)$ attached to the log-likelihood is activated whenever a violation is detected. However the estimation accuracy may degrades severely. We believe that in this case the true ML estimate applied to an extracted rectangular subimage, whenever feasible, can result in more reliable estimates of the GMRF parameters.

We have also observed that the likelihood surface defined on (β_1, β_2) is quite smooth. Consequently we may sample the (β_1, β_2) domain and perform the maximization over the resulting finite set of (β_1, β_2). Since the likelihood function may be evaluated from sufficient statistics and the shape descriptors, once they are computed it requires little extra effort to evaluate the likelihood functions associated with each (β_1, β_2) of the sample set. This latter approach was preferred and implemented in the segmentation algorithm.

2.A Appendix: Accuracy Study for the Partition Function Approximation

In Table 2.1 we show the values of the partition function (defined as $-\frac{1}{2}\ln|B|$) for the eight patches displayed in Fig. 6.11 using various (β_1, β_2). These patches are extracted from an MR image of the human brain. For each (β_1, β_2) the values in the three columns are computed respectively by the exact method, the accurate approximation through $\{\hat{\lambda}_k\}$ and the fast approximation through Taylor expansion. For $\beta_1 = \beta_2 = 0.25$, which is near the boundary of the valid domain for $\ln|B|$, the error for the first approximation is small while the error for the second becomes large. As the (β_1, β_2) moves further from the boundary, the error of the second approximation reduces rapidly $(\beta_1 = 0.15, \beta_2 = 0.3)$ and eventually becomes negligible $(\beta_1 = 0.1, \beta_2 = 0.15)$. Note that the approximation becomes exact when $\beta_1 = \beta_2 = 0$. Similar levels of approximation accuracy for partition

	Size	$\beta_1 = 0.25, \beta_2 = 0.25$			$\beta_1 = 0.15, \beta_2 = 0.30$			$\beta_1 = 0.10, \beta_2 = 0.15$		
		(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
a	298	21.849	21.550	19.127	17.754	17.735	16.625	4.207	4.213	4.191
b	615	52.601	50.940	43.606	40.839	40.704	37.629	9.440	9.439	9.384
c	559	46.492	45.448	39.157	36.707	36.656	33.914	8.496	8.502	8.453
d	489	37.034	36.346	32.053	29.325	29.216	27.310	6.951	6.951	6.915
e	569	46.217	45.395	39.289	36.485	36.430	33.802	8.513	8.514	8.466
f	477	35.987	35.740	31.410	28.935	29.059	27.124	6.803	6.864	6.828
g	414	31.524	31.384	27.440	25.679	25.858	24.085	6.016	6.045	6.013
h	602	55.461	54.815	45.653	42.456	42.904	39.298	9.698	9.790	9.728

Table 2.1: Values of partition function for patches shown in Fig. 6.11 calculated using : (1) direct method, (2) accurate approximation, (3) fast approximation

Size	$\beta_1 = 0.25, \beta_2 = 0.25$			$\beta_1 = 0.15, \beta_2 = 0.30$			$\beta_1 = 0.10, \beta_2 = 0.15$		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
4 × 4	1.0208	1.0712	0.9291	0.8426	0.8641	0.8020	0.2064	0.2075	0.2055
8 × 8	5.3090	5.2747	4.5084	4.1646	4.1860	3.8601	0.9728	0.9740	0.9673
16 × 16	24.314	23.789	19.700	18.341	18.366	16.788	4.1889	4.1899	4.1623
32 × 32	104.42	101.75	82.222	76.817	76.931	69.899	17.354	17.354	17.237
64 × 64	433.47	422.51	335.84	314.27	314.93	285.14	70.612	70.613	70.128

Table 2.2: Values of partition function for square lattices calculated using : (1) exact method, (2) accurate approximation, (3) fast approximation

functions defined on square lattices are also observed (Table 2.2).

A comparative study is also carried out for GMRF parameter estimation. The parameters are estimated respectively by maximizing the true likelihood and the approximate likelihood. In our experiment, we generate two realizations of first order Gauss-Markov random fields with $(\beta_1, \beta_2) = (0.2, 0.2)$ and $(\beta_1, \beta_2) = (0.1, 0.15)$, respectively. In both cases, $\mu = 128$ and $\sigma = 32$ are used, and their true values are used for estimating β_1 and β_2. The random fields are defined on a 511 × 511 square lattice generated by applying a 2-D discrete sine transform to *i.i.d.* Gaussian random variables weighted by the eigenvalues of the matrix B^{-1}. This GMRF synthesis method is similar to the algorithm used in [32], and the eigenvalues of B^{-1} are from [40]. Square subimages of different sizes are then extracted from the realizations for estimating β_1 and β_2. Here the use

Size	$\beta_1 = 0.20, \beta_2 = 0.20$			$\beta_1 = 0.10, \beta_2 = 0.15$		
	(1)	(2)	(3)	(1)	(2)	(3)
8 x 8	0.244053	0.243762	0.306052	0.191802	0.191399	0.199992
	0.202476	0.203267	0.154937	0.167056	0.166789	0.147094
16 x 16	0.221357	0.221372	0.245706	0.093866	0.093970	0.086112
	0.178942	0.179546	0.170371	0.188620	0.188378	0.208411
32 x 32	0.201495	0.201684	0.212936	0.065614	0.065632	0.072296
	0.186200	0.186556	0.183712	0.134186	0.134147	0.148668
64 x 64	0.185070	0.185481	0.188647	0.099275	0.099298	0.098364
	0.210321	0.210412	0.208736	0.133404	0.133393	0.131875

Table 2.3: Estimated values of (β_1, β_2) for realizations of 1st order GMRFs defined on square lattices by maximizing: (1) the true likelihood, (2) the approximate likelihood, and (2) the pseudo-likelihood.

	Sample mean			Sample standard deviation		
	(1)	(2)	(3)	(1)	(2)	(3)
$\beta_1 = 0.2$	0.199446	0.1997910	0.200814	0.0231955	0.02292836	0.0255177
$\beta_2 = 0.2$	0.200204	0.2005310	0.201290	0.0233693	0.02309547	0.0260651

Table 2.4: Sample mean and standard deviation of the estimates of (β_1, β_2) using 1000 32×32 subimages extracted from realizations of GMRFs generated with $\beta_1 = \beta_2 = 0.2$ by maximizing: (1) the true likelihood, (2) the approximate likelihood, and (3) the pseudo-likelihood.

of square subimages is to make the true ML estimation of β's possible. From the estimates of β's shown in Table 2.3, we note that the estimates resulting from the approximate likelihood are remarkably close to the true ML estimates. For reference, we also show the pseudo-likelihood estimates of β's. However we emphasize that no estimation accuracy should be concluded from Table 2.3, because the estimates are merely realizations of random variables. In order to assess the accuracy of the parameter estimates, we need to compute their variance. Since the exact value for the variances is very difficult if not impossible to compute, we have estimated their values using the following experiment. For a given set of parameter (β_1, β_2), we repeat the above parameter estimation procedure for 1000 GMRF realizations, and then compute the sample mean and sample variance for the resulting estimates of β's. For example, Table 2.4 shows the sample means and standard deviations of the resulting estimates of (β_1, β_2) using 32×32 subimages for GMRFs generated with $\mu = 128$, $\sigma = 32$ and $\beta_1 = \beta_2 = 0.2$. Note that the variance for the pseudolikelihood method is over 20% larger than the true ML method and the approximate ML method, while all three result in sample means very close to the true (β_1, β_2). As $|\beta_1| + |\beta_2| \rightarrow 0.5$, we have observed that both the approximate ML and pseudo-likelihood methods quickly become ineffective. For example, for a random field generated using $(\beta_1, \beta_2) = (0.35, 0.15)$ both methods result in invalid estimates of β_1 and β_2 (using a 64×64 subimage). When constrained to $|\beta_1| + |\beta_2| < 0.5$, the approximate ML estimates are $(0.3215638, 0.1784353)$, while the true ML estimates are $(0.341119, 0.158308)$. In this case, the numerical difference between the two estimates becomes quite large.

Chapter 3

Hierarchical Image Segmentation Based on Homogeneity Testing

3.1 Introduction

In this chapter an unsupervised hierarchical image segmentation scheme is developed. The image under study is modeled as a mosaic of regions where the pixel intensities in each region are characterized by a homogeneous GMRF with unknown parameters.

As a popular image model, Markov random field (MRF) models have been used to solve a variety of problems in image processing. In addition to image segmentation, they have also been used for image restoration [27], [28], [29], [30] and texture synthesis and classification [31], [32], [33]. Among the numerous papers on textured image segmentation in the literature, there are two general approaches which are closely related to this work: Bayesian and "maximum likelihood".

Within the Bayesian framework, a doubly stochastic process model is assumed: a prior Gibbs distribution for the region formation, and a second MRF for the texture of each region [34], [37]. The segmentation may be achieved by maximizing the aposteriori probability (MAP) or the posterior marginal probability (MPM) [48]. MAP solutions have been found using either iterative relaxation [29], [35], [36], [17], dynamic programming applied iteratively to strips of the image [34], [49], and stochastic relaxation through simulated annealing [27], [37], [50].

Another approach is hierarchical "maximum likelihood" segmentation as proposed by Cohen and Cooper [35], [36]. In this case the image is modeled as a mosaic of GMRFs with known parameters. The image is partitioned into small square windows (*e.g.* 16×16) and each window is assumed to contain pixels of at most two classes. Each window is represented by a quadtree and segmented independently using a hierarchical approach. Starting from the root node, each block is divided into four sub-blocks and the best grouping of four is then obtained by maximizing their joint likelihood. This procedure is repeated recursively for each of the sub-blocks so that the image can be segmented from coarse to fine scale. Except at the root level, the likelihood conditioned on the resulting classification at a coarser level of the surrounding blocks should be used to achieve a good segmentation. Due to the restriction of at most two textures in each window, this hierarchical segmentation method may perform well only for images without fine details.

The techniques mentioned above are mostly supervised. However unsupervised extensions have been reported [50] or may be achieved by incorporating some unsupervised learning algorithms suggested in [45], [51].

This chapter is focused on developing an unsupervised segmentation scheme based on the GMRF model. The unsupervised approach is often preferable to a supervised method in situations such as tissue classification for MR images, where the parameters for a given type of tissue may vary from patient to patient and from machine to machine, and may even vary within the same image depending on its relative location. The images are modeled as a mosaic of "homogeneous" subimages, each characterized by a first order GMRF with unknown parameters. The choice of a first order neighborhood was motivated by the need for computational tractability, however the variations within a region of an MR image, corresponding to a single tissue type, generally exhibit very local correlation rather than large scale textural characteristics and consequently this model should be appropriate in most cases. A preliminary version of the algorithm was reported in [51], and a later version in [52]. This proposed algorithm has some parallels to the hierarchical ML methods in [35], [36], namely that the images are modeled as a mosaic of "homogeneous" GMRFs; the joint likelihood plays an important role

in the segmentation; and the images are segmented in a hierarchical fashion.

The basis of the segmentation scheme is homogeneity testing for GMRFs. There are two cases to consider: labeled data and unlabeled data. Labeled data case refers to the situation where the data are already grouped into homogeneous subsets. One such example is the problem of deciding whether two homogeneous regions merge together to form a larger homogeneous region. The likelihood ratio test [53], [54] is chosen here as the homogeneity criterion. Similar tests have been used by other researchers for the same purpose [45]. Unfortunately when segmenting an image, the data are unlabeled. In other words, the data may be mixtures of samples from different populations, and a decision needs to be made about the occurrence of population mixtures. One of the commonly used strategy for testing homogeneity of unlabeled data is based on region/subregion similarity [19], [18]: divide the data into a few groups and test the homogeneity based on the similarity between statistics computed from those data groups. Chen and Pavlidis [19] used this approach for detecting inhomogeneity in their split-and-merge segmentation algorithm, where the image is modeled as uniform regions corrupted by additive white noise. Generalizations for correlated data have also been reported [55], [45]. Although this approach appears to be reasonable, the performance is usually poor regardless of the choice of the homogeneity criterion. This is contributed mainly by an implicit but often invalid assumption that the arbitrarily partitioned subregions are homogeneous themselves. We have computed the power function for the case where the likelihood ratio test is applied for comparing means in partitioned subregions, and we have found that the test has a low power in average of detecting inhomogeneity. Following a statistical analysis on the performance of various homogeneity tests for unlabeled data, we propose to use two more powerful tests: the dispersion test and the hierarchical likelihood ratio test. For the special case of white GMRFs, the power function for these tests can be computed and comparisons are made.

The segmentation algorithm seeks to group the image pixels into regions which, under a suitable hypothesis, are homogeneous GMRFs. Following an analysis of homogeneity testing, a three step unsupervised hierarchical segmentation algorithm is proposed. At first a familiar split-and-merge procedure [19] is used to

find large homogeneous regions. Its purpose is to allow the algorithm to learn the statistics of the different classes based on these large regions. A segmentation refinement step is then introduced through connected component finding and region growing. Finally a step-wise constrained maximum likelihood agglomerative clustering procedure is used to merge the resulting regions to a smaller number. In all three steps homogeneity testing is involved, for either labeled or unlabeled data.

As in most of the segmentation techniques which use the MRF model, the problem of computing the partition function of the MRF for regions of irregular shape is encountered. This problem is overcome using the new highly accurate approximation described in Chapter 2. Since the approximation is computed as a summation over the pixel sites of the region, where each term depends only on the pixel site and its neighborhood, its value can be updated efficiently when splitting or merging regions. The approximation is also used for estimating GMRF parameters during the process of segmenting the image.

3.2 Testing Homogeneity for GMRFs

Let X be a subimage defined on an arbitrary region consisting of N pixels. In this section we develop tests for whether X is a realization of a homogeneous GMRF (the null hypothesis). Under the alternative hypothesis H_1, X is a mosaic of homogeneous GMRFs, *i.e.* $X = (X_1, X_2, \cdots, X_K)^T$ where K is the number of GMRFs in X and X_i denotes the N_i samples from the i-th GMRF parameterized by $\theta_i = (\mu_i, \sigma_i^2, \beta_{1i}, \beta_{2i})$. The null hypothesis H_0 may be more conveniently stated as $\theta_i = \theta$, $\forall i$ for X partitioned into K groups. Here only tests against the class of equal variance alternatives, *i.e.* $\sigma_i^2 = \sigma^2$, $\forall i$, are studied. This class excludes images in which the variance is the only distinguishable feature.

3.2.1 Homogeneity tests for labeled data

When the data are labeled, *i.e.* the partition $\{X_i\}$ of region X is given, testing homogeneity falls within the conventional framework of statistical hypotheses testing. The tests for labeled data described here will be used in the segmentation

algorithm to decide whether two homogeneous regions should be merged together to form a larger one. For the limited class of alternative hypotheses stated above, the likelihood ratio test is used.

Let $\hat{\mu}$, $\hat{\sigma}^2$ and $\hat{B}$ denote the MLE of μ, σ^2 and B under H_0, and μ_i^*, σ_i^{*2} and B_i^* denote the MLE of μ_i, σ^2 and B_i under H_1 for the i-th subimage. The MLE of B refers to the matrix constructed using the MLE of β_1 and β_2. Then the likelihood ratio test [53], [45] for rejecting H_0 can be simplified to :

$$\tilde{F}_K = \frac{N-K}{K-1} \cdot \frac{|\hat{B}|^{-\frac{1}{N}} \cdot \hat{\sigma}^2 - |B^*|^{-\frac{1}{N}} \cdot \sigma^{*2}}{|B^*|^{-\frac{1}{N}} \cdot \sigma^{*2}} > C \tag{3.1}$$

where K is the number of classes, N is the number of pixels in X,
$B^* = \text{diag}(B_1^*, B_2^*, \cdots, B_K^*)$, and

$$\sigma^{*2} = \frac{1}{N}\sum_{i=1}^{K} N_i \sigma_i^{*2} = \frac{1}{N}\sum_{i=1}^{K} (X_i - \mu_i^* \cdot \underline{1})^T B_i^* (X_i - \mu_i^* \cdot \underline{1})$$

In the case where the common variance σ^2 is known, the corresponding likelihood ratio test for rejecting the null hypothesis H_0 is

$$\tilde{W}_K = \frac{1}{\sigma^2}(N\hat{\sigma}^2 - \sum_{i=1}^{K} N_i \sigma_i^{*2}) + \ln|B^*| - \ln|\hat{B}| > C \tag{3.2}$$

In order to use the tests $\tilde{F}_K$ and $\tilde{W}_K$ defined in (3.1) and (3.2), we must set the threshold C for any given significance test level α, the probability of the false detection $P(H_1|H_0)$. Unfortunately both $\tilde{F}_K$ and $\tilde{W}_K$ are usually complicated functions of the data, hence approximations are often needed for setting the threshold C.

Let us rewrite $\hat{\sigma}^2$ as :

$$\begin{aligned}
\hat{\sigma}^2 &= \frac{1}{N}\sum_{i=1}^{K} (X_i - \hat{\mu} \cdot \underline{1})^T \hat{B}_i (X_i - \hat{\mu} \cdot \underline{1}) \\
&= \frac{1}{N}\sum_{i=1}^{K} [(X_i - \hat{\mu}_i \cdot \underline{1})^T \hat{B}_i (X_i - \hat{\mu}_i \cdot \underline{1}) + (\underline{1}^T \hat{B}_i \underline{1}) \cdot (\hat{\mu} - \hat{\mu}_i)^2]
\end{aligned}$$

where $\hat{B}_i$ is the i-th diagonal block of $\hat{B}$, and

$$\hat{\mu}_i = \frac{X_i^T \hat{B}_i \underline{1}}{\underline{1}^T \hat{B}_i \underline{1}}.$$

Due to the consistency property of the MLE of β_1 and β_2 [40], if all subimages $\{X_i\}$ are sufficiently large, estimates of the β's, based on X or individual X_i, should be approximately equal under the null hypothesis H_0, *i.e.* $\hat{B} \approx B^*$. Hence under H_0,

$$\tilde{F}_K \approx \frac{N-K}{K-1} \cdot \frac{\sum_{i=1}^K (\underline{1}^T B_i^* \underline{1}) \cdot (\mu^* - \mu_i^*)^2}{\sum_{i=1}^K (X_i - \mu_i^* \cdot \underline{1})^T B_i^* (X_i - \mu_i^* \cdot \underline{1})}.$$

Note that the right hand side is approximately an $\mathcal{F}_{K-1,N-K}$ distributed random variable [53], because its numerator and denominator become asymptotically independent. Similarly,

$$\tilde{W}_K \approx \frac{1}{\sigma^2} \sum_{i=1}^K [(\underline{1}^T \hat{B}_i \underline{1}) \cdot (\hat{\mu} - \hat{\mu}_i)^2]$$

which is approximately χ^2_{K-1} distributed.

Another possible approximation for setting the threshold C is to use the property that the log of the likelihood ratio is asymptotically proportional to a χ^2, under the null hypothesis H_0 [53], [45].

Due to the complexity of the tests, evaluating their performance is very difficult. However there is a special sub-class of GMRFs for which the power functions for both tests can be computed. Assume all K GMRF's are white, *i.e.* $\beta_{1i} = \beta_{2i} = 0 \ \forall \ i$, then the homogeneity testing as formulated becomes the well studied problem of mean comparison for normal populations [53], [54]. In terms of image processing, our image model becomes uniform regions (constant mean for each region) with Gaussian additive noise. It can be shown that $\tilde{F}_K$ and $\tilde{W}_K$ are non-central $\mathcal{F}_{K-1,N-K}$ and non-central χ^2_{K-1} distributed, respectively. For both tests the non-centrality parameter vanishes under the null hypothesis H_0. Note that the approximations used earlier for $\tilde{F}_K$ and $\tilde{W}_K$ now become exact. For the

white random field case :

$$\tilde{W}_K = \frac{1}{\sigma^2} \sum_{i=1}^{K} N_i \cdot (\bar{X} - \bar{X}_i)^2. \tag{3.3}$$

Here $\hat{\mu}$ and $\hat{\mu}_i$ are replaced by the sample means $\bar{X}$ and $\bar{X}_i$ respectively.

Given the probability distributions, the power of $\tilde{F}_K$ and $\tilde{W}_K$ can be computed for any fixed K and $\{N_i\}$. The power function of a test is defined as the probability of detecting inhomogeneity at a given significance level. In Fig. 3.1 the power function of $\tilde{W}_2$ (curve a) is plotted for $N_1 = 4$ and $N_2 = 12$, against the normalized mean difference ($= \frac{\mu_1 - \mu_2}{\sigma}$). Other curves in Fig. 3.1 are described in the next subsection. The power function of $\tilde{F}_2$ was also computed but not plotted since the difference between the two are negligible. This suggests that the power of $\tilde{W}$ has a low sensitivity towards the true value of variance, even for a relatively small sample size. In this two population case both tests, $\tilde{F}_2$ and $\tilde{W}_2$, are uniformly most powerful (UMP) for their respective hypotheses [53], [54].

3.2.2 Homogeneity tests for unlabeled data

The tests described in the last section are valid only for labeled data X. However X is often unlabeled, which makes homogeneity testing more difficult. Several tests will be proposed and their performance will be analyzed here. In order to make the analysis simpler, we assume that the common variance σ^2 is known throughout this section.

Homogeneity tests for white GMRF

We begin the analysis focused on the limited sub-class of white GMRF for two reasons. Firstly homogeneity testing for white GMRFs is important in its own right. It would provide a solution to the problem of resolving population mixtures for a given data sample. Also useful analytic results for homogeneity testing can be derived for white GMRF, which may prove useful for solving the more general problem.

One common solution for testing homogeneity of unlabeled data is the region/subregion similarity approach [19], [18] : divide region of interest X into

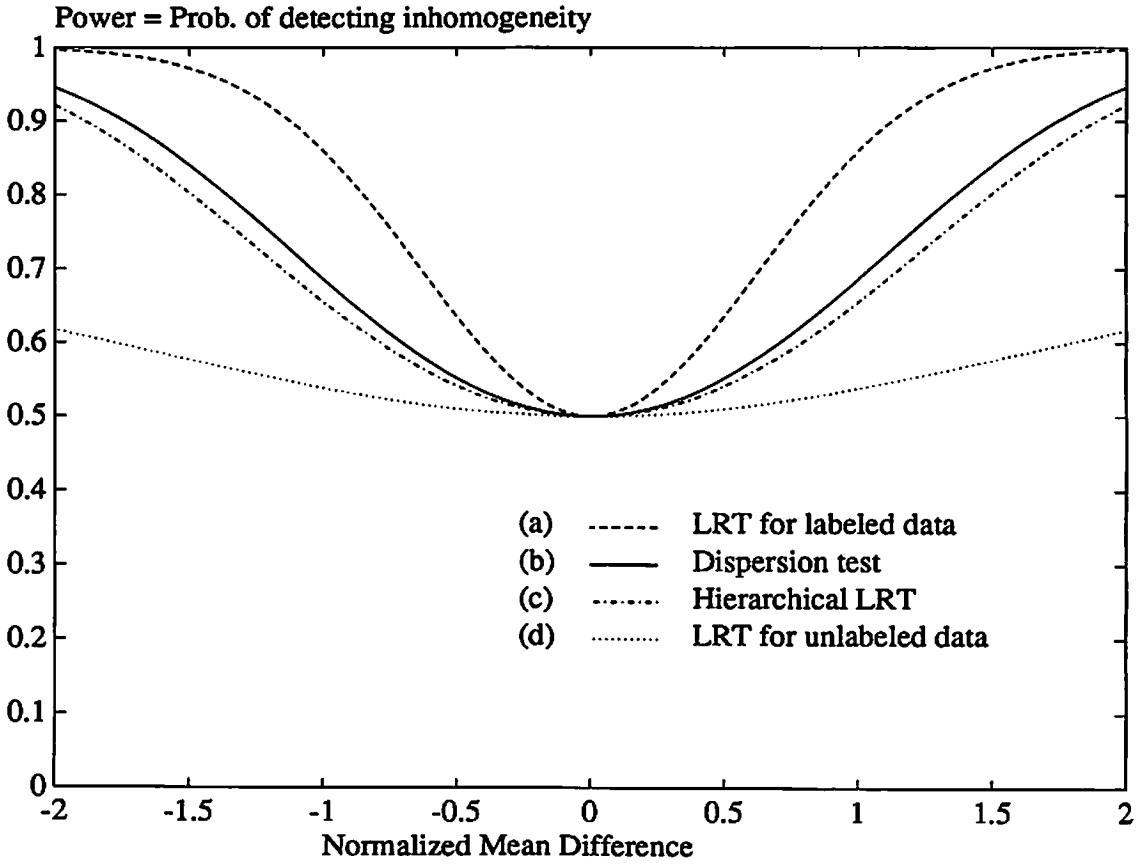


Figure 3.1: Example of power functions for (a) likelihood ratio test (LRT) for labeled data; (b) dispersion test; (c) hierarchical LRT; (d) LRT applied to unlabeled data with arbitrary partition

K' subregions $\{X_i'\}$, usually of equal size, then test the homogeneity of X based on the statistics of $\{X_i'\}$. Examples of homogeneity criteria include the $\tilde{W}_{K'}$ [45] and the maximum difference in the sample means of X and $\{X_i'\}$ [19]. The region/subregion similarity strategy is simple and seems reasonable, but homogeneity tests following this strategy may suffer from a poor performance. Note that there is always an implicit assumption that the $\{X_i'\}$ are homogeneous themselves, regardless of the choice of homogeneity criterion; such an assumption is often invalid. As a result, the power function can be greatly reduced.

Consider the case where $\tilde{W}_{K'}$ is used. For an arbitrary partition $\{X_i'\}$, $\tilde{W}_{K'}$ is a non-central $\chi^2_{K'-1}$ random variable, as for the case of labeled data, but often

with a much reduced non-centrality parameter. Let consider the example where $K = K' = 2$. Assume that the N pixels of X are partitioned into X_1' and X_2' with their respective sizes N_1' and N_2', then the critical region for rejecting the null hypothesis H_0 is

$$\tilde{W}_2 = \frac{N_1' N_2'}{N} \cdot \frac{(\bar{X}_1' - \bar{X}_2')^2}{\sigma^2} > C,$$

where $\bar{X}_i' = \frac{X_i'^T \mathbf{1}}{N_i'}$ is the sample mean of X_i'. Let $N_{i,j}'$ be the number of pixels in X_i' that are samples from the population j, then the centrality parameter of $\tilde{W}_2$ will be

$$\lambda_2' = \frac{N_1' N_2'}{N} \cdot (\frac{N_{1,1}' N_{2,2}' - N_{1,2}' N_{2,1}'}{N_1' N_2'})^2 \cdot (\mu_1 - \mu_2)^2,$$

which can be quite small for many choices of $N_{1,1}'$. Note that $N_{1,1}'$ uniquely determines $N_{1,2}'$, $N_{2,1}'$ and $N_{2,2}'$. Since $N_{1,1}'$ is unknown, the overall power of $\tilde{W}_{K'}$ has to be computed as the average of the power functions over all possible partitions $\{X_i'\}$ according to a prior distribution of $\{X_i'\}$. For the same two class example with $N_1 = 4$ and $N_2 = 12$ used in the previous section, we compute the power function with $N_1' = N_2' = 8$, at a significance level $\alpha = 0.5$. Split arbitrarily the 16 samples into two subsets of 8, and the least informative prior is used to compute the overall power of the $\tilde{W}_2$ test. In other words, we assume that all possible partitions of $\{X_i'\}$ are equally likely to occur without any spatial constraint. In this case the number of partitions with $N_{1,1}' = 0$, 1, 2, 3 and 4 samples from population 1 is equal to 70, 448, 784, 448 and 70, respectively. Note that the 4 samples in X_1 are most likely to be distributed evenly in both X_1' and X_2' ($N_{1,1}' = 2$), which results in a extremely low power of detecting the presence of population mean difference. As a result, the averaged power of the test is rather disappointing, as shown in Fig. 3.1 (curve d).

One remedy for increasing the test power is to make the subimages $\{X_i'\}$ smaller, so they have a higher probability of being homogeneous. Although more subsets must be tested, which contributes some power loss, the overall power is still improved. In the limit every subset has only one pixel, then the $\tilde{W}$ test becomes proportional to the sample variance of X, which has also been widely

used as a homogeneity criterion. Here we refer this special case of $\tilde{W}$ test as the dispersion test : reject H_0 if

$$\tilde{D}_N = \tilde{W}_N = \frac{N\hat{\sigma}^2}{\sigma^2} > C. \tag{3.4}$$

Despite its simple appearance, the $\tilde{D}_N$ test has a surprisingly good power. Figure 3.1 shows the power function of $\tilde{D}_N$ (curve b) when applied to the same two class problem, also at a significance level $\alpha = 0.5$.

A new and more elaborate approach for testing homogeneity of unlabeled data is also proposed and evaluated. This testing scheme is recursively defined: apply the dispersion test if the size of X is small; otherwise, divide X into K' groups and apply recursively the same testing scheme to all groups, then apply $\tilde{W}_{K'}$ to test the homogeneity among the groups only if all K' groups test homogeneous. This procedure is named here the hierarchical likelihood ratio test, and can be naturally implemented within a hierarchical data structure, *e.g.* a quadtree. In contrast to the dispersion test $\tilde{D}_N$, which actively avoids testing homogeneity between potentially inhomogeneous regions, the hierarchical likelihood ratio test seeks to distribute the total size of type I error, α, to all the involved tests. The $\tilde{W}_{K'}$ test used for unlabeled data can be viewed as a special case of the hierarchical test where all tests but one at root level have a zero significance level. By properly distributing the type I errors, the tests at the lower levels of the hierarchy can more effectively contribute to the overall power by detecting inhomogeneities occurred in smaller subregions.

In order to claim the proposed hierarchical test as a genuine hypothesis test, it is necessary to know how to set the involved thresholds so that the overall test has the desired significance level. The following theorem provides the solution to this problem.

Theorem 3.1 *Under either H_0 or H_1 all tests involved in the hierarchical likelihood ratio test scheme are statistically independent.*

In order to prove this theorem, let introduce the following lemma. The lemma is an extension to the well known result in statistics that the sample mean and the sample variance of i.i.d. normal random variables are independent. Its proof

is also similar following the proof of that result given in [53].

Lemma 3.1 *Let* $X = (X_1, X_2, \cdots, X_K)^T$ *be* K *independent random vectors, where* $X_i = (x_{i1}, x_{i2}, \cdots, x_{iN_i})^T$ *are* N_i *independent normal random variables with mean* μ_{ij} *and variance* σ^2*, then* $\bar{X}$ *and* $(\bar{X}_1 - \bar{X}, \bar{X}_2 - \bar{X}, \cdots, \bar{X}_K - \bar{X})$ *are statistically independent of each other, where the bar denotes the sample mean of the corresponding vector.*

Proof Write the joint characteristic function of the $K+1$ quantities involved:

$$\phi_{(\bar{X}, \bar{X}_1 - \bar{X}, \cdots, \bar{X}_K - \bar{X})}(t, t_1, \cdots, t_K) = \mathrm{E}\{\exp(j[t\bar{X} + \sum_{i=1}^{K} t_i(\bar{X}_i - \bar{X})]\}.$$

But observe that $\bar{X} = \sum \frac{N_i \bar{X}_i}{N}$ and define $\bar{t} = \frac{1}{N} \sum t_i$, hence

$$t\bar{X} + \sum_{i=1}^{K} t_i(\bar{X}_i - \bar{X}) = \sum_{i=1}^{K} [(\frac{t}{N} - \bar{t})N_i + t_i]\bar{X}_i = \sum_{i=1}^{K} a_i \bar{X}_i$$

which is a linear combination of independent, normal random variables $\{\bar{X}_i\}$ with coefficients

$$a_i = (\frac{t}{N} - \bar{t})N_i + t_i = \frac{N_i}{N}t + (t_i - N_i\bar{t})$$

having the properties

$$\sum_{i=1}^{K} \bar{\mu}_i a_i = \bar{\mu}t + \sum_{i=1}^{K} \bar{\mu}_i (t_i - N_i\bar{t})$$

and

$$\begin{aligned}
\sum_{i=1}^{K} \frac{a_i^2}{N_i} &= \sum_{i=1}^{K} \frac{1}{N_i} [\frac{t^2 N_i^2}{N^2} + \frac{2tN_i}{N}(t_i - N_i\bar{t}) + (t_i - N_i\bar{t})^2] \\
&= t^2 \sum_{i=1}^{K} \frac{N_i}{N^2} + 2t \sum_{i=1}^{K} \frac{t_i}{N} - 2t \cdot \bar{t} + \sum_{i=1}^{K} \frac{(t_i - N_i\bar{t})^2}{N_i} \\
&= \frac{t^2}{N} + \sum_{i=1}^{K} \frac{(t_i - N_i\bar{t})^2}{N_i}
\end{aligned}$$

The joint characteristic function of $\bar{X}$ and $(\bar{X}_1 - \bar{X}, \cdots, \bar{X}_K - \bar{X})$ can then be

written as

$$\begin{aligned}
\mathrm{E}[\exp(\sum_{i=1}^{K} j a_i \bar{X}_i)] &= \prod_{i=1}^{K} \exp(j\bar{\mu}_i a_i - \frac{\sigma_i^2}{2} a_i^2) \\
&= \exp(j \sum_{i=1}^{K} \bar{\mu}_i a_i - \frac{\sigma^2}{2} \sum_{i=1}^{K} \frac{a_i^2}{N_i}) \\
&= \exp[j\bar{\mu}t - \frac{\sigma^2 t^2}{2N}] \cdot \exp[j \sum_{i=1}^{K} \bar{\mu}_i (t_i - N_i \bar{t}) - \frac{\sigma^2}{2} \sum_{i=1}^{K} \frac{(t_i - N_i \bar{t})^2}{N_i}].
\end{aligned}$$

The last form of the joint characteristic function is separable in t and $(t_1, \cdots, t_N)$, therefore $\bar{X}$ and $(x_1 - \bar{X}, x_2 - \bar{X}, \cdots, x_N - \bar{X})$ are independent. □

Proof of Theorem 3.1 At an arbitrary point of the hierarchical testing scheme, the data X is divided into $(X_1, X_2, \cdots, X_K)$ and the test for homogeneity is

$$\tilde{W}_K = \frac{1}{\sigma^2} \sum_{i=1}^{K} N_i \cdot (\bar{X}_i - \bar{X})^2.$$

Since $\tilde{W}_K$ is exclusively a function of $\{\bar{X}_i - \bar{X}\}$, $\tilde{W}_K$ must be independent of $\bar{X}$ following lemma 3.1. Also note that the only statistic of X_i that $\tilde{W}_K$ uses is its sample mean $\bar{X}_i$. Similarly the test at one level above will only use the statistic $\bar{X}$ from the data set X. Therefore the test at one level above should be independent of $\tilde{W}_K$. By repeating the same argument, we show that all tests involved in the hierarchical scheme are independent. □

Using Theorem 3.1, the overall significance level of the hierarchical test can be computed from the significance levels of the individual tests involved in the testing scheme. It is, however, still an open question as to how to distribute the overall significance level within the hierarchical test so that the resulting test power is optimized. In our experiment, significance level are assigned empirically to all involved tests.

In order to illustrate how the hierarchical likelihood ratio test works, let apply a two-level hierarchical test to the same example of $(N_1 = 4, N_2 = 12)$, at a significance level of $\alpha = 0.5$. The 16 samples are arbitrarily split into 2 groups

of 8, and three tests are performed. The dispersion test $\tilde{D}_8$ is applied to each of the 2 sample groups. If both tests are successful, a third test $\tilde{W}_2$ is then applied to check the homogeneity between the 2 sample groups. In order to achieve an overall significance level of $\alpha = 0.5$, the significance levels for the three involved tests, α_1', α_2' and α_3', must satisfy the following:

$$1 - \alpha = (1 - \alpha_1') \cdot (1 - \alpha_2') \cdot (1 - \alpha_3').$$

In our experiment we use $(1-\alpha_1') = (1-\alpha_2') = (1-\alpha_3') = \sqrt[3]{1-0.5}$. Similarly the power function of the hierarchical test, π, is computed from the power functions of the 3 involved tests:

$$\pi = 1 - (1 - \pi_1') \cdot (1 - \pi_2') \cdot (1 - \pi_3').$$

The power function of the hierarchical test is computed for the least informative prior distributions of the partitions $\{X_i'\}$, and shown in Fig. 3.1 (curve c). One can observe that the hierarchical test is much more powerful than the $\tilde{W}$ test applied to unlabeled data, but slightly less powerful than the dispersion test. This result could be considered encouraging, since the power is computed for the least informative prior. In contrast, the dispersion test makes no use of a prior. Another advantage of using the hierarchical test is the possibility of detecting homogeneous portions even if the overall test fails.

An example of detecting homogeneous subimages is presented in Appendix 3.A.

Homogeneity tests for general unlabeled GMRF

When local interactions among neighboring pixels are also considered, the homogeneity issue becomes more complex. In addition to analytic complexity, there are important model defects of which we need to be aware. Consider two contiguous subimages, X_1 and X_2. According to our mosaic image model, they are uncorrelated when H_0 is rejected, but they become correlated if H_0 should be accepted. One way to avoid dealing with such an undesirable discontinuity is not to use inter-region statistics when testing homogeneity, as in the case of the hierarchical likelihood ratio test. Another defect is caused by the use of the GMRF model

itself as an approximation to a stationary spatial process. If observations are added or removed the resulting model is not compatible with the original model in the sense that the same pixels will have different characterization within the two models.

Based on the result obtained for white GMRFs (Fig. 3.1) the use of tests whose decision are based on statistics computed from arbitrarily formed sub-regions should be avoided due to their low power of detecting inhomogeneity. Therefore only the dispersion test and the hierarchical likelihood ratio test will be used for testing homogeneity.

Due to the difficulty of obtaining analytical results, these tests are only presented as generalizations of their counterparts in the white GMRF case. The dispersion test is defined as a scaled MLE of the variance σ^2, *i.e.* reject H_0 if

$$\tilde{D}_N = \frac{N\hat{\sigma}^2}{\sigma^2} > C. \tag{3.5}$$

Here $\hat{\sigma}^2$ is computed according to (2.11). Under the null hypothesis H_0, $\tilde{D}$ is approximately χ^2. Consequently C can be approximately computed for a given significance test level.

The hierarchical likelihood ratio test is defined in a similar manner as for the white GMRF case, except the test $\tilde{W}$ in (3.2) should be used in place of (3.3. For the general case, the thresholds for the tests involved in the hierarchical scheme can only be set approximately for a given significance level.

3.3 Hierarchical Image Segmentation Algorithm

The hierarchical image segmentation algorithm can be divided into three major components: split-and-merge learning, patch forming and patch clustering.

3.3.1 Split-and-merge learning

The main goal of the split-and-merge procedure is to allow the algorithm to learn and to be trained through hierarchically finding large homogeneous regions within the image. This is the most crucial part of our unsupervised segmentation scheme.

Let the image be represented by a quadtree with the top of the tree corresponding to the entire image, the next level to the four quadrants of the image and so on [56]. Starting from the top of the tree, we test homogeneity for the region corresponding to each node using either the dispersion test or the hierarchical test. The node is then labeled accordingly: if the region is homogeneous a label is attached together with the estimated GMRF parameters to that node and the tree is terminated at this node; if the region is inhomogeneous the node remains unlabeled and the same node labeling strategy is applied to its four quadrants. Furthermore, each newly labeled region is tested for possible merging with previously formed homogeneous regions. This allows larger homogeneous regions to be formed gradually, so that the GMRF parameter estimates become more and more reliable. Here the likelihood ratio test $\tilde{W}_2$ should be used. As the node size becomes small, the parameters estimated from large homogeneous regions can provide supervision for deciding whether to merge the node to bigger regions. In this way the algorithm can be "trained" by these early large homogeneous regions for later classifying the remaining "hard" (small) regions. This split-and-merge mechanism is illustrated in figure 3.2. In completion, the split-and-merge step produces an image partition.

When testing homogeneity two kinds of errors may occur : a region is labeled inhomogeneous while H_0 is true (type I error) or a region is labeled homogeneous while H_0 is false (type II error). The type II error is related to how accurately the algorithm can be trained. Once committed, this error is unlikely to be corrected. On the other hand, the type I error controls the rate of training for the algorithm and is often self-correctable. In addition a step-wise ML agglomerative clustering can also be applied to reduce the type I error after the split-and-merge. A large number of type I errors may however inhibit the early formation of large homogeneous regions, resulting in a very noisy segmentation. Therefore a balance between the two types of error is needed. A general rule is to keep the probability of type II error small while testing large regions, then gradually increase this probability as the region size becomes small.

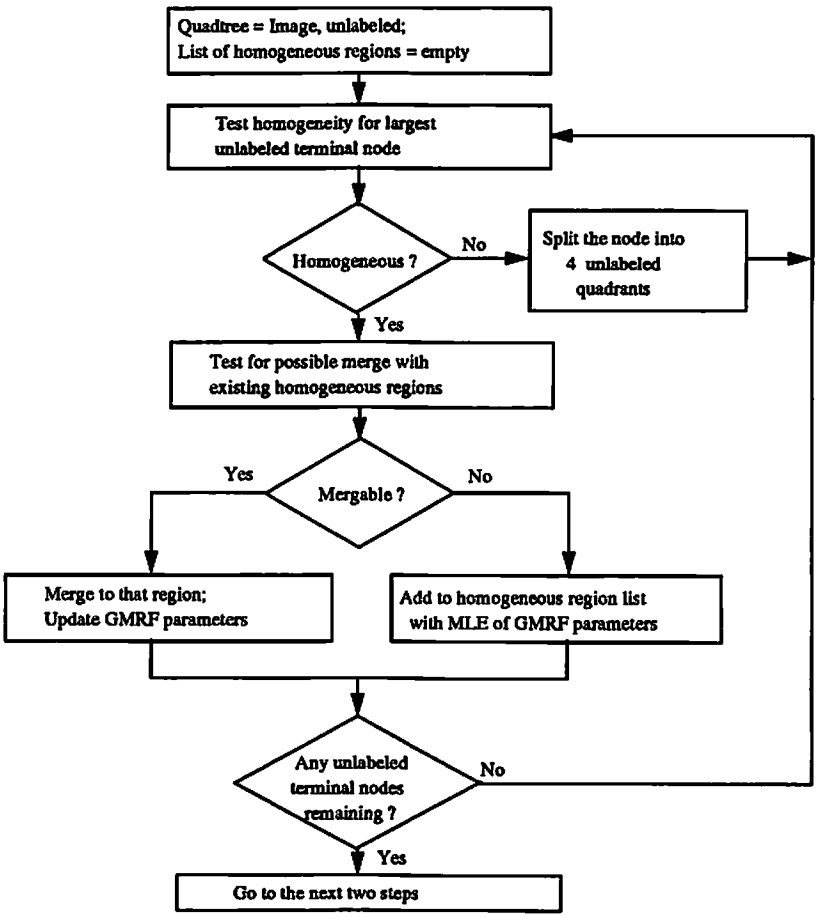


Figure 3.2: Block diagram for split-and-merge learning

3.3.2 Patch forming

The resulting segmentation from the split-and-merge is often noisy. This is not surprising since no smoothness constraint is imposed. Each segmented region (set of pixels assigned to the same label) may have many isolated subregions, most of which may contain very few pixels. Hence it is natural to think of breaking down each segmented region into connected subregions (patches) in order to get a smoother segmentation.

The patch forming algorithm consists of the following steps :

1. Patch labeling: For each resulting region from the split-and-merge, find all its spatially connected components (patches), and label them accordingly : a different label is assigned to each patch except those with very few pixels

($\leq$ 4 pixels in our experiment) which are left unlabeled.

2. Patch growing: Among all pairs of unlabeled patches and their neighboring labeled patches, select the one with the smallest mean difference and expand the labeled patch by attaching the unlabeled one to it. Repeat this step until the smallest mean difference exceeds a given threshold or no unlabeled patch is left.

In the above algorithm, the patch labeling is carried out using a modified version of the algorithm in [57].

3.3.3 Patch clustering

The number of resulting homogeneous patches resulting from the patch forming step is usually too large for the segmentation purpose. A clustering procedure is therefore applied to methodically group the patches together, thus reduces the number of labels. Two methods are considered in this dissertation: constrained agglomerative clustering and optimal graph theoretic clustering. The later approach will be described in Chapter 5.

Constrained ML agglomerative patch clustering

The constrained maximum likelihood agglomerative patch clustering is performed as follows :

> Among all pairs of labeled neighboring patches, select the one which when merged produces the smallest reduction in their joint likelihood while satisfying certain regularization constraints. A new patch is then formed by merging these two patches. The procedure is repeated until a stopping condition is met.

This clustering is a generalization of the clustering method described by Duda and Hart [58], in the sense that regularization constraints are imposed and the clustering is performed on sets of correlated pixels (patches).

When implemented the following specifications are used :

1. In addition to connectivity only one other regularizing constraint is used: the maximum gradient on the boundary between two candidate patches should not exceed a given threshold. Here the gradient is computed using only pixels that lie in the union of the candidate patches.

2. The stopping condition for clustering is that no pair of patches can be found which give a smaller likelihood reduction than a pre-set threshold while meeting the regularizing constraints.

3. A common variance is used for computing the likelihood for all patches. This variance is estimated as a weighted sum of ML variance estimates of the labeled patches.

When merging patches, the criterion of maximum likelihood seeks to put together patches with similar GMRF features, while the connectivity and edge constraints tend to limit the extension of resulting patches and to preserve the local contrast displayed between neighboring patches. The patch clustering is only stepwise optimal. After patch clustering, most of the resulting patches are large and distinct from their surrounding patches. Therefore when applied to the problem of tissue classification of MR images, the final tissue labeling of the patches can be easily performed through an interactive patch editing.

3.A Appendix: An Example of Detecting Homogeneous Subimages

In order to obtain a more intuitive understanding of homogeneity testing for unlabeled data, we illustrate the power of previously introduced tests through the detection of homogeneous subimages, which is an important step in slit-and-merge mechanism. Fig. 3.3.a shows a 4 region computer generated image of 128×128 pixels with 256 grey levels. The intensities for the 4 uniform regions are 80, 112, 144 and 176, respectively. There are 154 homogeneous blocks of 8×8 in Fig. 3.3.a, which are shown in white in Fig. 3.3.b. The image was then corrupted by Gaussian additive noise with a standard deviation $\sigma = 16$ (Fig. 3.3.c). Three

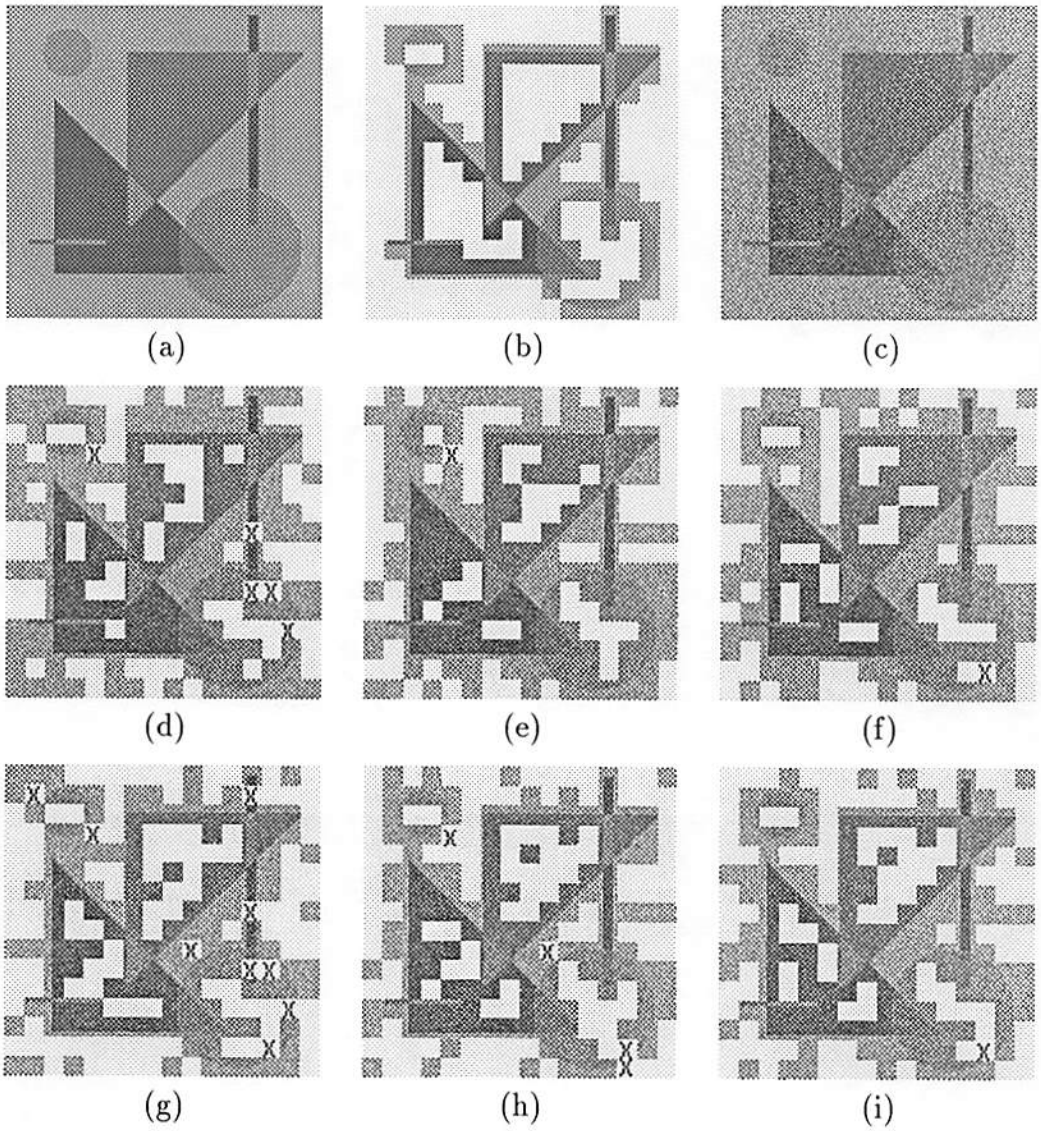


Figure 3.3: Detection of homogeneous subimages for a 4 region hand-draw image: (a) noise free image; (b) 154 homogeneous 8×8 blocks in (a) displayed in white ; (c) noisy image with $\sigma = 16$; (d) 82 blocks detected as homogeneous using the $\tilde{W}_4$ test at $\alpha = 0.5$, 5 erroneous blocks; (e) 82 blocks detected as homogeneous using the $\tilde{D}_{64}$ test at $\alpha = 0.5$, 1 erroneous block; (f) 81 blocks detected as homogeneous using the hierarchical test at $\alpha = 0.5$, 1 erroneous block; (g) 119 blocks detected as homogeneous using the $\tilde{W}_4$ test at $\alpha = 0.3$, 9 erroneous blocks; (h) 118 blocks detected as homogeneous using the $\tilde{D}_{64}$ test at $\alpha = 0.3$, 4 erroneous blocks; (i) 106 blocks detected as homogeneous using the hierarchical test at $\alpha = 0.3$, 1 erroneous block.

homogeneity tests, namely the likelihood ratio test $\tilde{W}_4$, the dispersion test $\tilde{D}_{64}$ and the hierarchical likelihood ratio test, are used to test the homogeneity of 8×8 blocks in the noisy image. For both the $\tilde{W}_4$ test and the hierarchical test, the 8×8 blocks are partitioned into 4 sub-blocks of 4×4. At a significance level $\alpha = 0.5$, the $\tilde{W}_4$ test detects 82 blocks as homogeneous with 5 mistakes (Fig. 3.3.d); the $\tilde{D}_{64}$ test also detects 82 blocks as homogeneous but with only 1 mistake (Fig. 3.3.e); the hierarchical test detects 81 blocks as homogeneous also with 1 mistake (Fig. 3.3.e). Within the hierarchical test, all 5 involved tests are set at a same significance level of $1 - \sqrt[5]{1 - 0.5} = 0.12945$. At $\alpha = 0.5$, we only expect to detect about half of the 154 homogeneous blocks, which is consistent with the experimental results shown above. When the test significance level is reduced, the number of detected homogeneous blocks is increased but so is the number of mistakes. At $\alpha = 0.3$, the $\tilde{W}_4$ test, the $\tilde{D}_{64}$ test and the hierarchical test respectively detect 119 (9 mistakes), 118 (4 mistakes) and 106 blocks as homogeneous with 1 mistake (Figs. 3.3.g-i). Although the results in Fig. 3.3 are not statistically sufficient to draw significant conclusions, we have observed from numerous experiments that the hierarchical test performed more favorably in general than the dispersion test in detecting small inhomogeneities within large subimages.

Chapter 4

Graph Theoretic Clustering

4.1 Introduction

Data clustering is an important methodology in exploratory data analysis. Numerous clustering methods have been reported in the literature (*e.g.* see [59], [58]). Given M data points, $X = \{x_1, x_2, \cdots, x_M\}$, the objective of clustering is to partition the data set into K non-empty subsets such that *alike* data are grouped together and data in different subsets or clusters are not *alike*. In this chapter we propose a new graph theoretic technique for data clustering, and then demonstrate its application to the problem of image segmentation in Chapter 5.

Many graph theoretic techniques have been proposed for cluster analysis. Commonly known techniques include: (i) single-link and multi-link hierarchical algorithms formulated and implemented using a threshold graph [60], [61]; (ii) forming clusters by breaking inconsistent arcs in the minimum spanning tree of the proximity graph [62] or graphs constructed based on limited neighborhood sets [63], and (iii) detecting clusters using directed trees [64]. The clustering technique presented in this dissertation is based on network flow theory. Here minimum cuts in an undirected adjacency graph are used for partitioning the data into clusters. This idea was first proposed in our previous paper [65], and was further developed in [66], [67].

The data to be clustered are represented by an undirected adjacency graph $\mathcal{G}$: each vertex of $\mathcal{G}$ corresponds to a data point, and an arc links two vertices in $\mathcal{G}$

if the corresponding data points are neighbors according to a given neighborhood system. A flow capacity is then assigned to each arc in $\mathcal{G}$. This flow capacity is chosen to reflect the feature similarity between the pair of linked vertices. The clustering is achieved by removing arcs of $\mathcal{G}$ to form mutually exclusive subgraphs. For the case of an unconstrained optimal K-subgraph partition of $\mathcal{G}$, the arcs selected for removal are those in a set of K$-$1 minimum cuts with the smallest K$-$1 values among all possible minimum cuts separating all pairs of vertices.

This new clustering strategy possesses some desirable properties. Unlike many other clustering techniques proposed in the literature, the resulting K-subgraph partition is a globally optimal K-partition of the adjacency graph $\mathcal{G}$. It minimizes the largest inter-subgraph maximum flow among all possible K-partitions of $\mathcal{G}$, hence minimizing the similarity between the subgraphs (clusters). The difficulty in reaching a globally optimal solution for a particular partitional clustering technique arises from the requirement that a huge number of possible K-partitions must be considered. Locally optimal solutions are often found instead using iterative, hill-climbing techniques. Attempts have been made to identify and reject large numbers of obviously non-optimal partitions, using techniques such as dynamic programming [68], branch-and-bound [69], and conditional clustering [70]. Despite a substantial reduction in the number of partitions that need to be evaluated, these techniques are computationally infeasible even for problems of moderate size. In contrast, our clustering strategy can be efficiently implemented to handle very large graphs with several hundred thousand vertices. In addition, this clustering technique does not only produce an optimal K-subgraph partition, but also a nested sequence of partitions which are optimal for cluster numbers ranging from 2 to K. This is especially attractive when the cluster number has to be determined from the data.

In order to make this new strategy work as a data clustering method, two important issues need to be considered: (i) finding an efficient implementation scheme to make the clustering technique practical, and (ii) constructing an adjacency graph $\mathcal{G}$ which can produce meaningful clusters. The minimum cuts of the undirected graph $\mathcal{G}$ can be computed from a flow and cut equivalent tree $\mathcal{T}^*$ of $\mathcal{G}$, which is constructed using the Gomory-Hu algorithm which was originally

developed for solving the multi-terminal maximum flow problem for undirected graphs [71]. The Gomory-Hu algorithm involves the successive solution of exactly M−1 maximum flow problems with M being the number of vertices in $\mathcal{G}$. Once $\mathcal{T}^*$ has been computed, the optimal K-partition of $\mathcal{G}$ can be equivalently obtained by simply disconnecting the K−1 arcs in $\mathcal{T}^*$ with the K−1 smallest arc capacities. This direct implementation is acceptable for graphs of moderate size ($\sim$ 1000). However, it quickly becomes impractical as the the size of $\mathcal{G}$ increases, due to the polynomial complexity of the algorithm. In order to overcome this problem, a fast hierarchical algorithm is developed that requires construction and partition of a partially equivalent tree $\mathcal{T}_c^*$ of greatly reduced size. This still results in an optimal solution equivalent to that obtained by partitioning the complete equivalent tree of $\mathcal{G}$. The algorithm is based on the observation that most of the minimum cuts found in $\mathcal{G}$ are never used since their associated values (the value of a cut is defined as the capacity sum of its arcs) are sufficiently large that the arcs in those cuts will not be removed to form subgraphs. New theorems providing sufficient conditions for subgraph condensation are derived. We show that many of the minimum cuts with large value can be identified using small local subgraphs, hence the vertices linked by them can be condensed *before* constructing the equivalent tree. Consequently the Gomory-Hu algorithm is applied only to graphs of much smaller size, but without compromising the overall optimality of the clustering algorithm. As a result the new clustering technique can be applied to partition very large graphs.

The problem of properly constructing an adjacency graph is highly application dependent and involves the appropriate selection of a neighborhood system and an arc capacity function. We will address the problem in the context of image segmentation in Chapter 5.

4.2 Review of Network Flow Theory

The problems in network flow theory which particularly interest us are (i) finding the maximum flow from one vertex to another and (ii) finding the maximum flow between every pair of vertices. The methods for solving these problems provide

the foundation of our graph theoretic algorithm for data clustering.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a graph with vertex set $\mathcal{V} = \{v_1, v_2, \cdots, v_M\}$ and arc set $\mathcal{A} = \{a_{ij}\}$ with a_{ij} denoting the arc between vertices v_i and v_j with capacity c_{ij}. A flow from an arbitrary vertex s to a vertex t with value F_{st} is defined by a set of numbers f_{ij} assigned to all $a_{ij} \in \mathcal{A}$ which satisfy the following conditions:

$$\sum_{v_j \in \Gamma(v_i)} f_{ij} - \sum_{v_k \in \Gamma^{-1}(v_i)} f_{ki} = \begin{cases} F_{st} & \text{if } v_i = s \\ -F_{st} & \text{if } v_i = t \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

and $f_{ij} \leq c_{ij}$ for all $a_{ij} \in \mathcal{A}$. In (4.1), F_{st} denotes the value of the net flow from s to t, $\Gamma(v_i) = \{v_j | a_{ij} \in \mathcal{A}\}$ and $\Gamma^{-1}(v_i) = \{v_k | a_{ki} \in \mathcal{A}\}$ respectively. In the clustering procedure described in Section 4.3.2, we are interested in the maximum net flow, $\bar{F}_{st}$, between all pairs of vertices, s and t. This maximum flow, for a specific pair s and t, is the maximum value of F_{st} for which there exists a set $\{f_{ij} \leq c_{ij}, \forall a_{ij} \in \mathcal{A}\}$ which satisfies (4.1).

Theorem 4.1 (Ford-Fulkerson theorem [72]) *The maximum flow from a vertex s to vertex t, $\bar{F}_{st}$, is equal to the value of the minimum cut $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ separating s and t.*

A cut $(\mathcal{V}_o \rightarrow \tilde{\mathcal{V}}_o)$ is a set of arcs in $\mathcal{A}$ which start from $\mathcal{V}_o$ and end in $\tilde{\mathcal{V}}_o$, where $\mathcal{V}_o$ and $\tilde{\mathcal{V}}_o$ are two vertex sets such that $s \in \mathcal{V}_o$, $t \in \tilde{\mathcal{V}}_o$ and $\mathcal{V}_o \cup \tilde{\mathcal{V}}_o = \mathcal{V}$. The value of the cut $(\mathcal{V}_o \rightarrow \tilde{\mathcal{V}}_o)$ is defined as:

$$\bar{F}(\mathcal{V}_o \rightarrow \tilde{\mathcal{V}}_o) = \sum_{a_{ij} \in (\mathcal{V}_o \rightarrow \tilde{\mathcal{V}}_o)} c_{ij}.$$

The minimum cut $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ for the vertex pair s and t is then the cut with the smallest such value. The proof of the theorem can be found in [72, Ch.1]. Ford and Fulkerson [72] have also developed an algorithm for solving the maximum flow problem through finding the minimum cut separating s from t. Observe that the minimum cut optimally partitions $\mathcal{V}$ into two subsets given the "seed" vertices s and t.

For an undirected graph $\mathcal{G}$, the problem of finding the maximum flow between

every pair of vertices or the "multi-terminal maximum flow problem" can be efficiently solved using an algorithm due to Gomory and Hu [71]. A complete description of their algorithm can be found in [73, Ch.11] and will not be given here. However, the following properties of undirected graphs are fundamental to the use of the Gomory-Hu algorithm in our work.

Theorem 4.2 ([71]) *Let $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ be a minimum cut separating s from t. Let r and r' be two arbitrary vertices in $\mathcal{V}_m$ (or in $\tilde{\mathcal{V}}_m$). Then the maximum flow and the minimum cut between r and r' may be computed from a smaller graph with all vertices in $\tilde{\mathcal{V}}_m$ (or in $\mathcal{V}_m$) condensed into a single vertex $\tilde{v}_c$ (or v_c).*

The condensation of $\tilde{\mathcal{V}}_m$ is such that for each vertex $v_i \in \mathcal{V}_m$ all arcs a_{ij} in $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ are replaced by a single arc $a'_{i,\tilde{v}_c}$ of capacity

$$c'_{i,\tilde{v}_c} = \sum_{v_j \in \tilde{\mathcal{V}}_m \cap \Gamma(v_i)} c_{ij}, \qquad \forall\, v_i \in \mathcal{V}_m.$$

The implication of the theorem is that if $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ is a minimum cut separating s from t, then there exists a minimum cut for each pair of vertices in $\mathcal{V}_m$ (or in $\tilde{\mathcal{V}}_m$) which will not further partition $\tilde{\mathcal{V}}_m$ (or $\mathcal{V}_m$). This is crucial to the optimality of our clustering algorithm, and to the efficiency of the Gomory-Hu algorithm.

Theorem 4.3 *Let $\mathcal{G}^* = (\mathcal{V}, \mathcal{A}^*)$ be a hypothetical complete graph with the capacity of arc $a^*_{ij} \in \mathcal{A}^*$ equal to the maximum flow $\bar{F}_{ij}$ between the corresponding vertices v_i and v_j of the original graph $\mathcal{G}$ and let $\mathcal{T}^*$ be a maximal spanning tree of $\mathcal{G}^*$. Then $\mathcal{T}^*$ is flow-equivalent to the original graph $\mathcal{G}$. Furthermore the flow-equivalent tree $\mathcal{T}^*$ constructed using the Gomory-Hu algorithm is also cut-equivalent to $\mathcal{G}$.*

Cut-equivalency (flow-equivalency) means that $\mathcal{G}$ and $\mathcal{T}^*$ are indistinguishable as far as the minimum cut (maximum flow) between vertices is concerned. A proof of this theorem is given by Gomory and Hu [71]. A more comprehensive proof of Theorem 4.3 can be found in [73, Ch.11] for flow-equivalency and in [74, Ch.2] for cut-equivalency. From now on we will refer to this cut and flow equivalent tree $\mathcal{T}^*$ simply as an equivalent tree.

The construction of the equivalent tree $\mathcal{T}^*$ using the Gomory-Hu algorithm involves the successive solution of precisely $M-1$ maximum flow problems with M being the number of vertices in $\mathcal{G}$. In addition, most of these maximum flow problems involve much smaller graphs due to vertex condensation. Once $\mathcal{T}^*$ has been constructed, the maximum flow between any pair of vertices s and t is equal to the minimum of the capacities among arcs on the unique path in $\mathcal{T}^*$ which leads from s to t. Any arc in that path with capacity equal to the maximum flow between s and t can be selected to form a minimum cut separating s and t.

4.3 Formulation

In this section, we show how the problem of clustering is formulated in terms of the optimal partitioning of an undirected graph into a number of subgraphs. The new clustering technique is described here in the context of image segmentation. The algorithm is however applicable to other clustering problems where a neighborhood relationship between data points can be established and the clustering objective can be properly defined in terms of an arc capacity function.

4.3.1 Clustering rationale

In this and the following section, we will assume that the data set can be adequately represented by an adjacency graph $\mathcal{G}$: each vertex corresponds to a data point (a component); an arc indicates a neighborhood relationship between the two linked components, and a similarity measure between these two is assigned to the arc as its flow capacity. In the case of image segmentation, one could construct such a graph in which each vertex represents a pixel in the image and arcs are placed between pairs of neighboring pixels.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be an adjacency graph, formed from the data set, with vertex set $\mathcal{V} = \{v_1, v_2, \cdots, v_M\}$ and arc set $\mathcal{A} = \{a_{ij}\}$ with a_{ij} denoting the arc between vertices v_i and v_j with capacity c_{ij}. Here c_{ij} is a similarity measure between v_i and v_j, *i.e.* the larger the arc capacity c_{ij}, the more similar are the vertices v_i and v_j. The purpose of a clustering algorithm is to group together the components

into a reduced number of clusters. Each cluster contains components with similar features. This problem can be formulated equivalently in terms of the adjacency graph $\mathcal{G}$ formed from the components: divide $\mathcal{G}$ into a number of unconnected subgraphs by removal of the arcs joining the subgraphs. The set of vertices in each subgraph then represents a single cluster. In the case of image segmentation, each of the remaining subgraphs contains a set of connected vertices or components whose union represents a spatially connected region of the image. Recall that the arc capacities are defined as a measure of similarity between each component and its connected neighbors, *i.e.* the larger the arc capacity, the more similar are the two linked vertices. Suppose we wish to partition the graph $\mathcal{G}$ into two subgraphs with as dissimilar features as possible. A good criterion for choosing such a partition is to minimize the maximum flow between the two subgraphs. Following Theorem 4.1, such an optimal partition can be found by removing arcs in $\mathcal{G}$ corresponding to the cut set with the smallest cut value among all minimum cuts between every pair of vertices in $\mathcal{G}$. Note that any other cut would result in greater or equal maximum flow (and hence similarity) between the two subgraphs. Finding these minimum cuts in an undirected graph is a well known problem in graph theory and has been efficiently solved by Gomory and Hu [71] through the construction of an equivalent tree $\mathcal{T}^*$.

One can extend this principle to the subdivision of the graph $\mathcal{G}$ into K subgraphs. In this case the largest inter-subgraph maximum flow is minimized among all possible K-partitions. This optimal K-partition of $\mathcal{G}$ is obtained by choosing the $K-1$ smallest minimum cuts from the set of all minimum cuts between each pair of vertices. From Theorem 4.3, the equivalent tree $\mathcal{T}^*$ has arcs with capacity equal to the maximum flow between the vertices. Thus one can easily remove arcs in $\mathcal{T}^*$ in order of increasing maximum flow to partition the original graph into any number of subgraphs. Equivalently, a K-partition of the graph using the method described above, produces a segmentation of the image into the K regions which are most dissimilar (in the sense defined by the capacity function) among all K-region segmentations of the image.

A proof of the optimality of this clustering procedure follows from Theorem 4.3,

and will be given in the next subsection. To clarify the practical utility of this results we illustrate the clustering approach for the case of 3 clusters. The objective here is to partition a graph $\mathcal{G}$ into three subgraphs where the largest inter-subgraph maximum flow is minimized. Let $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ be a minimum cut with smallest cut value among all minimum cuts separating each pair of vertices of $\mathcal{G}$, and let r and r' be any other pair of vertices. Note that if r and r' lie on opposite sides of $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$, then $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ has to be a minimum cut between them. Therefore in order to partition $\mathcal{G}$ further, we need only consider those pairs of vertices, r and r', such that both r and r' belong to either $\mathcal{V}_m$ or $\tilde{\mathcal{V}}_m$. If both r and r' are in $\mathcal{V}_m$ (or in $\tilde{\mathcal{V}}_m$), the minimum cut between r and r' can partition further only $\mathcal{V}_m$ (or $\tilde{\mathcal{V}}_m$) according to Theorem 4.2, which implies a partition of $\mathcal{G}$ into exactly three subgraphs. Hence the optimal 3-partition is obtained by choosing the second cut with the smallest possible cut value. Similar argument applies to cases with more clusters.

4.3.2 The clustering algorithm and its properties

Basic Clustering Algorithm

Step 1 : Form the adjacency graph $\mathcal{G}$ from the homogeneous components.
Step 2 : Find an equivalent tree $\mathcal{T}^*$ for $\mathcal{G}$.
Step 3 : Successively remove arcs in $\mathcal{T}^*$ in order of increasing arc capacity until a given stopping condition is met.

This clustering algorithm produces an optimal partition of the graph $\mathcal{G}$ into a given number of subgraphs, in the sense that the maximum of the inter-subgraph flows is minimized among all possible partitions of $\mathcal{G}$ into the same number of subgraphs.

Corollary 4.1 (Corollary to Theorem 4.3) *The K-partition of an undirected graph G, obtained by removing the K-1 arcs with the smallest capacity in the equivalent tree $\mathcal{T}^*$ of $\mathcal{G}$, minimizes the largest inter-subgraph maximum flow among all possible K-partitions of $\mathcal{G}$.*

Proof of Corollary 4.1 Following Theorem 4.3, the maximum flow between

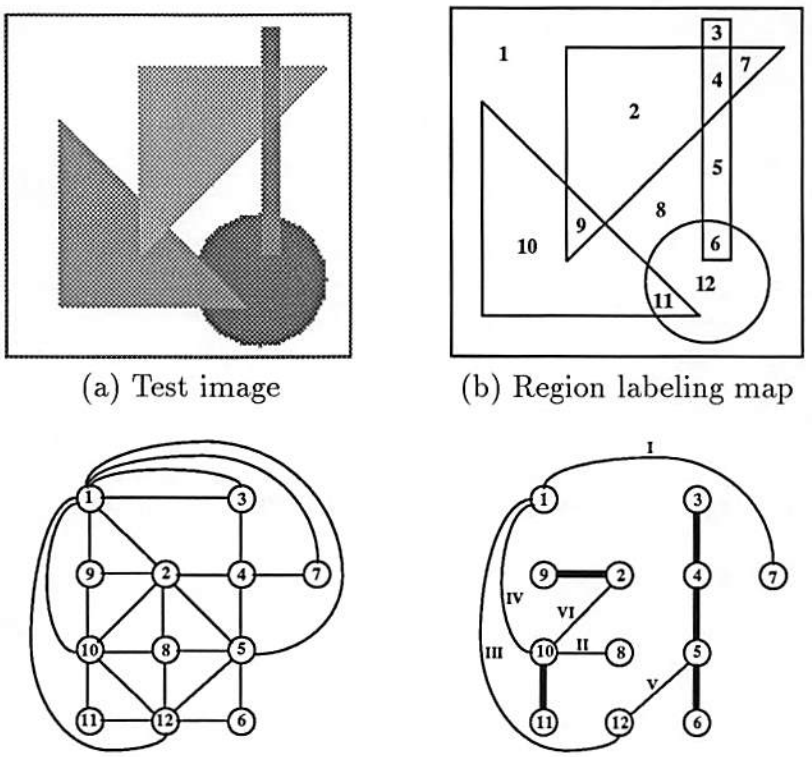
any two subgraphs is equal to the minimum of the capacities among arcs on the unique path in $\mathcal{T}^*$ which connects the subgraphs. Along that path, at least one of the arcs must have been marked for removal in order to disconnect the two subgraphs. Hence the maximum flow between any subgraph pair cannot exceed the largest capacity of those K-1 marked arcs. But note also that between those subgraph pairs linked only by a single marked arc, the maximum flow must be equal to the capacity of that marked arc. Therefore we conclude that the largest inter-subgraph maximum flow is equal to the largest of the K-1 marked arc capacities, which is minimized when the K-1 arcs in $\mathcal{T}^*$ with the smallest capacity are marked for removal. □

Note also that in this optimal subgraph partition, the maximum flow between any pair of vertices in the same subgraph (intra-subgraph maximum flow) is always greater than or equal to the maximum flow between vertices in two different subgraphs (inter-subgraph maximum flow). A given undirected graph $\mathcal{G}$ may have more than one equivalent tree $\mathcal{T}^*$, but $\mathcal{T}^*$ will be unique if all of its arcs have distinct capacities [72, Ch.4]. In the case of a unique $\mathcal{T}^*$ a unique optimal K-partition is indeed guaranteed. However the optimal K-partition may still be unique even if $\mathcal{T}^*$ is not.

Corollary 4.2 (Corollary to Theorem 4.3) *The K-partition of an undirected graph $\mathcal{G}$, obtained by removing the K-1 arcs with the smallest capacity in the equivalent tree $\mathcal{T}^*$ of $\mathcal{G}$, is unique if all K-1 arcs in $\mathcal{T}^*$ marked for removal have capacity strictly less than the remaining unmarked arcs.*

Proof Let $\mathcal{F}_T$ be the smallest arc capacity of all unmarked arcs in $\mathcal{T}^*$. In other words, the maximum flow between any pair of vertices in the same subgraph is at least $\mathcal{F}_T$. However the maximum flow between any pair of vertices belonging to two different subgraphs is strictly less than $\mathcal{F}_T$ by assumption. Hence any other K-subgraph partition would have to result in at least one intra-subgraph maximum flow less than $\mathcal{F}_T$ and hence the optimal partition is unique. □

As the number of subgraphs increases, so does the number of corresponding regions in the segmented image. Since the goal of clustering is to produce a small



(a) Test image (b) Region labeling map

(c) Adjacency graph $\mathcal{G}$ (d) Equivalent tree $\mathcal{T}^*$

Figure 4.1: The test image, its adjacency graph $\mathcal{G}$ and the equivalent tree $\mathcal{T}^*$.

number of regions from the image, a stopping condition should be used to prevent the formation of too many regions. This could be achieved either interactively by terminating the arc cutting procedure when sufficient regions have been formed, or by using a threshold so that arcs with a maximum flow above a given threshold are not cut. The problem of systematically choosing a suitable threshold is clearly important. Currently, we choose the threshold empirically.

4.3.3 A clustering example

A simple example is described here to illustrate how the clustering algorithm works for image segmentation. Fig. 4.1.a shows a computer generated image of 128×128 pixels. Different intensities are assigned to the background, the 4 objects and their overlapped areas. The image is thus partioned into 12 connected regions (patches) of constant intensity (Fig. 4.1.b). The respective grey scale

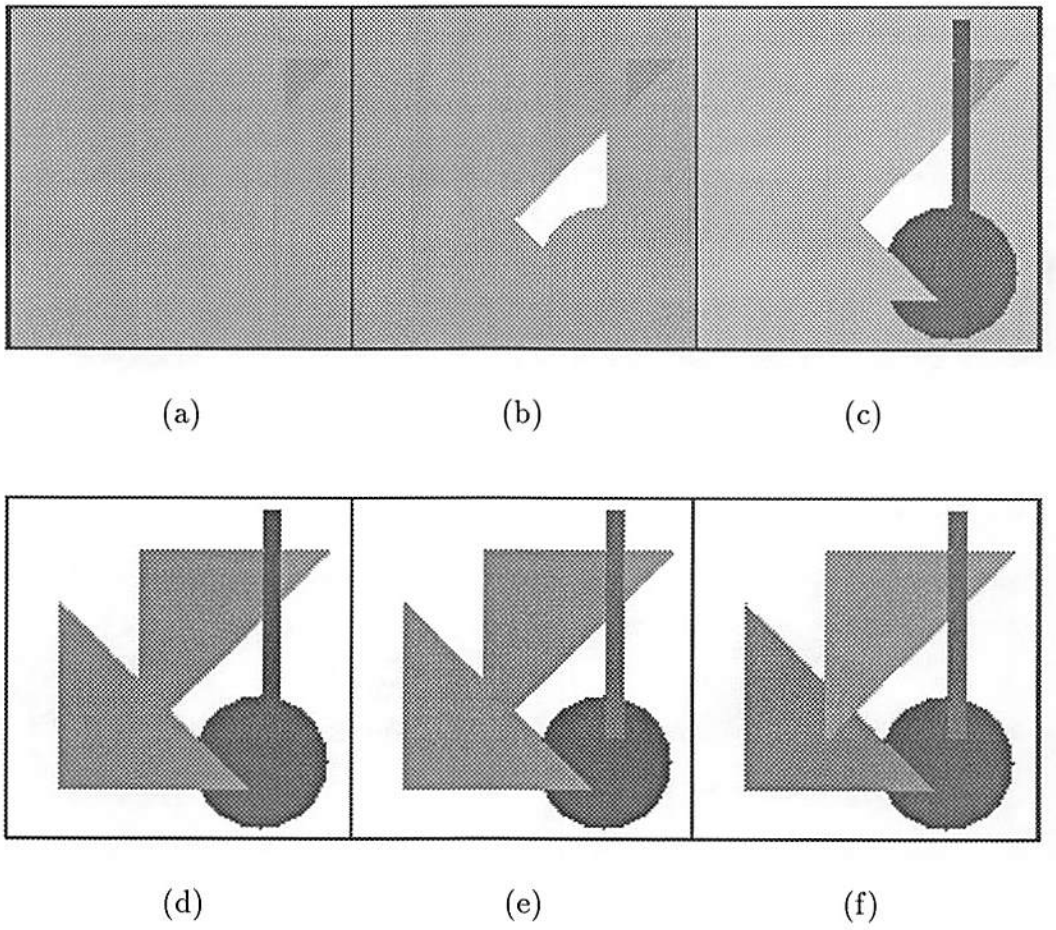


Figure 4.2: Resulting image partitions at each step of patch clustering. All pixels are displayed as their respective subtree's mean intensity. This figure shows regions formed after removal of 1 (4.2a), 2 (4.2b), 3 (4.2c), 4 (4.2d), 5 (4.2e), 6 (4.2f) arcs of the spanning tree $\mathcal{T}^*$ shown in Fig. 4.1d.

Patch	Intensity
1	255
2	176
3	112
4	120
5	112
6	120
7	176
8	255
9	184
10	144
11	136
12	80

(a)

Order	Arc	Capacity
I	$a^*_{1,7}$	7.40×10^{-11}
II	$a^*_{8,10}$	1.56×10^{-10}
III	$a^*_{1,12}$	3.78×10^{-10}
IV	$a^*_{1,10}$	4.32×10^{-10}
V	$a^*_{5,12}$	7.56×10^{-5}
VI	$a^*_{2,10}$	1.12×10^{-4}
VII	$a^*_{3,4}$	3.1638
VIII	$a^*_{5,6}$	3.1638
IX	$a^*_{4,5}$	6.8555
X	$a^*_{2,9}$	11.073
XI	$a^*_{10,11}$	12.128

(b)

Table 4.1: (a) Grey scale intensities for the regions or patches in Fig 4.1; (b) Values of the capacity for the arcs of the equivalent tree shown in Fig. 4.1d.

intensities for these patches are listed in Table 4.1.a. Note that the smallest intensity difference between neighboring overlapped and non-overlapped areas is 8. An adjacency graph, $\mathcal{G}$, is formed from these patches, as shown in Fig. 4.1.c. Each vertex of the graph corresponds to a patch. An arc is placed between a pair of vertices if their corresponding patches are neighbors of each other, *i.e.* if they are spatially connected. A flow capacity is then assigned to each arc. The capacities may be any non-negative symmetric function. For the purposes of this example, we choose these functions as a measure of similarity between the patches (vertices). In particular the following capacity function, with the control parameter $\sigma = 10$, is used in our example:

$$c_{i,j} = k_{i,j} \cdot \mathrm{e}^{-(\frac{\mu_i - \mu_j}{\sigma})^2} \tag{4.2}$$

where $k_{i,j}$ denotes the number of neighboring pixels between vertices v_i and v_j, and μ_i and μ_j are the sample means of patches i and j. An equivalent tree $\mathcal{T}^*$ is then generated using the Gomory-Hu algorithm; $\mathcal{T}^*$ is shown in Fig. 4.1.d. The

capacity values for all arcs of $\mathcal{T}^*$ are listed in increasing order in Table 4.1.b.

Arcs are now removed, in order of increasing capacity, from the equivalent tree $\mathcal{T}^*$. The first arc to be removed is $a^*_{1,7}$, which results in a subtree consisting of the region 7 and another containing the rest of the image. Fig. 4.2.a shows the resulting image partition with all pixels displayed as their respective subtree's mean intensity. Similarly the arcs $a^*_{8,10}$, $a^*_{1,12}$, $a^*_{1,10}$, $a^*_{5,12}$ and $a^*_{2,10}$ are removed to further partition the image (Fig. 4.2.b-f). Since Fig. 4.2.f correctly classifies neighboring regions with similar features (in our case the pixel intensity), no additional clusters are needed. The remaining arcs in $\mathcal{T}^*$ are shown in Fig. 4.1.d with bold links. Note that regions 1 and 8 are not clustered together because they are not spatially connected. The same is true for regions 2 and 7. The final number of clusters is 7.

4.4 Hierarchical Implementation

A direct implementation of the new graph theoretic clustering method is practical only for graphs of moderate size. The number of maximum flow problems involved in the Gomory-Hu algorithm increases linearly with the number of nodes in the graph; the time needed for solving those problems increases at a much faster rate. This is due to the fact that finding the maximum flow between two vertices has polynomial complexity with respect to the graph size (see [75] for complexities of various maximum flow algorithms). In our experiments on a Sun SPARCstation, our direct implementation of the clustering method could comfortably handle graphs with up to a couple of thousand vertices and a comparable number of arcs, i.e. computation times on the order of ten minutes. Unfortunately, a graph constructed from a 256×256 image, with each node containing only one pixel, will have 65,536 vertices.

The hierarchical implementation developed here is based on the observation that most of the minimum cuts found in the equivalent tree are never used since their values are sufficiently large that the arcs in those cuts will not be removed to form subgraphs. Consequently one may consider the problem of identifying these minimum cuts and condensing the vertices linked by them *before* constructing the

equivalent tree. This would greatly reduce the size of the graph to which the Gomory-Hu algorithm is applied, without compromising the overall optimality of the clustering algorithm.

4.4.1 Extensions on network flow theory

The maximum flow between any pair of vertices in a connected graph $\mathcal{G}$ is a global characteristic of $\mathcal{G}$. Its value must be computed using the complete graph or an appropriately condensed graph. The Gomory-Hu algorithm provides a way to construct a condensed graph, but this condensation process itself has to start with the complete graph. It would clearly be desirable to know if a subgraph may be condensed without processing the complete graph $\mathcal{G}$. The following new theorem and its corollary will provide sufficient conditions for condensation of $\mathcal{G}$ using arbitrary subgraphs of $\mathcal{G}$.

Theorem 4.4 *Let $\mathcal{G}_o$ be a subgraph of an undirected graph $\mathcal{G}$, and $\bar{F}_{o,\min}$ be the minimum value of the maximum flows between all pairs of vertices in $\mathcal{G}_o$. Let s and t be an arbitrary pair of vertices in $\mathcal{G}$ such that the value of the maximum flow between s and t, $\bar{F}_{st}$, is smaller than $\bar{F}_{o,\min}$. Then $\bar{F}_{st}$ can be equivalently computed from the graph, $\mathcal{G}_c$, which is obtained by condensing all vertices of $\mathcal{G}_o$ into a single vertex.*

Proof Let us first show that the minimum cut separating s and t, $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$, may not contain any arcs in $\mathcal{G}_o$. Assume the contrary to be true, *i.e.* there exists at least one arc $a_{rr'}$ in $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ such that both r and r' belong to $\mathcal{G}_o$. Then by definition, $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ is also a cut (not necessarily minimum) separating r and r'. However its cut value, $\bar{F}_{st}$, is smaller than $\bar{F}_{rr'}$, the maximum flow between r and r', because $\bar{F}_{st} < \bar{F}_{o,\min}$ by assumption, and $\bar{F}_{o,\min} \leq \bar{F}_{rr'}$ by definition. This contradicts the definition of $\bar{F}_{rr'}$ as the value of the minimum cut between r and r'. Consequently, $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ cannot contain any arc in $\mathcal{G}_o$ and therefore $\mathcal{G}_o$ should be entirely contained by either $\mathcal{V}_m$ or $\tilde{\mathcal{V}}_m$. It then follows that condensing vertices in $\mathcal{G}_o$ will have no effect on the minimum cut $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ which separates s from t. □

Theorem 4.4 can be modified to provide a result which is more useful in developing the hierarchical clustering algorithm. This result is contained in the following corollary.

Corollary 4.3 *Let $\mathcal{G}_o$ be a subgraph of an undirected graph $\mathcal{G}$, $\bar{F}'_{o,\min}$ be the minimum value of the maximum flows between all pairs of vertices in $\mathcal{G}_o$ computed using an arbitrary subgraph, $\mathcal{G}'_o$, which contains $\mathcal{G}_o$, and s and t be an arbitrary pair of vertices in $\mathcal{G}$. If the value, $\bar{F}_{st}$, of the maximum flow between s and t, computed using $\mathcal{G}$, is smaller than $\bar{F}'_{o,\min}$, then $\bar{F}_{st}$ can be equivalently computed from the condensed graph $\mathcal{G}_c$ with all vertices in $\mathcal{G}_o$ condensed into a single vertex.*

Before we proceed to prove the corollary, let us introduce the following lemma. The lemma establishes an important relationship between the maximum flows computed using the complete graph and those using a local subgraph. Proof of Corollary 4.3 then follows by combining this lemma with Theorem 4.4.

Lemma 4.1 *Let $\mathcal{G}'_o$ be a subgraph of an undirected graph $\mathcal{G}$, and $\bar{F}'_{st}$ be the value of the maximum flow between a pair of vertices, s and t, in $\mathcal{G}'_o$ computed using $\mathcal{G}'_o$. Let $\bar{F}_{st}$ be the maximum flow between s and t computed using $\mathcal{G}$. Then*

$$\bar{F}'_{st} \leq \bar{F}_{st}.$$

Proof Let $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ be a minimum cut separating s and t computed in $\mathcal{G}$. The subset of $(\mathcal{V}_m \rightarrow \tilde{\mathcal{V}}_m)$ whose arcs are in $\mathcal{G}'_o$ is also a cut separating s from t in $\mathcal{G}'_o$. By definition, the value of this cut should be greater than or equal to $\bar{F}'_{st}$, the value of the minimum cut between s and t computed in $\mathcal{G}'_o$. □

Proof of Corollary 4.3 If $\bar{F}_{st} < \bar{F}'_{o,\min}$, then $\bar{F}_{st} \leq \bar{F}_{o,\min}$, since $\bar{F}'_{o,\min} \leq \bar{F}_{o,\min}$ from Lemma 4.1. Therefore the conditions for Theorem 4.4 are satisfied. □

The implication of Corollary 4.3 is significant since the condition under which a subgraph, $\mathcal{G}_o$, may be condensed can be checked using only a local subgraph which contains $\mathcal{G}_o$. Using this result a hierarchical clustering method may be developed which uses subgraphs for condensation (local processing) and the Gomory-Hu algorithm for optimal clustering applied to the condensed graph (global processing).

The theoretic foundation of such a clustering technique is presented below as a theorem.

Theorem 4.5 *Let $\mathcal{G}_o$ be a subgraph of an undirected graph $\mathcal{G}$, and $\bar{F}'_{o,\min}$ be the minimum value of the maximum flows between all pairs of vertices in $\mathcal{G}_o$ computed in a subgraph $\mathcal{G}'_o$ which contains $\mathcal{G}_o$. Let $\mathcal{G}_c$ be the graph formed from $\mathcal{G}$ with all vertices in $\mathcal{G}_o$ condensed into a single vertex and let $\mathcal{T}_c^*$ be an equivalent tree constructed from $\mathcal{G}_c$. Then $\mathcal{T}_c^*$ is a partially cut and flow equivalent tree of $\mathcal{G}$, at threshold $\bar{F}'_{o,\min}$, in the sense that the maximum flow, $\bar{F}_{st}$, and the corresponding minimum cut can be computed from $\mathcal{T}_c^*$ for all pairs of vertices s and t in $\mathcal{G}$ such that $\bar{F}_{st} < \bar{F}'_{o,\min}$. Furthermore for all vertex pairs, s and t, in $\mathcal{G}$ with flow $\bar{F}_{st} \geq \bar{F}'_{o,\min}$, the maximum flow value computed in $\mathcal{T}_c$ is also greater than or equal to $\bar{F}'_{o,\min}$.*

Proof Let s and t be an arbitrary pair of vertices in $\mathcal{G}$, and assume the maximum flow between s and t, $\bar{F}_{st}$, is less than $\bar{F}'_{o,\min}$. From Corollary 4.3, $\bar{F}_{st}$ and the corresponding minimum cut separating s and t may be computed from $\mathcal{G}_c$. Since $\mathcal{G}_c$ and $\mathcal{T}_c^*$ are flow and cut equivalent, the maximum flow $\bar{F}_{st}$ and the minimum cut can also be computed from $\mathcal{T}_c^*$. Now assume that $\bar{F}_{st} \geq \bar{F}'_{o,\min}$. From Lemma 4.1, the maximum flow computed in $\mathcal{G}_c$ is at least as large as $\bar{F}_{st}$. □

So far we have derived sufficient conditions for subgraph condensation when we are only interested in finding minimum cuts with small cut values. In the remainder of this subsection, we will derive additional conditions for subgraph condensation with no cut value constraints attached.

Let $\mathcal{G}'_o$ be a subgraph of an undirected graph $\mathcal{G}$ and $\mathcal{T}'^*_o$ be an equivalent tree computed from $\mathcal{G}'_o$. We wish to construct an equivalent tree $\mathcal{T}^*$ for $\mathcal{G}$ using as much information from $\mathcal{T}'^*_o$ as possible. Specifically we want to identify those minimum cuts in $\mathcal{T}'^*_o$ which are also minimum cuts in $\mathcal{T}^*$. If successful, we can initialize the Gomory-Hu algorithm at a much more advanced stage, since these identified minimum cuts need not be computed again, and thus greatly reduce the computation time.

Theorem 4.6 *Let $\mathcal{T}'^*_o$ be an equivalent tree of $\mathcal{G}'_o$, a subgraph of an undirected graph $\mathcal{G}$, and $\mathcal{T}_o^*$ be a branch of $\mathcal{T}'^*_o$ consisting exclusively of interior vertices of $\mathcal{G}'_o$.*

Let $\mathcal{G}_c$ be the graph formed from $\mathcal{G}$ with all vertices in $\mathcal{T}_o^*$ condensed into a single vertex v_o^*. The equivalent tree $\mathcal{T}^*$ of $\mathcal{G}$ can be constructed as follows: construct an equivalent tree $\mathcal{T}_c^*$ for the condensed graph $\mathcal{G}_c$, and then replace the vertex v_o^* by $\mathcal{T}_o^*$ connected at its root vertex.

Here an interior vertex of a subgraph $\mathcal{G}_o'$ is a vertex such that all of its neighboring vertices also belong to $\mathcal{G}_o'$. A non-interior vertex is called a boundary vertex. The root vertex of the branch $\mathcal{T}_o^*$ refers to the only vertex in $\mathcal{T}_o^*$ which neighbors a boundary vertex in $\mathcal{T}_o'^*$. The proof of Theorem 4.6 is based on the Gomory-Hu algorithm and its optimality.

Proof Let s denote the root vertex of the branch $\mathcal{T}_o^*$, and t denote the parent vertex of s in $\mathcal{T}_o'^*$. Since all vertices in $\mathcal{T}_o^*$ are interior vertices, the minimum cut (in $\mathcal{G}_o'$) separating s and t, which is equivalent to the arc $a_{st}'^*$ in $\mathcal{T}_o'^*$, should also separate s and t in the complete graph $\mathcal{G}$. Denote this minimum cut as $(\mathcal{G}_o \rightarrow \tilde{\mathcal{G}}_o)$, where $\tilde{\mathcal{G}}_o$ is the subgraph of $\mathcal{G}$ which contains all vertices not in $\mathcal{G}_o$.

Apply the Gomory-Hu algorithm to $\mathcal{G}$ as follows. Choose s and t as the first vertex pair between which the maximum flow will be computed. Since $(\mathcal{G}_o \rightarrow \tilde{\mathcal{G}}_o)$ is a minimum cut separating s and t, the resulting tree at this stage should be two condensed vertices v_o^* and $\tilde{v}_o^*$ linked by an arc with capacity equal to the cut value of $(\mathcal{G}_o \rightarrow \tilde{\mathcal{G}}_o)$, where v_o^* and $\tilde{v}_o^*$ are condensed from $\mathcal{G}_o$ and $\tilde{\mathcal{G}}_o$ respectively. Now proceed with the algorithm by selecting only vertex pairs in $\tilde{\mathcal{G}}_o$ until every condensed vertex except v_o^* has become a single vertex of $\mathcal{G}$. Note that we have just constructed the equivalent tree $\mathcal{T}_c^*$ for the graph $\mathcal{G}_c$, which is obtained from $\mathcal{G}$ by condensing all the vertices in $\mathcal{G}_o$ into a single vertex v_o^*.

Finally we need to compute the maximum flows between all vertex pairs in $\mathcal{G}_o$. But the resulting maximum flows and the minimum cuts should be identical to those computed using the subgraph $\mathcal{G}_o'$, because all the vertices in $\mathcal{G}_o$ are interior vertices of $\mathcal{G}_o'$. Therefore in order to complete the equivalent tree $\mathcal{T}^*$ of $\mathcal{G}$, the condensed vertex v_o^* can be expanded in exactly the same manner as when constructing $\mathcal{T}_o'^*$ from $\mathcal{G}_o'$. □

4.4.2 A clustering algorithm using hierarchical implementation

In order to perform clustering of the vertices of a graph $\mathcal{G}$ to minimize the largest inter-subgraph maximum flow, it is not necessary to know the exact arc capacities (minimum cut values in $\mathcal{G}$) for all arcs in the equivalent tree $\mathcal{T}^*$. This is because only those arcs with small capacities will be removed during the clustering process. In other words, if an arc capacity is greater than a given threshold, $\mathcal{F}_T$, we are no longer interested in its exact value since the arc will not be cut. The value of this threshold can be chosen as any value greater than the largest minimum cut which will be made in partioning the graph into its subgraphs. Using this approach and Theorem 5, a partially equivalent tree can be constructed more efficiently than $\mathcal{T}^*$. This produces a saving in computation time for the overall clustering process without any loss in optimality. In this subsection we develop an efficient clustering technique based on hierarchically constructing a partially equivalent tree for the graph $\mathcal{G}$.

Let us illustrate the subgraph condensation procedure using the example shown in Fig. 4.3. Let $\mathcal{G}$ be an adjacency graph constructed from an image with vertices and arcs in $\mathcal{G}$ correspond to the image pixels and edges between neighboring pixels, respectively. Arbitrarily partition $\mathcal{G}$ into smaller subgraphs, $\{\mathcal{G}'_k\}$, and then construct equivalent trees, $\{\mathcal{T}'^*_k\}$, for each of the subgraphs. If the vertices in $\mathcal{G}'_k$ are clustered by removing all the arcs corresponding to cuts whose capacities are smaller than $\mathcal{F}_T$, then the maximum flow (computed in $\mathcal{G}'_k$) between any pair of vertices belonging to the same cluster must be at least $\mathcal{F}_T$. From Theorem 4.5, the equivalent tree $\mathcal{T}^*_c$, computed from the graph with each of the clusters in $\{\mathcal{G}'_k\}$ condensed permanently into a single vertex, can be used in place of $\mathcal{T}^*$ for clustering vertices in $\mathcal{G}$. Here $\mathcal{T}^*_c$ is referred to as a *partially* equivalent tree of $\mathcal{G}$, because $\mathcal{T}^*_c$ is cut and flow equivalent to $\mathcal{G}$ only between vertex pairs with maximum flow values smaller than $\mathcal{F}_T$.

The construction of this partially equivalent tree $\mathcal{T}^*_c$ can also make full use of the minimum cuts in $\{\mathcal{T}'^*_k\}$ that enclose interior vertices of the subgraph $\{\mathcal{G}'_k\}$. A branch of $\mathcal{T}'^*_k$ that contains only interior vertices is referred to here as an interior branch of $\mathcal{T}'^*_k$. A maximal interior branch of $\mathcal{T}'^*_k$ is then an interior branch that

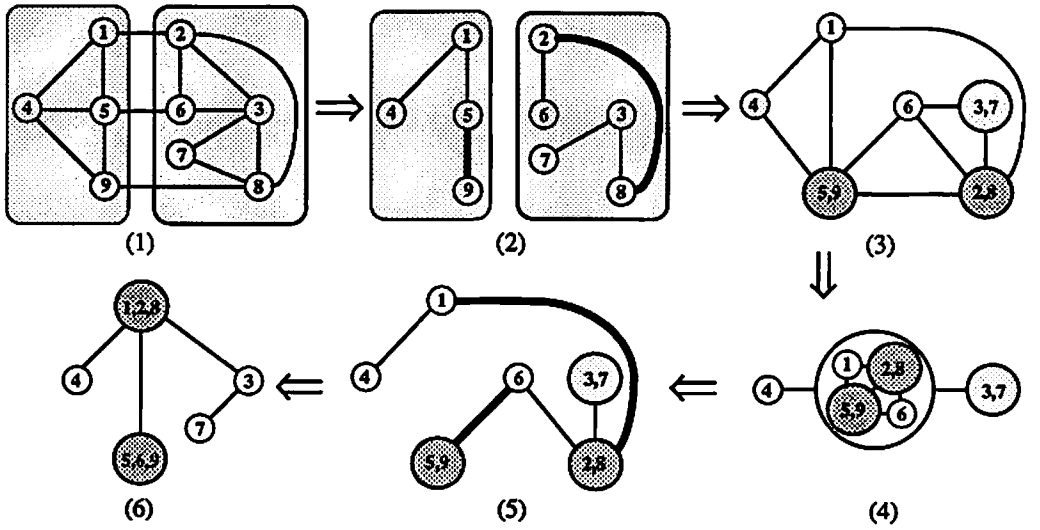


Figure 4.3: Illustration of the subgraph condensation procedure: vertices and arcs in the graph $\mathcal{G}$ correspond to the pixels and edges between neighboring pixels, respectively. The tasks performed in the six steps are: (1) partition $\mathcal{G}$ into subgraphs $\mathcal{G}_1'$ and $\mathcal{G}_2'$; (2) construct equivalent trees $\mathcal{T}_1'^*$ and $\mathcal{T}_2'^*$ (Arcs shown as wide lines indicate that their capacities exceed the threshold $\mathcal{F}_T$, and the vertices linked by them are condensed permanently.); (3) form the condensed graph $\mathcal{G}_c$ with interior branches in $\mathcal{T}_1'^*$ and $\mathcal{T}_2'^*$ condensed temporarily; (4) initialization of Gomory-Hu algorithm for $\mathcal{G}_c$; (5) construct the equivalent tree of $\mathcal{G}_c$; (6) form the partially equivalent tree $\mathcal{T}_c^*$ of $\mathcal{G}$ by expanding the interior branches.

cannot be contained by any other interior branch of $\mathcal{T}_k'^*$. In our example, $\{v_4\}$ and $\{v_3, v_7\}$ are maximal interior branches in their respective equivalent trees. From Theorem 4.6, all vertices in any maximal branch of $\mathcal{T}_k'^*$ can be temporarily condensed into a single vertex when computing $\mathcal{T}_c^*$. Furthermore the minimum cut between these condensed vertices and the rest of $\mathcal{T}_c^*$ is identical to that computed in $\mathcal{T}_k'^*$. Hence the Gomory-Hu algorithm may be initialized using the following tree: every maximal interior branch of $\{\mathcal{T}_k'^*\}$ corresponds to a condensed vertex in this tree, and each is linked to a condensed vertex t_0 that contains the rest of the vertices of $\mathcal{G}_c$ (step 4 in Fig. 4.3). The capacity of each arc in this tree is equal to the value of the minimum cut between the corresponding interior branch and the

rest of the graph. The Gomory-Hu algorithm is only used to expand the vertex t_0. The rest of the condensed vertices are expanded using the interior branches of $\{\mathcal{T}_k'^*\}$ (step 6 in Fig. 4.3). Note that not only is the Gomory-Hu algorithm applied to a graph of reduced size due to the condensation of the maximal interior branches of $\{\mathcal{T}_k'^*\}$, but it is also initialized to solve only as many maximum flow problems as the number of vertices in t_0 minus one.

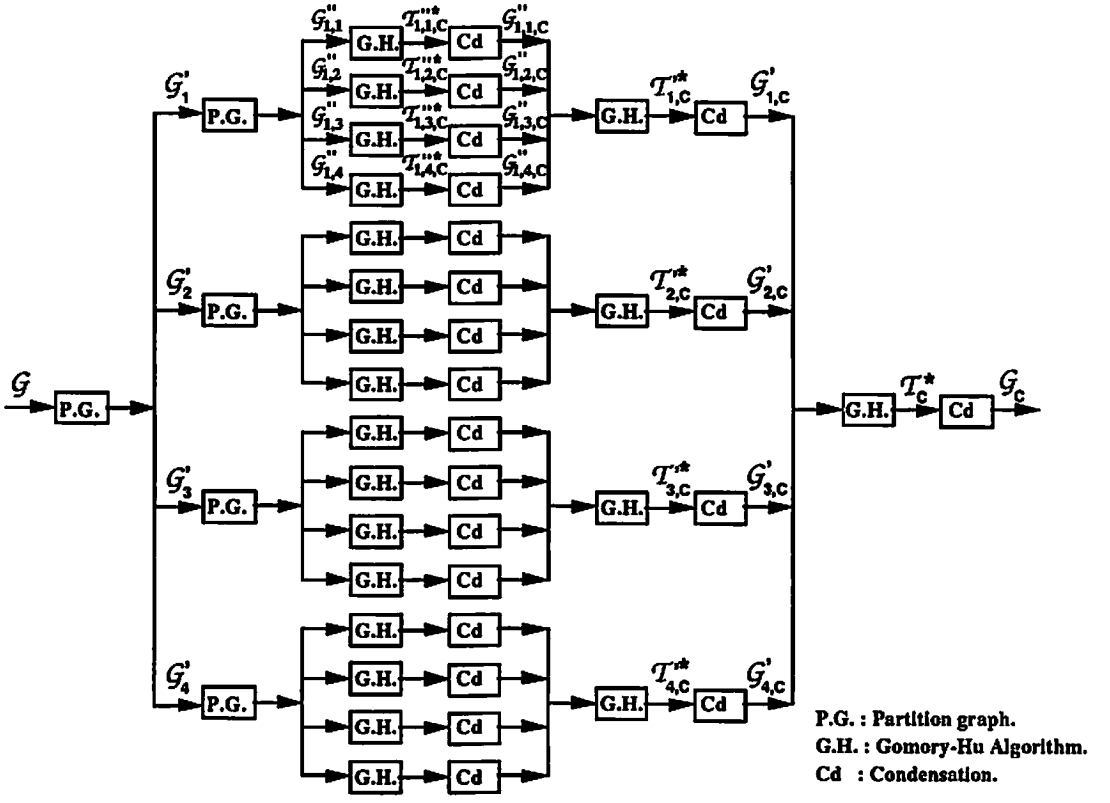


Figure 4.4: Block diagram for the construction of a partially equivalent tree using a three level hierarchy.

The above condensation procedure can be applied in a hierarchical fashion, i.e. $\mathcal{G}$ may be partioned into a large number of smaller subgraphs with the above procedure applied to each. The resulting condensed subgraphs are then combined in groups of two or more and the process repeated in a hierarchical fashion as

illustrated in Fig. 4.4. For a clustering procedure in which the maximum inter-subgraph flow is constrained to be less than $\mathcal{F}_T$, the result obtained using this hierarchical approach is identical to that which would be obtained by applying the Gomory-Hu algorithm directly to the complete graph. The computational advantage is significant - in our experiments CPU times for the direct Gomory-Hu algorithm in excess of 12 hours are reduced to the order of 10 minutes using the hierarchical approach.

Clustering Algorithm Using Hierarchical Implementation

Step 1 : Form the adjacency graph $\mathcal{G}$, and then partition $\mathcal{G}$ into a number of small subgraphs, $\{\mathcal{G}'_{0,m}\}$, usually of similar size. Let $i = 0$.

Step 2 : For each subgraph $\mathcal{G}'_{i,m}$ find an equivalent tree $\mathcal{T}'^{*}_{i,m}$ using the Gomory-Hu algorithm.

Step 3 : Permanently condense all the subtrees in $\mathcal{T}'^{*}_{i,m}$ which are linked by arcs with capacities $\geq \mathcal{F}_T$. Also temporarily condense all the maximal interior branches of this resulting tree to form a condensed subgraph $\mathcal{G}'_{i,m,c}$.

Step 4 : If there is only one subgraph in $\{\mathcal{G}'_{i,m,c}\}$, then denote the resulting equivalent tree as $\mathcal{T}'^{*}_{c}$ and go to Step 6; otherwise continue to Step 5.

Step 5 : Group $\{\mathcal{G}'_{i,m,c}\}$ into subsets, each containing a number of neighboring subgraphs of $\{\mathcal{G}'_{i,m,c}\}$. By connecting the subgraphs in each subset using the inter-subgraph arcs, we form a new subgraph set $\{\mathcal{G}'_{i+1,m}\}$. Let $i = i + 1$, and go to Step 2.

Step 6 : Expand back all the temporarily condensed vertices in $\mathcal{T}'^{*}_{c}$ to construct the partially equivalent tree $\mathcal{T}^{*}_{c}$ of the original graph $\mathcal{G}$.

Step 7 : Remove successively arcs in $\mathcal{T}^{*}_{c}$ in order of increasing arc capacity until a given stopping condition is met. A clustering constraint may be attached in this last step.

4.4.3 Incorporation of constraints

So far we have only discussed unconstrained data clustering. Often we also want the resulting data clusters to possess certain desirable characteristics, *e.g.* clusters should not be too small.

A direct approach to incorporating these requirements is to impose "hard" constraints on the clustering formulation as follows: find the K-subgraph partition that minimizes the largest inter-subgraph maximum flow, while each of the K subgraphs meets the specified requirements. Unfortunately the optimal solution would have to be chosen from a possible $C_{M_c-1}^{K-1}$ different combinations, where M_c is the number of vertices in $\mathcal{T}_c^*$. Branch-and-bound algorithms should be a suitable choice for solving this problem.

Alternatively, we may impose constraints on the minimum cuts which are used to partition the graph $\mathcal{G}$. For example, only minimum cuts containing at least a given number of arcs in the original graph may be selected for graph partitioning. For this type of constraint, the optimal solution can still be found in a sequential fashion: remove the arcs of $\mathcal{T}^*$ in order of increasing capacity unless the corresponding cut set in $\mathcal{G}$ does not satisfy the constraint, in which case no action takes place. This process is repeated until K-1 arcs have been removed. The resulting solution is locally optimal, since the equivalent tree of $\mathcal{G}$ may not be unique. However if either all or none of the arcs in $\mathcal{T}_c^*$ with the same capacity are considered for removal, the solution becomes globally optimal following similar arguments used to prove Corollary 4.2.

Another approach is to select arc capacities in the adjacency graph $\mathcal{G}$ in such a way that the minimum cuts become less likely to result in undesirable subgraphs. When the constraints are not strict, this penalty approach may be a preferred one. In addition, we may also want to give special treatment to those clusters that do not meet the constraints. For example, when segmenting images we do not want to form many small regions. In practice, we find that these small regions often correspond to boundary pixels between large objects. Therefore we may simply choose to allow these small regions to be formed in the clustering stage, and then merge them into large neighboring regions in a refining step.

Chapter 5

Image Segmentation Using Graph Theoretic Clustering

In this chapter we will demonstrate how we can apply the graph theory based clustering algorithm to the image segmentation problem. It has been a common practice to segment images based on clustering techniques. Many of the thresholding based segmentation techniques [76] are also implicitly related to clustering. The basic approach is to compute a feature vector for each pixel of the image and then to segment the image by clustering these computed feature vectors together [77], [78], [14]. Often little or no spatial information about the image is used. In a related work, Haddon and Boyce [79] propose to segment the image by partitioning a cooccurrence matrix (as opposed to a histogram) followed by a relaxation procedure which incorporates conditions on neighborhood consistency. Here a new approach for image segmentation is developed based on our clustering algorithm. The segmentation is achieved by effectively searching for closed contours of strong edges. The edges are defined as separating elements between adjacent pixel pairs and computed using a local derivative operator. An adjacency graph $\mathcal{G}$ is constructed from the pixels, with strong edges mapped into small arc capacities and weak edges mapped to large capacities. Our clustering algorithm is then applied to find minimum cuts in $\mathcal{G}$ with small value, or equivalently closed contours containing mostly strong edges. This approach is able to accurately locate region boundaries, and at the same time guarantees the formation of closed

edge contours.

Depending on the availability of prior knowledge about the image, supervised or unsupervised approaches may be used for computing the edges. In the unsupervised case, the edges are estimated as a function of intensity differences between adjacent pixel pair and between pixels in their vicinity. In the supervised case, the image is modeled as a mosaic of uniform regions corrupted by an additive noise, where the intensities for the uniform regions and the characterization of the noise are both known. Using this image model, edge detection is formulated and solved as a hypothesis testing problem. The likelihood ratio test is proposed for detecting edges, and a detailed performance analysis of applying this test to the case of two uniform regions with Gaussian noise is presented. Following this analysis, the likelihood ratio between the hypotheses of edge absence and edge presence is suggested as an edge strength measure.

In this chapter we also apply the graph theoretic clustering algorithm to the hierarchical segmentation algorithm for patch clustering. Experimental results show that the graph theoretic clustering technique performs more favorably than the ML agglomerative clustering technique of Chapter 3.

5.1 Unsupervised Graph Theoretic Segmentation

The goal of the segmentation algorithm described here is achieved by effectively searching for closed contours of strong edges in the image. The algorithm intends to combine the advantages of two types of edge detectors.

There is a class of edge operators, such as the Marr-Hildreth method [11], which are based on zero crossings that can guarantee the formation of closed edge contours. Ehricke [21] and Bommans *et al.* [22] have developed segmentation algorithms for tissue classification in MR images based on regions enclosed by the edge contours resulting from the Marr-Hildreth operator. Despite the remarkable success of this type of edge detectors for many applications, it does not appear to be appropriate to use these edge contours as the basis for MR image segmentation. Edge finding using zero crossings, equivalently locates edges at the local

maxima or minima of the response of the image to a first order derivative operator. Some of these maxima or minima may however correspond to small fluctuations in the image rather than the true edges. Consequently the edges found at the zero crossings must often be subjected to a false edge elimination procedure [80], [81], which inevitably breaks the otherwise closed edge contours. Even without considering this effect of the edge authentication, the edges from zero crossings usually suffer from inaccurately located region boundaries because the edge operators are tuned to a prespecified spatial resolution [11]. Consequently boundaries between objects with other spatial resolutions may either be missed or inaccurately located. Furthermore, there is an intrinsic drawback in finding connected regions using this type of edge operator. Region finding through zero crossings is essentially image coloring with only two colors. This forces all closed contours to be shared by exactly two connected regions. Consequently, complex region formations in an image may be poorly encoded and regions with distinct features may be falsely linked together. On the other hand, edge detectors based on local intensity differences, *e.g.* the Sobel operator [82], are able to locate edges more accurately at region boundaries, often at the expense of thick edges and false edges. Edge linking and thining are usually needed to alleviate these problems [83]. As expected, they generally cannot provide closed edge contours for region formation. The edge based, graph theoretic segmentation method proposed in this chapter intends to combine the advantages of both types of edge detectors, *i.e.* to accurately locate region boundaries and at the same time guarantee the formation of closed edge contours. Here we propose to compute edges using a local derivative operator, and then apply the optimal graph theoretic clustering algorithm to link strong edges together to form closed contours by searching for minimum cuts with small value in an adjacency graph.

In order to apply the graph theoretic clustering algorithm to image segmentation, we define an edge element between every pair of neighboring pixels in the image as a separating element, in a similar manner to the line processes used in Gibbs random field models [27], and then compute the edge strength using a local derivative operator. In this dissertation only a first order neighborhood system

is used although there is no inherent restriction on using a higher order neighborhood. An adjacency graph $\mathcal{G}$ is constructed from the image: each vertex of $\mathcal{G}$ corresponds to a pixel of the image, and an arc is placed between two vertices if their associated pixels are neighbors of each other. Note that there is a one-to-one correspondence between an arc in $\mathcal{G}$ and an edge element in the image. The edge strength is computed as a function of intensity differences between the pixel pair and between pixels in their vicinity, by using masks similar to those shown in Fig. 5.1. The bigger the intensity difference, the stronger is the edge element separating the pixel pair.

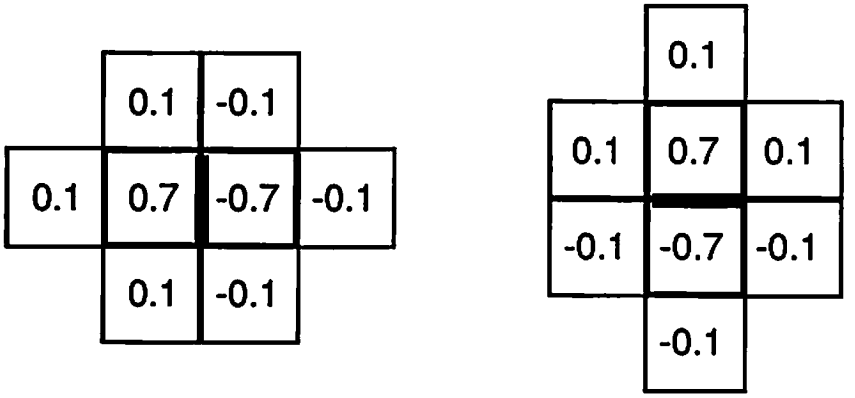


Figure 5.1: Masks for computing the edge strength: (a) edge between horizontal neighboring pixels; (b) edge between vertical neighboring pixels.

Let $x_{i,j}$ be the grey scale intensity of the pixel at coordinate (i,j), and let $D^H_{i,j}$ and $D^V_{i,j}$ be the edge elements defined between the pixel pairs $(x_{i,j}, x_{i,j+1})$ and $(x_{i,j}, x_{i+1,j})$ respectively, then

$$D^H_{i,j} = \left| \frac{\delta \cdot (x_{i,j} - x_{i,j+1}) + (x_{i-1,j} - x_{i-1,j+1}) + (x_{i+1,j} - x_{i+1,j+1}) + (x_{i,j-1} - x_{i,j+2})}{(\delta + 3) \cdot \sigma} \right| \quad (5.1)$$

$$D^V_{i,j} = \left| \frac{\delta \cdot (x_{i,j} - x_{i+1,j}) + (x_{i,j-1} - x_{i+1,j-1}) + (x_{i,j+1} - x_{i+1,j+1}) + (x_{i-1,j} - x_{i+2,j})}{(\delta + 3) \cdot \sigma} \right| \quad (5.2)$$

where δ and σ are control parameters. The parameter δ controls the smoothing effect of the edge masks. For the brain MR images used in our experiments, we have observed that $\delta = 7$ seems to be a good compromise between reducing noise sensitivity and preventing thick edge lines. For noisier images, $\delta = 1$ and/or larger edge masks may be necessary.

Given the adjacency graph $\mathcal{G}$, the problem of image segmentation is solved by partitioning $\mathcal{G}$ into mutually exclusive subgraphs such that the largest inter-subgraph maximum flow is minimized. Note that each of these subgraphs will contain a set of connected pixels whose union represents a spatially connected region of the image. The minimum cuts used for partitioning the adjacency graph $\mathcal{G}$ also have an appealing interpretation in image segmentation. For an arbitrary cut in the graph $\mathcal{G}$, the edge elements corresponding to the arcs belonging to the cut form a closed edge contour. The value of a cut is equal to the sum of the capacities of its arcs. Hence if a small arc capacity is assigned to a strong edge element and *vice versa*, then the cuts with small value correspond to closed contours which contain strong edge elements. Conversely, isolated strong edges will not produce boundaries in the segmented image, since there is a high cost associated with the inclusion of weak edges (large arc capacities) necessary to form a closed boundary through these isolated edges. Ideally, the capacity function for a cut should not penalize long boundaries, but rather penalize the presence of gaps in the boundaries. This is particularly important in the segmentation example for an MR image of the brain presented in Chapter 6. In that case, the boundary between the cortex and CSF and between the cortex and the white matter is highly convoluted. Fortunately this penalization of long boundaries can be compensated for to an acceptable, if not negligible, level by properly selecting a function that maps the edge strength into the arc capacity. Previously we used the negative exponential of the square of the edge strength to define the arc capacity [66]. Although the results have generally been satisfactory, we note that the capacity function drops to zero too quickly as the edge strength increases, which results in indistinguishable capacities (10^{-12} is used as the lower bound for arc capacities to prevent underflow). Consequently isolated regions with very few pixels are more likely to occur when clustering pixels. Of course this problem may be alleviated by increasing the value of the control parameter σ. However a small value for σ is necessary to make a clear distinction between strong and weak edges. Based on our experiments, we found the following arc capacity function to be a more appropriate choice. Let $v_{i,j}$ be the vertex corresponding to pixel $x_{i,j}$, then the capacities for the arcs connecting $v_{i,j}$ to $v_{i,j+1}$ and to $v_{i+1,j}$, $C^H_{i,j}$ and $C^V_{i,j}$

respectively, are defined as:

$$C^p_{i,j} = \begin{cases} e^{-(D^p_{i,j})^2} & \text{if} \quad 3 > D^p_{i,j}, \\ e^{-3D^p_{i,j}} & \text{if} \quad D^p_{i,j} \geq 3, \end{cases} \qquad p = H \text{ or } V. \tag{5.3}$$

The relationships between elements of the adjacency graph $\mathcal{G}$ and the image are summarized in Fig. 5.2. Note that all minimum cuts in $\mathcal{G}$ form edge contours of the image; but the converse is not necessarily true.

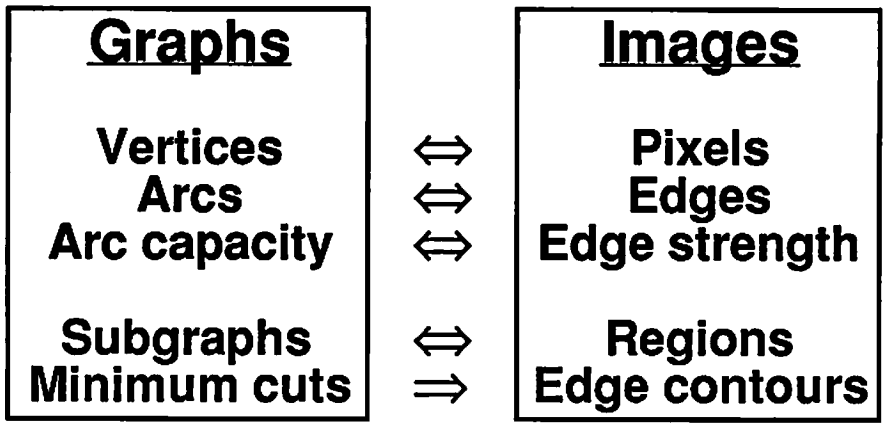


Figure 5.2: Correspondences between elements of the graph and the image

5.2 Supervised Graph Theoretic Segmentation

In the previous section, we showed how the graph theoretic clustering algorithm can be applied for searching for closed edge contours. The edges were computed using a differential operator defined on a small image window. This type of edge detector is simple and often robust, but usually lacks the flexibility to easily incorporate prior information about the image. In this section we intend to incorporate knowledges on pixel intensity distributions into the segmentation algorithm, achieved by formulating the edge detection as a hypothesis testing problem. In particular, we will study in detail the case where the images are modeled as a mosaic of uniform intensity regions corrupted by additive white Gaussian noise.

5.2.1 Edge detection using hypothesis testing

Edge detection using a hypothesis testing approach has been proposed and studied by many researchers. Yakimovsky [84] defines the edges as separating elements between adjacent pixels, and formulates the edge detection as testing the hypothesis that pixels on opposite sides of the edge site have distinct features. Pixels in two small windows, one on each side of the edge, are used in the proposed likelihood test to obtain improved detection power. Kundu and Mitra [85] also suggest a similar approach using rank statistics. Related work also includes detecting edges using a facet image model [86] and determining moving edges in an image sequence [87]. Here we propose a homogeneity testing approach to edge detection that is similar to Yakimovsky's method [84] but is able to incorporate prior knowledge of the pixel intensity distribution.

Formulation

Assume that all pixels of the image under study are random samples from the same family of probability densities with the sole exception of the values of some parameters. To be more specific, we make the following assumptions:

1. Any pixel x_i of the image X is a sample from one of K classes.

2. The least informative prior P_k is assumed for each class, *i.e.* $P_k = \frac{1}{K}$ for $k = 1, \cdots, K$. This assumption is due to a lack of information rather than a prior belief of the image.

3. The class-conditional probability densities $f(x_i|\theta_k)$ are known for $k = 1, \cdots, K$, where θ_k is the parameter vector for the k-th class.

The edges are defined as separating elements between adjacent pixel pairs. Let x_i and x_j be two neighboring pixels in the image. Detecting edge presence between x_i and x_j is formulated as a statistical hypothesis testing problem. Under the null hypothesis H_0, x_i and x_j are samples from the same class, hence there is no edge between them; under the alternative hypothesis H_1, x_i and x_j are samples from different classes hence there is an edge element placed between the pixel pair. Although the Neyman-Pearson test is the most powerful for testing

these hypotheses [53], the complexity involved with the test makes it very difficult to manipulate. Here we choose instead the likelihood ratio test (LRT): reject H_0 if

$$\Lambda = \frac{\max_k \ f(x_i|\theta_k)f(x_j|\theta_k)}{\max_{k_1 \neq k_2} f(x_i|\theta_{k_1})f(x_j|\theta_{k_2})} < C, \tag{5.4}$$

where C is determined by a given significance level α of the test, which is defined as the probability of the type I error $P(H_1|H_0)$:

$$\alpha = P(H_1|H_0) = \frac{1}{K}\sum_{k=1}^{K} \int_{\Lambda<C} f(x_i|\theta_k)f(x_j|\theta_k)dx_i dx_j. \tag{5.5}$$

As illustrated in the case study presented below, incorporating the pixel intensity distribution can substantially improve the power of detecting the edge presence between adjacent pixels.

Case study: two normal populations with equal variance

Here we will present an analysis on the behavior and the performance of the likelihood ratio test (LRT) given in (5.4) for detecting edges in images that are modeled as two arbitrarily shaped uniform regions corrupted by white Gaussian additive noise. In other words, we assume the two class-conditional probability densities be $\mathcal{N}(\mu_1, \sigma^2)$ and $\mathcal{N}(\mu_2, \sigma^2)$ where μ_1, μ_2 and σ^2 are all known. Without loss of generality, we can assume $\mu_1 < \mu_2$ and $\sigma = 1$.

For this special case the LRT for rejecting H_0, defined in (5.4), becomes:

$$\Lambda = \frac{\max\left(\mathcal{N}(x_i|\mu_1,1)\mathcal{N}(x_j|\mu_1,1), \mathcal{N}(x_i|\mu_2,1)\mathcal{N}(x_j|\mu_2,1)\right)}{\max\left(\mathcal{N}(x_i|\mu_1,1)\mathcal{N}(x_j|\mu_2,1), \mathcal{N}(x_i|\mu_2,1)\mathcal{N}(x_j|\mu_1,1)\right)} < C. \tag{5.6}$$

Because of the symmetry between x_i and x_j, we only need to consider $x_i \leq x_j$. Note that due to the monotonicity of the normal density [88],

$$\mathcal{N}(x_i|\mu_1,1)\mathcal{N}(x_j|\mu_2,1) \geq \mathcal{N}(x_i|\mu_1,1)\mathcal{N}(x_j|\mu_1,1)$$

is always true when $x_i \leq x_j$. Hence there are only two cases to be considered. If $\mathcal{N}(x_i|\mu_1,1)\mathcal{N}(x_j|\mu_1,1) \geq \mathcal{N}(x_i|\mu_2,1)\mathcal{N}(x_j|\mu_2,1)$, which can be shown to be

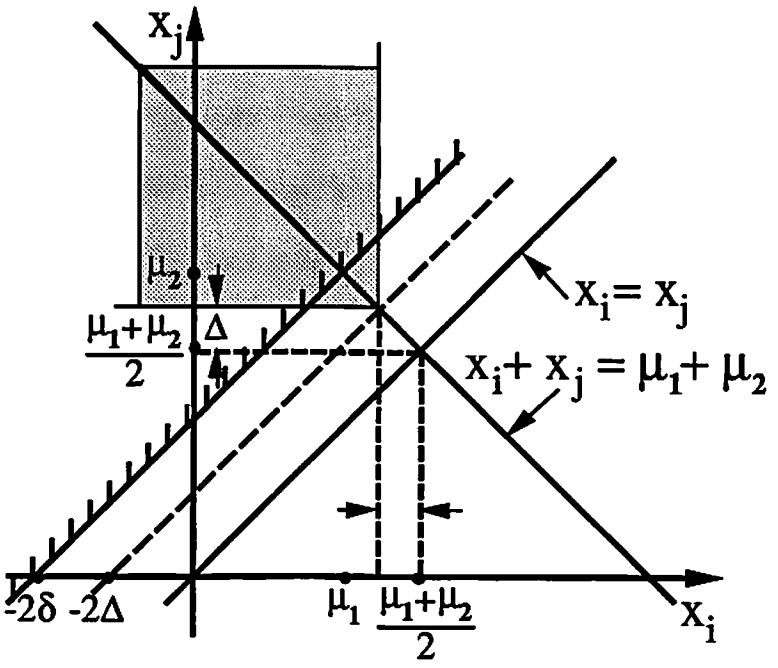


Figure 5.3: The critical region for the LRT

equivalent to $x_i + x_j \leq \mu_1 + \mu_2$, the LRT can be simplified to

$$x_j > \frac{\mu_1 + \mu_2}{2} + \Delta \quad \text{for rejecting } H_0, \text{ if } x_i + x_j \leq \mu_1 + \mu_2. \tag{5.7}$$

Alternatively, if $\mathcal{N}(x_i|\mu_1, 1)\mathcal{N}(x_j|\mu_1, 1) < \mathcal{N}(x_i|\mu_2, 1)\mathcal{N}(x_j|\mu_2, 1)$, or equivalently $x_i + x_j > \mu_1 + \mu_2$, the LRT becomes

$$x_i < \frac{\mu_1 + \mu_2}{2} - \Delta \quad \text{for rejecting } H_0, \text{ if } x_i + x_j > \mu_1 + \mu_2. \tag{5.8}$$

Combining the above results, the critical region for rejecting the null hypothesis H_0 is

$$x_i < \frac{\mu_1 + \mu_2}{2} - \Delta, \quad \text{and} \quad x_j > \frac{\mu_1 + \mu_2}{2} + \Delta \tag{5.9}$$

provided $x_i \leq x_j$ (Fig. 5.3). Note that the knowing the population means drastically changes the resulting critical region, which otherwise would be $|x_j - x_i| > 2\Delta' > 2\Delta$ for the same significance level. Given the value of Δ, the significance

level α for the test defined in (5.9) can be evaluated using

$$\alpha(\Delta) = \begin{cases} 2 \cdot \mathcal{Q}(\frac{\mu_2-\mu_1}{2} - \Delta) \cdot \mathcal{Q}(-\frac{\mu_2-\mu_1}{2} - \Delta) & \text{if } \Delta \geq 0 \\ 1.0 - \mathcal{Q}^2(\frac{\mu_2-\mu_1}{2} + \Delta) - \mathcal{Q}^2(-\frac{\mu_2-\mu_1}{2} + \Delta) & \text{if } \Delta < 0 \end{cases} \tag{5.10}$$

where

$$\mathcal{Q}(t) = \int_{-\infty}^{t} \mathcal{N}(x|0,1)dx.$$

Fig. 5.4 shows the plot of $\alpha(\Delta)$ for $\mu_2 - \mu_1 = 2$.

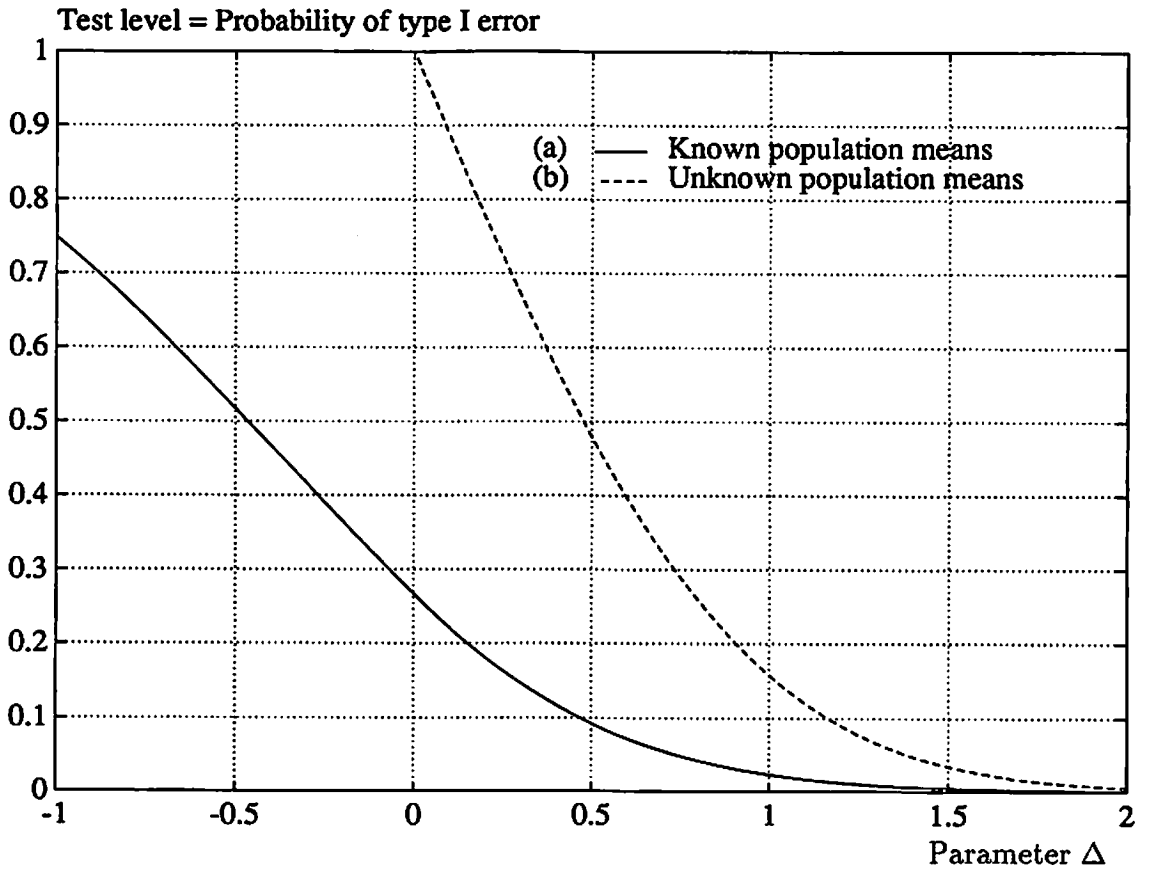


Figure 5.4: Significance levels of the LRTs for different values of Δ given $\mu_2 - \mu_1 = 2$ and $\sigma = 1$: (a) LRT for known population means; (b) LRT for unknown population means.

Now let look at the power of the test, or equivalently the size of the type II

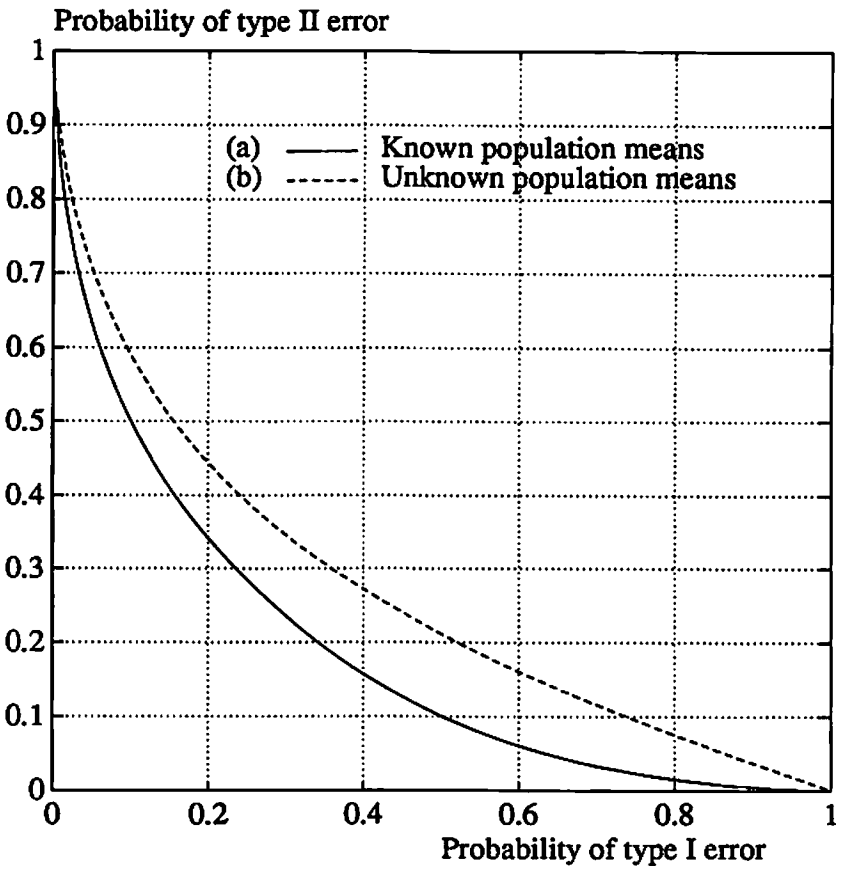


Figure 5.5: The β-α plot given $\mu_2 - \mu_1 = 2$ and $\sigma = 1$ for: (a) LRT for known population means; (b) LRT for unknown population means.

error β. By definition, it can easily show that β is computed as:

$$\beta(\Delta) = P(H_0|H_1) = \begin{cases} 1.0 - Q^2(\frac{\mu_2-\mu_1}{2} - \Delta) - Q^2(-\frac{\mu_2-\mu_1}{2} - \Delta) & \text{if } \Delta \geq 0 \\ 2 \cdot Q(\frac{\mu_2-\mu_1}{2} + \Delta) \cdot Q(-\frac{\mu_2-\mu_1}{2} + \Delta) & \text{if } \Delta < 0 \end{cases} \quad (5.11)$$

Note that increasing Δ decreases α but increases β. The conflict between α and β is evident in the plot of β versus α, parameterized by Δ. This β-α plot is shown in Fig. 5.5 for $\mu_2 - \mu_1 = 2$. One advantage of using the β-α plot in place of the $\beta(\Delta)$ curve is that the performance of different tests can be easily compared.

In order to evaluate the performance improvement of the LRT with the population means incorporated, let us also compute the sizes of the type I and II

errors for the LRT without knowning the population means. In this case, it can be easily shown that the critical region for rejecting H_0 is

$$x_j - x_i > 2\Delta' \geq 0, \quad \text{given} \quad x_i \leq x_j.$$

where the threshold Δ' is determined by the test level α'. Note that only pixel intensity difference is used here by the edge detector. After proper substitution into Eq. (5.5), α' can be simplified to

$$\alpha'(\Delta') = 2 \cdot \mathcal{Q}(-\sqrt{2} \cdot \Delta'). \tag{5.12}$$

Similarly the size of the type II error β' is computed as

$$\beta'(\Delta') = \mathcal{Q}(\sqrt{2} \cdot (\frac{\mu_2 - \mu_1}{2} + \Delta') - \mathcal{Q}(\sqrt{2} \cdot (\frac{\mu_2 - \mu_1}{2} - \Delta'). \tag{5.13}$$

For the same mean difference of $\mu_2 - \mu_1 = 2$, the $\alpha'(\Delta')$ curve is also plotted in Fig. 5.4, and the β'-α' plot is shown in Fig. 5.5. By comparing the plots in these figures, we can observe that given a test level α, incorporating the population means allows the detection of edges with relatively small intensity difference, and the size of the type II error is also noticeably reduced.

The above analysis can be extended to cases of multiple normal populations and non-Gaussian populations. Similar improvement in performance is expected.

An example

Fig. 5.6.a shows a 4 region computer generated image of 128 × 128 pixels with 256 grey levels. The intensities for the 4 uniform regions are 80, 112, 144 and 176, respectively. The image was then corrupted by Gaussian additive noise with a standard deviation $\sigma = 16$ (Fig. 5.6.b). For reference, the pixel-wise maximum likelihood classifier applied to this noisy image (thresholds at 96, 128, 160) results in a 21.1% error percentage. Edge strengths are computed using small windows, shown in Fig. 5.1, to reduce noise sensitivity. For the case of unknown population means, edges are computed using Eqs. (5.1) and (5.2) with $\delta = 1$, and are then thresholded at approximately 1.4 times the standard deviation which corresponds

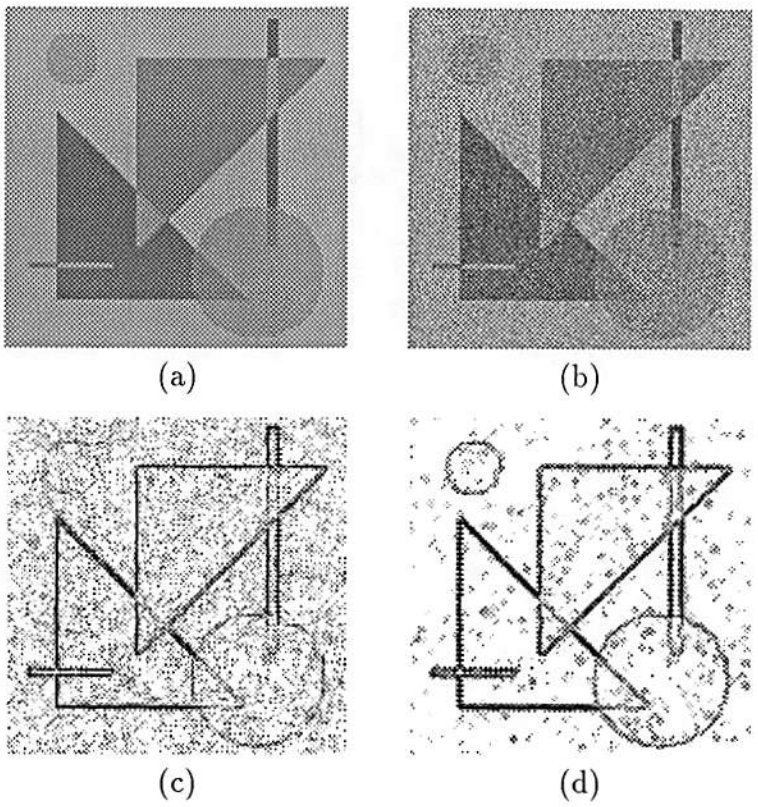


Figure 5.6: Edge detection of a 4 region hand-draw image: (a) noise free image; (b) noisy image; (c) detected edges without knowing the 4 population means ($\alpha = 0.05$); (d) detected edges using the 4 population means ($\alpha = 0.05$).

to a significance level $\alpha = 0.05$. Note that the standard deviation has been reduced by half due to the use of the window. The detected edges are displayed in Fig. 5.6.c, where the maximum of $D^H_{i,j}$ and $D^V_{i,j}$ are shown at coordinate (i,j). For the case of known population means, Eqs. (5.15) and (5.16) are used to compute edge strength with $\delta_X = \delta' = 1$. For the same test level $\alpha = 0.05$, the computed edges are thresholded at approximately 0.75 times the standard deviation. The resulting edges are displayed in Fig. 5.6.

5.2.2 Segmentation algorithm

Here we propose a supervised algorithm for segmenting images that can be applied to the image model described in the previous subsection, *i.e.* a mosaic of uniform regions corrupted by noise. Based on the performance analysis of detecting edges using homogeneity testing, we suggest the use of the negative logarithmic of the likelihood ratio between the hypotheses of edge absence and edge presence as a measure of edge strength. Weighted intensity averages of pixels within a small window are used for computing the edge strength in order to reduce its noise sensitivity. For the case of K equal variance normal populations (*i.e.* uniform regions with additive Gaussian noise), $D^H_{i,j}$ and $D^V_{i,j}$, the edge elements defined between the pixel pairs $(x_{i,j}, x_{i,j+1})$ and $(x_{i,j}, x_{i+1,j})$ respectively, are computed as follows:

$$\begin{aligned}(D^H_{i,j})^2 &= -\sigma'^2 \log \frac{\max_k \mathcal{N}(y_{i,j}|\mu_k, \sigma^2_Y)\mathcal{N}(y'_{i,j}|\mu_k, \sigma^2_Y)}{\max_{k_1,k_2} \mathcal{N}(y_{i,j}|\mu_{k_1}, \sigma^2_Y)\mathcal{N}(y'_{i,j}|\mu_{k_2}, \sigma^2_Y)} \\ &= \frac{\sigma'^2}{2\sigma^2_Y}\{\min_k [(y_{i,j} - \mu_k)^2 + (y'_{i,j} - \mu_k)^2] \\ &\quad - \min_{k_1,k_2} [(y_{i,j} - \mu_{k_1})^2 + (y'_{i,j} - \mu_{k_2})^2]\}\end{aligned} \tag{5.14}$$

and

$$\begin{aligned}(D^V_{i,j})^2 &= -\sigma'^2 \log \frac{\max_k \mathcal{N}(z_{i,j}|\mu_k, \sigma^2_Z)\mathcal{N}(z'_{i,j}|\mu_k, \sigma^2_Z)}{\max_{k_1,k_2} \mathcal{N}(z_{i,j}|\mu_{k_1}, \sigma^2_Z)\mathcal{N}(z'_{i,j}|\mu_{k_2}, \sigma^2_Z)} \\ &= \frac{\sigma'^2}{2\sigma^2_Z}\{\min_k [(z_{i,j} - \mu_k)^2 + (z'_{i,j} - \mu_k)^2] \\ &\quad - \min_{k_1,k_2} [(z_{i,j} - \mu_{k_1})^2 + (z'_{i,j} - \mu_{k_2})^2]\}.\end{aligned} \tag{5.15}$$

where

$$
\begin{aligned}
y_{i,j} &= \frac{\delta \cdot x_{i,j} + x_{i-1,j} + x_{i+1,j} + x_{i,j-1}}{\delta + 3}, \\
y'_{i,j} &= \frac{\delta \cdot x_{i,j+1} + x_{i-1,j+1} + x_{i+1,j+1} + x_{i,j+2}}{\delta + 3}; \\
z_{i,j} &= \frac{\delta \cdot x_{i,j} + x_{i,j-1} + x_{i,j+1} + x_{i-1,j}}{\delta + 3}, \\
z'_{i,j} &= \frac{\delta \cdot x_{i+1,j} + x_{i+1,j-1} + x_{i+1,j+1} + x_{i+2,j}}{\delta + 3}.
\end{aligned}
$$

In (5.15) and (5.16), $\sigma_Y^2 = \sigma_Z^2 = \frac{\delta^2+3}{(\delta+3)^2}\sigma_X^2$ and σ'^2 is a tuning parameter. Note that we implicitly assume that all four of the samples used for computing $y_{i,j}$ ($y'_{i,j}$, $z_{i,j}$ or $z'_{i,j}$) are drawn from the same population.

Once $D_{i,j}^H$ and $D_{i,j}^V$ are computed, Eq. (5.3) is used to convert them into corresponding arc capacities. Note that the resulting arc capacity is closely related to the likelihood ratio of the hypothesis of edge absence against the hypothesis of edge presence. After applying the clustering algorithm to the resulting adjacency graph $\mathcal{G}$, each of the clusters can be assigned to one of the the K classes using the maximum likelihood classifier: the cluster is classified to the k-th class if its sample mean is closest to μ_k.

So far we have assumed that the K population means $\{\mu_k\}$ are known. In practice, these parameters often need to be estimated from the image under study. Like most other supervised segmentation techniques, we assume that these parameters can be estimated from the image with reasonable accuracy. Depending on the particular images under study, the pixel intensity distributions may be estimated from the histogram or co-occurrence matrices, or training regions.

5.3 Graph Theoretic Clustering Applied to Hierarchical Segmentation

In Chapter 3, we described a hierarchical image segmentation algorithm. The algorithm consists of three major steps: split-and-merge, patch forming and patch clustering. In the patch clustering step, a constrained ML agglomerative clustering

technique is used to reduce the number of regions by properly grouping the patches together. Here we will apply instead the graph theoretic clustering technique for the same purpose.

An adjacency graph is formed as follows: each vertex of the graph corresponds to a patch of the image; an arc is placed between a pair of vertices if their associated patches are spatially connected to each other. The capacity function, assigned to each arc, is chosen to reflect the likelihood of the patches (vertices) belonging to the same cluster. Having formed the graph, the final clustering is performed as described in Section 4.3.

As expected, the choice of a particular arc capacity function plays a decisive role in the clustering outcome. We choose the capacity function to be be proportional to the number of neighboring pixels between linked patches, in a similar manner to the edge based case described in Section 5.1, and proportional to the negative exponential of a distance measure between the GMRF feature estimates of the two patches. One of the simplest capacity functions of this kind is shown in Eq. (4.2) in Section 4.3, which was also used in our early work [65]. Based on our experimental results, we note that the following capacity function performs better:

$$C_{i,j} = \begin{cases} k_{i,j} \cdot \mathrm{e}^{-(\frac{\mu_i - \mu_j}{\sigma})^2} & \text{if} \quad 3 > \left|\frac{\mu_i - \mu_j}{\sigma}\right| \\ k_{i,j} \cdot \mathrm{e}^{-3|\frac{\mu_i - \mu_j}{\sigma}|} & \text{if} \quad \left|\frac{\mu_i - \mu_j}{\sigma}\right| \geq 3 \end{cases} \tag{5.16}$$

where $k_{i,j}$ is the number of neighboring pixels between vertices v_i and v_j, μ_i and μ_j are the sample means of components i and j.

In our experiments, where we are primarily interested in segmenting MR images, the above arc capacity function has been found to be appropriate. For other types of images where the means of the GMRFs are insufficient for distinguishing different regions, more elaborated capacity functions may be formulated in terms of the likelihood ratio involving all the model parameters and the boundary conditions.

Chapter 6

Experimental Results

This chapter presents experimental results of applying the hierarchical and the graph theoretic segmentation algorithms, described in Chapters 3 and 5 respectively , to computer generated images and then to MR images of the human brain.

6.1 Segmentation of Computer Generated Images

Although the problems addressed in this section are somewhat artificial, the experiments can help to evaluate the performance of the segmentation schemes.

Figure 6.1.a shows a 4 region computer generated image of 128 × 128 pixels with 256 grey levels. This image is adapted from the example of [50], but it is not an exact replica due to the lack of specification. The intensities for the 4 uniform regions are 80, 112, 144 and 176, respectively. This image is then corrupted by a Gaussian additive noise with a standard deviation equal to 16 which corresponds to a signal-to-noise ratio SNR = 2 (Fig. 6.1.b). The SNR is defined as the ratio between the minimum intensity difference between the mean of these uniform regions and the standard deviation of the noise. The pixel-wise maximum likelihood classifier applied to this noisy image (thresholds at 96, 128, 160) gives a 21.1% error percentage. This error rate is provided as a reference for other segmentation schemes. In contrast, if the same ML classifier is applied to

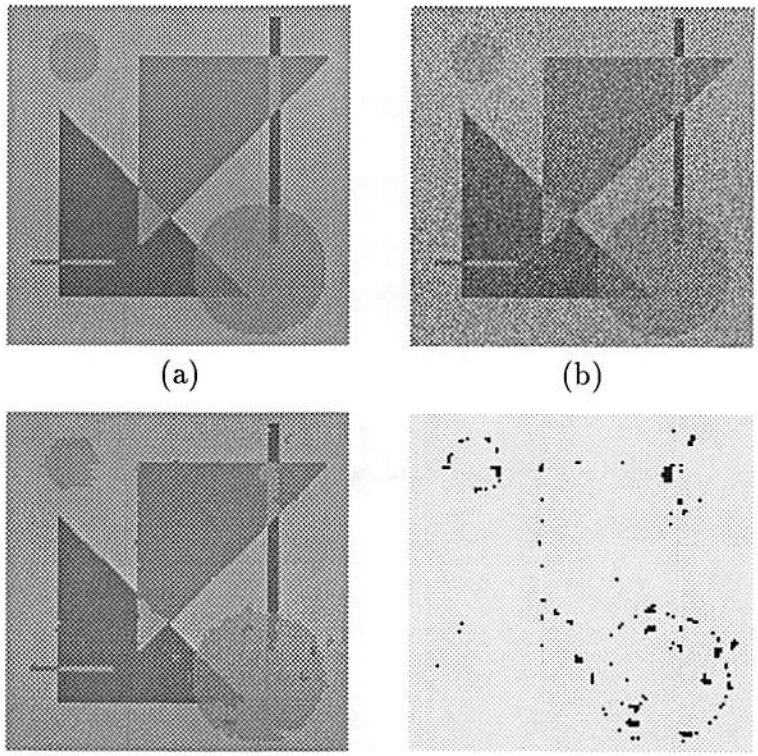


Figure 6.1: Hierarchical segmentation of 4 region hand-drawn image: (a) noise-free image, (b) noisy image (SNR = 2), (c) segmentation, (d) error (1.3%)

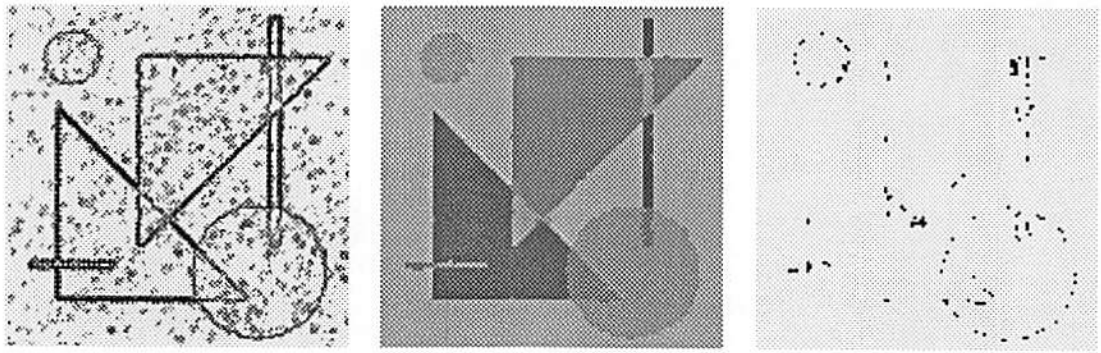


Figure 6.2: Supervised graph theoretic segmentation a noisy 4 region hand-draw image (SNR=2): (a) computed edge image before clustering; (b) 4 region segmentation; (c) error (0.946%).

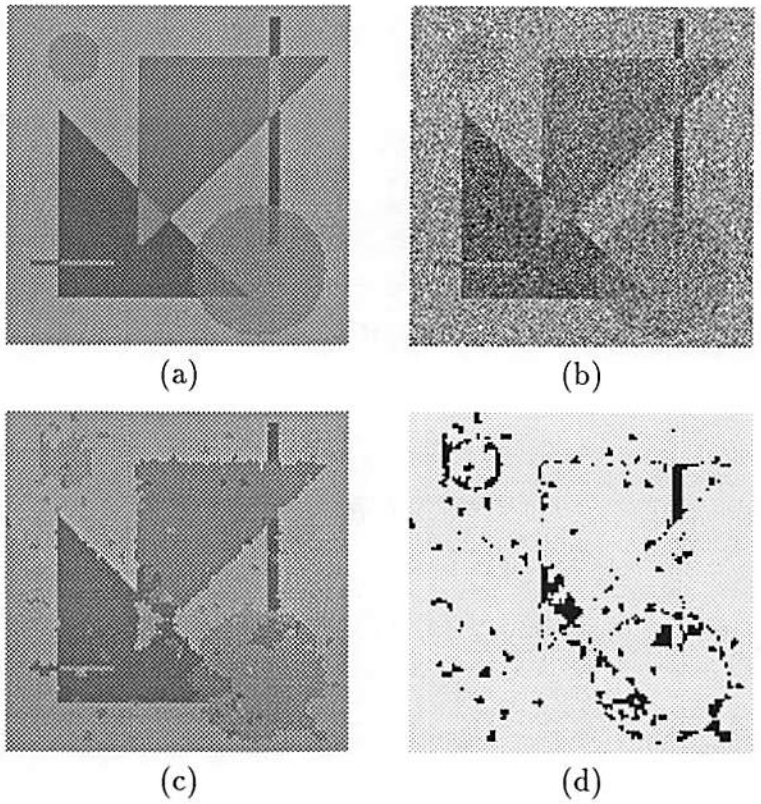
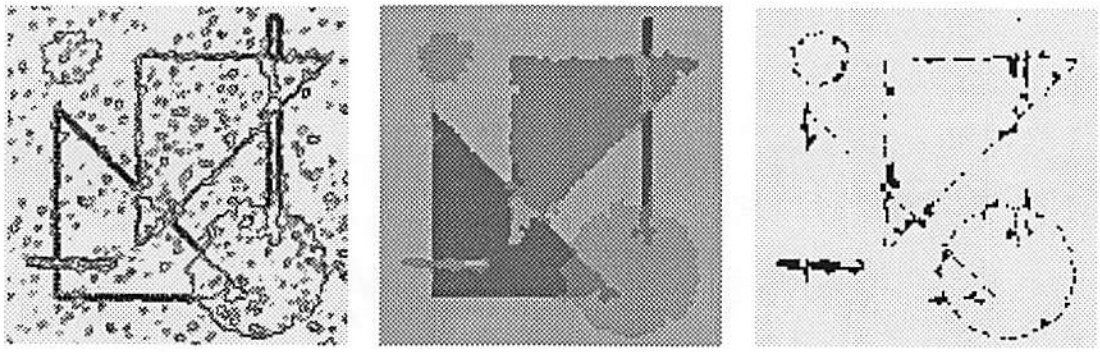


Figure 6.3: Hierarchical segmentation of 4 region hand-drawn image: (a) noise-free image, (b) noisy image (SNR = 1), (c) segmentation, (d) error (7.3%)

the patches resulting from the unsupervised hierarchical segmentation algorithm described in Chapter 3, the error drops to only 1.3% (Fig. 6.1.d).

The same noisy image is also segmented using the supervised version of the graph theoretic segmentation algorithm. The arc capacities are computed according to Eqs. 5.15, 5.16 and 5.3 with $\sigma'^2 = 2\sigma_Y^2 = 2\sigma_Z^2$, $\delta = 2$. Fig. 6.2.a shows the computed edges before applying graph theoretic clustering. A partially equivalent tree is constructed at threshold $\mathcal{F}_T = 0.1$. After merging clusters with very few pixels into their most similar neighboring clusters, the final segmentation (Fig. 6.2.b) achieves an error percentage of 0.946%. The erroneously classified pixels are displayed in Fig. 6.2.c.

A second case is shown in Fig. 6.3, where the noise level is doubled which corresponds to a SNR = 1 (Fig. 6.3.b). The hierarchical segmentation algorithm results in a segmentation error of 7.3% (Fig. 6.3.d); and the supervised graph



(a) (b) (c)

Figure 6.4: Supervised graph theoretic segmentation a noisy 4 region hand-draw image (SNR=1): (a) computed edge image before clustering; (b) 4 region segmentation; (c) error (3.99%).

theoretic segmentation algorithm reduces the error rate to 3.99% (Fig. 6.4). In comparison, the error rate of using the pixel-wise ML classifier is 40.3%.

6.2 MR image segmentation

MR imaging is a powerful non-invasive diagnostic tool which produces anatomical images of the human body based on the nuclear magnetic resonance properties of hydrogen. This imaging modality is able to provide sharp, high resolution images of soft tissues of the human body. There are however undesirable characteristics of MR images which make the problem of segmentation and tissue classification in MR images a difficult one. For instance, the parameters in MR images for a given type of tissue may change from patient to patient, from scanner to scanner and with variations with the imaging protocol. Moreover they may also vary as a function of the tissue's location within the same image, due to the radiometric variability. The last is particularly disturbing, since this variation may results in undistinguishable features (*e.g.* pixel intensity) for two different types of tissue if they are far apart from each other. This constitutes a strong motivation for segmenting the image into connected patches so that local contrast displayed between adjacent tissue regions (neighboring patches) can be preserved.

The problem of feature ambiguity in MR images is experienced in the example

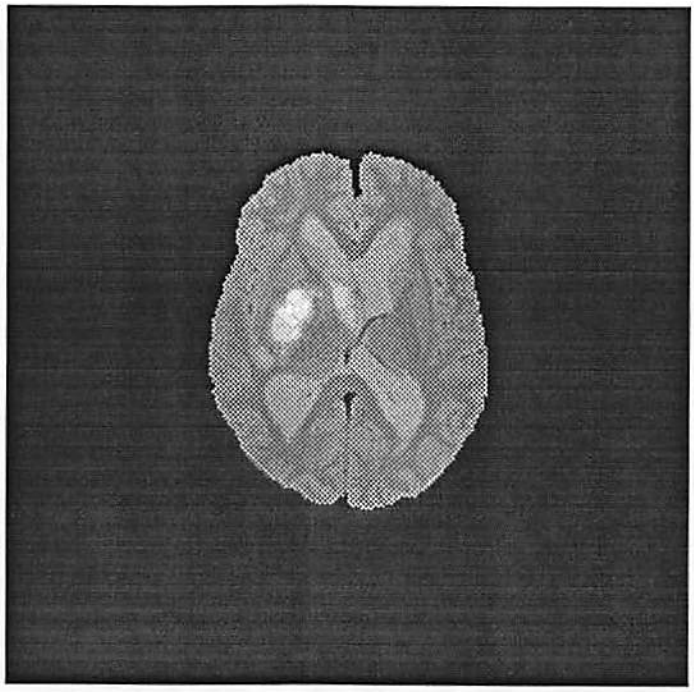


Figure 6.5: A cross-section MR image of the brain for a patient with brain tumor.

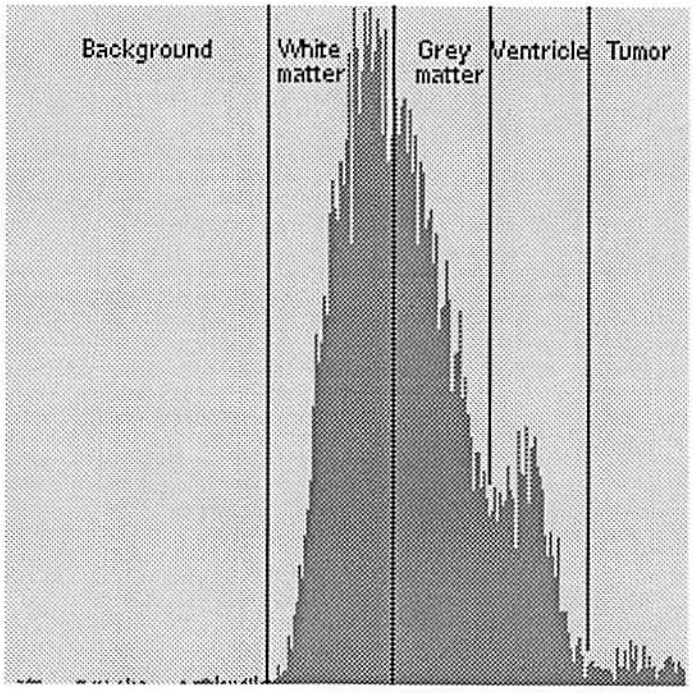
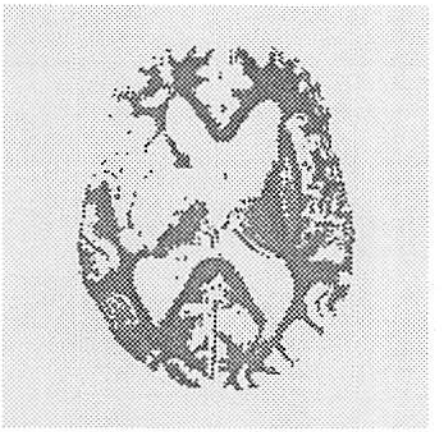
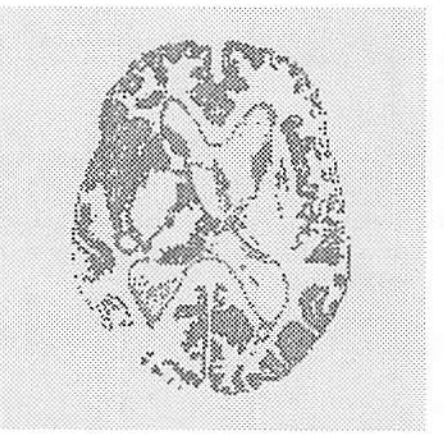


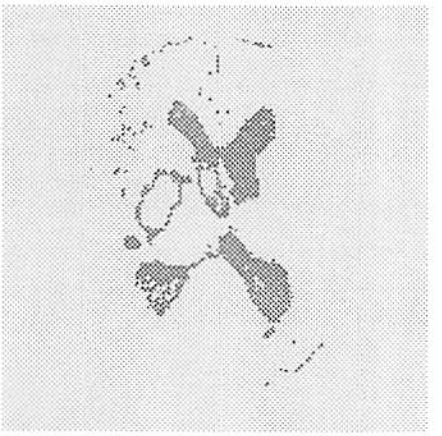
Figure 6.6: Histogram computed from the MR image in Fig. 6.5.



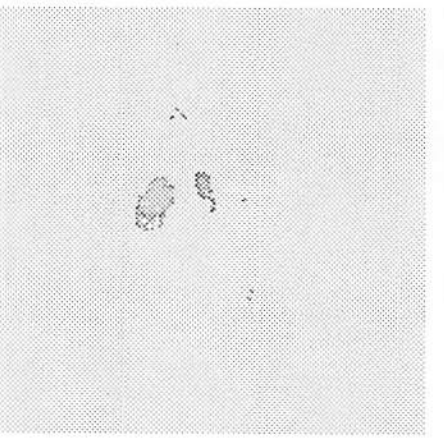
(a) White matter



(b) Grey matter



(c) Ventricles



(d) Tumor

Figure 6.7: The MR image in Fig. 6.5 segmented using thresholding.

shown in Figs. 6.5, although it is barely noticeable to our eyes. For the purpose of illustrating this problem, let us try to segment the MR image using pixel intensity thresholding. Fig. 6.5 shows a typical MR image of the human brain for a patient with a large lesion on the left side of the brain. The image size is 256×256 quantized to 8 bits per pixel. Note that the skull portion of the image has been removed from the image by finding the contour of the boundary between the skull and the enclosed soft tissues. In addition to the tumor, we can also identify other soft brain tissues such as ventricle, grey matter and white matter, which are shown in order of decreasing pixel intensity in that MR image. Ideally we would expect the histogram, computed from the image in Fig. 6.5, to show four well separated masses with clearly identifiable peaks and valleys. The true histogram (Fig. 6.6) is far from this ideal. The pixel intensity distributions for white and grey matters are so closely located that virtually no separation can be observed. By trial and error, the thresholds are set at 99.5, 144.5, 181.5 and 217.5 which achieved, to our subjective judgement, the least misclassified pixels. We can observe from Fig. 6.7 that a significant portion of the pixels, especially among those corresponding to white and grey matters, have been misclassified.

Numerous techniques have been proposed for MR image segmentation. One commonly used approach is to segment the image based on statistics of regions corresponding to different tissue types. Some researchers model the image as a doubly stochastic process and achieve the segmentation goal by maximizing the *aposteriori* probability (MAP) [16], [17]. Others seek to cluster pixels with similar features into regions while taking into account spatial relationships between adjacent pixels [20], [14], [15]. Despite much improved performances, these techniques share the same implicit assumption with the global intensity thresholding technique: the pixels corresponding to the same type of soft tissue have similar features. Their success depends very much on the validity of such a belief. Unfortunately for the MR images under study, this assumption is hardly justifiable as illustrated in Fig. 6.7.

There is another class of segmentation techniques which are based on edge detection. Ehricke [21] and Bomans *et al.* [22] have developed MR segmentation algorithms based on region finding through the Marr-Hildreth operator. Their

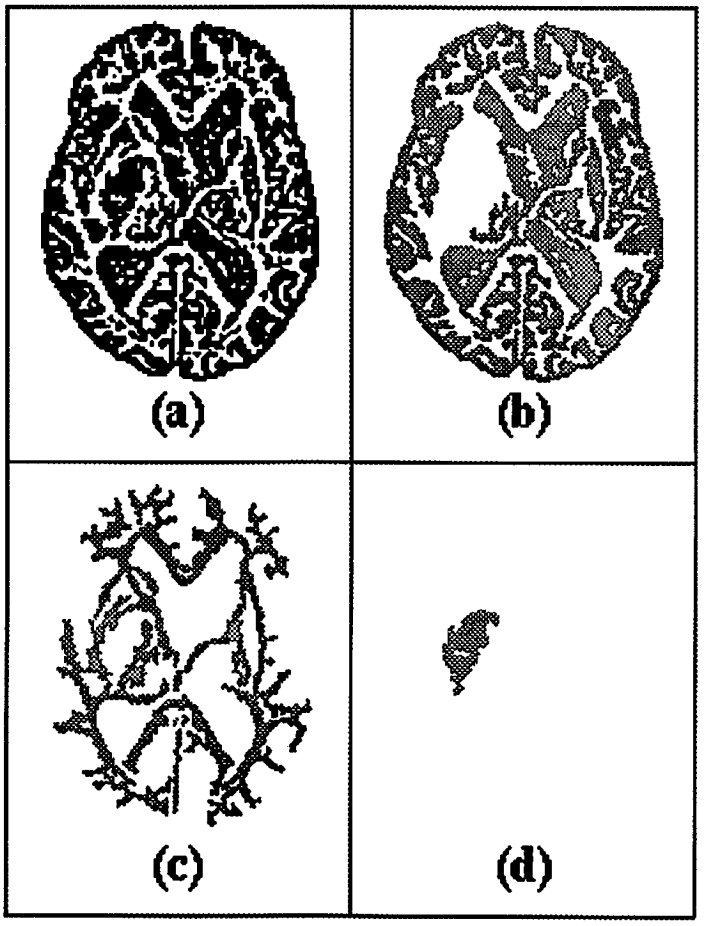


Figure 6.8: Results of applying Marr-Hildreth operator to the image in Fig. 6.5: (a) Regions from Marr-Hildreth operator; (b) - (d) the three largest connected regions in (a) after region growing.

results are among the most promising. However these techniques may still suffer from drawbacks associated with the Marr-Hildreth operator. While this operator guarantees the formation of regions with closed boundaries at zero crossings, these boundaries are often inaccurate. Fig. 6.8.a shows the filtered image of the MR image in Fig. 6.5 using the Marr-Hildreth operator. Fig. 6.8 displays a binary image where pixels in black correspond to negative responses to the operator, and pixels in white correspond to positive responses. By trial and error we found that a Marr-Hildreth operator with the scale parameter $\sigma = 1$, implemented using a 7×7 window, achieved the best results. All connected regions in Fig. 6.8.a were then extracted, and those with very small region size (≤ 4 in this case) were merged to their neighboring regions. The three largest resulting regions (excluding the background) are shown in Fig.6.8.b-d. While this method is effective at finding certain boundaries, it also mislocates many other boundaries. For instance, we can observe in Fig.6.8.c that edges for large regions of white matter are often located a few pixels inside the true boundary. This is due to the fact that a larger scale parameter is needed for locating more accurately those edges. A most noticeable problem here is that one must break connections between collections of pixels with distinct features. For example, in Fig. 6.8.b pixels corresponding to grey matter and the ventricles are mistakenly linked; these connections must be broken for accurate segmentation and labeling. This problem is mainly due to an intrinsic drawback of techniques based on zero crossings. All closed contours have to be shared by exactly two connected regions and thus more complex structures, where three or more regions are connected, will be forced into at most 2 regions. Various attempts to alleviate some of the problems mentioned above are also discussed in [21] and [22].

In the remainder of this section, we will present and discuss the results of MR image segmentation using our hierarchical and graph theoretic approaches.

6.2.1 MR image segmentation using the hierarchical approach

In order to illustrate how the hierarchical segmentation method works, we will present and discuss the partial results produced at each step of the algorithm

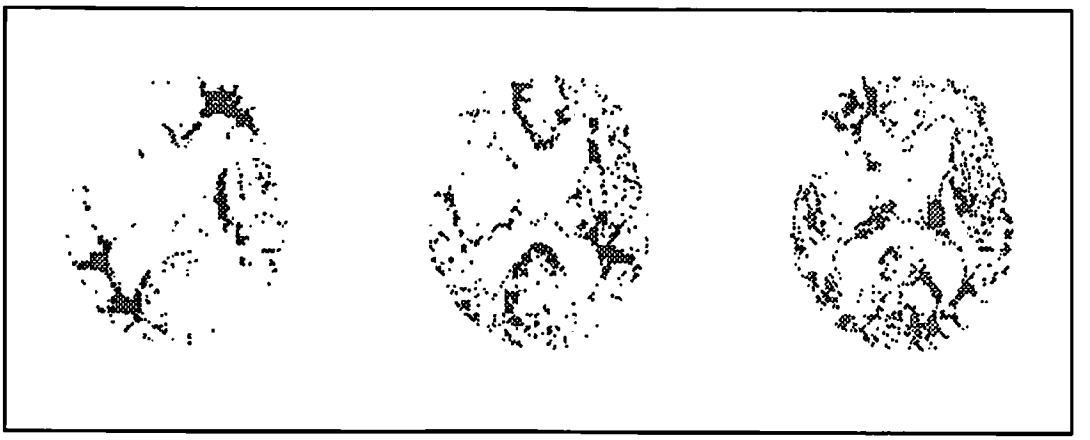


Figure 6.9: Partial results of split-and-merge.

when applied to the image in Fig. 6.5. In these examples all GMRF parameters are estimated from the images and no prior information of region partition is assumed.

A step-by-step segmentation example:

Here we apply the hierarchical segmentation algorithm to the MR image in Fig. 6.5, and show the results produced after each step of the algorithm.

Results of the split-and-merge: The split-and-merge algorithm produces 16 regions. Figure 6.9 shows three of these regions whose pixels correspond mostly to the white matter in the brain. Note that they are rather noisy. However our main goal at this point, as stated before, is not to produce a smooth segmentation but to find large homogeneous regions.

Results of patch forming: The connected component labeling algorithm is applied to each of the regions obtained from the split-and-merge. Figure 6.10.a shows all labeled patches displayed with their mean grey level while the unlabeled patched are shown in white. Note that the unlabeled patches fall mainly into the boundaries between different tissues. The unlabeled patches are then assigned a label through patch growing, the resulting patch mean image is shown in figure 6.10.b. Note that the main features of the original image are preserved. At this stage, the total number of labeled patches is 385.

Results of ML agglomerative patch clustering: Fig. 6.11 shows a few

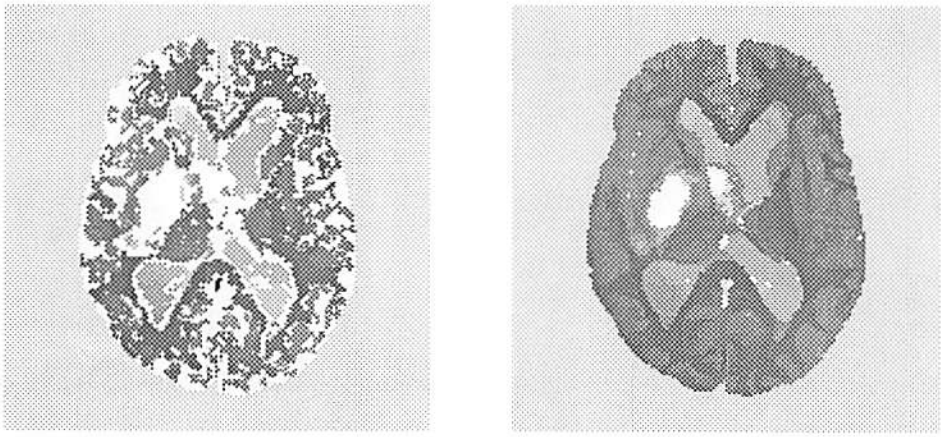


Figure 6.10: Resulting patches before and after patch growing.

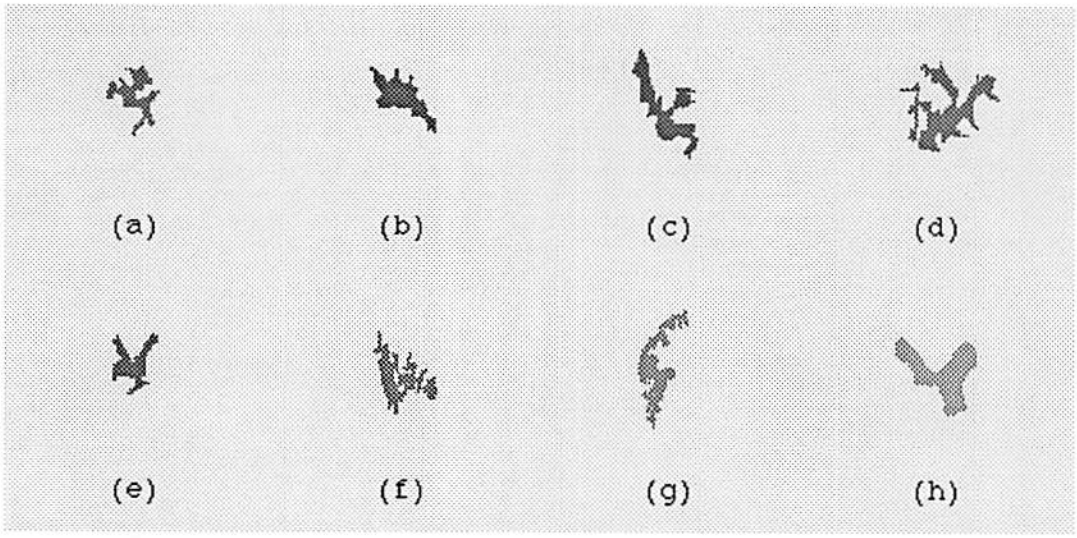
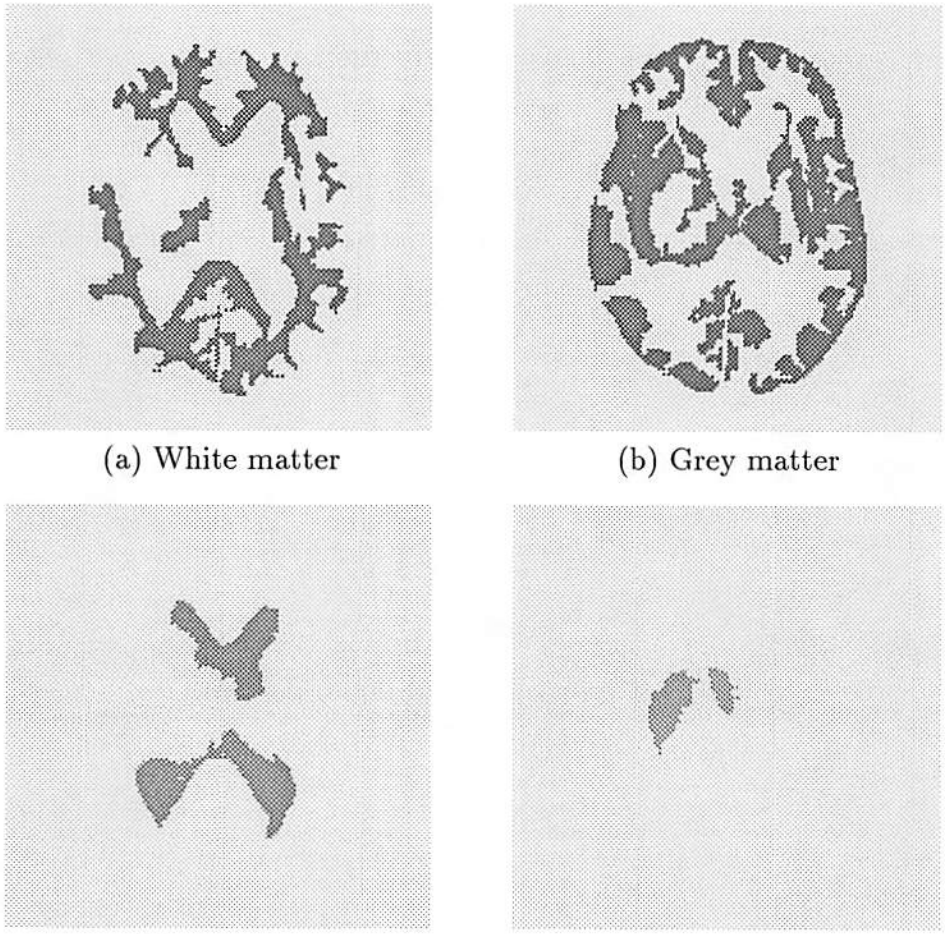


Figure 6.11: Examples of regions formed after ML agglomerative patch clustering.



(a) White matter (b) Grey matter

(c) Ventricles (d) Tumor

Figure 6.12: Final tissue classification after interactive labeling of regions found after ML agglomerative clustering.

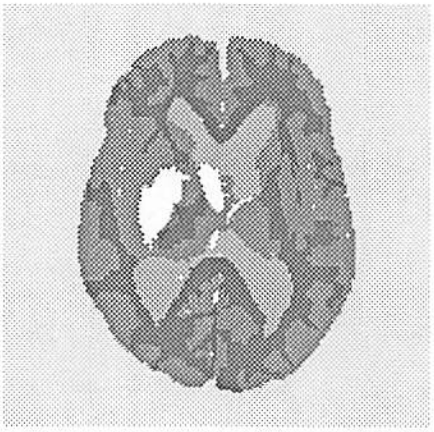


Figure 6.13: Combined segmentation map using the hierarchical segmentation approach (using agglomerative clustering).

resulting patches after constrained ML agglomerative patch clustering. In Figures 6.11.a-e the patches belong to white matter; in Fig. 6.11.f-g, grey matter; and in Fig. 6.11.h, ventricle. In this case we further classified the patches interactively into one of four tissue types. Figures 6.12.a-d shows the resulting segmented regions corresponding respectively to (a) white matter; (b) grey matter; (c) ventricle; and (d) tumor. In each figure only pixels classified to that label are displayed while the rest are set to white. The combined segmentation map is shown in Fig. 6.13.

Results of graph theoretic patch clustering: In place of ML agglomerative patch clustering, we also used the graph theoretic clustering algorithm to reduce the number of patches resulting from the patch forming step. An adjacency graph is constructed from the patches resulting from the split-and-merge and patch forming steps. The arc capacities are computed using Eq. (5.16). The patch clustering step reduces the number patched from 385 to 83. Fig. 6.14 shows four of the resulting regions after clustering. To obtain the final results, these regions were further classified interactively into one of four types. Figs 6.15.a-d show the segmented regions corresponding respectively to (a) white matter; (b) grey matter; (c) ventricle; and (d) tumor. In each figure, only pixels classified to that label are displayed while the rest are set to white. The union of the four regions is shown in Fig. 6.16.

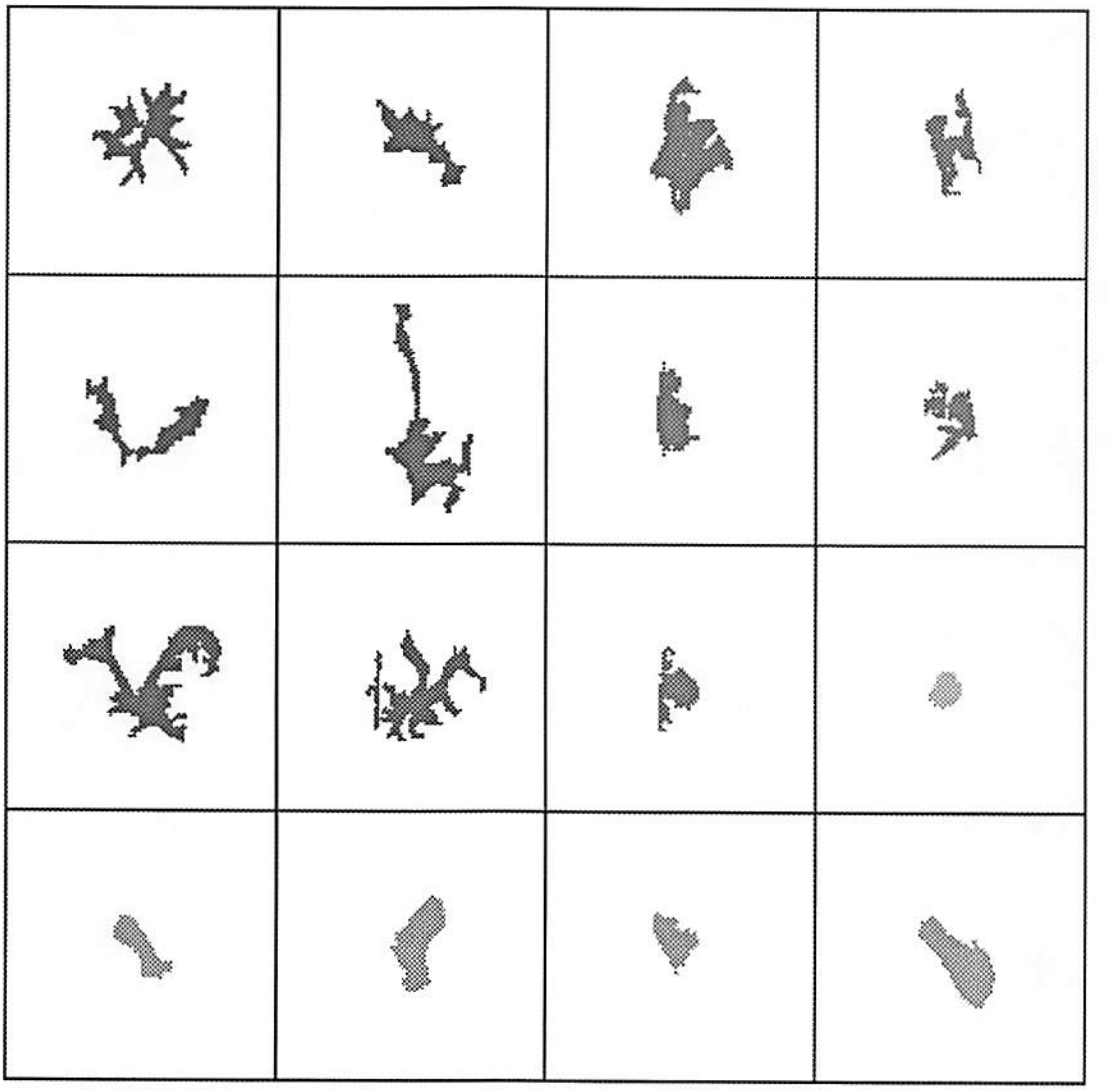
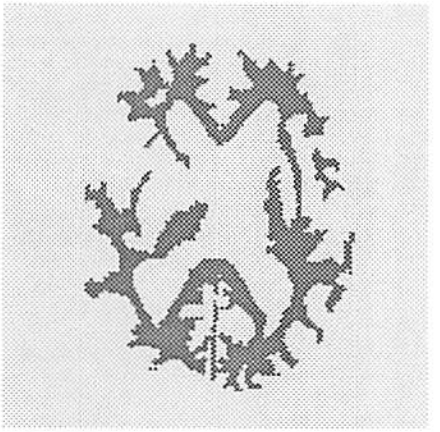
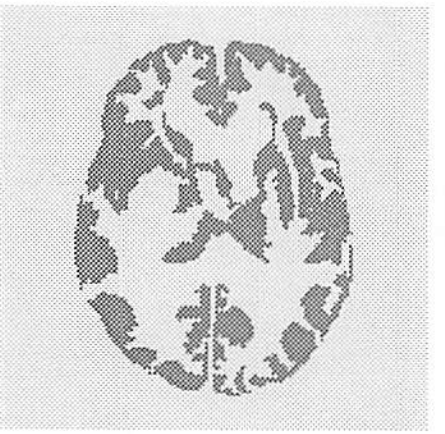


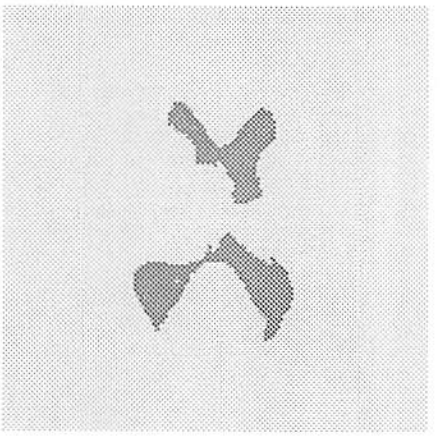
Figure 6.14: Examples of regions formed after graph theoretic patch clustering.



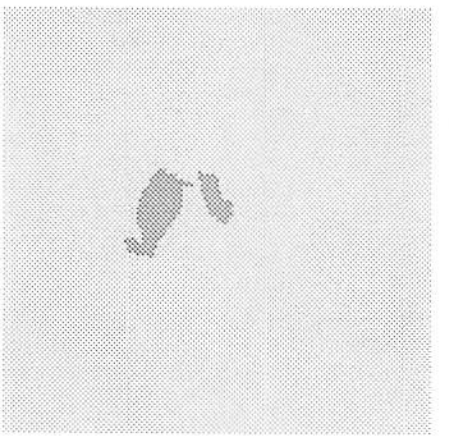
(a) White matter



(b) Grey matter



(c) Ventricles



(d) Tumor

Figure 6.15: The final segmentation after interactive labeling of regions found after graph theoretic patch clustering.

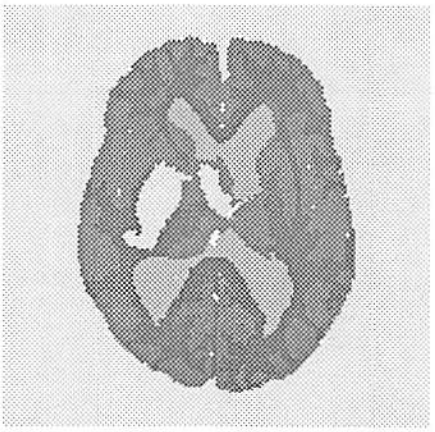


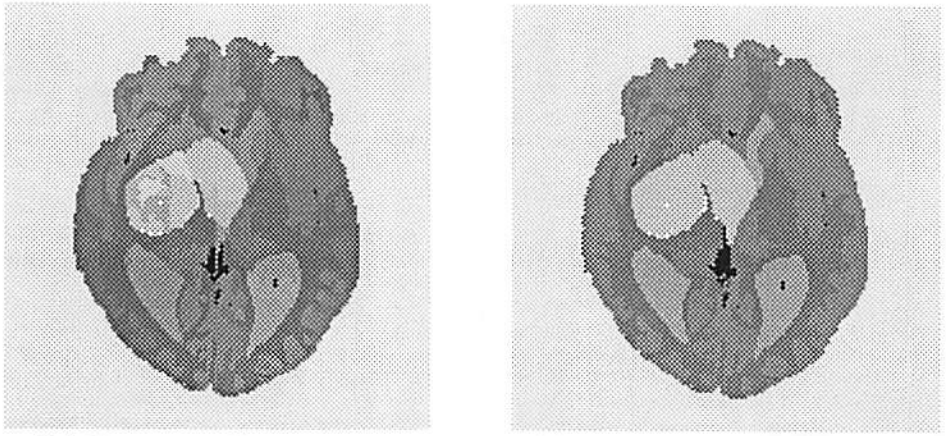
Figure 6.16: Combined segmentation map.

By comparing results from agglomerative clustering and graph theoretic clustering, we note that the latter approach produces sharper and more detail-preserving regions.

Another example:

Figure 6.17: Another axial MR image from the same patient as in Fig. 6.5.

The image, shown in Fig. 6.17, is another axial MR image from the same patient as in Fig. 6.5. The hierarchical approach is used to segment this image. After the split-and-merge learning and patch forming steps, the algorithm results in a total number of 418 patches. The graph theoretic patch clustering algorithm is then



(a) Segmented brain tissue (b) Tissue classification map

Figure 6.18: Hierarchical segmentation results of the image in Fig. 6.17: (a) segmented image of the brain tissue; (b) combined tissue classification map.

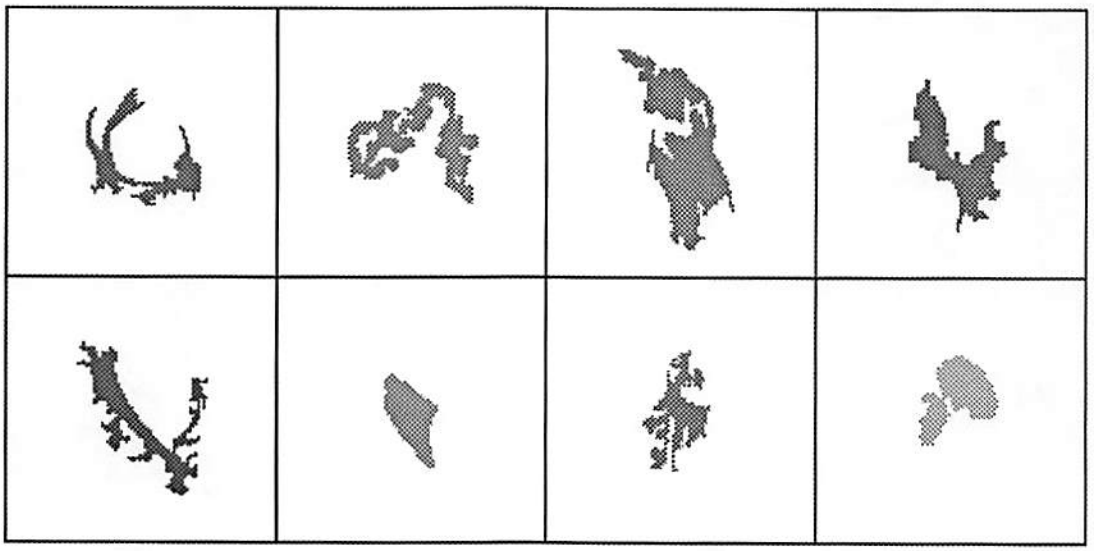
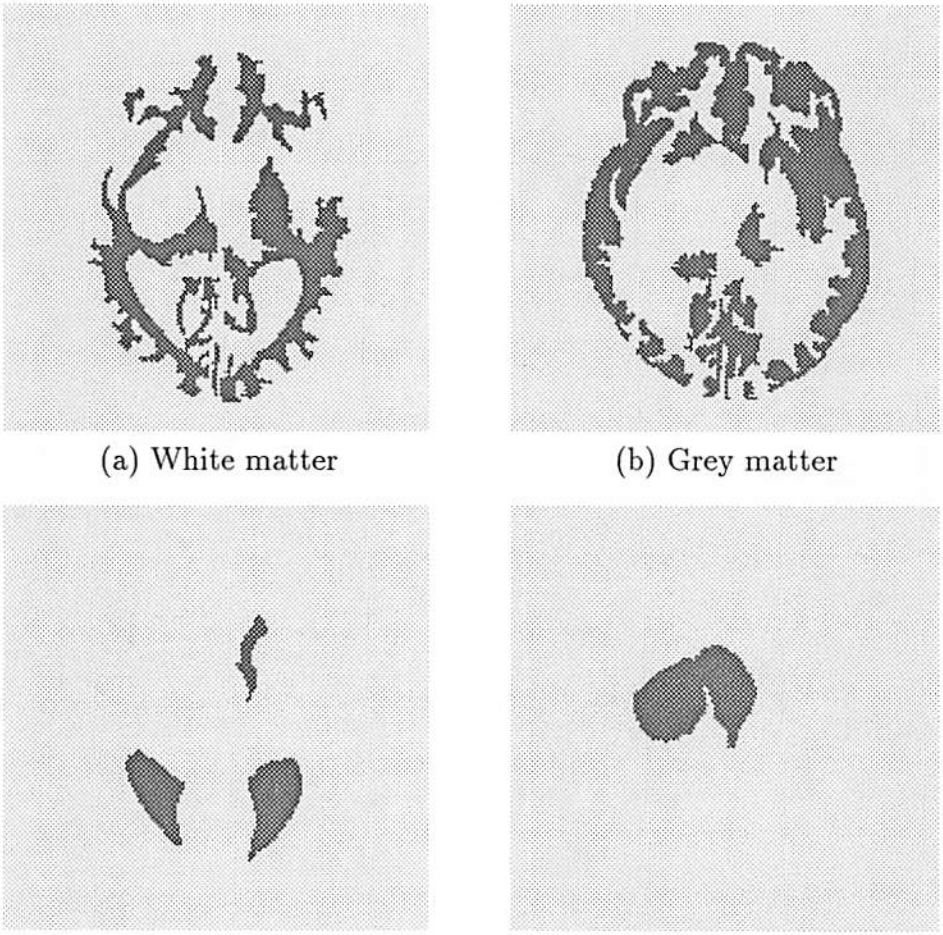


Figure 6.19: Examples of regions (patch clusters) formed by the hierarchical segmentation scheme using graph theoretic patch clustering.



(a) White matter (b) Grey matter

(c) Ventricles (d) Tumor

Figure 6.20: The final tissue classification after interactive labeling of regions found using hierarchical segmentation scheme.

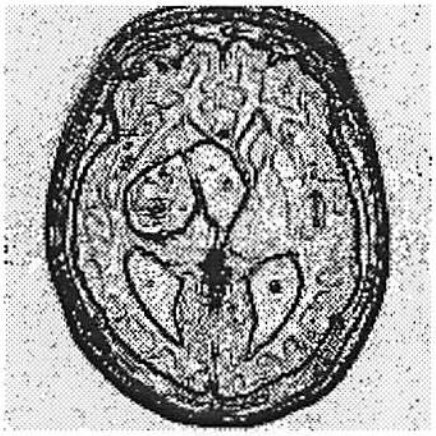
applied to reduce the number of patches. An adjacency graph is constructed with the arc capacity function as defined in (5.16) with $\sigma = 3.5$, and then an equivalent tree $\mathcal{T}^*$ is computed. Patch clusters are formed as follows: removing arcs in $\mathcal{T}^*$ in order of increasing arc capacity if the arc capacity is smaller than $\mathcal{F}_T = 10^{-9}$ and the cut does result in patches with fewer pixels than $\mathcal{N}_T = 5$; repeat the procedure for the remaining arcs in $\mathcal{T}^*$ for $(\mathcal{F}_T, \mathcal{N}_T) = (10^{-7}, 9)$, $(10^{-5}, 13)$, $(10^{-3}, 17)$ and $(10^{-1}, 21)$, respectively. This patch clustering step reduced the cluster number to 128, where 69 of them correspond to the brain tissue after removing the skull and the external tissue (Fig. 6.18.a). Fig. 6.19 shows 16 of the resulting patch clusters. To obtain the final result, these regions were further classified interactively into one of four types. Figs 6.20.a-d show the segmented regions corresponding respectively to (a) white matter; (b) grey matter; (c) ventricle; and (d) tumor. In each figure, only pixels classified to that label are displayed. The union of the four regions is shown in Fig. 6.18.b.

6.2.2 MR image segmentation using the graph theoretic approach

Now let apply the graph theoretic approach to segment the same images, Figs. 6.5 and 6.17. For both images, the unsupervised version of the segmentation algorithm is applied.

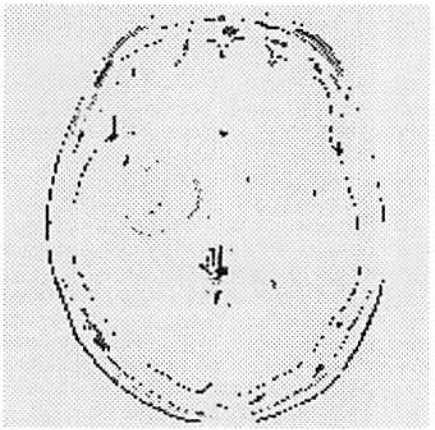
Segmenting the image in Fig. 6.17:

In order to segment the image in Fig. 6.17, an adjacency graph $\mathcal{G}$ is constructed from the image as described in Section 5.1. The edge elements were computed using (5.1) and (5.2) with control parameter $\sigma = 3.0$ (Fig.6.21.a). Since there are almost twice as many edge elements as there are pixels, the following convention is used to display the edge image in Fig.6.21.a. At each pixel site (i, j), the maximum of $D^H_{i,j}$ and $D^V_{i,j}$ is displayed. The capacity function for the arcs in $\mathcal{G}$ is defined according to (5.3). A partially equivalent tree $\mathcal{T}_c^*$ of $\mathcal{G}$ is then constructed at threshold $\mathcal{F}_T = 0.1$. Clusters of pixels are formed by removing all arcs in $\mathcal{T}_c^*$ with capacity smaller than $\mathcal{F}_T$. When no constraint is attached to these clusters, there



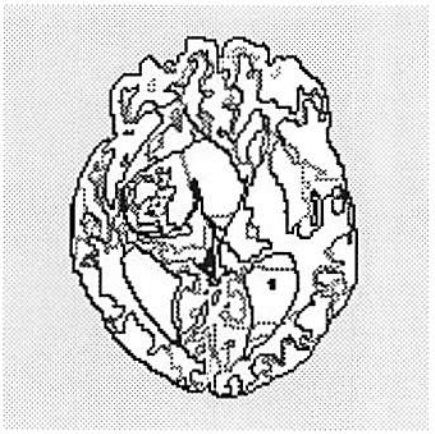
(a) Edges before clustering

(b) Segmented image

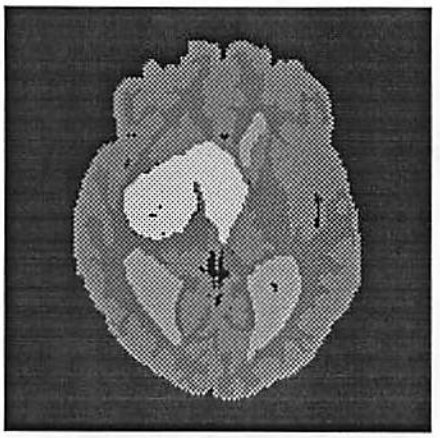


(c) Unclassified pixels

(d) Brain tissue



(e) Edges after clustering



(f) Tissue classification map

Figure 6.21: Segmentation results using the graph theoretic approach: (a) edges before clustering; (b) segmented image with each pixel cluster displayed in its mean; (c) unclassified pixels; (d) pixels corresponding to brain tissue; (e) edges after clustering; (f) combined tissue map.

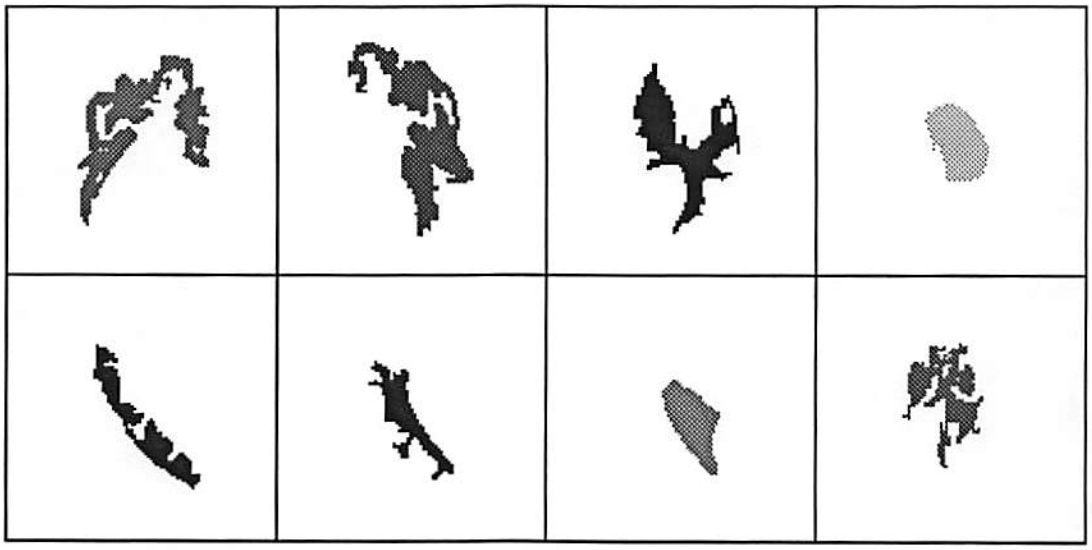
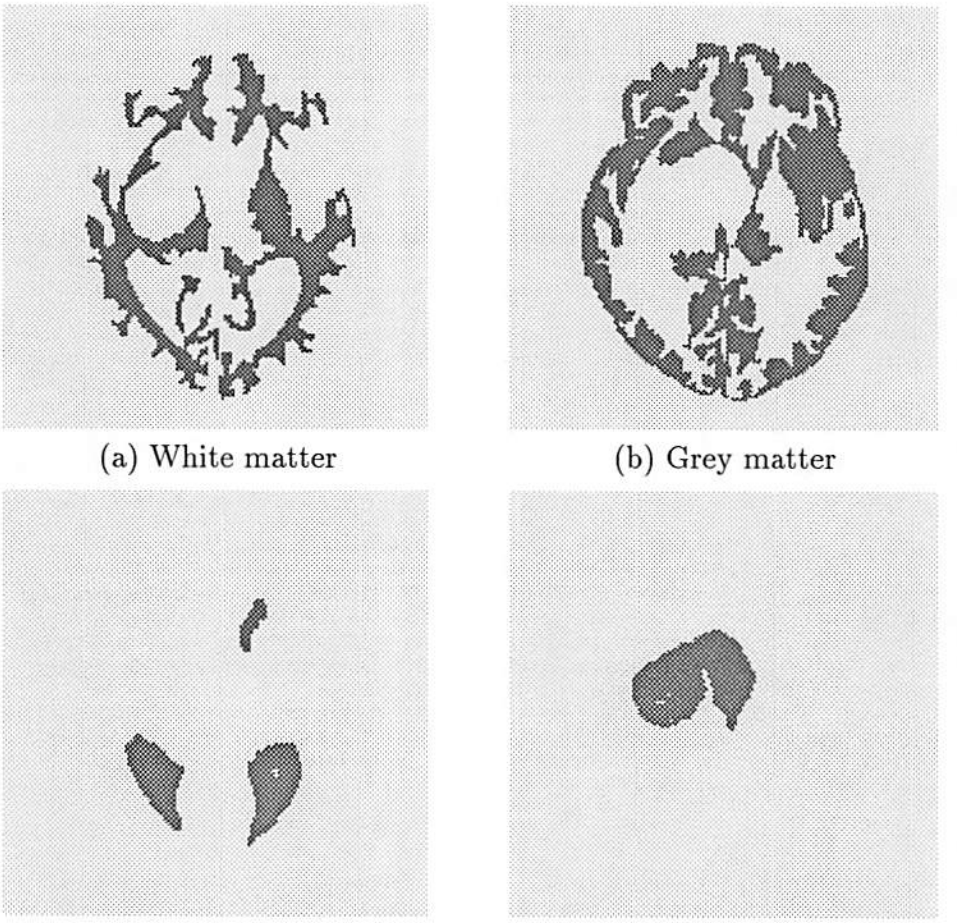


Figure 6.22: Examples of regions (pixel clusters) formed by linking edge elements using graph theoretic clustering

may be a large number with very few pixels. In our experiment, the minimum size for a cluster is restricted to 5 pixels. All clusters with less than 5 pixels, except those clusters connected with other clusters in $\mathcal{T}_c^*$ through arcs with capacities much smaller than $\mathcal{F}_T$, are merged to the most similar neighboring cluster which has more than 5 pixels, using the merging procedure described in [52]. Unmerged clusters that have less than 5 pixels will be labeled as unclassified. The algorithm produced a total of 161 clusters (connected regions) excluding the unclassified clusters. Fig. 6.21.b shows the segmented image with the pixels in each cluster replaced by their sample mean and Fig.6.21.c shows the unclassified pixels which are also displayed in black within the image of Fig.6.21.b. Using the equivalent tree of the adjacency graph of the segmented image, the skull and the external tissue can be easily removed from the image. The remaining 79 clusters correspond to brain tissue (Fig.6.21.d). Eight of these clusters are shown in Fig. 6.22. Fig.6.21.e shows the edge image associated with the 79 pixel clusters. To obtain the final tissue classification, these pixel clusters are further labeled interactively into one of five types. Figs 6.23.a-d show the segmented regions corresponding respectively to (a) grey matter; (b) white matter; (c) ventricle; and (d) tumor. In each figure, only pixels classified to that label are displayed while the rest are set to white.



(a) White matter (b) Grey matter

(c) Ventricles (d) Tumor

Figure 6.23: The final tissue classification after interactive labeling of regions found after edge linking

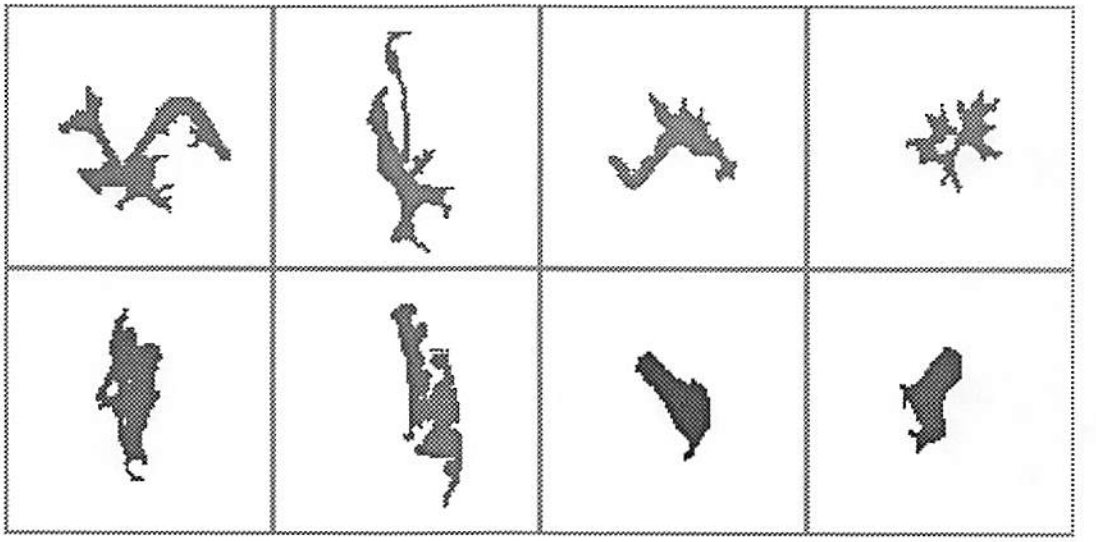
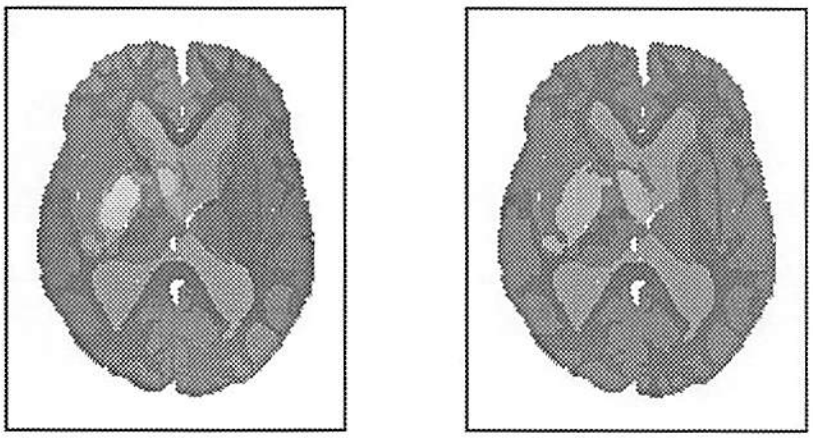


Figure 6.24: Examples of regions (pixel clusters) of the image in Fig. 6.5 formed by the graph theoretic segmentation algorithm.

The clusters that do not fit in any of the four tissue types are labeled as the null type, "none-of-the-above". The union of the four regions is shown in Fig. 6.21.f, with the black spots within the image indicating unclassified pixels or pixels of the null type.

Segmenting the image in Fig. 6.5

An adjacency graph is constructed from the pixels of the image in Fig. 6.5, where the capacity functions are computed according to (5.1), (5.2) and (5.3) with a control parameter $\sigma = 3.5$. A partially equivalent tree $\mathcal{T}_c^*$ is constructed with a threshold $\mathcal{F}_T = 0.1$. Clusters of pixels are formed by removing all arcs in $\mathcal{T}_c^*$ with capacity smaller than $\mathcal{F}_T$. In our experiment, the minimum size for a cluster is again restricted to 5 pixels except in the case where a cluster is connected with other clusters in $\mathcal{T}_c^*$ through arcs with capacities much smaller than $\mathcal{F}_T$. In this case the cluster, which has less than 5 pixels, will be labeled as unclassified. The algorithm produced a total of 92 clusters (connected regions) excluding the unclassified clusters. Fig. 6.25.b shows the segmented image with the pixels in each cluster replaced by their sample mean. The unclassified pixels are shown as white spots within the image. Fig. 6.24 shows eight of the clusters. To obtain



(a) (b)

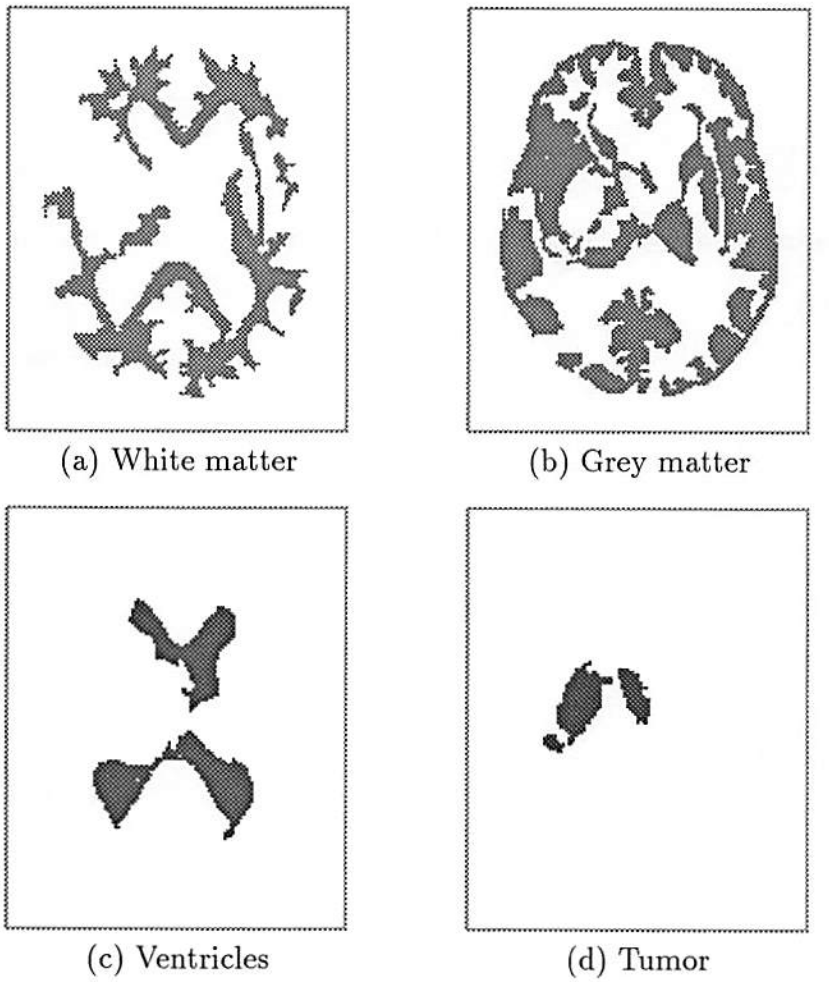
Figure 6.25: Segmentation results: (a) Image formed by pixel clusters displayed in their sample mean; (b) Combined tissue classification map.

the final tissue classification, these pixel clusters are further labeled interactively into one of four types. Figs 6.26.a-d show the segmented regions corresponding respectively to (a) grey matter; (b) white matter; (c) ventricle; and (d) tumor. In each figure, only pixels classified to that label are displayed while the rest are set to white. The union of the four regions is shown in Fig. 6.25.c. Again the white spots within the image correspond to unclassified pixels.

6.2.3 Discussions

Comments on the final segmented regions

As we have observed, the proposed segmentation algorithms do not classify the pixels of an MR image into a few tissue types. Instead the pixels are grouped into a number of connected regions (patches). Most of these regions are large and distinct from their surrounding regions, and preserve details of the boundaries between different soft tissues. For the examples presented in this section, the total number of segmented regions is around 80. We could obtain fewer regions by performing fewer cuts in the equivalent tree (or merging more patches together) during the clustering step. However, early termination of the patch clustering procedure, may result in the clustering, and hence incorrect labeling, of inhomogeneous regions.



(a) White matter (b) Grey matter

(c) Ventricles (d) Tumor

Figure 6.26: The final tissue classification after interactive labeling of regions found using the graph theoretic segmentation approach.

This problem, we believe, reflects the limited nature of the information present in an MR image, i.e. without providing any form of supervision to the algorithm (the algorithms do not 'know' that the image is an MR scan), it is conceivably not possible to obtain an accurate segmentation and clustering into only four regions (tumor, white matter, grey matter and ventricles). In the examples, the final tissue classification is obtained by interactively labeling the segmented regions found after clustering. Rather than attempting to force the current algorithm to form only the desired four regions, i.e. to perform only three cuts on the equivalent tree (which in practice gives a fairly poor result), our goal is to use the results of the patch clustering algorithm as the input to a tissue labeling algorithm which compares the features of the regions, such as size, shape, location and spatial relationship with the other regions, with a stored brain map and label the regions accordingly.

Agglomerative *vs.* graph theoretic patch clustering

Within the hierarchical segmentation scheme, we have proposed two competing approaches for patch clustering: constrained ML agglomerative clustering and graph theoretic clustering. From Figs. 6.11-6.16, we can observe that the graph theoretic approach was able to produce sharper and more accurately segmented regions than the agglomerative approach while resulting in a comparable number of regions. Similar conclusions were also drawn from other sets of MR images that we have segmented.

It is not just a coincidence that graph theoretic clustering has a superior performance over agglomerative clustering. When deciding whether two neighboring patches should be merged, the latter approach relies only on a similarity measure between the patch pair while ignoring any interaction between these patches and the rest of the image. In contrast, the graph theoretic approach evaluates the similarity between all patches that lie on the opposite sides of a potential separating contour (minimum cut). Furthermore being a divisive but globally optimal clustering technique, a poor decision (in a subjective sense) at a particular clustering step does not have a propagating effect on subsequent steps. Unfortunately the same is not true for the step-wise optimal agglomerative approach.

In addition the graph theoretic approach is a very robust technique. As we have mentioned in Section 3.3, two types of error may occur when segmenting an image. The type I error occurs if a homogeneous region is labeled inhomogeneous and hence divided into smaller subregions; and the type II error occurs if a region consisting of subregions with distinct features is labeled homogeneous. Since patch clustering cannot correct any type II error, we have to keep the number of type II errors fairly small during split-and-merging and patch forming at the expense of a larger number of type I errors. The goal of patch clustering is then to reduce the number of type I errors without significantly increasing the type II errors. A robust patch clustering technique should be expected to perform well even for very large numbers of initial type I errors. As we can observe from the examples of applying the edge based graph theoretic segmentation algorithm, which is equivalent to push the number of the type I errors to its upper limit, the obtained segmentation results are still comparable with those in Figs. 6.14-6.16.

Hierarchical *vs.* graph theoretic segmentation

Here the hierarchical segmentation algorithm refers to the version which uses graph theoretic patch clustering. Since both hierarchical and graph theoretic segmentation algorithms involve the same clustering technique, it is important to discuss the similarities and differences between them and point out their respective strengths and weaknesses.

Although the hierarchical algorithm can degenerate to the graph theoretic algorithm under extreme conditions, the latter approach is by no means a mere special case of the former. These two algorithms are developed using two different rationales. The hierarchical approach is a region based algorithm which seeks to segment an image by grouping pixels with similar features together to form homogeneous region. As a result, it is quite capable of resolving distinct regions connected by transitory boundaries. In addition, this algorithm can easily be extended to incorporate different image models to segment images containing large textures and/or ramp surfaces. However it often has difficulty in resolving small regions surrounded by larger regions. On the other hand, the graph theoretic approach is edge-based, which segments an image by searching and linking the

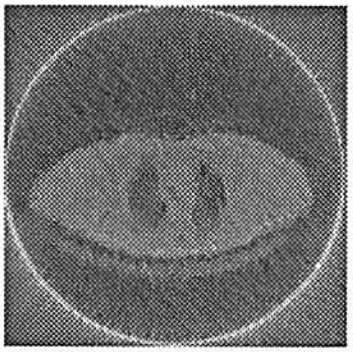
boundaries between different regions. As opposed to the hierarchical approach, the resulting segmentation is optimal for a well defined cost function computed from the image pixels. Furthermore when extending this algorithm to 3-D image segmentation, not only will the optimality be retained but also the solution method will be identical since the clustering algorithm is applicable to any undirected adjacency graphs. Since the edges (or arc capacities) are computed locally using a small mask, this approach is usually unable to detect smooth transitions between regions with distinct features.

As far as segmenting MR images is concerned, the experimental results produced by either approach have been satisfactory and promising. Each algorithm has its own merits and deficiencies. However if we were forced to make a choice between the two based on their performance, we would have selected the hierarchical approach (with graph theoretic patch clustering) because it is more efficient and less sensitive to noise.

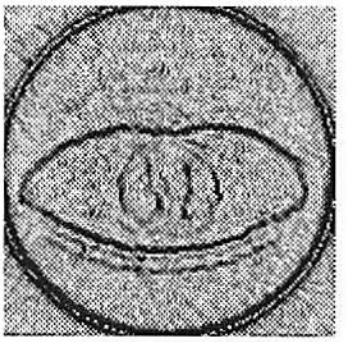
6.3 Miscellaneous Examples of Applying Graph Theoretic Clustering

Segmentation of a PET image

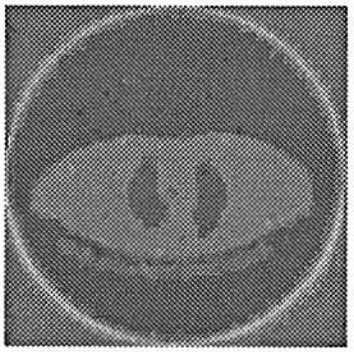
Fig. 6.27.a shows a cross-sectional transmission PET (Positron Emission Tomography) image of the human body, at the liver level. The transmission scan is used in PET for attenuation correction. This image is very noisy and has relatively low spatial resolution, which are characteristics of PET images. To reduce noise, a smoothed version of the image in Fig. 6.27.a, generated using a 3×3 Gaussian window, is used for segmentation. Fig. 6.27.b shows the edges computed using (5.1) and (5.2) with $\sigma = 3.5$ and $\delta = 2$. An adjacency graph is constructed using the arc capacity defined in (5.3) and a partially equivalent tree is then computed at threshold $\mathcal{F}_T = 0.2$. The resulting pixel clusters are shown in Fig. 6.27.c, displayed in the cluster's sample mean intensity, without any post-processing. For reference, the edges resulting from applying the Marr-Hildreth operator are shown



(a) Original PET image



(b) Computed edges before clustering



(c) Segmented image

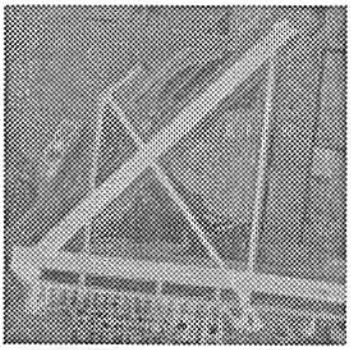
(d) Edges by Marr-Hildreth

Figure 6.27: Segmentation results for a transmission PET image of the liver: (a) original PET image; (b) computed edges before clustering; (c) segmented PET image; (d) edges computed using the Marr-Hildreth operator ($\sigma = 2$).

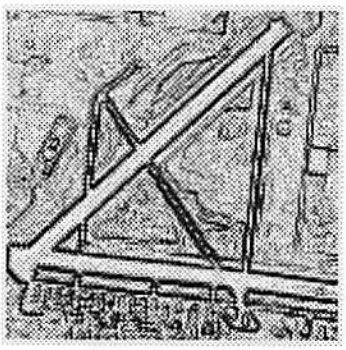
in Fig. 6.27.b. By trial and error, the best result was achieved with $\sigma = 2$ implemented using a window of 13×13 pixels. Although the structures of interest are detected with reasonable accuracy, there also exist many false edges which tends to make all regions enclosed by zero crossings have similar widths close to half of the window size.

Edge detection of an airport image

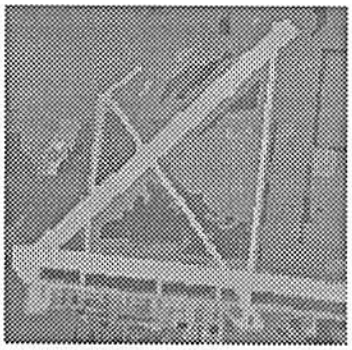
The edge based segmentation algorithm using graph theoretic clustering has also been applied to edge detection for aerial photographs. Fig. 6.28.a shows a photograph of an airport, and Fig. 6.28.b shows its edge image computed using (5.1)



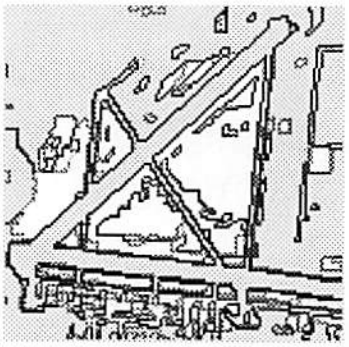
(a) Original airport image



(b) Computed edges before clustering



(c) Segmented image



(d) Edges after clustering

Figure 6.28: Application to aerial photographs: (a) original airport image; (b) computed edges of the airport image before clustering; (c) segmented airport image; (d) edges after clustering.

and (5.2). An adjacency graph is constructed using the arc capacity defined in (5.3) with $\sigma = 3.5$, and a partially equivalent tree is then computed at threshold $\mathcal{F}_T = 0.1$. Clusters of pixels are formed by removing all arcs in $\mathcal{T}_c^*$ in order of increasing arc capacity with capacity smaller than $\mathcal{F}_T$, unless doing so will result in a cluster with less than 5 pixels. This procedure produces a total of 321 clusters. These are shown in Fig. 6.28.c with pixels in each cluster displayed in the cluster's sample mean intensity. The edge image corresponding to Fig. 6.28.c is shown in Fig. 6.28.d. Again the edge images only display the maximum of $D_{i,j}^H$ and $D_{i,j}^V$ at each pixel site (i,j).

Chapter 7

Conclusions and Future Research

7.1 Conclusions

This dissertation has addressed the problems of image segmentation and pattern classification, and their application to the problem of tissue classification of magnetic resonance (MR) images of the human brain. Two new image segmentation schemes are proposed: hierarchical and graph theoretic. The hierarchical scheme is an unsupervised, region based approach. It seeks to group the image pixels into regions which, under a suitable hypothesis, are homogeneous. The latter scheme is based on a novel graph theoretic approach to data clustering, also developed in this research work. The images are represented by an undirected graph and the segmentation is achieved by effectively searching for closed contours of edge elements.

An unsupervised hierarchical segmentation scheme was developed for segmenting textured images. The images are modeled as a mosaic of "homogeneous" first order GMRFs, and the segmentation seeks to group the image pixels into homogeneous regions. Based on homogeneity testing a three step segmentation scheme was developed : a split-and-merge procedure for finding large homogeneous regions, a patch forming for classifying boundary pixels and a final patch clustering to reduce the number of resulting patches. This approach allows the algorithm to

learn about the image by first clustering the "easy" pixels while delaying decisions on the "hard" pixels until more information has been gathered. Depending on the image model and the homogeneity criterion, this scheme may be applied to segment either fine-textured or non-textured images.

A statistical analysis on testing homogeneity for GMRFs was presented. For the case of labeled data the likelihood ratio test was used; and for the unlabeled data case we proposed to use the dispersion test and the new hierarchical likelihood ratio test. In particular, we have computed the power functions for the above two tests as well as the likelihood ratio test based on the similarity between the pixel region and arbitrarily partitioned subregions for the case of white Gaussian random fields. We have found that the commonly used region/subregion similarity approach usually has a low power of detecting inhomogeneity, while the other two tests have much better performance.

Accurate approximations have been proposed for computing the joint likelihood for the first order GMRFs defined on irregularly shaped regions. This problem is equivalent to computing the partition function for Gibbs distributions for the case where the random field is Gaussian.

We have also presented a novel graph theoretic approach for data clustering based on network flow theory, and demonstrated an application to the problem of image segmentation. The data to be clustered are represented by an undirected adjacency graph $\mathcal{G}$. A flow capacity is assigned to each arc in $\mathcal{G}$ to reflect the similarity between the neighboring data points linked by that arc. Given $\mathcal{G}$, clustering is achieved by removing arcs corresponding to minimum cuts between vertex pairs in $\mathcal{G}$ to form mutually exclusive subgraphs. For an unconstrained optimal K-subgraph partition, the arcs selected for removal are those corresponding to the K$-$1 minimum cuts with the smallest K$-$1 cut values among all minimum cuts between every pair of vertices in $\mathcal{G}$. The resulting K-partition minimizes the largest inter-subgraph maximum flow among all possible K-partitions of $\mathcal{G}$, hence minimizing the similarity between the subgraphs (clusters). At the same time, this procedure also results in a nested sequence of partitions which are optimal for cluster numbers ranging from 2 to K, respectively. This is a desirable characteristic of a clustering technique, especially when the cluster number has to be

determined from the data.

The minimum cuts of $\mathcal{G}$ are computed from a flow and cut equivalent tree $\mathcal{T}^*$ of $\mathcal{G}$, which can be constructed using the Gomory-Hu algorithm. Given $\mathcal{T}^*$, the optimal K-partition can be equivalently obtained by disconnecting the K$-$1 arcs in $\mathcal{T}^*$ with the smallest K$-$1 arc capacities. Based on the observation that most of the minimum cuts found in $\mathcal{T}^*$ are never used, since their values are sufficiently large that the arcs in those cuts will not be removed to form subgraphs, a fast hierarchical algorithm has been developed which constructs only a partially equivalent tree $\mathcal{T}_c^*$ of greatly reduced size. New theorems for subgraph condensation have been derived. Using these theorems, many of the minimum cuts with large values can be identified using local subgraphs, hence the vertices linked by them can be condensed *before* constructing the equivalent tree. Consequently the Gomory-Hu algorithm is applied to graphs of much smaller size without compromising the overall optimality of the clustering algorithm.

This graph theoretic clustering algorithm has been applied to the problem of image segmentation. Segmentation is achieved through seeking closed contours of edge elements of the image. By properly assigning arc capacities as a function of the computed edge strength, a minimum cut with small cut value, found in the adjacency graph formed by the image pixels, should contain mostly strong edges. We have observed that the proposed segmentation approach is able to accurately locate region boundaries while guaranteeing the formation of closed edge contours. The segmentation is unsupervised, but it also offers the flexibility for incorporating prior information about the image through the arc capacity function. For example when segmenting MR brain images, the pixel intensity ranges for different tissues and their relative position within the brain may be reflected in the capacity function to provide improved segmentation results. Another advantage of this approach is that it is directly applicable to 3-D images segmentation, except that the arc capacities need to be assigned differently than in the 2-D case.

The two proposed segmentation schemes have been applied to tissue classification in MR images of the human brain. The preliminary results are very promising.

7.2 Future Research

One of the long term objectives of this dissertation is to develop an automated system for the tissue classification of 3-D MR images. In order to reach this goal, we believe that it is crucial to complete the following research works:

- Extensions to 3-D MR image segmentation.
- Incorporation of anatomical information into tissue classification.

3-D MR image segmentation

The next step in our research is to extend the proposed segmentation schemes to segment 3-D MR images of the human brain. Extending a 2-D segmentation scheme to 3-D is more than simply adding one extra dimension to the data structure. There are some practical problems of concern. The 3-D MR image data are available to us as a stack of 2-D images (slices) of the head. Often the resolution between slices are lower than the resolution within the slices. This resolution difference in different directions has to be taken into account by the segmentation scheme. Also we have to handle properly the huge amount of memory required by a 3-D image. A typical MR scan can have over 60 slices of 256×256 pixels, each coded with 2 bytes. The image alone will occupy about 8 MBytes.

It is conceptually simple to extend the graph theoretic approach for the segmentation of 3-D MR images. Once an undirected graph $\mathcal{G}$ has been properly constructed from the 3-D data, the hierarchical clustering algorithm may be applied directly. However some practical concerns need to be addressed. The selection of an arc capacity function will become more difficult, because the increased connectivity between vertices can make the clustering outcome more sensitive to the arc capacities.

The hierarchical approach may also be extended to the 3-D case, at least in theory. However we do not think that a direct extension is practical, because the increased complexity due to the added third dimension may make it computationally infeasible. Alternatively a hybrid approach may be taken. Because of the usually large spacing between adjacent slices, a weak correlation is expected

among voxels in neighboring slices. Therefore we may first process the slices individually to find homogeneous regions (use the split-and-merge and patch forming steps), then construct an adjacency graph $\mathcal{G}$ from the patches resulting from all the slices, and finally apply the graph theoretic clustering algorithm to partition $\mathcal{G}$ to form 3-D regions.

Incorporating anatomical information

The new segmentation schemes are able to provide large patches, however human intervention, through an interactive patch labeling process, is still needed to assign a tissue type to each resulting patch.

In order to make this tissue type identification automatic, which include the classification for normal tissue types and the detection and possible identification of abnormal tissue types (*e.g.* tumors), anatomical information on the human brain has to be incorporated. Appropriate knowledge may include, but is not limited to, the following :

- Topological structure of tissues inside the brain.
- Geometric shapes of certain types of tissue.
- Relative MR imaging characteristics for different types of tissue.

References

[1] S. W. Young, *Magnetic Resonance Imaging: Basic Principles.* Raven Press, New York, 1988.

[2] W. S. Hinshaw and A. H. Lent, "An introduction to MRI imaging: From the Bloch equation to the imaging equation," *Proceedings of the IEEE*, vol. 71, pp. 338–350, 1983.

[3] E. J. Farrell, W. C. Yang, and R. A. Zappulla, "Animated 3D CT imaging," *IEEE Computer Graphics and Applications*, pp. 26–32, Dec. 1990.

[4] M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29–37, 1988.

[5] D. N. Levin *et al.*, "Surface of the brain: Three-dimensional MR images created with volume rendering," *Radiology*, vol. 171, no. 1, pp. 277–280, 1989.

[6] M. Brummer, A. V. Est, and W. Menhardt, "The accuracy of volume measurements from MR imaging data," *in Proc. 8th Ann. Meeting SMRM, Amsterdan*, 1989.

[7] D. N. Levin *et al.*, "Retrospective geometric correlation of MR, CT and PET images," *Radiology*, vol. 169, no. 3, pp. 817–823, 1988.

[8] W. Menhardt and K. H. Schmidt, "Automated interpretation of transxial MR-images," *in Proc. CAR 87, H.U. Lemke et al. (eds.)*, pp. 386–390, 1987.

[9] R. M. Hylton and D. A. Ortendahl, "Information extraction and representation in MRI," *in Proc. IEEE 7th Annual Conf. of the Eng. in Med. Bio. Soc.*, 1985.

[10] G. Gerig, J. Martin, R. Kikinis, O. Kubler, M. Shenton, and F. Jolesz, "Automating segmentation of dual-echo MR head data," *in Proc. 12th Int. Conf. on Information Processing in Medical Imaging*, 1991.

[11] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Royal Soc. of London*, vol. 207, pp. 187–217, 1980.

[12] M. W. Vannier, C. M. Speidel, and D. L. Rickman, "Magnetic resonance imaging multispectral tissue classification," *NIPS*, vol. 3, Aug. 1988.

[13] M. W. Vannier *et al.*, "Validation of magnetic resonance imaging (MRI) multispectral tissue classification," *in Proc. of 9th Conf. on Pattern Recognition, Rome, Italy*, pp. 1182–1186, Nov. 1988.

[14] D. A. Ortendahl and J. W. Carlson, "Segmentation of magnetic resonance images using fuzzy clustering," *in Proc. 10th Int. Conf. on Information Processing in Medical Imaging*, 1988.

[15] S. P. Raya, "Low-level segmentation of 3-D magnetic resonance brain images - A rule-based system," *IEEE Trans. on Medical Imaging*, vol. 9, pp. 327–337, Sept. 1990.

[16] H. S. Choi, D. R. Haynor, and Y. Kim, "Multivariate tissue classification of MRI images for 3-D volume reconstruction - A statistical approach," *in Proc. SPIE Medical Imaging III: Image Processing*, 1989.

[17] R. H. Lee and R. Leahy, "Multi-spectral tissue classification of MR images using sensor fusion approaches," *in Proc. SPIE Conf. on Medical Imaging IV*, 1990.

[18] A. Rosenfeld and A. C. Kak, *Digital Picture Processing.* Academic Press, Inc., 1982.

[19] P. C. Chen and T. Pavlidis, "Image segmentation as an estimation problem," *Comp. Graphics Image Processing*, vol. 12, pp. 153–172, 1980.

[20] D. A. Ortendahl, N. M. Hyton, L. Kaufman, and L. E. Crooks, "Tissue characterization using intrinsic NMR parameters and a hierarchical processing algorithm," *IEEE Trans. on Nuclear Science*, vol. 32, pp. 875–879, Feb. 1985.

[21] H.-H. Ehricke, "Problems and approaches for tissue segmentation in 3D-MR imaging," *in Proc. SPIE Medical Imaging IV: Image processing*, 1990.

[22] M. Bomans, K.-H. Hohne, U. Tiede, and M. Riemer, "3-D segmentation of MR images of the head for 3-D display," *IEEE Trans. on Medical Imaging*, vol. 9, pp. 177–183, June 1990.

[23] J. W. Woods, "Two-dimensional discrete Markovian fields," *IEEE Trans. Information Theory*, vol. 18, pp. 232–240, Mar. 1972.

[24] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Stat. Soc., Series B*, vol. 36, pp. 192–236, 1974.

[25] A. K. Jain, "Advances in mathematical models for image processing," *Proceedings of the IEEE*, vol. 69, pp. 502–528, May 1981.

[26] B. D. Ripley, *Statistical Inference for Spatial Processes.* Cambridge University Press, 1988.

[27] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[28] H. Derin, H. Elliott, R. Cristi, and D. Geman, "Bayes smoothing algorithms for segmentation of binary images modeled by Markov random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 707–720, Nov. 1984.

[29] J. Besag, "On the statistical analysis of dirty pictures," *J. Roy. Stat. Soc., Series B*, vol. 48, pp. 259–302, 1986.

[30] R. Chellappa and R. L. Kashyap, "Digital image restoration using spatial interaction model," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 30, pp. 461–472, Jun. 1982.

[31] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, pp. 25–39, Jan. 1983.

[32] R. Chellappa and R. L. Kashyap, "Texture synthesis using 2-D noncausal autoregressive models," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 33, pp. 194–203, Feb. 1985.

[33] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random field," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 33, pp. 959–963, Aug. 1985.

[34] H. Derin and W. S. Cole, "Segmentation of textured images using Gibbs random fields," *Comp. Vision Graphics Image Processing*, vol. 35, pp. 72–98, 1986.

[35] F. S. Cohen and D. B. Cooper, "Real time textured-image segmentation based on noncausal Markovian random field models," *in Proc. of 3rd Int. Conf. on Robot Vision & Sensory Control*, Nov. 1983.

[36] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithm for segmenting noncausal Markovian random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 195–219, Mar. 1987.

[37] T. Simchony and R. Chellappa, "Stochastic and deterministic algorithms for MAP texture segmentation," *in Proc. IEEE ICASSP*, 1988.

[38] P. Whittle, "On stationary processes in the plane," *Biometrika*, vol. 41, pp. 434–449, 1954.

[39] X. Guyon, "Parameter estimation for a stationary process on a d-dimensional lattice," *Biometrika*, vol. 69, pp. 95–105, 1982.

[40] K. Ord, "Estimation methods for models of spatial interaction," *J. Amer. Stat. Soc.*, vol. 70, pp. 120–126, Mar 1975.

[41] J. Besag and P. A. P. Moran, "On the estimation and testing of spatial interaction in Gaussian lattice processes," *Biometrika*, vol. 62, pp. 555–562, 1975.

[42] K. V. Mardia and R. J. Marshall, "Maximum likelihood estimation of models for residual covariance in spatial regression," *Biometrika*, vol. 71, pp. 135–146, 1984.

[43] J. Besag, "Statistical analysis of non-lattice data," *The Statistician*, vol. 24, pp. 179–195, 1975.

[44] R. L. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial-interactive models of images," *IEEE Trans. Information Theory*, vol. 29, pp. 60–72, Jan. 1983.

[45] J. F. Silverman and D. B. Cooper, "Bayesian clustering for unsupervised estimation of surface and texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 482–495, July 1988.

[46] J. Besag, "Efficiency of pseudolikelihood estimation for simple Gaussian fields," *Biometrika*, vol. 64, pp. 616–618, 1977.

[47] D. Luenberger, *Linear and Nonlinear Programming.* Addison-Wesley, 2nd ed., 1984.

[48] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Amer. Stat. Soc.*, vol. 82, pp. 76–89, Mar 1987.

[49] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images and Gibbs random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 39–55, Jan. 1987.

[50] R. Lankhmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random field using simulated annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 799–813, Aug. 1989.

[51] Z. Wu and R. Leahy, "A new unsupervised hierarchical segmentation algorithm for textured images," *in Proc. IEEE ICASSP*, Apr. 1990.

[52] Z. Wu and R. Leahy, "Unsupervised hierarchical segmentation of textured images based on homogeneity testing." USC - Signal & Image Processing Institute Technical Report #157, June 1990.

[53] B. W. Lindgren, *Statistical Theory.* McMillan Publishing Co., Inc., New York, 3rd ed., 1976.

[54] E. L. Lehamann, *Testing Statistical Hypotheses.* John Wiley & Sons, Inc., 2nd ed., 1986.

[55] P. C. Chen and T. Pavlidis, "Segmentation by texture using correlation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, pp. 64–69, 1983.

[56] H. Samet, "The quadtree and related hierarchical data structures," *Computing Surveys*, vol. 16, pp. 187–260, 1984.

[57] H. Samet, "Connected component labeling using quadtree," *J. of the Association for Computing Machinery*, vol. 28, pp. 487–501, 1981.

[58] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[59] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* Prentice Hall, Inc., New Jersey, 1988.

[60] L. J. Hubert, "Some applications of graph theory to clustering," *Psychometrika*, vol. 38, pp. 435–475, 1974.

[61] D. W. Matula, "Graph theoretic techniques for cluster analysis algorithms," *In* Classification and Clustering, *J. Van Ryzin, ed., Academia Press, Inc., New York*, pp. 95–129, 1977.

[62] C. T. Zahn, "Graph-theoretic methods for detecting and describing gestalt clusters," *IEEE Trans. on Computers*, vol. 20, 1971.

[63] R. Urquhart, "Graph theoretical clustering based on limited neighborhood sets," *Pattern Recognition*, vol. 15, 1982.

[64] W. L. Koontz, P. M. Narendra, and K. Fukunaga, "A graph-theoretic approach to nonparametric cluster analysis," *IEEE Trans. on Computers*, vol. 24, pp. 936–944, Sept. 1976.

[65] Z. Wu and R. Leahy, "Tissue classification in MR images using hierarchical segmentation," *in Proc. IEEE Int. Conference on Medical Imaging*, Oct. 1990.

[66] Z. Wu and R. Leahy, "A graph theoretic approach to image segmentation of MR images," *in Proc. SPIE/SPSE's Symp. on Elect. Image Sci. & Tech.*, vol. SPIE-1450, Feb. 1991.

[67] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation." submitted for publication in IEEE Trans. on Pattern Anal. Machine Intell., July 1991.

[68] R. E. Jensen, "A dynamic programming algorithm for cluster analysis," *Operations Research*, vol. 17, pp. 1034–1057, 1969.

[69] W. L. Koontz, P. M. Narendra, and K. Fukunaga, "A branch and bound clustering algorithm," *IEEE Trans. on Computers*, vol. 24, pp. 908–915, Sept. 1975.

[70] L. P. Lefkovitch, "Conditional clustering," *Biometrics*, vol. 36, pp. 43–58, 1980.

[71] R. E. Gomory and T. C. Hu, "Multi-terminal network flows," *SIAM Journal on Appl. Math.*, vol. 9, pp. 551–570, 1961.

[72] L. R. Ford, Sr. and E. Fulkerson, *Flows in Networks.* Princeton University Press, Princeton NJ, 1962.

[73] N. Christofides, *Graph Theory : An Algorithmic Approach.* Academic Press, New York, 1975.

[74] L. Lovász and M. D. Plummer, *Matching Theory.* Elsevier Science Publishers B.V., Amsterdam, 1986.

[75] R. K. Ahuja and J. B. Orlin, "A fast and simple algorithm for the maximum flow problem," *Operations Research*, vol. 37, pp. 748–759, 1989.

[76] P. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comp. Vision Graphics Image Processing*, vol. 41, pp. 233–260, 1988.

[77] G. K. Coleman and H. C. Andrews, "Image segmentation bu clustering," *Proceedings of the IEEE*, vol. 5, pp. 773–785, May 1979.

[78] R. L. Cannon, J. V. Dave, J. C. Bezdek, and M. M. Trivedi, "Segmentation of a thematic mapper image using the fuzzy c-means clustering algorithm," *IEEE Trans. Geoscience and Remote Sensing*, vol. 24, pp. 400–408, May 1986.

[79] J. F. Haddon and J. F. Boyce, "Image segmentation by unifying region and boundary information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 928–948, Oct. 1990.

[80] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 679–698, 1986.

[81] J. J. Clark, "Authenticating edges produced by zero-crossing algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 43–57, 1989.

[82] R. C. Gonzalez and P. Wintz, *Digital Image Processing.* Addison Wesley, 1987.

[83] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Comp. Graphics Image Processing*, vol. 13, pp. 257–269, 1980.

[84] Y. Yakimovsky, "Boundary and object detection in real world images," *J. of the Association for Computing Machinery*, vol. 23, pp. 599–618, 1976.

[85] A. Kundu and S. K. Mitra, "A new algorithm for image edge extraction using a statistical classifier approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 569–577, 1987.

[86] R. M. Haralick, "Edge and region analysis for digital image data," *Comp. Graphics Image Processing*, vol. 12, pp. 60–73, 1980.

[87] P. Bouthemy, "A maximum likelihood framework for determining moving edges," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 499–511, May 1989.

[88] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, New York, 2nd ed., 1980.