Abstract

Feature and Module Integration for Image Segmentation

Amit Chakraborty

Yale University

1996

A systematic approach towards the problem of designing integrated methods for image segmentation has been developed in this thesis. This is aimed towards the analysis of underlying structures in an image which is crucial for a variety of image analysis and computer vision applications. However, a robust identification and measurement of such structure is not always achievable by using a single technique that depends on a single image feature. Thus, it is necessary to make use of various image features, such as gradients, curvatures, homogeneity of intensity values, textures, etc. as well as model-based information (such as shape). Integration provides a way to make use of the rich information provided by the various information sources, whereby consistent information from the different sources are reinforced while noise and errors are attenuated. As a first step, integration is achieved in this work by using region information in addition to gradient information within the deformable boundary finding framework. This considerably increases the robustness of the final boundary output to noise and initial estimate. This feature integration paradigm for deformable boundary finding is then further developed through the addition of curvature information which makes the boundary solution better localized. Next, a more general integration framework is considered, whereby computational modules are associated with the boundary and region processes which are simultaneously updated. This is achieved through the use of a new game-

theoretic procedure where the modularity of the underlying objectives are retained. The integration problem is framed as a family of coupled and coexisting objectives using a Bayesian strategy whereby the output of one module depends upon the previous outputs of the other modules. This mode of information sharing, where only the final decisions of the different modules (decision makers) are broadcast to the other decision makers, is not only technically more general than other single objective function approaches, but is also computationally less burdensome especially in cases like the present one where incommensurate objectives are involved. This further improves both the region and boundary estimates.

Feature and Module Integration for Image Segmentation

A Dissertation

Presented to the Faculty of the Graduate School

of

Yale University

in Candidacy for the Degree of

Doctor of Philosophy

by

Amit Chakraborty

Dissertation Director: James S. Duncan

May 1996

Acknowledgements

At the outset I would like to profoundly thank my advisor, James Duncan for introducing me to Image Processing & Computer Vision and for providing continued guidance, encouragement, direction and support. I thank Larry Staib for not only serving on my committee and for all the stimulating discussions that we had, but also for having the patience to address the numerous computer related questions that I had. Thanks to Peter Schultheiss for providing encouragement and for all the enlightening discussions that we had- I'm privileged to have him on my committee. I also thank Roman Kuc for serving on my committee. Thanks to Alan Yuille for agreeing to be my external reader.

This work was made thoroughly enjoyable by the friendly and congenial atmosphere of the Image Processing and Analysis Group. I thank all the past and present members that made this possible: Amir Amini, Ravi Bansal, Isil Bozma, Sudhakar Chelikani, Rupert Curwen, Gene Gindi, Chuck Harrell, Mindy Lee, John McEachen, Francois Meyer, Wiro Neissen, Xenios Papademetris, Suguna Pappu, K. Purosottam, N. Rajeevan, Anand Rangarajan, Glynn Robinson, Peng Cheng Shi (for all the technical and non-technical discussions and his friendship), Hemant Tagare, Frans Vos, Yongmei Wang, Marcel Worring, Lei Zhang and George Zubal. I would also like to thank Carl Jaffe, Bob Schultz, Al Sinusas, Eliot Heller and Don Dione with whom I had the opportunity to collaborate. Many thanks to Carolyn Meloling for providing administrative assistance.

Finally, I thank my parents and my brother for their support and encouragement.

This work was supported in part by NIH-NHLBI grant R01-HL-44803, NIMH-P01MH grant 49351-01 and NLM grant R01-LM-05007.

List of Figures

Contents

Chapter 1

Introduction

1.1 Introduction to the problem

Segmentation and analysis of underlying structures in an image is of importance in a variety of image analysis and computer vision applications including robot vision, pattern recognition and biomedical image processing. However, any individual image analysis tool is not likely to achieve reliable results under all circumstances. This is especially true if the images were obtained under different conditions and have different content. This work is concerned with the precise and robust segmentation of underlying structures from natural images using integrated methods.

It has been observed that when different segmentation techniques are applied to an image, they are likely to produce different segmentation maps. Integrated methods give us ways to resolve these differences, thereby allowing us to make better use of the rich information provided by the various sources. The idea is that consistent information from the different sources is reinforced while noise and errors are attenuated.

The bulk of the existing work on image segmentation can be categorized into two basic approaches: region-based and gradient-based [5]. Region based methods [45] rely on the homogeneity of spatially dense localized features and other pixel statistics. Simple thresholding is the most basic and easiest region-based segmentation method [108, 120]. However even after a careful choice of the threshold [133, 131], for most practical images that are noisy, thresholding produces poor results. Split and merge techniques [65, 20] carry out seeded region growing, but under noisy circumstances suffer from the problems of over-segmentation and are often too sensitive to the choice of the seed points. Following the seminal work of Geman & Geman [47], Markov Random Field (MRF) based methods have been widely used by researchers for region segmentation [47, 41, 40, 91]. While more details about region-based segmentation using the MRF formalism will be discussed later (chapters 2 and 3), we would like to note here that there are two commonly used variants of this formalism. The first method employs energy minimizing techniques [91, 14] to reconstruct a piece-wise flat image from the noisy data field. The other method is more versatile in the sense that it can deal with homogeneity in other features like texture (besides grey level intensity values) and uses probabilistic classification to divide the image. Incidentally, anisotropic diffusion methods [101] can be used to obtain region estimates similar to the ones obtained via some of the MRF based methods [91] (see details in the following chapter). Further, while some of the region methods have performance superior to the others, they share a common problem - that of over-segmentation and poorly localized region boundaries.

The other group of work uses gradient-based edge detection methods [21, 84] followed by edge following techniques [107]. Gradient methods concentrate on regions of the image where there is a sharp local difference in the grey scale values. Thus instead of looking at the similarities as in the previous region-based case, these methods focus on the differences/transitions. However, the problem with these methods is that almost always they result in false and broken edges. Thus they are likely to produce edges at locations where there are no objects to be found. This is the case mainly because of the local nature of the procedure whereby any location where the gradient is high could be considered to be an edge point even if it is a local noise artifact and is not really a part of an object boundary. Also, performing a reasonable edge following [107] could often be very difficult. Local operations without any notion of shape cannot guarantee the segmentation of whole structures from images without further processing. More details regarding the various segmentation methods will be provided in Chapter 2.

One of the solutions for the above lies in using deformable whole boundary methods [116, 62] which rely on the gradient features at a subset of the spatial positions of an image (near an object boundary). By imposing a global shape to the problem, one augments missing or noisy local information. Once again, there are a number of whole boundary methods that can be used [62, 116, 82]. In the snakes method [62, 3] the spatial coordinates of a continuous deformable curve are adjusted from the initial to the final position guided by image-derived forces as well as internal spline forces. Improvements to this basic approach can be found in [87, 33]. The shape-based deformable models introduced in [116] use an orthogonal boundary representation [116] and have the capability to use shape priors easily. (This is the approach that we have chosen, more details of which will be provided in chapter 3.) Other shape-based methods such as [37] or [119] (which in turn is a development based on [116]) use pattern matching techniques to constrain both the overall shape and the possible deformations. Recently, a new breed of topologically adaptable active contour methods [94, 22, 82, 85] and their variants (such as the reaction-diffusion approach of [121]) have been used that are adaptable to changes in the topology of the target object.

This work primarily focuses on segmentation of whole structures from images and is consequently based on a whole boundary method. However, while the above-mentioned whole boundary methods are free of broken edges, they still suffer from a variety of problems similar to that of edge detection as long as they use gradient information alone to detect the true boundary. Specifically, when the high frequency information in an image is either noisy or missing, the gradient-based boundary finder tends to drift away from the actual boundary. This problem becomes acute when the signal to noise ratio (SNR) is low, because the gradient is a derivative process that amplifies noise. Another associated problem with these methods is that of initialization. Due to the presence of other objects in an image and also due to noise, these methods typically need an initial guess from where they converge to the actual boundary. Especially when the noise is high, due to the presence of multiple local minima, they may not converge unless the initialization is very close.

One of the ways to address the problem of a low SNR for deformable boundary finding is through the addition of region information. Since this information can be obtained (as in the present case) directly from the raw image data (without taking derivatives), it is less susceptible to noise. Also, region information is a more global source of information since image classification is done based on the statistical properties of the different regions in an image, unlike the gradient information which only uses the pixel values of a particular site. This global information also makes the problem more robust to the problem of poor initialization.

Thus, an integrated deformable boundary finding system using both gradient and region information is likely to perform better. While the details will be provided in the next chapter, we note here that there is a limited amount of previous work available seeking to integrate region-based methods with boundary-based methods. Among the various methods [29, 53, 47, 91, 14, 121], only the recent effort of [138] tries to integrate region and boundary information within a solid mathematical framework even though the emphasis here is more on the region module.

We take the view that integration is best achieved via mathematical optimization. In its simplest form, additional information can be introduced as extra constraints within the same optimization framework. This is similar to the situation in a least squares fitting problem when extra observations become available. We will define this form of integration, where the optimization is carried out over the same set of parameters, but the solution depends on a larger information set (here, image-derived features) as *feature integration* (within the deformable boundary finding framework). This is so because of the use of additional features. Alternatively, a more general approach can be used where computational modules are associated with the different features. Module parameters are then updated through the optimization process. This is what we denote the *feature-based module integration* framework. Here, parameters associated with the different modules are updated simultaneously. In the *feature integration* framework, new features are used as additional sources of information within the same computational framework (i.e. optimization is carried out over the same set of parameters). However, in *feature-based module integration* different computational modules using different information sources are combined and the optimization is carried out over the parameters of both the modules. Clearly, *feature integration* as defined here, thus can be considered to be a special case of the just defined *feature-based module integration* framework.

As mentioned before, one of our main goals is to make the boundary estimate more robust to noise and poor initialization. Now, since we have already observed that region information has better noise properties, our first effort involves feature integration within the boundary finding framework by using the region segmented image as an additional source of information. A simple way to incorporate this information consists of first producing the region information and then using this in the boundary finding process (see chapter 4 for the details). The important thing to note is that the use of this additional feature do not change the parameters over which the optimization is carried out.

Our next effort is aimed at addressing yet another problem that often plagues the gradient-based boundary-finder. Boundary finding often has a tendency to smooth out the sharp corners in an object resulting in large localization errors. There are a couple of reasons as to why this happens. First, we emphasize smoothness of the boundary to reduce the effects of noise. The other is that if we consider the boundary to lie at locations where the local pixel-wise difference is a maximum, by using a gradient-based boundary finder we are in fact ignoring all the other higher order information. Higher order information such as grey level curvature along with gradient could be incorporated as a further feature in addition to gradient and region homogeneity within the boundary finding framework (see chapter 5 for the details).

Feature integration within the boundary finding framework improves the boundary estimate by using the region information as an additional feature besides the gradient information. If that is indeed the case, the dual situation might be true as well. In other words, it is likely that using information from the boundary finder as an additional feature for the region process will refine the region estimate. Now this improved region information can be fed back to the boundary finder to further improve its output. We can continue to do this until the system reaches a stable equilibrium. Clearly, such a *feature-based modular integration* would produce a superior output for both the modules than what we might obtain by *feature integration* alone. As said before, one can visualize *feature integration* as a special case of the more complete *feature-based module integration.* While in *feature integration* the output of one of the computational modules (i.e. boundary finding) is improved, in the more general case of *feature-based modular integration* all the computational models are simultaneously updated.

Thus effectively, we are considering an integrated effort that could potentially improve both methods through information feedback. However, any integration always results in a larger, more complicated system. In general, analyzing large and complex information systems is a daunting task. Probably the easiest way is to divide the system into functional modules and analyze them separately. Hence, the reverse problem, i.e. the development of integrated methods often consists of intelligently assembling smaller well-understood modules, leading to what can correctly be termed *integrated* systems. Here, *integration* refers not only to succinctly describing the mathematical relations between these modules but also describing algorithms that can perform the integration.

In the general case, we have a team of modules (two for the present segmentation problem) working in a coordinated fashion in a possibly hostile and uncertain environment towards common or at least related goals. It is important to note that the members do not necessarily acquire the same information and thus operate in a decentralized mode of decision making. The formulation of such a problem consists of modeling the objectives of the modules as cost functions that have to be optimized. Modeling the uncertain environment and the possible measurements made by the modules in this environment is done in the form of a probability space along with a suitable information structure [56].

If we assume the presence of a common probability space, thereby allowing the modules to share a similar view of the world, this could lead to probably the most common method of optimizing multi-objective functions, namely the single objective function approach. Here, all the objective functions of the different modules are combined into a single cost function which is then optimized [10]. Besides the single objective approach, other proposed approaches have been based on artificial intelligence techniques and systems that are a mixture of mathematical, control-theoretic and algorithmic techniques [81]. While sometimes such techniques may provide constructive solutions, often their utility is limited by their inadequate analytic and computationally tractability [18].

While the single objective approach often seems to be the method of choice in the vision literature [81, 14], we note that there are occasions when it is inadequate or at least not the best approach. One of the important concerns is the robustness of such models and the *optimum* solutions that it produces under small variations in the underlying assumptions. This could be important especially in situations where the models perceive the outside world differently i.e. if they have different probability descriptions of the environment. This would then lead to different expected cost functions for the different modules, making the use of a single objective approach at best sub-optimal. A big advantage of treating the problem within a game- theoretic framework, in particular as a stochastic non-zero sum game with the modules taking on the roles of decision makers having different subjective probability measures [7], is that there is no need for calculating the joint probability distribution function.

Another advantage of using this game-theoretic approach is that any one module need not know the cost functions of the other modules. This approach keeps them decoupled. The flow of information is restricted to passing only the results of the decisions between the modules. This allows us to combine incommensurate objectives through this approach which might not be possible using the conventional single objective approach. However, the outputs of the two approaches (i.e. the game theoretic approach and the single objective approach) may not necessarily be the same as has been pointed out in [17]. Thus the application of either will depend on the appropriateness of the solution that might be achieved using either of the methods. Further, the decoupled nature of the system has the practical benefit of being computationally light.

All this is of interest for our segmentation problem. The probability distributions of the gradient and the region process are definitely different, even if they are related. Thus, in order to use the global objective function appropriately, we need to calculate the joint probability distribution function of the boundary and the region process, which in itself is very complicated. On the other hand, using the game theoretic approach, this problem can be circumvented because when optimizing the cost function of one of the methods, the output of the other is only used as a prior, in the maximum *a posteriori* framework that we use. Thus, the boundary and region processes remain reasonably decoupled, yet they influence each other through mutual information sharing.

Despite resulting in complementary sets of information, the region and boundary processes involve conflicting and incommensurate objectives, as region-based segmentation attempts to capitalize on homogeneity properties whereas boundary finding techniques use the non-homogeneity of the same data as a guide. Thus, combining them into a global objective function is non-trivial. However, as mentioned before, using the game theoretic framework, there isn't any need to combine them. Further, it is also important that it is computationally light because a large number of parameters are generally involved.

1.2 Main Contributions

This thesis is concerned with robust and reliable methods for image segmentation. We have developed a unique new approach to integration that can accommodate appropriate features and feature-based modules resulting in segmentation methods that are considerably superior to the existing methods. To obviate some of the problems associated with integrating image segmentation methods (which were mentioned in the last section), our work looks into the problem from two inter-related viewpoints, *feature integration* and *feature-based module integration.* An important thing to note is that all the optimizations are motivated by the maximum *a posteriori* (MAP) formalism. The main contributions of the work are listed below.

First, we have developed an integrated deformable boundary finding method that uses region information in addition to the gradient to locate the final boundary. We have done this by emphasizing region homogeneity within the area enclosed by the deformable boundary along with high gradient at the boundary location. It is not necessary for the regions to be perfectly homogeneous, but instead assumes that the intra-region variation should be smaller than the inter-region variation. This use of region information has resulted in a method that is significantly more robust to both noise and poor initialization than the gradient-based boundary finder of [116] as the experiments in chapter 4 demonstrate. We achieved this without significantly increasing the computational overhead through appropriate application of Greens theorem. We note that the above was done within the *feature integration* paradigm defined before and has been reported in chapter 4.

Second, we have further developed the deformable boundary finder to improve its localization accuracy by using curvature, a second order feature in addition to gradient and region information. This was done once again, within the *feature integration* framework. Especially for sharply bending objects, it resulted in considerable improvement. This has been reported in chapter 5. In this chapter, we have also developed a model to select which feature should be emphasized and under what circumstances via theoretical and experimental analysis.

Next, we have developed a *feature-based module integration* framework for the image segmentation problem using game theory. This section of the work reported in chapter 6 makes two important contributions. The use of game theory within the image analysis setting was first introduced in [18]. The primary contribution here is in extending the game theoretic framework to integrate region and boundary procedures. Computational modules assigned to the above processes took on the role of players and the final decision was obtained as the Nash Equilibrium of the resulting non-zero sum two-person non-cooperative game. The second contribution relates to the conditions for the existence and stability of the Nash equilibrium for problems like the present one (i.e. integrating boundary and region process) where the cost functions of the two modules take a particular structure. We state and prove the condition under which cost functions having the above structure will have a stable Nash equilibrium. It shows how the existence of the Nash equilibrium can be controlled by the appropriate choice of constants.

Finally, we have developed a new approach to 3D deformable surface finding that uses in addition to prior and gradient information, region information as well. We took the Fourier parameterized surface finding algorithm developed in [118] and integrated it with region based methods in a way similar to the two dimensional case discussed earlier. This made the surface finding procedure more robust to noise and initialization.

1.3 Overview

This thesis investigates integration methods for robust and reliable image segmentation. The present chapter serves to introduce the problem of integration and our analysis as to what the possible solutions could be.

Chapter 2 reviews the relevant background literature on the different methods for image segmentation.

In chapter 3 we discuss in mathematical detail the segmentation methods, both region-based and boundary-based that we use in the subsequent chapters. It is these methods that we integrate. In particular, the Fourier parameterized deformable boundary finding method and Markov Random Field based region segmentation techniques that we use throughout the rest of the thesis are discussed here.

Chapter 4 formulates how region information can be integrated within a deformable boundary finding framework [25, 24, 26]. The motivation and the mathematical formulation is followed by extensive experimental analysis to demonstrate decisively that the Greens theorem-based formulation indeed improves the performance of the deformable boundary finder both for synthetic as well as real images.

We extend the feature integration paradigm (for deformable boundary finding) in chapter 5 by introducing curvature into the deformable boundary finding framework [27]. We also present analytical as well as theoretical results to show how much of which information should be used under a particular circumstance.

Chapter 6 takes up the problem of modular integration which is considered to be a more complete method for image segmentation, with a view towards generating better boundary solutions. Game theoretic modular integration is suggested as an alternative to the traditional single global objective function minimization procedure. The chapter starts with some of the basic definitions of Game Theory which will be necessary for the subsequent development. This is followed by modeling the segmentation problem as a two person game where the computational models take on the roles of decision makers [23]. We also discuss the stability and existence of the Nash equilibria.

Finally, in chapter 7, we extend some of the integration methods that we had suggested to surface finding for three dimensional images. We briefly explain the relevant surface parameterization, give the mathematical formulation and show experimental results.

Chapter 2

Related Work

In this chapter, we describe briefly the available literature on the topics that are relevant to this work. Over the years a number of different approaches has been developed for image segmentation, excellent reviews of which can be found in [45, 55, 97]. Here, we first will describe some of those approaches. Next, we will describe briefly the available work on integrated image segmentation methods which have recently become popular in view of their ability to overcome some of the limitations faced by any one of these above-mentioned methods. In the same context, we will also describe earlier work on applications of game- theoretic integration to computer vision.

2.1 Related Work in Image Segmentation

2.1.1 Region based Methods

We classify region-based methods as those that exploit homogeneity of spatially dense information, e.g. pixel-wise grey level values, texture properties etc. to produce the segmented image. Thus these methods primarily depend on the underlying consistency of any relevant feature in the different regions of an image. Formally, the underlying principle behind these approaches can be defined as follows [97]: if F is the set of all pixels and $P()$ is a uniformity predicate (of any one of the above mentioned properties) on groups of connected pixels, then region-based segmentation is a partitioning of the set F into a set of connected subsets or regions $(S_1, S_2, ..., S_n)$ such that

$$\bigcup_{i=1}^{n} S_i = F \ with \ S_i \cap S_j = \phi, \ i \neq j \tag{2.1}$$

The uniformity predicate $P(S_i) = true$ for all regions S_i and $P(S_i \cup S_j) = false$, when S_i is spatially adjacent to S_j.

2.1.1.1 Thresholding

Thresholding is one of the oldest and most widely used tools for image segmentation. It can either be done based on information derived from the whole image (e.g. grey level histogram of the entire image) or it can be done based on local image characteristics (cooccurance matrix) of the image [108, 120]. If the image is composed of distinct regions, characterized by their grey level, then the histogram is generally characterized by sharp peaks and deep valleys, thus providing natural choices for threshold levels. But often in a noisy image that is not the case, making the job of threshold selection non-trivial. Different methods have been tried. Otsu [95] maximizes a measure of class separability, Nakagawa and Rosenfeld [92] minimized the total misclassification error under assumptions of normal distributions. Others used more sophisticated measures like the image entropy [131, 78] and so on.

However, the basic assumption behind a global thresholding scheme, that pixels higher than a certain grey level fall into one region and those that are lower than the threshold, belong to another region may not often be true, particularly when the image is noisy or the background is uneven and the illumination is poor. This image variability makes adaptive thresholding techniques necessary where local thresholds are chosen for local image blocks which are then interpolated to produce a threshold surface [133]. However, despite the use of adaptive thresholding, segmentation by thresholding is extremely noise sensitive and unreliable, producing isolated regions of mis-classified data, jagged boundaries etc.

2.1.1.2 Region Growing

Region growing methods have some similarity to the adaptive thresholding methods in the sense that they too are generally dependent on the pixel statistics over localized areas of the image. The most widely used region growing methods are the split and merge techniques [65, 68]. The idea here is to first plant some seeds at different points in the image by choosing some pixels as representative of the attributed regions. Then the regions associated with these points are allowed to grow based on their homogeneity properties. The problem however is that the resulting segmentation depends considerably on the choice of the seed points and the region's shape is dependent on the choice of the actual algorithm used. Also, more often than not, these methods result in an over-segmented image, although this can be corrected through the use of domain specific but rather ad-hoc rules [20, 77].

While region growing methods perform better than thresholding procedures, they continue to have similar problems, producing artificial and poorly localized boundaries. Thus, to produce reliable boundary estimates, further processing often becomes necessary.

2.1.1.3 MRF based approaches

Region-based image segmentation techniques using spatial interaction models like Markov Random Field (MRF) or Gibbs random field to model the image [47, 41, 40] have become very popular recently. This started with the influential work of Geman & Geman [47] who forged an elegant link via statistical mechanics between mechanical systems like the soap films or splines (that minimize energy) and probability theory. They have shown that in effect, signal estimation by least square spline fitting results in optimal estimates if certain *a priori* probabilistic beliefs about the world in which the signal originated is to be expected, especially if the estimated signal can be modeled as samples from a Markov Random Field and Gaussian noise was assumed to have been added during the process of data generation. The MRF assumption is useful because it gives a probabilistic description of interaction. It says that the probability of a point being in a particular state is entirely dependent on the probabilities of the states of its neighbors. However, despite this local description, global interaction is still possible as a result of propagation. Further, it is also possible to directly include interaction from distant points [48].

Formally, Geman & Geman [47] and related efforts showed that the link between the energy of the spline reconstruction E and the probability of the corresponding image realization is given by [14]

$$P \propto e^{-E/T} \tag{2.2}$$

(T is a constant). Thus, the lower the energy of a particular signal (that was generated by a particular MRF), the more likely it is to occur. This is so because the energy is computed as a measure of the distance between the model and the raw image data.

Further, Geman & Geman [47] also developed a restoration algorithm based on stochastic relaxation and annealing. Introduction of the above result, which is also known as the Hammersley-Clifford theorem resulted in a number of region-based image segmentation techniques. Most of them can be categorized into two main groups brief discussions of which follow.

The first version considers the problem of simultaneously smoothing the image and detecting discontinuities. It can be formulated in terms of a simple energy function that has the appropriate interaction between the data field and the line processes (which are defined as a discontinuity field). A simplified version of the model is given by the weak membrane energy which for the continuous case was proposed by Mumford & Shah [91] and for the discrete lattice by Geman & Geman [47], Blake & Zisserman [14] etc. The most basic formulation given by Mumford & Shah [91] considers minimizing:

$$E = \int\int_I (f(x) - g(x))^2 dx + \alpha \int\int_{I-C} ||\nabla f(x)||^2 dx + \gamma \int_C ds \tag{2.3}$$

where $f()$ is the smoothed image, $g()$ is the noisy image data, $I(\subset \mathcal{R}^2)$ is the image domain, and C is the set of curves that divide up the image domain into a set of regions. Here, the first term is the data fidelity term, the second term the smoothness term and the last term is the penalty term for introducing edges. Thus it produces a smooth reconstruction of the image except at discontinuities. While in general this method and its variants [46, 14] might produce better results than some of the approaches mentioned before, they still suffer from the problems of unwanted and poorly localized edges that are characteristic of any region-based method.

Another problem of the above approach is that it is ill-suited for the problem of texture segmentation in its basic form. This brings us to the second approach that is also MRF based but can naturally handle more diverse region descriptions than just uniformity in the grey-level values as in the previous case. These methods can be considered as classification procedures that label a pixel as belonging to one of a finite number of classes [11, 41, 39, 83, 73]. There are two things that need to be estimated for this class of problems. First, we need to estimate the class properties, following which the actual classification is done. For a simple grey level image, the class properties correspond to the mean and the variance, whereas for texture images, a larger number of parameters are used to characterize the texture. Thus, these methods are more general than the methods discussed before. However, the problem lies in estimating the class properties for which the segmented image is necessary. To avoid this problem, both the class properties and the pixel classification can be solved simultaneously as in [73], or one can first calculate the class properties from a crude estimate of the segmented image and then do the classification [83, 98]. The optimization process is more complex in these types of region-based methods, because unlike the previous case, here the optimization cannot be posed as an energy minimization problem because the probability function generates a combinatorial explosion. While these methods can be more general than the region-based methods discussed earlier, in the sense that they have the ability to handle a larger class of image features, they continue to suffer from the problems of poor localization and unwanted boundaries resulting from misclassified pixels.

In either of the above cases, global optima can be achieved only through the use of a global optimization method like simulated annealing which is computationally very expensive [47, 73]. However, suboptimal methods like iterated conditional modes (ICM) algorithm [11, 83] often produce comparable results. More elaborate discussions of the

MRF based approaches used in this work can be found in chapter 3.

2.1.1.4 Non-linear Diffusion methods

Non-linear diffusion methods [101, 49, 1], like the MRF based methods discussed before [91, 14] are used for doing *edge preserved smoothing.* Non-linear diffusion can be considered to be a natural extension of scale space filtering. Linear scale space methods solve the linear diffusion equation and were first developed by Witkin [130] and Koenderink [67] and further developed in [4, 135, 58]. The essential idea here is to produce a multiscale description of the image i.e. embed the original image in a family of derived images $I(x, y, \sigma)$ obtained by convolving the original image $I_0(x, y)$ with a Gaussian kernel $G(x, y, \sigma)$ i.e.

$$I(x, y, \sigma) = I_0(x, y) * G(x, y, \sigma) \tag{2.4}$$

Now this one parameter family of derived images can be viewed as the solution of the isotropic diffusion equation,

$$I_\sigma = \Delta I \tag{2.5}$$

where Δ is the Laplacian operator. However, the problem (as noted in [101]) with this method is that the true location of a boundary is not directly available at a non-zero scale image because the smoothing generated does not respect the existence of edges. To obviate this problem, the anisotropic diffusion equation

$$I_\sigma = div(c(x, y, \sigma)\Delta I) = c(x, y, \sigma)\Delta I + \nabla c \cdot \nabla I \tag{2.6}$$

was suggested [101, 1]. Here, the smoothing is controlled by the function $c()$ which is nonlinear. When constant, it reduces to the isotropic diffusion equation. Different choices for $c()$ have been suggested, the main idea being to let $c()$ decay to small values where the gradient is high, thereby preventing smoothing across edges [101, 106]. Clearly, the motivation is similar to the ideas in the MRF based energy minimizing methods [14, 91] described before. Mathematical derivation of this equivalence can be obtained in either [101] or [46].

It would be appropriate to mention here on a related note that another class of non-linear methods that generate medialness description has also recently been advocated as a plausible representation for structures in an image [80, 44]. Any structure is represented by a line passing through the center of that object and the radial width at all points on that central axis. This forms what has been described as the "core" of the object. Often this is done over multiple scales and is used for instance in image registration.

In the next section, we will discuss the boundary methods that have better localization properties than the region methods but suffer from high noise sensitivity.

2.1.2 Boundary Methods

Boundary based methods rely on the pixel-wise difference to guide the process of segmentation. Thus they try to locate points of abrupt changes in the grey tone images. There are a number of approaches that use boundary information some of which are discussed below.

2.1.2.1 Edge Detection

Edge detectors are model-free local boundary finders where for any image the edge strength is calculated by using a local differential operator. Important requirements for a good edge operator are that it should be a differential operator and it should be capable of being tuned to act at any desired scale. There are a variety of edge operators such as Robert's, Sobel, Prewitt etc. [107]. One of the most influential methods use the zero-crossings of the Laplacian of the Gaussian of the image [84]. It produces continuous closed boundaries (being the zero-set of an implicit function), but since it is a second derivative operator, it is extremely noise sensitive. Also, it is difficult to distinguish between more and less probable edges and the only way of discarding false edges would be to do so interactively by an human operator.

The Canny edge detector [21] seeks to optimize by making the best trade-off between detection and localization of edges, the two main yet opposing goals of edge detection, in addition to producing one and only one response for a single edge point. He showed that the ideal edge detector can be approximated by first taking the gradient of the image convolved with the Gaussian kernel and then choosing the maxima of the gradient image in the direction of the gradient. However, except under ideal circumstances, it results in broken edges and thus identifying objects would require grouping those edges either interactively or by using some other algorithm [107].

2.1.2.2 Hough Transform / Rigid Templates

Hough transform may be considered to be an alternative to boundary analysis. It is essentially a mapping from the image to a parameter space. It was originally used for parameterizable shapes like curves, but was extended later to include general shapes [5].

Likely boundary points are identified first in the image, which are then mapped to the parameter space. Maxima in the Hough space correspond to possible shape instances. The advantage of this method is that it is relatively insensitive to gaps and noise. The drawback however is the computational requirement that goes up geometrically with an increase in the number of parameters. And for deformable boundaries, it becomes even more unsuitable.

This finally brings us to the whole boundary methods that we shall briefly describe below and then in more detail in chapter 3.

2.1.2.3 Whole Boundary Methods

Whole boundary methods [116, 62, 3, 37] rely mainly on the gradient features at a subset of the spatial positions of an image (near an object boundary) for segmentation of structures from an image. By considering the boundary as a whole, a global shape measure is imposed on the problem that simplifies the task. Thus gaps are prohibited and overall consistency is emphasized. Once again, there are a number of approaches, some of which we will describe here, more details about which can be found in the book by Blake & Yuille [13] which contains an excellent collection of papers on deformable contour methods. We will also discuss some of the most recent work on this topic using curve evolution methods [82, 22, 63].

One of the first instances of this approach used parameterized templates called rubber masks by Widrow [129]. Yuille [136, 134] used a parameterized template to model the features of an human face. In both these cases, the overall object was broken down to parts that were subsequently modeled. These methods were not very generic, in other words, a new model was necessary for every new object.

2.1.2.3.1 Energy minimizing snakes One of the most generic and popular methods of detecting whole boundaries is the snakes approach due to Kass et al [62]. A snake is a continuously deformable curve that can be used as a mechanism to locate features of interest in an image. It's shape is controlled both by internal forces (the implicit model) and external or image forces. If $v(s) = (x(s), y(s))$ is the parametric description of the snake ($s \in [0, 1]$)), its total energy can be written as

$$E_{snake} \quad = \quad \int_0^1 \left[E_{int}(v(s)) + E_{image}(v(s))\right] ds \tag{2.7}$$

where

$$E_{int}(v(s)) = \frac{1}{2}\left(\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2\right) \tag{2.8}$$

and

$$E_{image}(v(s)) = w_{line}E_{line}(v(s)) + w_{edge}E_{edge}(v(s)) + w_{term}E_{term}(v(s)) \tag{2.9}$$

The internal energy, E_{int}, represents the smoothing forces on the curve, and E_{image} represents the image-derived forces that constrain the curve to take the shape of the features present in the image. The internal energy forces $v(s)$ to be a controlled continuity spline [123] with its first order membrane term favoring points to become close to one another and its second order thin plate term favoring low overall curvature.

The image energy consists of a linear combination of three image-derived terms. The line energy attracts the contour to lower or higher intensity values in the image depending on the sign of w_{line} with $E_{line} = I(x, y)$. The edge energy calculated as

$E_{edge} = -|\nabla I(x,y)|^2$ attracts the contour to image points with high gradient. E_{term} being the curvature of the level curves in a Gaussian smoothed image, attracts the contour towards line terminations. Minimization of the above was achieved by using the variational method that involves solving for the corresponding Euler equations.

The problem with this initial version was that if the initial curve was not close enough to an edge, it had difficulty in being attracted by it. Also, the curves in the original snakes version had a tendency to shrink on themselves. To improve the convergence properties, [2, 3] used dynamic programming to do the energy minimization. Also, [3] extended the objective function to include hard local constraints. Some robustness to the problem of finding a good initial guess was achieved by the addition of an extra force which made the contour have a more dynamic behavior. In other works, the curve was considered to be a balloon [33] (in 2D) that was being inflated. From an initial oriented curve, an extra pressure force was added that pushed the curve outside as if air was being introduced inside the closed contour.

However, despite all these improvements, this direct representation still had the problem of having too many parameters (here the position vectors along the curve) to optimize over. To alleviate this problem, a parameterized B-spline [6] approximation was used in [87] where the curve was replaced by its B-spline approximation and the energy of the approximation was minimized.

2.1.2.3.2 Shape-constrained deformable models As an alternative to the snakes method, orthogonal representations for curves were proposed in [115, 116] where a curve was expressed in terms of an orthogonal series. Thus the curve was expressed as a weighted sum of some known functions. An orthonormal set is desirable because it makes the parameters uncorrelated. This reduction in redundancy becomes particularly helpful during the optimization process. In our work here, we chose to use this Fourier parameterized boundary approach mainly because of its suitability for the types of problems we were interested in, its compact representation, and its ability to incorporate prior shape information. Since, this constitutes our primary approach, we will describe this method in more detail in the next chapter (chapter 3).

While the use of prior shape can be used as an optional constraint in the above method of [116], the "active shape model" of [37] constrains not only the overall shape but also any deformation to be consistent with a model built from a training set. Thus for every single object that needs to be detected, a model is created that learns the pattern of variability from a training set. Thus only those variations that have already been experienced before can be detected. While this pattern matching procedure makes the process more robust, it takes away some of the flexibility as well. Thus even for high quality images, one has to depend primarily on the model. Also, only those objects for which a model has already been created, can be detected, making the process very task-specific.

2.1.2.3.3 Curve Evolution methods While the last two deformable boundary finding approaches (energy minimizing snakes [62, 3] and Fourier parameterized boundary method [116]) can be used successfully in a variety of image processing/computer vision applications, they have one important limitation: they are unable to handle changes in topology of the deforming contour. To circumvent that, a new family of deformable boundary methods, based on the curve evolution techniques of Osher & Sethian [94] have recently been introduced [82, 22, 63].

The essential idea here is to first represent the contour that we are interested in as a front $\gamma(t)$. Now, given a moving closed contour $\gamma(t)$, the idea is to produce an Eulerian formulation for the motion of this contour propagating along its normal direction with speed F, where F can be a function of the curve characteristics (like the curvature, normal direction etc.) or it could be a function of the image characteristics (e.g. the grey level gradient etc.). This is done by embedding this propagating interface as the zero level set of a higher dimensional function Ψ defined by

$$\Psi(\mathbf{x}, t = 0) = d \tag{2.10}$$

where d is the distance from $\mathbf{x}$ to $\gamma(t = 0)$. The evolution equation of Ψ, inside which our contour is embedded as the zero level set is given by:

$$\Psi_t + F|\nabla\Psi| = 0 \tag{2.11}$$

where the speed function is given by:

$$F = k_I\,(F_A + F_G) \tag{2.12}$$

k_I is the image derived term and is often a function of the reciprocal of the gradient, the purpose being to stop the front at locations where the gradient is high. The term F_A is a constant and is independent of the moving front's geometry. F_G depends on the geometry of the front (such as the local curvature) and is responsible for smoothing out high curvature points on the curve.

While the above is the basic idea behind the approaches of [94, 82, 22, 110, 137], a variation of the above is the reaction-diffusion method of [121]. Here random bubbles are introduced throughout the image, which then propagate using the evolution equation given before. Whenever these bubbles meet they produce “shocks”, which are then handled using “shock-capturing” numerical algorithms [109, 94].

Besides these curve evolution methods, in [85] a variation of the original snakes approach have been reported that handles changes in the topology of the contour by embedding the snake in a simplicial domain decomposition that handles the splitting or merging of contours in a topologically meaningful way.

However, it is important to remember that the whole boundary methods have some limitations as well. Being gradient-based, they suffer from high noise sensitivity. Also, if the high frequency information in an image is either absent or is missing, gradient-based methods tend to diverge. The following section addresses some of these problems via the use of integrated methods.

2.2 Image Segmentation using Integrated methods

There has been only very limited previous work seeking to integrate either different features or different modules to improve the segmentation process.

Simultaneous consideration of smoothness and contrast was first used in [89] which applied dynamic programming for minimizing the weighted sum of a contrast measure and smoothness measure. The snakes approach of Kass et. al. [62] solves explicitly a regularization problem to locate object boundaries.

However, seeking to integrate region and boundary information has only been tried relatively recently. The difficulty is that even though the two methods yield complimentary information, they involve conflicting and incommensurate objectives, as region-based segmentation attempts to capitalize on homogeneity properties whereas boundary finding techniques use the non-homogeneity of the same data as a guide. Thus, as previously observed [99], even though integration has long been a desirable goal, achieving this goal is non-trivial. Among the available methods, AI-based techniques have been used where production rules are invoked for conflict removal [99]. Other efforts have used probability-based approaches (see e.g. [47, 46, 91, 14, 66, 53]) where often the aim is to maximize the *a posteriori* probability of the region classified image given the raw image data by optimization methods like simulated annealing. Integration here is achieved in the local or dense field sense where the edges are used as line processes and the optimization is achieved both over the location of the line processes as well as the pixel classification. These methods, as mentioned before, are thus primarily aimed at doing edge-preserved smoothing rather than truly integrating edge and region processes. Furthermore, these methods are concerned only with the grey-level homogeneity, and are not suitable for handling other features like texture, similarity in the optical flow etc. which are more incommensurate with the gradient-based boundary methods.

In some methods [99, 53], region growing is done first followed by an binary edge detection step. There are a few disadvantages to this procedure. First, a region classified image is often over-segmented due to the presence of noise. Thus, one needs a validating scheme to distinguish between true and false edges. Also, such a scheme has no way of differentiating between multiple regions as it deals with the binary edge map obtained from the region grown image. Further, such a method may suffer from the effects of poor edge localization as is often the case with region-based segmentation. Some of these problems can be alleviated using the method in [29], but it still is an edge-based approach with associated drawbacks.

The reaction-diffusion method of [121] has bubbles (described earlier) deforming unless encountered by changes in any one of the low level process that it endeavors to integrate. However, the problem of using such a method is that if any one of the processes make an error (e.g. a false edge), that is propagated to the final solution. Also, there is no notion of global shape attached to it. Further, decision regarding the final object boundary is made here by considering the whole space of reaction-diffusion images, something that can get very complicated.

However, the recent work of [138] has similar motivations as ours even though here integration is achieved through the single objective method and the final implementation is similar of the weak membrane method of [14, 91]. Also, while it is constrained to produce closed boundaries, it does not handle the issue of shape in a similar way as is done over here.

2.3 Game Theoretic approaches for Image Analysis

Most of the work involving integration in image analysis systems [29, 81, 14] have used the single objective approach that combines the different objectives into a single global objective function which is then optimized. However, as noted in the introduction and in [7], this may not always be appropriate especially when the modules that are being integrated have different views of the world, characterized by different probability spaces. Also, a global objective approach may be computationally complex. Expectation Maximization methods [38, 74] can be used, but for the general case of unknown joint probability distribution, use of nonparameteric methods become necessary that could result in algorithms that might be as complicated and as sub-optimal as the global objective function approach.

In view of the above, game theoretic methods were suggested in [7, 18, 17] for module integration. Game-theoretic methods for vision and image analysis in particular were used in [18] for image segmentation by integrating locally curvilinear edge-like structures to boundaries. Besides that work [18], further references to which will be made in a later chapter (chapter 6), little work using game-theoretic ideas has been performed within the field of computer vision.

In the present work, we will use a model-based game-theoretic integration framework to integrate the region and boundary models.

2.4 Three Dimensional Object Segmentation

Segmentation of three dimensional images is of particular importance in the medical imaging domain, where 3D images are routinely acquired. A large number of the segmentation ideas discussed above can be extended from two dimensional images to three dimensional images at least theoretically. This is particularly true for most of the region-based methods. In [30] a multispectral voxel classification method is used in conjunction with connectivity, to segment the brain into different tissue types from 3D MR images. This method is limited by the assumption of normality in the probability distributions of the tissues. Gerig *et al.* [50] used a similar approach. Raya [104] uses multispectral classification in conjunction with a rule base to classify MR brain images. In [19, 126] classification via histogram thresholding for 3D MR images.

However, unlike in the region-based case, extension of boundary methods to 3D images is non-trivial. The first such model, the Generalized cylinders proposed by Binford can only be applied to a very restrictive class of objects that are naturally cylindrical [103]. The 3D snakes model of [124] is more flexible because it has deformation parameters to control the elasticity of the main axis and the walls of the cylinder, but it still suffers from some of the same problems. Superquadrics model objects with deformable ellipsoids

[112] with parameters for such operations as twisting, bending and tapering. While they have a compact description with a small number of parameters, the nature of the basis used, highly constrain the range of shapes that can be modeled. Strain modes [100] and wavelet basis [127] have been used to augment their modeling capability. Another related development uses hyperquadrics [72], which has been shown to be more flexible than the superquadrics. The Fourier parameterized surface model of [117, 118], a direct extension of the 2D Fourier parameterized boundary method, describes the surface by the leading coefficients of a Fourier series taken of the surface coordinates. More details on this parameterization will be provided in a later chapter (chapter 7). A big problem with many of these methods however, is the one of proper initialization. To circumvent that, recently a method very similar to the pattern learning mechanism of [37] was suggested [119], that creates a 3D Fourier surface model for the structures of interest from a set of training images, which also constrain the possible variations to those already experienced within the training set. As in the 2D case, however, while it makes the problem more robust, it also severely limits the flexibility of the surface finder.

Chapter 3

Segmentation Models

3.1 Introduction

In the previous chapter we gave an overview of the different methods that have been used to perform image segmentation. In this chapter we will describe in mathematical detail the specific boundary-based and the region-based segmentation methods that we will try to integrate in the subsequent chapters.

3.2 Boundary Finding

As has already been indicated, our interest lies in a whole boundary method. Segmentation by boundary finding using only local information has often been confounded by among other things, poor contrast, occlusion, adverse viewing conditions and noise. A model-free interpration is hence problematic due to the under-constrained nature of the problem. This makes appropriate modeling of the boundary important. One of the biggest advantages that a good model can bring into the boundary finding procedure is that it constrains the optimization process to handle missing or noisy data [62, 116].

Also, it allows us to introduce prior information about the shape if available. Without these constraints the optimization process might get unnecessarily complicated particularly because our aim is to find boundaries of continuously deformable objects (examples of which are abundant in the medical imaging domain) as opposed to shapes that have an obvious decomposition [52].

As has already been discussed in the previous chapter, various boundary parameterizations are available in the literature. While none of them can be considered to be optimal under all circumstances, each one has its own advantages and disadvantages. Each has particular properties that make it suitable for a certain class of problems. For example, operations such as the calculation of geometric properties such as area or moments are directly available from some representations. Similar is the case with the calculation of an inside-outside function that specifies the relation of a point with respect to the object boundary. On the other hand, such approaches also impose a limitation on the domain of shapes that can be handled. Some restrictions, such as smoothness can actually be advantageous as they build in a necessary constraint directly into the representation. For problems, such as matching, it is important that a bijective mapping exist between the shape and its parameterization. This mapping should also be continuous and stable so that small deformations in shape result in small changes in the parameterization. This would allow for matching to take place in the parameter space. It is also desired, especially for optimization purposes, that the parameterization be concise. This saves both memory and time.

In any case, the point to note is that there are various parameterizations available and the choice of any one of them should be guided by the problem at hand. In the last chapter we gave an overview of the different methods available. Here, we will describe for the sake of completeness the shape constrained deformable boundary model of Staib & Duncan [116] which we use for the class of problems that we are interested in. This is primarily so, because we are mainly concerned with objects, whose shape often vary around an average shape, an information that can easily be used as shape priors within this formulation.

3.2.1 Shape-based Deformable Boundary Finding

The shape-based deformable boundary finding method of Staib & Duncan [116] uses orthogonal representations to describe the boundary. Thus, the curve is expressed as a weighted sum of some known functions. Use of an orthonormal set makes the parameters distinct. This becomes particularly helpful during the optimization process.

The function $X(t)$ on the interval (a, b) in terms of the basis $\phi_k(t)$ can be expressed as,

$$X(t) = \sum_{k=1}^{\infty} p_k \phi_k(t) \tag{3.1}$$

where

$$p_k = \int_a^b X(t)\phi_k(t)dt \tag{3.2}$$

The coefficients $\vec{p}$ (the projection of $X(.)$ onto the basis functions) form the parameter vector. In order to use this representation however, the sum must be truncated. We shall have more to say about it in a later section.

Sinusoids are the most popular basis functions [102] even though other orthogonal basis could be used as well. The main advantage with sinusoids is their ease of representation and their familiar association with frequencies making the parameterization more meaningful.

The Fourier representation has a few advantages that have motivated us in using it for the work presented here. It is a good representation in the sense that closed curves are naturally periodic in arclength and a Fourier parameterization captures this inherently. It can be made invariant to the starting point, scale, and 2D translation and rotation [116]. Also, it has the capability to incorporate prior information as we shall soon see. Being concise and orthogonal, it makes the optimization easier through redundancy removal. For most biomedical applications that we are interested in, we only need a few parameters simplifying the optimization process even further. Finally, the conversion between the parameterization and the actual coordinates can be easily and directly obtained via the fast Fourier transform.

The standard Fourier representation of [116] is based on (3.1) and (3.2) using sinusoids as the basis functions. Hence,

$$\phi = \left\{ \frac{1}{2\pi}, \frac{cos\, x}{\pi}, \frac{sin\, x}{\pi}, \frac{cos\, 2x}{\pi}, \frac{sin\, 2x}{\pi}, ... \right\} \tag{3.3}$$

These can be used along with the direct representation to parameterize closed curves. Many biological objects like cells, organelles, parts of the brain (like the corpus callosum) and the heart (like the epicardium and the endocardium) which often interest us and form the main application areas for this work can be delineated using a closed boundary. A closed curve can be represented by two periodic functions of t, where $t \in [0, 2\pi]$, $x(t)$ and $y(t)$, one for each of the coordinates. The Fourier decomposition in the matrix form gives the elliptic Fourier representation [51, 69].

$$\begin{aligned} v(t) &= \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \\ &= \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} cos(kt) \\ sin(kt) \end{bmatrix} \end{aligned} \tag{3.4}$$

where, $v(t)$ is the contour vector consisting of the x and y coordinates and a_k, b_k, c_k and d_k are the corresponding Fourier coefficients given by:

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} x(t)dt \tag{3.5}$$

$$c_0 = \frac{1}{2\pi} \int_0^{2\pi} y(t)dt \tag{3.6}$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} x(t)cos(kt)dt \tag{3.7}$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} x(t)sin(kt)dt \tag{3.8}$$

$$c_k = \frac{1}{\pi} \int_0^{2\pi} y(t)cos(kt)dt \tag{3.9}$$

$$d_k = \frac{1}{\pi} \int_0^{2\pi} y(t)sin(kt)dt \tag{3.10}$$

The closed curve is thus represented by

$$\vec{p} = (a_0, c_0, a_1, b_1, c_1, d_1, ...) \tag{3.11}$$

This particular version of Fourier boundary representation has the advantages already mentioned before. Also, a geometric interpretation in terms of ellipses is provided in [116] that can help better understand the effects of the individual parameters.

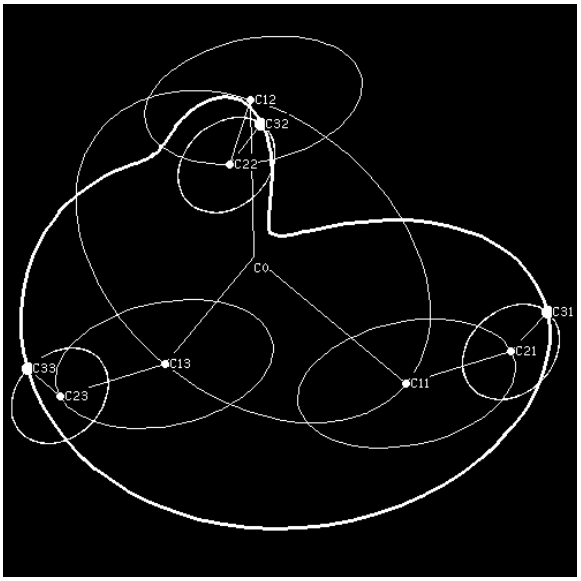


Figure 3.1: Contour (dark line) is constructed from three component ellipses shown at three different times (taken from [116])

In (3.4) the first two coefficients a_0 and c_0 determine the overall translation of the shape. Each term in the above summation is the parametric form for an ellipse. If $a_k d_k - b_k c_k = 0$, the ellipse degenerates to a straight line. The matrix determines the characteristics of the ellipse. The boundary can be viewed as a composition of rotating phasors, each defining an ellipse, the speed of rotational being proportional to their harmonic number k. This can be seen in Figure 3.1, where a contour is shown to be constructed from three component ellipses forming a sort of a planetary system. The straight lines represent the phasors for each ellipse at three different time instants. Thus, the point $C_{i,j}$ traces out the ith ellipse at the jth time instant. Each point is the center of the next higher ellipse C_0 is the center of the first ellipse. Points C_{31}, C_{32} and C_{33} are three different points on the final curve.

It is desired that the curves be continuous. Discontinuities slow down the speed of convergence due to the presence of the high frequencies. Also, since the contour is closed, the functions are periodic, with a period of 2π.

Number of Harmonics

In a practical situation, it is necessary to work only with a limited number of harmonics. This smoothes the curve but decreases the accuracy of the representation. Since, now only smooth representations are possible, high frequency variations are automatically filtered out when we use this representation to estimate the boundary of an object from a noisy image. This is analogous to regularization in ill-posed problems. The difference is that, unlike in regularization, where a functional is devised that adds smoothing constraints, here the solution space is restricted to accommodate only smooth functions depending of course on the number of harmonics used. Smoothing, by reconstructing a truncated Fourier elliptic representation is in general a good method because it avoids the shrinkage problem that is accompanied with normal smoothing. The choice of the number of harmonics is a design trade-off that is made between accuracy, conciseness and the degree of smoothing. Most of the biological forms that we are interested in are relatively smooth and unconvoluted. Hence, they can be represented by a limited number of harmonics. However, one might choose to have a larger number of harmonics for more complicated structures at higher computational cost. A good selection of the number of harmonics necessary, can be made by calculating the modeling error for the types of objects under investigation and choosing the least number (of harmonics) that produces an error smaller than a pre-determined limit.

The Parameter Probability Distributions

In order to use the prior information, probability distributions are associated with the parameters. This prior information can then bias the boundary finder to search for a particular range of shapes. When a sample is available, using experience, the prior knowledge can be obtained. However, an initial estimate of the boundary will still be necessary. The images in a sample will differ due to the variability in the object shape and the view. The importance of the prior is that it considerably reduces the search space by enforcing a very reasonable assumption which states that instances of the same object found under different circumstances should have similar shapes. For example, it is reasonable to assume that neuroanatomical structures such as the corpus callosum for different patients (as seen in MRI) should have similar shapes. This knowledge can be built in as a prior with a distribution around a mean shape. This prior distribution can be obtained by decomposing the boundaries of the sample objects in terms of their parameters and then taking statistics. In order to do so, the boundaries of the sample objects can be obtained either by manual segmentation or by running this method without using prior information.

[116] models the prior by using a multivariate Gaussian distribution with known

means and variances for the N parameters. Hence,

$$P(\vec{p}) = \prod_{i=1}^{N} P(p_i) = \prod_{i=1}^{N} \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(p_i - m_i)^2}{2\sigma_i^2}} \tag{3.12}$$

Here, m_i and σ_i are the mean and variance respectively of p_i, the i^{th} component of the parameter vector, $(\vec{p})$.

3.2.2 Boundary Estimation

The estimation process involves finding optimum values of the parameters that describe the boundary given the image data.

This is based on the expectation that the target object can be differentiated from the background by some measure of the boundary strength and direction (if available), computed from the image. This section describes the objective function that is being optimized based on a maximum *a posteriori* (MAP) formulation after [115].

A maximum *a posteriori* formulation is used since this method is interested in using prior shape knowledge while estimating the boundaries from the true image data. Let $I_b(x, y)$ be the image which depicts a set of objects and $t_{\vec{p}}(x, y)$ be an image template that corresponds to the parameter vector $\vec{p}$, which needs to be estimated. The goal is to detect the object boundary, which in turn is given by the most probable such boundary given the prior shape knowledge and the image information. This is done by maximizing $P(t_{\vec{p}}|I_b)$, the probability of the template given the image, where the maximization is done over $\vec{p}$. Using Baye's rule, this is equivalent to:

$$\arg\max_{\vec{p}} P(t_{\vec{p}}|I_b) = \arg\max_{\vec{p}} \frac{P(I_b|t_{\vec{p}})P(t_{\vec{p}})}{p(I_b)} \tag{3.13}$$

where in the above MAP formulation, $P(t_{\vec{p}})$ is the prior probability of the template and $P(I_b|t_{\vec{p}})$ is the likelihood that the template conforms to the cues available from the image. Ignoring the denominator of the RHS (which is not a function of $\vec{p}$, with respect to which the optimization is carried out), and taking the logarithm of the above gives,

$$\arg\max_{\vec{p}} M(I_b, t_{\vec{p}}) = \arg\max_{\vec{p}}[lnP(t_{\vec{p}}) + lnP(I_b|t_{\vec{p}})] \tag{3.14}$$

In the above, $M(.)$ is the general form of the objective function that is being optimized. The first term in the RHS is obtained from the prior information, and can be computed from (3.12). The other term is the data-driven likelihood term and is discussed below.

A Gaussian noise model assumption is used. Consequently, the image I_b is assumed to consist of one of these templates $t_{\vec{p}}$ corrupted by additive white Gaussian noise (AWGN), i.e. $I_b = t_{\vec{p}} + n$. Hence,

$$P(I_b|t_{\vec{p}}) \equiv P(I_b = t_{\vec{p}} + n) = P(n = I_b - t_{\vec{p}}) \tag{3.15}$$

and the noise at each pixel $n(x, y)$ is given by:

$$n(x, y) = I_b(x, y) - t_{\vec{p}}(x, y) \tag{3.16}$$

Due to the white noise assumption (which implies independence between the pixels), the joint probability over the entire area A is identical to the product of their individual probabilities. The conditional probability of obtaining I_b given the underlying structure is that of the template is then given by

$$P(I_b|t_{\vec{p}}) = \prod_A P(n(x,y)) \tag{3.17}$$

If the mean of the Gaussian is zero and the standard deviation is σ_n, then

$$P(I_b|t_{\vec{p}}) = \prod_A \frac{1}{\sigma_n\sqrt{2\pi}} e^{-\frac{(I_b(x,y)-t_{\vec{p}}(x,y))^2}{2\sigma_n^2}} \tag{3.18}$$

By taking the logarithm and substituting the result in (3.14), the expanded objective function becomes

$$M(I_b, t_{\vec{p}}) = lnP(t_{\vec{p}}) + \sum_A ln\frac{1}{\sigma_n\sqrt{2\pi}} - \sum_A \frac{(I_b(x,y) - t_{\vec{p}}(x,y))^2}{2\sigma_n^2} \tag{3.19}$$

This represents the MAP function for the images under the AWGN assumption. The first term is the prior term, the second one is a constant, and the last one represents the data likelihood term. It is easy to see that as long as $\sum_A t_{\vec{p}}^2(x,y)$ does not vary too much, the last term in (3.19) is similar to a correlation term.

As already indicated before, the boundary of the target object in this method [116] is represented by the template $t_{\vec{p}}(x,y)$. The templates are assumed to form a continuum, each having a corresponding value of the parameter vector $\vec{p}$. Essentially, the boundary is 1D, but it is embedded in a 2D image by assuming that $t_{\vec{p}}(x,y)$ is constant along the contour and is zero everywhere else. Thus, $I_b(x,y)$ is considered to be a boundary measure applied to the raw image data i.e. $I_b(x,y) = b(I(x,y))$. Generally, the boundary measure is given by the gradient magnitude or, by both the gradient magnitude and the direction of the original image. Only points that lie on the contour are considered because those are the only points at which the template is non-zero. Now, (3.19) becomes:

$$\begin{aligned} M(I_b, t_{\vec{p}}) &= lnP(t_{\vec{p}}) + \sum_A ln\frac{1}{\sigma_n\sqrt{2\pi}} \\ &- \frac{1}{2\sigma_n^2}\left(\sum_A I_b^2(x,y) + \sum_{C_{\vec{p}}}\left(-2I_b(x,y)t_{\vec{p}}(x,y) + t_{\vec{p}}^2(x,y)\right)\right) \end{aligned} \tag{3.20}$$

where $C_{\vec{p}}$ is the curve defined by the boundary $(x(\vec{p}), y(\vec{p}))$ in the template $t_{\vec{p}}$. Since $t_{\vec{p}}(x,y)$ is constant over the curve that it defines,

$$\begin{aligned} M(I_b, t_{\vec{p}}) &= lnP(t_{\vec{p}}) + \sum_A \left(ln\frac{1}{\sigma_n\sqrt{2\pi}} - \frac{I_b^2(x,y)}{2\sigma_n^2}\right) \\ &+ \frac{1}{2\sigma_n^2}\sum_{C_{\vec{p}}}\left(2I_b(x,y)t_{\vec{p}}(x,y)k - k^2\right) \end{aligned} \tag{3.21}$$

where k is the template magnitude at any point which is assumed to be a constant and is chosen to be the maximum boundary response. It can be observed that the first summation term in the above doesn't change with respect to variations in $\vec{p}$. Further, the last term that involves k^2, is proportional to the length of the contour which is not assumed to change appreciably when compared to changes in the boundary measure. In view of that, we ignore that term, and redefine (after simplification) $M()$ as:

$$M(I_b, t_{\vec{p}}) = lnP(t_{\vec{p}}) + \frac{1}{\sigma_n^2}\sum_{C_{\vec{p}}} I_b(x,y)t_{\vec{p}}(x,y)k \tag{3.22}$$

where the first term in the RHS is once again the prior, the second one being the likelihood term. As a note, it would be appropriate to mention here that the above equations can also be written in the continuous form (simply by replacing the summation by an integral) which for (3.22) becomes (see [116] for the details)

$$M(I_b, t_{\vec{p}}) = lnP(t_{\vec{p}}) + \frac{k}{\sigma_n^2} \int_{C_{\vec{p}}} I_b(x, y) ds \tag{3.23}$$

It is important to mention here that this white noise assumption is valid only if the template or model has a shape that fairly closely resembles that of the true object. However, when that is not the case, i.e. the boundary is not close to the object boundary, the white noise assumption may not hold, but, the resultant matching term given by (3.23) can still be used to guide the contour to the true object. This interpretation is consistent with the fact that it is necessary to initialize the contour close to the true object, or otherwise it is likely to fail. Use of a region-based term (see next chapter) makes the process more robust to this initialization.

3.3 Region-Based Segmentation

The goal of region-based image segmentation is to partition the image into contiguous regions, based on some similarity criteria (see the previous chapter for a formal definition). Segmentation in this case is performed by assigning each pixel one of the allowed classes (or region types) based on some local processing on the neighborhood of the pixel. As we have already pointed out in the background section, statistical techniques have recently become popular. The images are modeled as realizations of random fields and for segmentation statistically optimal estimation techniques, such as minimum mean squared error (MMSE), maximum likelihood (ML), and maximum *a posteriori* (MAP) estimation are used [12, 31, 41, 40, 47, 54]. Optimization is achieved by using dynamic programming [40, 54], stochastic relaxation/simulated annealing [47, 31], and deterministic relaxation

[12, 31].

Here, in this work we shall use the Markov Random Field (MRF) formalism to do our region based segmentation. Following, we first represent the MRF formalism and then discuss two related algorithms to do the segmentation.

3.3.1 Markov random field image model

MRFs are an important class of stochastic models and have been applied to problems like image estimation and texture segmentation. Once the model-variables have been chosen, the MRF is completely specified by a joint distribution over these variables. In this section, the MRF image model similar to the ones in [11, 12, 47, 73, 75] is presented.

Assume that all the images are defined on an $M \times M$ lattice $S = \{(i,j)\, 1 \leq i \leq M,\, 1 \leq j \leq M\}$. Let $x_{i,j}$ denote a random variable associated to the site $(i,j) \in S$. Thus the image

$$\mathbf{x} \,=\, \{x_{i,j}\, 1 \leq i \leq M,\, 1 \leq j \leq M\} \,\in R^{M\times M} \tag{3.24}$$

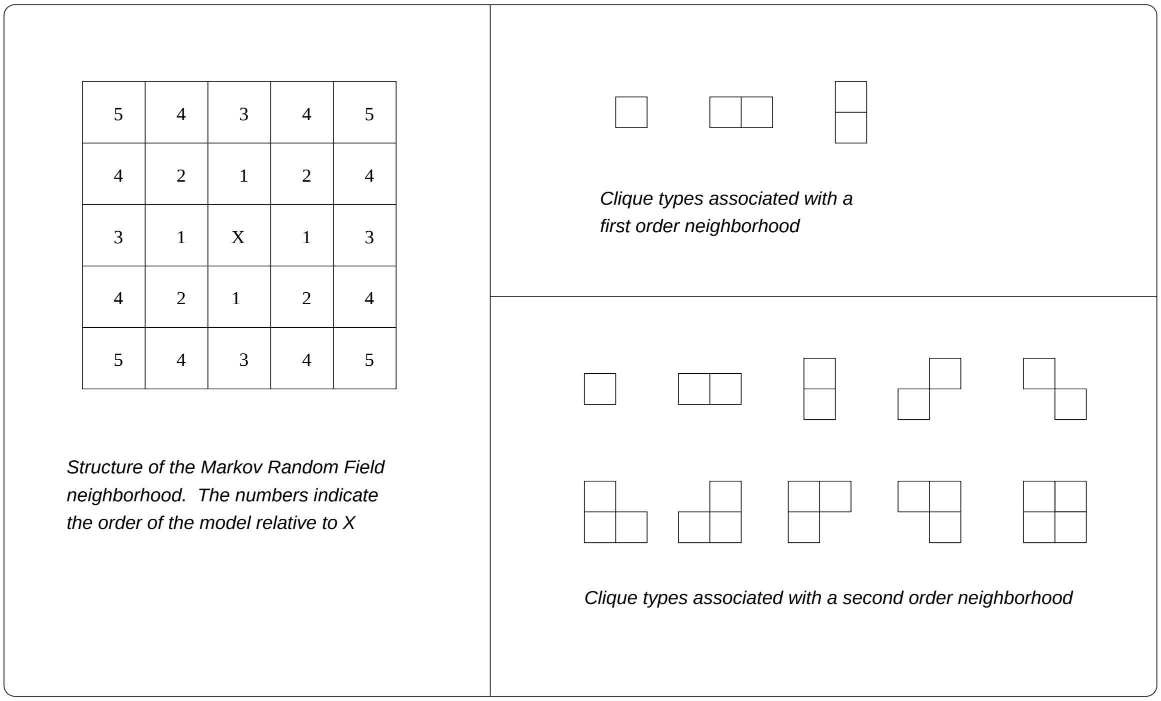
may be considered as a collection of $M \times M$ random variables forming a random field on the lattice S. By definition, the image $\mathbf{x}$ is a Markov random field if

$$P(x_{i,j}|\mathbf{x}_{S/(i,j)}) \,=\, P(x_{i,j}|\mathbf{x}_{N_{i,j}}) \tag{3.25}$$

where,

$$\mathbf{x}_{S/(i,j)} \,=\, \{x_{k,l}\; k = 1,2,..,i-1,i+1,..,M;\, l = 1,2,..,j-1,j+1,..,M\} \tag{3.26}$$

i.e. $S/(i,j)$ denotes the set of all sites excluding (i,j), and $N_{i,j} \subseteq S$ denotes a set of sites forming a neighborhood of site (i,j) and $\mathbf{x}_{S/(i,j)}$ and $\mathbf{x}_{N_{(i,j)}}$ denote respectively the random variables associated with these sites. A first order neighborhood consists of the four nearest pixel sites (Figure 3.2a), a second order neighborhood consist of the eight nearest neighborhood sites (Figure 3.2a), and so on. In order to be able to specify the joint probability distribution function, the concept of clique is necessary[47]. A clique is a set of one or more sites such that each site in the clique is a neighbor of all the other sites in the clique. The cliques for an isotropic first and second order MRFs are shown in Figures 3.2b and 3.2c respectively. MRF's on a 3D lattice can be defined in a similar



5	4	3	4	5
4	2	1	2	4
3	1	X	1	3
4	2	1	2	4
5	4	3	4	5

Structure of the Markov Random Field neighborhood. The numbers indicate the order of the model relative to X

Clique types associated with a first order neighborhood

Clique types associated with a second order neighborhood

Figure 3.2: (a) Left: Neighborhood structure of a Markov random field; (b) Right,top: Clique types associated with a first order neighborhood; (c) Right,bottom: Clique types associated with a second order neighborhood [111, 47].

way.

Since the idea is to do probabilistic optimization, it is necessary to define the joint probability of $\mathbf{x}$. To do this without violating the assumption in equation (3.25), one needs to use the Hammersley-Clifford Theorem [11, 113] which states that a random field defined on a lattice and satisfying the condition that if $P(x_i) > 0 \ \forall i$, then $P(x_1, x_2, ...) > 0$, is a MRF if and only if the joint distribution is a Gibbs function of the form,

$$P(\mathbf{x}) \ = \ \frac{1}{Z} exp(-U(\mathbf{x})/\beta) \tag{3.27}$$

where Z and β are constants and $U(.)$, called the energy function is of the form

$$U(\mathbf{x}) \ = \ \sum_{c \in C} V_c(\mathbf{x}) \tag{3.28}$$

where $V_c(\mathbf{x})$ are a set of potential functions depending only on the values of $\mathbf{x}$ at the sites in the clique c, C denotes the set of all cliques, and Z, the normalizing constant

$$Z = \sum_{\mathbf{x}} exp(-U(\mathbf{x})/\beta) \tag{3.29}$$

is also called the partition function. Finally, β stands for "temperature" and it controls the degree of peaking in the probability density, i.e. larger the value, larger is the peaking. Thus, if this model is used as an image prior, a large value of β would mean that the prior is very strong, resulting in a minimal emphasis on the data.

The power of this result is that the potential functions could be arbitrary as long as the RHS in equation (3.28) is summable (integrable) over all allowed values of $\mathbf{x}$. The potential functions are chosen to reflect the desired properties of the image so that the more likely images have a lower energy and are thus more probable. Also, the Gibbs energy formalism has the added advantage that if the likelihood term is given by an exponential, and the prior is obtained through a MRF model, the posterior probability continues to be a Gibbsian. This makes the MAP problem equivalent to an energy minimization.

In the following sections we will discuss two popular algorithms that rely on the MRF formalism to do image segmentations and later point out in subsequent chapters how we tend to use them.

3.3.2 Image segmentation through image estimation

The problem can be succinctly stated as follows: *it is necessary to obtain an estimate of a noisy image which at the same time yields a convenient representation in terms of piecewise homogeneous regions.* Adopting the MRF viewpoint, the degradation is assumed to be corruption by additive noise and the prior belief is that the image is composed of piecewise homogeneous regions. Thus once we can remove the noise through the process of estimation, we are left with contiguous homogeneous regions. This leads to what is called the *weak membrane* model [91, 14, 46, 90].

3.3.2.1 Weak membrane model

Here, it is assumed that the image can be modeled by

$$y(i,j) = x(i,j) + n(i,j); \quad 1 \leq i \leq M; \quad 1 \leq j \leq M \tag{3.30}$$

where $n(i,j)$ corresponds to additive white Gaussian noise (AWGN). Now, as for $x(i,j)$, a further assumption that it can be modeled as a collection of homogeneous regions of uniform or slowly varying intensities is made. Note that this is the same assumption that was made at the beginning of the present section which was simply restated here for continuity. At the edges one assumes that the variation can be approximated by a Gaussian step edge profile as in [59]. In this method, besides the intensity $\mathbf{x}$, there are also the binary- valued vertical $\mathbf{v}$ and the horizontal $\mathbf{h}$ line processes which locate the discontinuities in the intensity process which are to be estimated.

For recovering piecewise smooth functions as hypothesized in this problem, a method similar to the weak membrane model [91, 14] is used. Hence, maximizing the probability $P(\mathbf{x}|Y)$ (where $\mathbf{x}$ is the segmented image and Y is the image data) becomes equivalent to minimizing the energy function (to estimate $\mathbf{x}$,

$$\begin{aligned} U(\mathbf{x},\mathbf{v},\mathbf{h}) &= \sum_{i,j}[y_{i,j} - x_{i,j}]^2 \\ &+ \lambda^2 \sum_{i,j} \Big[(x_{i,j} - x_{i-1,j})^2(1 - v(i,j)) + (x_{i,j} - x_{i,j+1})^2(1 - h(i,j))\Big] \\ &+ \alpha \sum_{i,j}[v(i,j) + h(i,j)] \end{aligned} \tag{3.31}$$

The first term in the above equation is the data fidelity term, a measure of how closely the estimated process fits the observed data. The next term is the smoothness term and measures how much the function is deformed and the last term is the penalty term that is a measure of the discontinuities in the intensity function. If discontinuities were absent, then the reconstruction would be like a membrane that is continuous everywhere giving it the name, *weak membrane.*

Actually the above can be simplified through the elimination of the line variables giving the following,

$$\begin{aligned} U(\mathbf{x}) &= \sum_{i,j} [y_{i,j} - x_{i,j}]^2 \\ &+ \sum_{i,j} g^*((x_{i,j} - x_{i-1,j})^2) + \sum_{i,j} g^*((x_{i,j} - x_{i,j-1})^2) \end{aligned} \tag{3.32}$$

where,

$$g^*(f_p) = \begin{cases} \lambda^2 f_p^2 & \lambda^2 f_p^2 < \alpha \\ \alpha & \lambda^2 f_p^2 \geq \alpha \end{cases} \tag{3.33}$$

Here, f_p is used as a generic symbol for the intensity gradient. Thus, $f_x(i,j) = (x_{i,j} - x_{i-1,j})$ and $f_y(i,j) = (x_{i,j} - x_{i,j-1})$.

3.3.2.2 Optimization

The energy function as represented in equation (3.32) is non-convex due to the nature of the $g^*(.)$ function. What makes the problem difficult is that discontinuities are incorporated into the reconstructed functions. Several researchers [14, 76] have proposed a variety of continuation methods which essentially are convex formulations to the non-convex functionals at hand. Following, we shall first discuss the optimization for problems that do not have discontinuities and then indicate how the same can be modified to handle discontinuities.

If it is not necessary to solve for the discontinuities then a gradient descent based algorithm will do the job. That is equivalent to switching the line variables off resulting in

$$\begin{aligned} U(\mathbf{x}) &= \sum_{i,j}[y_{i,j} - x_{i,j}]^2 \\ &+ \lambda^2 \sum_{i,j} \left[(x_{i,j} - x_{i-1,j})^2 + (x_{i,j} - x_{i,j+1})^2 \right] \end{aligned} \tag{3.34}$$

If for some reason it is known that there are discontinuities along some contour, the energy function is modified by clamping the line variables "on" at those particular locations.

When all the line variables are *fixed* (either on or off as the case may be), this energy minimization becomes a classical problem of solving for zero gradient i.e. the desired minimum, $\mathbf{x}$ is obtained from

$$\frac{\partial U}{\partial x_{i,j}} = 0 \quad \forall (i,j) \tag{3.35}$$

Since the energy is quadratic in $\mathbf{x}$, this results in a linear system of equations. Further, $U(.)$ is strictly convex guaranteeing an unique solution.

There are many ways of minimizing such a function, most of which involve changing the variables to decrease the energy until no further change decreases the energy. A good way to reduce the search space is to use the gradient $\frac{\partial U}{\partial x_{i,j}}$ as a guide to how $x_{i,j}$ should be changed in order to reduce the energy fastest. Terzopoulos [122] used the *Gauss-Seidel* algorithm which is a special case of the Successive Over-relaxation (SOR) algorithm (see [14] for details).

When the line variables are not turned off, one can still use the above method using the $g^*()$ functions defined before in (3.32). However, as has already been pointed out, in that case the above in itself will not achieve good optimization results. Thus the graduated non-convexity (GNC) algorithm was proposed which optimizes a sequence of functions derived from the original objective function. There is a control parameter which at it's initial setting produces a very smooth convex approximation to the original energy function. This is first optimized and the result from this is passed on as an initial estimate of the next objective function of the sequence, which is obtained by a different parameter setting that makes it closer to the original function. Then this function is optimized and the result is passed on to the next function in the sequence and so on. This process is continued until the control parameter setting is such that the derived objective function actually is identical to the actual objective function. While this method will not guarantee a global optimization, it however leads to good sub-optimal results satisfactory for most applications.

Besides the graduated non-convex method other methods have been tried to optimize non-convex optimization problems. The first of these is simulated annealing [47]. Here, instead of direct descent, some randomness is introduced in the descent path. This avoids sticking to the local minima. The degree of the randomness is controlled by a temperature parameter, which is initially high. As the system approaches the global minima, the system is allowed to cool down. For a detailed description, the reader is referred to [47, 88]. The second one is Hopfield'd neural network construction [57] which is a compromise between the true energy function and some convex approximation to the same. That procedure is actually similar to the GNC algorithm [14] described before.

3.3.3 Image segmentation through region labeling

In this method, the objective is to classify the pixels in the image as belonging to one of a variety of classes. This method is not just limited to the segmentation (classification) of intensity images, but also can be used for segmenting texture images. It is important to be able to handle texture images because a large group of naturally occurring images have information about underlying objects that can be distinguished only by analyzing the texture properties and not just the intensity distributions. In the medical imaging domain, ultrasound images are instances of such images.

The main difficulty in classifying images is that the segmentation and the parameters necessary to do so, both are unknown. To compute the parameters effectively, the segmented image is needed and to compute the segmented image it is necessary to have a good estimation of the parameters. One way to solve this is to do simultaneously both parameter estimation and image classification. But that often turns out to be computationally too expensive. To get around this problem it has been suggested that first parameter estimation be carried out on known parts of the image and then use those parameters to do the classification. As has been shown in [83] there is almost no noticeable difference in performance by breaking up the process and doing them sequentially rather than doing them simultaneously as performed in [73].

Before describing the mathematical details of the model, it is important to remark that there are two random fields involved here. One is the raw image and the other is a label set that defines the classified image. Below, we describe the mathematical formulation of the method that we use, part of which has been borrowed form [83, 111] and the rest from [75].

3.3.3.1 Image model

The intensity image is modeled as a Gaussian Markov random field (GMRF). This model has been used by many researchers to model texture and other image variations [28]. It is used here to model the conditional probability density of the image intensity given the classification.

As already defined, S denotes the $M \times M$ lattice, i.e. $S = \{(i,j),\, 1 \leq i,j \leq M\}$. Let $\{L_s, s \in S\}$ and $\{Y_s, s \in S\}$ denote the labels and the zero mean array obtained from the image data respectively. Let N_s denote the neighborhood of a site s. The neighborhood scheme is as shown in Figure (3.2). Thus for the second order neighborhood there are eight neighbors. Now, if it is further assumed that all the nearest neighbors of s also have the same label as s, one can write the following expression for the conditional density of the intensity at the pixel site s [83, 111]:

$$P(Y_s = y_s | Y_r = y_r, r \in N_s, L_s = l) \quad = \quad \frac{exp(-U(Y_s = y_s | Y_r = y_r, r \in N_s, L_s = l))}{Z(l | y_r, r \in N_s)} \tag{3.36}$$

where $Z(l|y_r, r \in N_s)$ is the partition function of the conditional Gibbs distribution, and

$$U(Y_s = y_s | Y_r = y_r, r \in N_s, L_s = l) = \frac{1}{2\sigma_l^2} \left(y_s^2 - 2 \sum_{r \in N_s} \Theta_{s,r}^l y_s y_r \right) \tag{3.37}$$

In (3.37), σ_l and Θ^l are the GMRF model parameters of the l^{th} region class. Also, the model parameters satisfy:

$$\Theta_{s,r}^l = \Theta_{s-r}^l = \Theta_{r-s}^l = \Theta_r^l$$

Further, the joint probability in a window W_s, centered at s can be written as:

$$P(Y_s^* | L_s = l) = \frac{-U_1(y_s^* | L_s = l)}{Z_1(l)} \tag{3.38}$$

where $Z_1(l)$ is the partition function for that window, and

$$U_1(y_s^*|L_s = l) = \frac{1}{2\sigma_l^2} \sum_{r \in W_s} \left\{ \sum_{r \in N^* | r+\tau \in W_s} \Theta_r^l y_r (y_{r+\tau} + y_{r-\tau}) \right\} \tag{3.39}$$

y_s^* represents the intensity array in the window W_s. The above assumes a free boundary model. For a more complex boundary model, one can look at [61]. N^* is a set of shift vectors corresponding to the second order GMRF model:

$$N^* = \{\tau_1, \tau_2, \tau_3, \tau_4\} = \{(0,1), (1,0), (1,1), (-1,1)\}$$

3.3.3.2 Parameter estimation

In the last section, the model that was used to characterize the intensity values of the image was described. However, before it can be used to do the classification, it is necessary to estimate the parameters of the model. For every region class, there are a set of parameters that identify that class. Either one can do a rough segmentation of the image and use that as a mask on which to base the estimation of the parameters, or as was done here, have the user identify points representative of a particular class, and then base the estimation on a window around that point.

The parameter estimation is done in a similar way as has been done in [111] and for completeness is described below.

There are many methods to estimate the parameters of a GMRF, but none of them can guarantee consistency (estimates converging to the true value of the parameters) and stability (the covariance matrix in the expression for the joint probability density of the MRF must be positive definite) together. Normally an optimization algorithm is used to obtain stable estimates. The GMRF parameters are used for obtaining certain measures for segmentation and not for doing the region/texture synthesis. Hence, it makes sense to use computationally less expensive methods that can provide reasonable estimates of these parameters for the segmentation process to work on, even if they do not necessarily guarantee the stability of these estimates. A least squares estimate [61] was used as a tradeoff between simplicity and stability.

Consider a region of size $K \times K$ consisting of a single region/texture. (For our implementation, this could mean providing a seed point around which a $K \times K$ window is chosen.) Let Ω be the lattice under consideration, and let Ω_I be the interior region of Ω, i.e. if Ω_B is the boundary,

$$\begin{aligned} \Omega_I &= \Omega - \Omega_B, \\ \Omega_B &= \{s = (i,j), \\ &\quad s \in \Omega \ and \ s \pm \tau \notin \Omega \ for \ at \ least \ some \ \tau \in N^*\} \end{aligned} \tag{3.40}$$

Let,

$$Q_s = [y_{s+\tau_1} + y_{s-\tau_1}, ..., y_{s+\tau_4} + y_{s-\tau_4}]^T \tag{3.41}$$

Then the least squares estimate of the parameters are:

$$\hat{\Theta} = \left[\sum_{\Omega_I} Q_s Q_s^T\right]^{-1} \left[\sum_{\Omega_I} Q_s y_s\right] \tag{3.42}$$

$$\hat{\sigma}^2 = \frac{1}{N^2} \sum_{\Omega_I} \left[y_s - \hat{\Theta}^T Q_s\right]^2 \tag{3.43}$$

If $\hat{\mu}$ is the mean in the sub-image under consideration, then the feature vector for the region is denoted by:

$$F = (\theta_1, \theta_2, \theta_3, \theta_4, \hat{\mu}, \hat{\sigma}^2) \tag{3.44}$$

3.3.3.3 Optimization

The label field L_s is modeled as a first or second order MRF. It is not involved directly in the parameter estimation described before. If $\hat{N}_s$ denotes the appropriate neighborhood for the label field, then the distribution function for the region/texture label at site s conditioned on the labels on the neighboring sites can be written as [111]:

$$P(L_s|L_r, r \in \hat{N}_s) = \frac{exp(-U_2(L_s|L_r))}{Z_2} \tag{3.45}$$

where Z_2 is a normalizing constant and

$$U_2(L_s|L_r, r \in \hat{N}_s) - \beta \sum_{r \in \hat{N}_s} \delta(L_s - L_r), \;\; \beta > 0 \tag{3.46}$$

where β denotes the degree of clustering and $\delta(i - j)$ is the Kronecker delta. Now, using the Bayes rule one can get,

$$P(L_s|Y_s^*, L_r, r \in \hat{N}_s) = \frac{P(Y_s^*|L_s)P(L_s|L_r, r \in \hat{N}_s)}{P(Y_s^*)} \tag{3.47}$$

Since, Y_s^* the data in the observation window is known, the denominator in (3.47) is just a constant. The numerator is a product of two exponential functions and can be expressed as [111]

$$P(L_s|Y_s^*, L_r, r \in \hat{N}_s) = \frac{1}{Z_p} exp(-U_p(L_s|Y_s^*, L_r, r \in \hat{N}_s) \tag{3.48}$$

where Z_p is the partition function and $U_p(.)$ is the posterior energy corresponding to (3.47). From equations (3.39) and (3.46) it is possible to write,

$$U_p(L_s|Y_s^*, L_r, r \in \hat{N}_s) = w(L_s) + U_1(Y_s^*|L_s) + U_2(L_s|L_r, r \in \hat{N}_s) \tag{3.49}$$

The second term in (3.49) relates the observed pixel intensities to the region/texture labels and the last term specifies the label distribution. The bias term $w(L_s) = logZ_1(L_s)$ is dependent on the region/texture class and it can be evaluated explicitly for the GMRF model considered here [61]. Finally, the posterior distribution of the texture labels for the entire image given the intensity array is:

$$P(L|Y^*) = \frac{P(Y^*|L)P(L)}{P(Y^*)} \tag{3.50}$$

Maximizing (3.50) gives the optimal Bayesian estimate. Stochastic relaxation techniques [47, 73] would yield an optimal solution at the cost of huge computational overhead. However, sub-optimal results obtained using methods like deterministic relaxation [111] or the ICM approach [75, 98] produces almost similar results without the computational burden that is often associated with global optimization procedures like simulated annealing or dynamic programming.

In our results, we have used the coordinate-wise descent, similar to the iterated conditional mode (ICM) algorithm [11, 12]. This is adapted from the method given in

[75].

As already stated, the aim is to maximize the *a posteriori* probability given in (3.50). But in the coordinate wise descent algorithm, one instead tries to find the local optimum. In this algorithm, one starts with an initial labeling, L^0 and then iterates sequentially (or in parallel) through *each pixel, replacing the current label at that pixel with the label that maximizes* $P(L_s|Y^*, L_{S/s})$ where $L_{S/s}$ represents the label set for the whole image except the site s, and the other symbols have the same connotations as before.

Thus, the aim is to maximize,

$$P(L_s|Y^*, L_{S/s}) = P(L_s|Y^*, L_r, r \in \hat{N}_s) \tag{3.51}$$

The above is obtained through the use of the Markov property. Now, using the Bayes theorem we get:

$$\begin{aligned} P(L_s|Y^*, L_{S/s}) &= \frac{P(Y^*, L_s, L_r, r \in \hat{N}_s)}{P(Y^*, L_r, r \in \hat{N}_s)} \\ &= \frac{P(Y^*|L_s, L_r, r \in \hat{N}_s)P(L_s, L_r, r \in \hat{N}_s)}{P(Y^*, L_r, r \in \hat{N}_s)} \\ &= \frac{P(Y^*|L_s, L_r, r \in \hat{N}_s)P(L_s|L_r, r \in \hat{N}_s)P(L_r, r \in \hat{N}_s)}{P(Y^*, L_r, r \in \hat{N}_s)} \\ &= \frac{P(Y^*|L_s, L_r, r \in \hat{N}_s)P(L_s|L_r, r \in \hat{N}_s)}{P(Y^*|L_r, r \in \hat{N}_s)} \\ &\propto P(Y^*|L_s, L_r, r \in \hat{N}_s)P(L_s|L_r, r \in \hat{N}_s) \end{aligned} \tag{3.52}$$

The last step is valid because the denominator is essentially a normalizing term. Thus, using Bayes theorem on the first term of the RHS in (3.52) gives:

$$
\begin{aligned}
P(Y^*|L_s, L_r, r \in \hat{N}_s) &= P(y_s^*, Y_{S/s}^*|L_s, L_r, r \in \hat{N}_s) \\
&= \frac{P(y_s^*, Y_{S/s}^*, L_s, L_r, r \in \hat{N}_s)}{P(L_s, L_r, r \in \hat{N}_s)} \\
&= \frac{P(y_s^*|Y_{S/s}^*, L_s, L_r, r \in \hat{N}_s)P(Y_{S/s}^*, L_s, L_r, r \in \hat{N}_s)}{P(L_s, L_r, r \in \hat{N}_s)} \\
&= P(y_s^*|Y_{S/s}^*, L_s, L_r, r \in \hat{N}_s)P(Y_{S/s}^*|L_r, r \in \hat{N}_s) \qquad (3.53)
\end{aligned}
$$

The last equation follows because the vector $Y_{S/s}^*$ does not include y_s^* and while optimizing over L_s second order effects are ignored. Consequently, the second term (in the multiplication) on the right hand side in (3.53) is independent of the label at site s.

Thus combining equations (3.52) and (3.53), the ICM algorithm proposed here, needs to compute

$$\max_{L_s} P(y_s^*|Y_{S/s}^*, L_s, L_r, r \in \hat{N}_s)P(L_s|L_r, r \in \hat{N}_s) \qquad (3.54)$$

for each of the sites s.

The functional in (3.54) is optimized at each step where the first term is the conditional probability (likelihood) term and is given by (3.37) and the second term is the prior term given by (3.46). To summarize the algorithm, at each iteration, each pixel is updated to maximize (3.54) using the data at site s and the neighborhood, and the current estimate of the labels of the neighborhood. This continues so long as the number of changes is above a certain fraction of the image size. The issue of the number of classes is left as an user choice, making it a supervised classification algorithm as opposed to a non-supervised classification algorithm where the algorithm has the ability to choose the number of classes. From a practical point of view, the number of classes is chosen by first assigning one class to the target region and then the rest of the image is divided into the least number of classes that is necessary to keep the separate identity of the target region.

3.4 Conclusions

In this chapter we discussed some of the popular segmentation methods in detail that we shall use in the following chapters. First, we discussed a shape-based deformable boundary finding method that we have adopted for our work. Then region-based segmentation methods that have widely been used in the literature and which we plan to use in our work were discussed. In the following chapters we will show how the information from these methods can be integrated resulting in superior techniques.

Chapter 4

Deformable Boundary Finding influenced by Region Information

4.1 Introduction

The objective of this chapter is to develop an integrated method so that region information can be used in addition to gradient information within the boundary finding framework.

As we have already mentioned in chapter 1 two popular approaches to image segmentation [5] are region-based segmentation and gradient-based boundary finding.

While the presence of noise is always a limiting factor for any image processing method, region-based methods are less affected by it than gradient-based boundary finding as the gradient is very noise sensitive. Also for an image, if the high frequency information is either missing or is unreliable, boundary finding is more error-prone compared to region-based segmentation. Shape variations, on the other hand, can be better handled using a deformable boundary finding framework when we consider them to vary around an average shape. Such information can easily be incorporated as priors [116].

Further, since conventional boundary finding relies on changes in the grey level, rather than their actual values, it is less sensitive to changes in the grey scale distributions such as spatially-varying shading artifacts than is region-based segmentation. Also, gradient-based methods in general do a better job of edge localization. Hence, we observe that both the methods have their limitations and advantages.

Integration can produce better results by removing some of these limitations through the combination of the strengths of these methods. As the results will show, the integrated approach performs remarkably better both against increasing noise and poor initialization.

4.2 Integration

In this section, we first discuss the motivation behind our approach and then describe it mathematically.

As an input to the problem, we have both the actual image I and the region classified image I_s, which is obtained from I after passing it through either one of the region-based segmentation steps discussed in the last chapter. This information is introduced as an added prior into the gradient-based boundary finding framework. In its simplest form, this region term forces the boundary to enclose a single region in I_s. As we shall later see, this assumption is not strictly necessary, but for the sake of simplicity, we will continue with it. Modifications to it will be discussed later. The traditional boundary finding problem does not use the original image directly. Being a gradient-based approach, it uses instead the gradient image I_g. As in the Staib and Duncan [116] approach, which we have explained in the previous chapter, we shall use the magnitude of the gradient vector at each pixel location. In the last chapter, we used a generic description I_b as the image derived from the original image on which the boundary finding procedure is based. For the following development, we have $I_b = I_g$. Gradient direction along with the magnitude can also be used (see Worring et. al [132]), but for the simplicity of analysis here, we shall stick to the previous approach [116] of using gradient magnitude only. We obtain I_g by first convolving the image I with the derivative of a Gaussian and then taking the magnitude of the corresponding vector image. Thus finally, the input to the system is the gradient image I_g and the region classified image I_s.

The above boundary estimation problem using gradient and region homogeneity information can be posed in the maximum *a posteriori* framework. The aim is to maximize $P(\vec{p}|I_g, I_s)$, where as described in the previous section, $\vec{p}$ is the vector of parameters used to parameterize the contour.

Now,

$$P(\vec{p}|I_g, I_s) = \frac{P(\vec{p}, I_g, I_s)}{P(I_g, I_s)} \tag{4.1}$$

$$= \frac{P(I_s|I_g, \vec{p})P(p, I_g)}{P(I_g, I_s)} \tag{4.2}$$

$$= \frac{P(I_s|I_g, p)P(I_g|\vec{p})P(p)}{P(I_g, I_s)} \tag{4.3}$$

Thus ignoring the denominator which does not change with '$\vec{p}$' it is necessary to determine,

$$\vec{p}^* = \arg\max_{\vec{p}} P(\vec{p}|I_g, I_s) \quad \propto \quad \arg\max_{\vec{p}} P(I_s|I_g, \vec{p})P(I_g|\vec{p})P(\vec{p}) \tag{4.4}$$

or,

$$\arg\max_{\vec{p}} P(\vec{p}|I_g, I_s) \quad \equiv \quad \arg\max_{\vec{p}} [lnP(\vec{p}) + lnP(I_g|\vec{p}) + lnP(I_s|I_g, \vec{p})] \tag{4.5}$$

In the last equation (4.5), we have just taken the natural logarithm, which is a monotonically increasing function. Knowledge of I_g may be used to calculate I_s, through the use of line processes [47, 14]. However, if we do not use that information, we are effectively discarding information rather than assuming extra information. Thus, finally, the above can be written in the following form:

$$\arg\max_{\vec{p}} M(\vec{p}, I_g, I_s) \quad = \quad M_{prior}(\vec{p}) + M_{gradient}(I_g, \vec{p}) + M_{region}(I_s, \vec{p}) \tag{4.6}$$

where as just mentioned, we have assumed that the calculation of I_s, assumes the knowledge of I_g. Exact probabilistic definitions of the above, if available can be used. However, that is non-trivial. Consequently, we use intuitively appealing analogous matching terms instead (i.e. we try to match the estimated boundary with the gradient and the region-segmented image), designed after the probabilistic terms.

4.2.1 Prior shape term

The first term in equation (4.6) corresponds to the prior shape term. This can be obtained from previous experience or from expert knowledge about the natural structures that exist in the image. When it is non-uniform, it biases the model towards a particular range of shapes. Use of prior boundary information can especially be important in clinical applications like the outlining of the corpus callosum or the endocardium of the heart, where the nature of the shape does not change a great deal from individual to individual even though the exact reconstruction is different. Prior boundary information under such circumstances can be obtained from previously outlined boundaries. These are first parameterized and then the mean and variance of these parameters are calculated to obtain a multivariate Gaussian prior (as discussed in the last chapter), which could then be used to constrain the optimization at a later stage for further instances of similar structures (see [115, 116] for more details). Since there might be other objects in the image, we would always need an initial estimate of the boundary to start the optimization process. The prior shape, if available, can be used for this purpose as well. The prior information could be of particular importance if at some point in an image the boundary is ill-defined. We will show one such example later.

4.2.2 Gradient term

The second term in equation (4.6) depends on the gradient image. It is a measure of the likelihood of the contour of the described object being the true boundary, once the parameters defining the boundary are given. This is expressed in terms of a contour integral where the integral is computed over $C_{\vec{p}}$, the curve described by the boundary $(x(\vec{p},t), y(\vec{p},t))$. At each point on the contour, the strength of the boundary is evaluated by the magnitude of the gradient at that particular point, given by the gradient image. Thus the likelihood of the whole contour being the true boundary becomes proportional to the sum of the magnitude of the gradients at all the points that lie on the boundary. Thus the match between the estimated contour and the gradient image is expressed as the following line integral (Staib and Duncan [116]):

$$M_{gradient}(I_g, \vec{p}) \propto \int_{C_{\vec{p}}} I_g(x(\vec{p},t), y(\vec{p},t))dt \tag{4.7}$$

We note that the first and the second term in equation (4.6) together form the right hand side of equation (3.23) that was developed in the last chapter with the assumption the $I_b = I_g$.

4.2.3 Region term

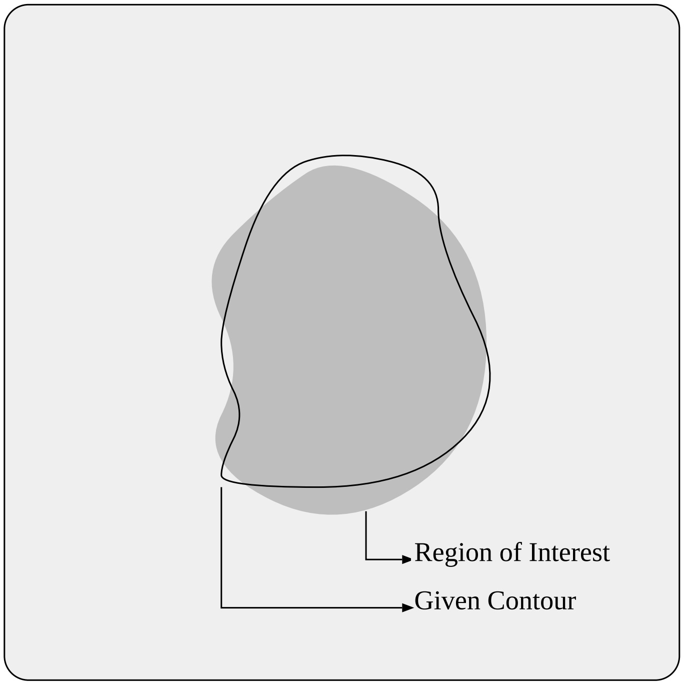


Figure 4.1: An example demonstrating the use of region information in the boundary finding process

The last term (of equation (4.6)) incorporates the region information into the boundary finding framework. Once again, we note here that the As we have already mentioned, we would like the boundary to enclose within it a homogeneous region. In other words, we expect the interior of the boundary to be filled with the region of a single class. This realizes the expectation that the variations within an object are assumed to be smaller than those between objects. We emphasize here that this similarity measure may be based on image intensity, texture properties or any other image attribute that can be handled by the region process. However, this method can also deal with situations like outlining an annulus, which has the shape of a torus, where the central part might belong to a separate class.

To describe this term mathematically, let us consider the cartoon situation shown in Figure 4.1 showing a hypothetical region segmented image and the boundary at any instance during the optimization process. For simplicity's sake let us assume there are only two regions and we want to segment out the object in the center from the surrounding background. As we can see, the boundary does not include the entire central region. At some points it includes the background and at other places it excludes points that belong to the central region. We would like to penalize cases where points from the surrounding regions are included and would like to reward if more and more of the central region is included. A very simple way of doing this would be to do the following. Let us consider for this simple image, all points that lie inside the central region to have value 1.0 and all points that lie outside a value -1.0. Now as the criteria function sums up the values of all the points that are inside the contour, it is clear that the sum achieves a maxima when the contour is so placed such that it includes all of the points with a value 1.0 and excludes all of the points that have a value of -1.0, that lie on the surrounding region. Even though this might appear to be a separate step, it can be incorporated into the region classification step.

If more than one region is involved, all pixels of the region that needs to be segmented out can be assigned a positive value and the remaining ones negative values, the magnitudes of which reflect how much one expects the target region to be dissociated from the remaining regions. Hence, remote regions are expected to have high negative values, representing larger penalty for including remote points. This way multiple regions can be handled. For an annulus, e.g. the myocardium in a transaxial heart image, it does not matter what value the points have in the region within the inner boundary the annulus circumscribes. Further, we can also relax the assumption that the interior of a region needs to have a single region. If we know that the target object consists of more than one region, than all those regions are assigned positive values and those that lie outside are assigned negative ones. Thus the only requirement is that the points immediately inside the boundary needs to be of a different type than those that are to be found immediately outside.

The above assignments work very well when the region-segmented image is obtained by region classification or a labeling procedure. Alternately, a region-segmented image might be obtained through image estimation (noise removal) along with the piecewise continuity assumption as has been described in the previous chapter. In that case, if we know the means of the different regions, then we could translate the image intensities in such a way that the target region has positive values and the other regions have negative values.

Thus, from the above discussion, we find that region can be expressed as:

$$M_{region}(I_s, \vec{p}) \propto \int\int_{A_{\vec{p}}} I_s(x, y) dA \tag{4.8}$$

4.2.4 Simplified Integration using Greens theorem

The objective function involving all the above three terms can be expressed as:

$$\arg\max_{\vec{p}} M(\vec{p}, I_g, I_s) = \arg\max_{\vec{p}} [M_{prior}(\vec{p}) + M_{gradient}(I_g, \vec{p}) + M_{region}(I_s, \vec{p})]$$

$$
\begin{aligned}
&\equiv \quad \arg\max_{\vec{p}} \left[M_{prior}(\vec{p}) + K_1 \int_{C_{\vec{p}}} I_g(x,y)dt \right. \\
&+ \quad \left. K_2 \iint_{A_{\vec{p}}} I_s(x,y)dA \right] \qquad (4.9)
\end{aligned}
$$

where K_1 and K_2 are the weighting constants which signifies the relative importance of the two terms in the above equation. Normally, one should choose them such that the contributions from each one of the terms are comparable. A better choice will be suggested in the following chapter.

Of the last two terms in (4.9), one is an area integral and the other is a line integral. In general, computing a line integral is much less expensive compared to an area integral. Thus we would save a lot of computation, especially when we carry out an iterative optimization procedure, if we could convert the area integral to a line integral which we have to compute anyway, as the second term which is present even in the original boundary finding method involves computing a line integral. Actually, the above can be obtained using Greens theorem [9] as follows.

Let,

$$
M_s(x,y) = \int_0^x I_s(z,y)dz \qquad (4.10)
$$

and,

$$
N_s(x,y) = -\int_0^y I_s(x,z)dz \qquad (4.11)
$$

Hence, using Greens Theorem,

$$\int\int_{A_{\vec{p}}} I_s(x,y)dA \quad = \quad \frac{1}{2}\int_{C_{\vec{p}}} [N_s(x,y)\,\frac{\partial x}{\partial t} + M_s(x,y)\,\frac{\partial y}{\partial t}]dt$$

Thus, finally we have,

$$\begin{aligned} \arg\max_{\vec{p}} M(\vec{p}, I_g, I_s) &= \arg\max_{\vec{p}} [M_{prior}(\vec{p}) \\ &+ \int_{C_{\vec{p}}} (K_1 I_g(x,y) \\ &+ K_2\{N_s(x,y)\,\frac{\partial x}{\partial t} + M_s(x,y)\,\frac{\partial y}{\partial t}\}\,)dt] \end{aligned} \tag{4.12}$$

Thus, in this section we have presented a boundary finding procedure that incorporates information that we might obtain from region-based segmentation. Further, using Greens theorem we reduce the whole problem to computing line integrals only rather than both line and area integrals. Since, M_s and N_s are evaluated **only once for every image, and is not repeated at every iteration** the computational speedup due to the use of the Green's theorem formalism is from $O(area)$ to $O(perimeter)$ where the area and the perimeter measures refer to those of the target object.

It is essential to point out here that in practice, the functions M_s and N_s are evaluated by simply adding up the pixel values of I_s in either the x or y direction. Thus, $M_s(i,j) = \sum_{k=0}^{i} I_s(k,j)$. We do the same thing along the other coordinate for $N_s()$. Hence, even though I_s is discontinuous, $M_s()$ and $N_s()$ are continuous in the x and y direction respectively. Further, the derivatives of these functions which are necessary for the gradient calculation yield I_s. Thus for most of the computations, these differentiations are not being carried out numerically due to the way the objective function is constructed.

The calculation of the gradient terms that need to be used for the optimization are done in Appendix A which justifies the above statement. We emphasize here that at no stage are we taking derivatives of I_s, which is discontinuous. We would also like to note here that even though I_s is discontinuous, it is still integrable (summable) in a discrete sense because it is defined for all the pixel locations. Optimization of the above expression (eqn. (4.12)) is achieved using the conjugate gradient method.

4.3 Results

Experiments were carried out both with synthetic and natural images to verify the performance of the above mentioned method. No prior information was used for the synthetic images. For the real images, only in Figure 4.11 prior information was used.

4.3.1 Synthetic Images

The output of all these experiments are object boundaries. We have set up experiments using synthetic images to quantitatively evaluate the method.

For our purposes we created a synthetic image that has one target object in the center surrounded by a background as shown in Figure 4.2(a). To be noticed is that the target object has both convex and concave parts to it. Further, it also has some high curvature points to make the boundary finding process non-trivial. To make it even more complicated we have smoothed the image using a Gaussian kernel so that the edges become fuzzy. On top of that white noise was added which would again affect the boundaries the most as they were smoothed out. Thus, in a way it represents many of the difficulties associated with structure segmentation in biomedical images. The advantage of using synthetic images is that we know exactly the actual boundary location. No prior information was used for these images. As for the region-based segmentation, it was carried out using only the mean and the variances of the two regions under consideration.

For evaluation we used the following procedure. First, the true boundary was evenly sampled into 256 points. The boundary finding process was initialized with a boundary that is spatially distant from the true boundary and depending upon the situation results in an output boundary. This was then evenly sampled into as many points. To find out how closely the output boundary approximated the true boundary, we need to calculate the distance between them. To solve the problem of pointwise correspondence, we keep the true boundary fixed, and vary the starting point of the other boundary point by point, calculating at each step the total distance as a sum of the distances between each corresponding points. Thus for example, in the first instance, point 1 of the first boundary is matched with point 1 of the second, point 2 with point 2 of the second and so on. Next the second boundary was shifted by one point so that point 1 of the first goes with point 2 of the second, point 2 matches with point 3 of the second and point 256 compares with point 1 of the second boundary. Thus we end up with 256 values of probable distances between the two boundaries. The minimum of these is considered to be the reported value of the distance.

The comparisons were done using three versions of the objective function. When only the gradient-based term in the objective function was used, we have the traditional gradient-based boundary finding. The second method only uses the region based term, where information only from the region classified image is used. Finally, the proposed method, uses both of the above terms in a combined way.

4.3.1.1 Experiment 1

Here the aim is to compare the performance of the three methods under varying amounts of noise. Each time, the noise in the same image was increased and then the three methods were tried upon it using the same initial boundary placement. The three methods were allowed to run roughly for the same number of iterations or as long as the change between successive iterations is above a certain predetermined threshold.

Figure 4.3 shows a comparison of the three methods. Here, the X-axis corresponds to the noise level given by the standard deviation of the noise used. The SNR varies from 4.0 to 1.0. The Y-axis gives a measure of the distance between the approximated contour and the true one. (This is a scaled version of the square of the distance between the contours calculated as described above.) Here in this and all the following plots, 'Gradient' represents the gradient-based method, 'Region' corresponds to the boundary found based only on the region classified image, and 'Combined' refers to the proposed integration method. As we might expect, the performance of all the three methods worsen as we increase the amount of noise. However, as can be seen, the combined method is the least sensitive to noise. Also, it performs much better compared to the traditional gradient-based method. Since the gradient-based method relies upon the first derivative, it is more susceptible to noise than the region-based method, which uses homogeneity within the image. On the other hand, the combined method seems to give uniformly better results.

In Figure 4.2 we show the result for a particular value of the noise given by $SNR = 2.0$. Here we can see the comparison between the different methods visually. The initial boundary is considerably displaced and disfigured as compared to the actual boundary. At some places, it is inside the central region and at some other places it is moved in

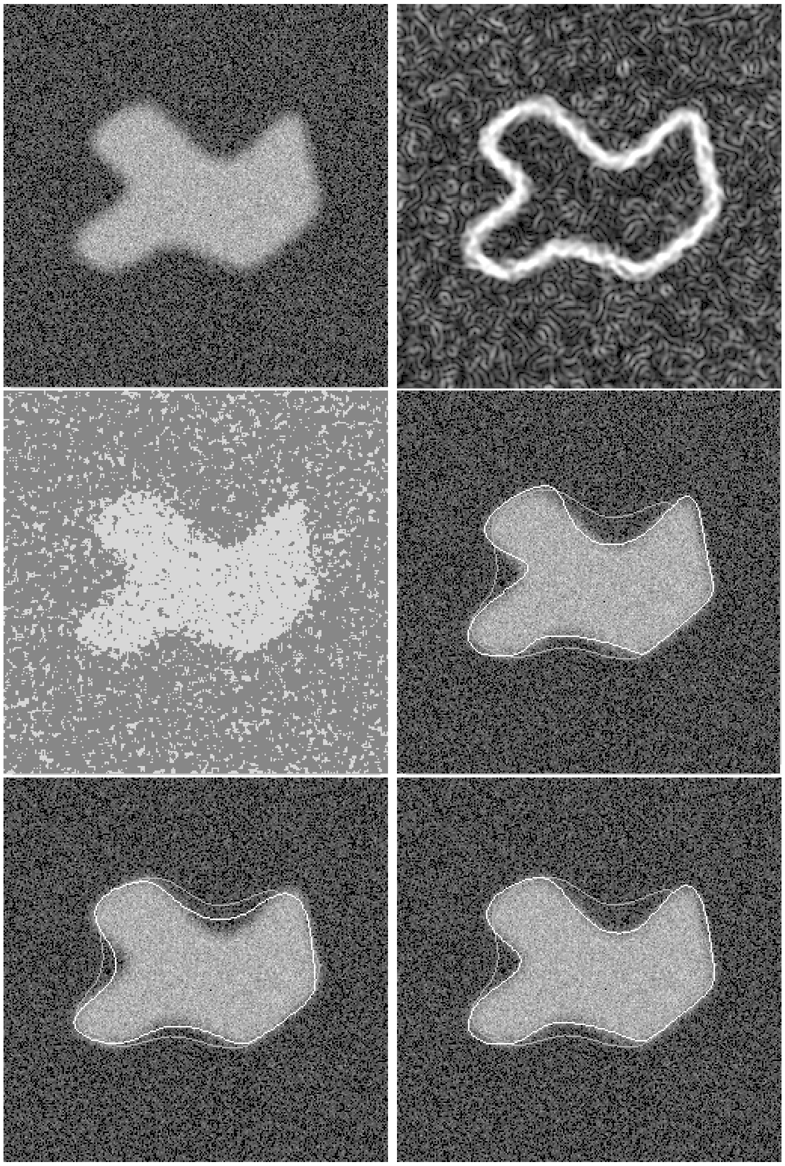


Figure 4.2: Performance of the three methods for a noisy image ($SNR = 2.0$). (a)Top,Left: Original image; (b)Top,Right: Gradient image; (c)Middle,Left: Region Grown image; (d)Middle,Right: Output of Boundary finding using gradient information only; (e)Bottom,Left: Output of Boundary finding using region information only; (f)Bottom,Right: Output of Boundary finding using an integrated approach. In the last three images, the more gray contour represents the initial boundary placing and the brighter one the final derived boundary.

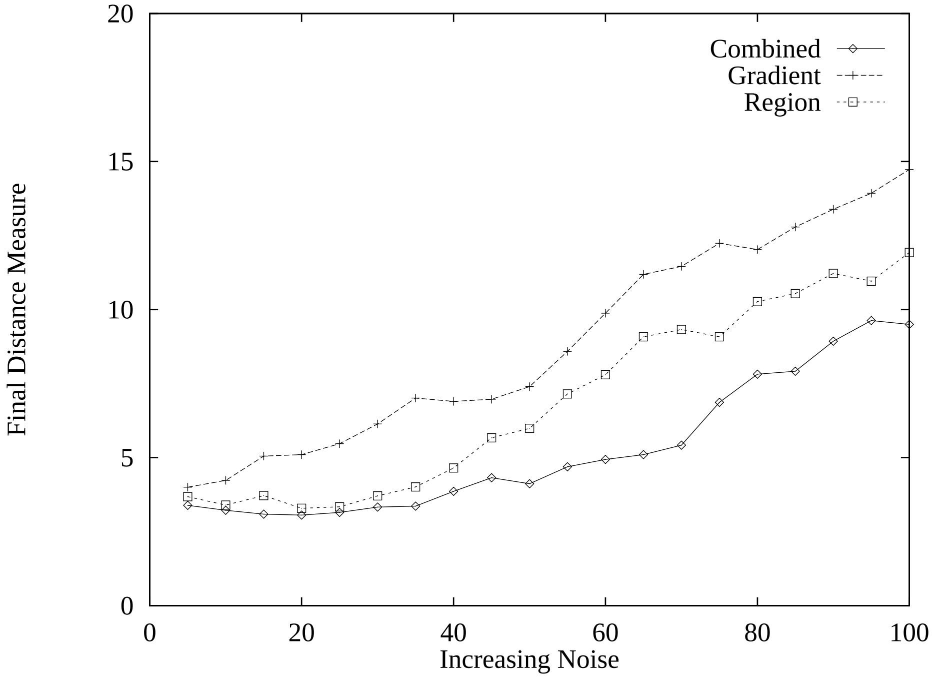


Figure 4.3: Performance under noise using the three methods (SNR varies from 4.0 to 1.0)

the background. Neither region-based segmentation nor gradient-based boundary finding works well for this noisy image. However, as we can easily see, the combined method produces the best result.

The next part of the experiment is to see how fast the methods converge. Figure 4.4 shows the situation when the image used has a noise level given by SNR=2.0 and has the same starting position shown in Figure 4.2. Clearly, the combined method performs better than the other two, especially compared to gradient-based boundary finding. Due to the global nature of the region term, initially the convergence is faster for the region-only method. But once it comes close to the actual boundary, due to the better localization property of the integrated method (which comes from the gradient term),

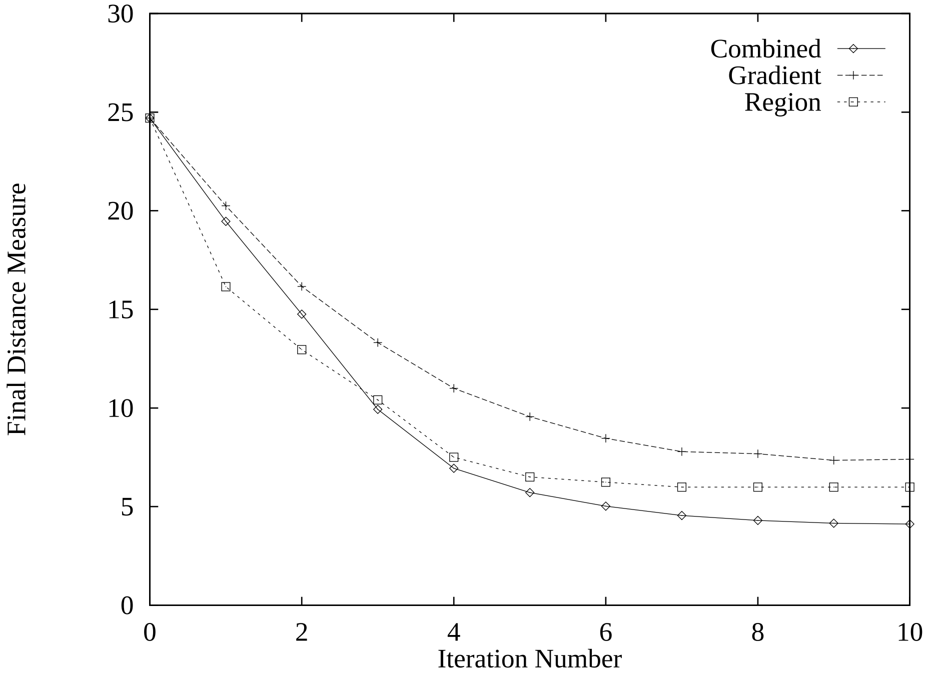


Figure 4.4: Convergence speed using the three methods (SNR=2.0)

the integrated method outperforms the region-only method. As for the gradient-only method, the effect of noise and fuzzy boundaries resist it from getting the best match.

The next plot, Figure 4.5 shows the situation under different starting positions. The initial contours vary from each other not only by translation and rotation, but also in terms of their exact shapes. We can observe that when the initial contour is close to the actual one, there is very little to choose among the three methods. However, when the initialization is far enough, the integrated method does considerably better than the gradient-only method. The region-only method comes close, the difference being due to the better localization property of the integrated approach. It must be noted that for the results shown in Figures 4.3, 4.4, 4.5 and 4.7 the experiments were repeated ten times under exactly the same settings, the only variation being that the noise sample is different

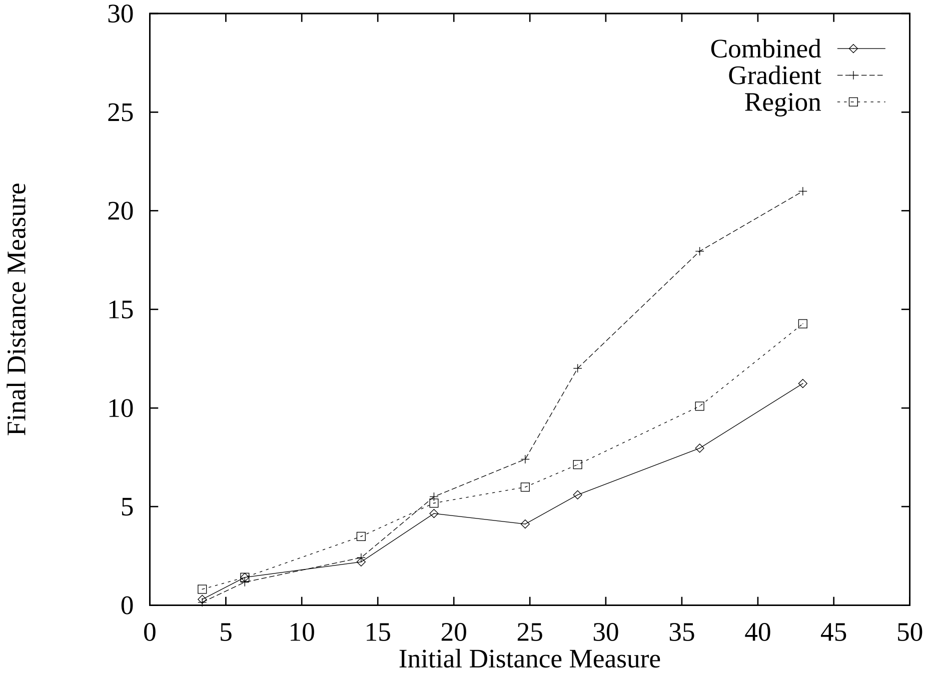


Figure 4.5: Performance under different Initializations (SNR=2.0)

even though the noise distribution is the same. The reported results were obtained by taking the average of the above which filtered out minor variations that might have arisen. One needs to do this because even though the noise distribution remains the same, the pixel values of the noise could be different. Thus it makes the plots more realistic.

4.3.1.2 Experiment 2

In the previous experiment, we were handling situations where both the methods that were combined produced reasonable results at least under low noise situations. As we increased the amount of noise, the gradient-based method started making more and more errors. However, there might be situations where one or both the methods might produce unrealistic results. To investigate such a situation, we devised an experiment with a synthetic occlusion.

If we look at Figure 4.6 (a) we can see that for a part of the image the intensity is lowered to a level slightly above the background. However, as it touches the background there is a further fall in the intensity level. On top of this as in the previous case, we smooth the boundaries and add noise to it. A similar thing might happen in a medical X-ray image where an object may be just behind part of another object which is of interest. Partial volume effect might also result in such spatial variations. Now if we do region based segmentation under the assumption that there are two regions, then that part of the central region beyond the artificial occlusion boundary will be classified with the background as can be seen in the figure. Thus as far as the region-based segmentation is concerned, it sees an occlusion. Now, if we base the boundary finding on this region classified image, it would tend to pull its boundary to the occlusion boundary. For those parts where there is no occlusion, it would still work. But for those parts where there is occlusion it would make gross mistakes resulting in a huge overall error. But the gradient-based method doesn't see much of a difference as it searches for a local gradient maxima, which is still there. Since the initialization is closer to the actual boundary than the occlusion boundary, the gradient-based method ignores the occlusion boundary. Under these circumstances, if we try to get the performance plot in the same way as we did in the previous case, we get the plot in Figure 4.7 where the axes and the legends have the same meaning as before.

Once again we can observe that the combined method performs much better compared to all the other methods. As expected, under low noise situations, the gradient-based method and the proposed integrated method are close. But as we increase the amount of noise the deterioration is very rapid for the gradient method. But the combined

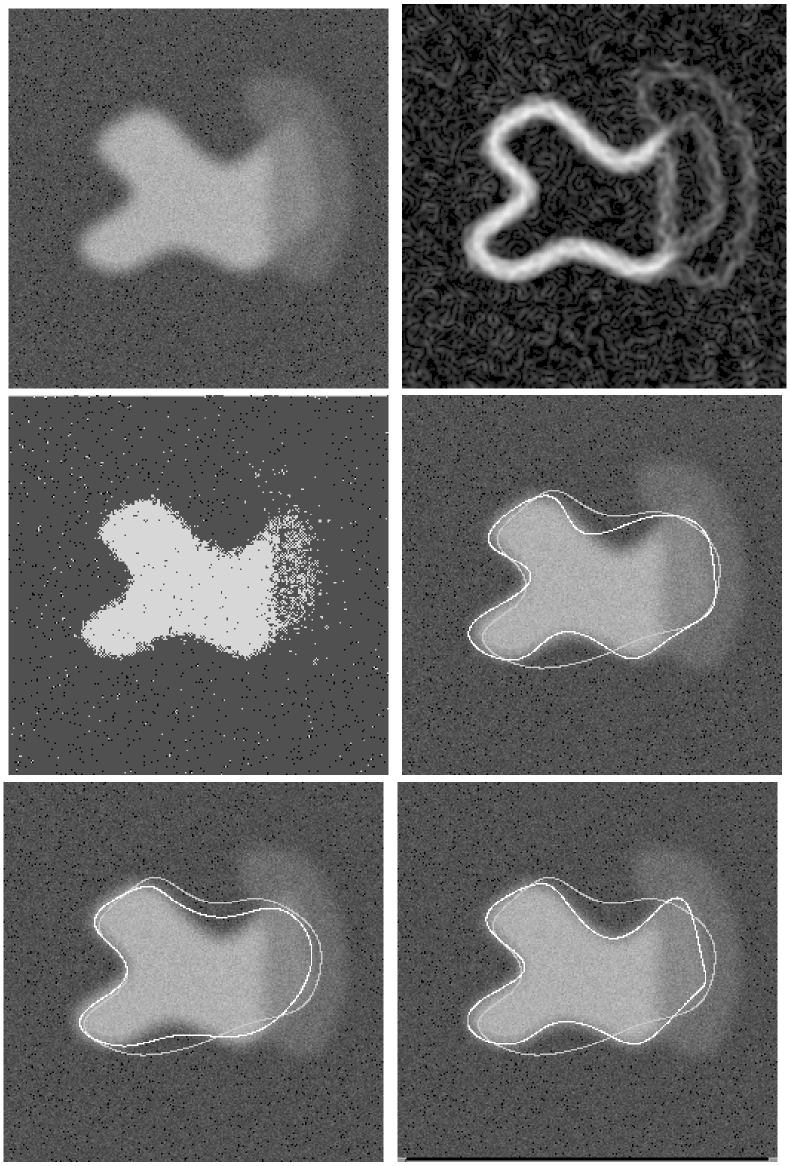


Figure 4.6: Performance of the three methods for an occluded image. (a)Top,Left: Original image; (b)Top,Right: Gradient image; (c)Middle,Left: Region Grown image; (d)Middle,Right: Output of Boundary finding using gradient information only; (e)Bottom,Left: Output of Boundary finding using region information only; (f)Bottom,Right: Output of Boundary finding using an integrated approach. In all the above, the darker contour represents the initial boundary placing and the brighter one the final derived boundary.

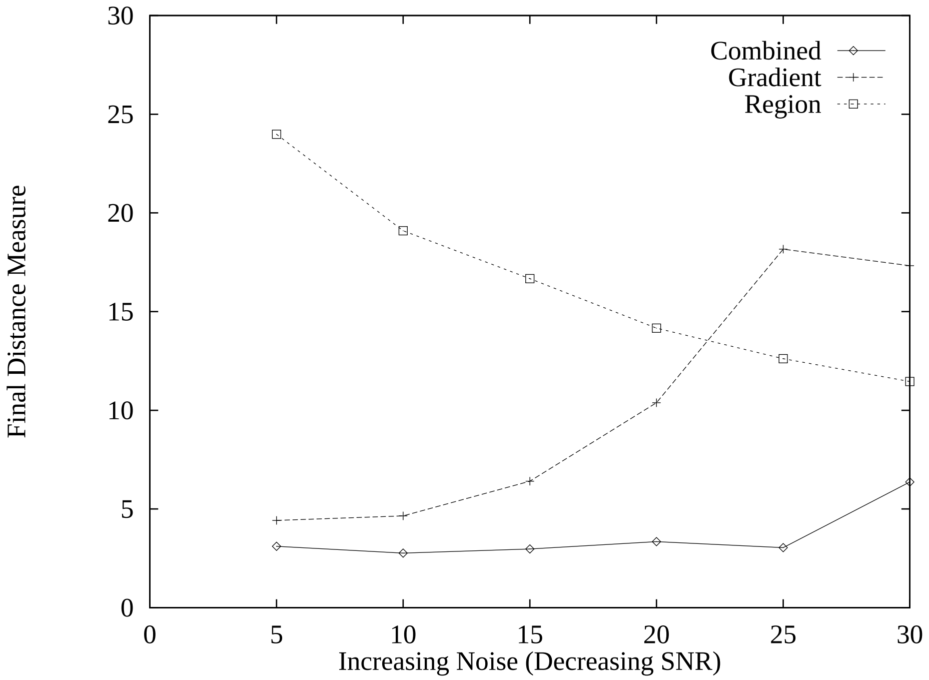


Figure 4.7: Performance of the three methods under occlusion

method shows little sensitivity to increased amounts of noise. This confirms the results that we had in the last experiment. But a very interesting thing happens with the region-based method. The plot seems to suggest an unreasonable thing: as the amount of noise increases, the performance improves. This apparent anomaly can be explained as follows. Since there is an occlusion, under low noise, the boundary comes close to the actual one where there is no occlusion, but where the occlusion is present, it pulls the boundary to the occlusion boundary. This results in large amounts of error. As we increase the amount of noise it becomes more probable for the optimization process to get trapped in a local minima resulting in smaller amount of movement. Thus it makes more mistakes at those positions where there is no occlusion, but at positions where there is occlusion it makes less error compared to the low noise case. This is so as long as the true boundary is closer to the initial boundary than the occlusion boundary. However, as we keep increasing the noise level, the characteristics flatten out and would again start rising for a further increase in noise. Figure 4.6 shows the occluded image and the performance of the different methods as described previously when applied on this image. It is easy to observe that the combined method performs the best.

Thus even if one of the methods fail partly, the integrated approach still seems to give reasonable results as it uses information from both the methods. We note here that a potential further work would be to use this method in conjunction with methods to handle occlusions (see [90] for example, which preserves depth information) for better results. We also note that if the initialization was close to the occlusion boundary, of course, none of the methods would be working anymore and for all the cases including the integrated method we would get grossly wrong results. It is precisely in situations like this that the importance of the prior boundary information can be realized which can serve as an initial boundary estimate as well. Let us note here that one might use a third region label to characterize the occluded region, and then try to find the boundary in such a way that the interior could be either of two regions used to characterize the target object.

4.3.2 Real Images

In this section we apply the algorithm to real world clinical images. As is often the case, neither gradient-based boundary finding nor region-based segmentation may give satisfactory results on their own.

Figure 4.8a shows such an image which is a short axis Magnetic Resonance (MR) image of a dog heart. The aim is to outline the endocardium. The region classified image is shown in Figure 4.8b. Figure 4.8c shows the initial contour used. Figure 4.8d

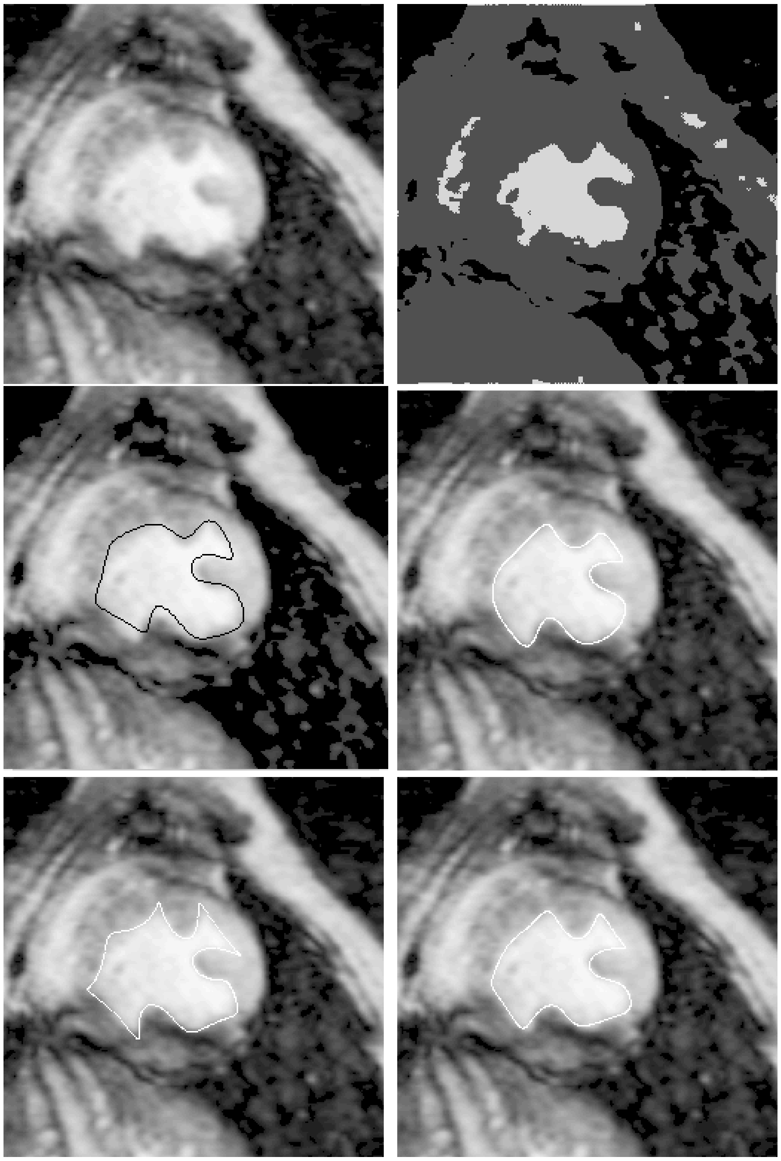


Figure 4.8: Example with an MR heart image. (a)Top,Left: Original image; (b)Top,Right: Region classified image; (c)Middle,Left: Original Image with the initial contour; (d)Middle,Right: Original image with the contour of the endocardium drawn by an expert; (e)Bottom,Left: Output of Boundary finding using gradient information only; (f)Bottom,Right: Output of Boundary finding using an integrated approach using the same initialization.

shows the probable edge as outlined by an expert. We display this only for the sake of comparison. It wasn't used otherwise. Region-based segmentation was done for three regions- endocardium, epicardium and the background using only the means and the variances. Once again, as in the synthetic examples, in this example and in the other examples no prior information was used. While we maintain that the use of appropriate prior information will only make the solution better, the purpose of this effort is to find out how the algorithm performs based solely on the image-derived information sources, without the introduction of expert knowledge. As we can see the image quality is very poor and the edges seem to be very fuzzy. Thus the gradient information is very weak. If we apply gradient-based boundary finding, due to a lack of strong edge information the boundary seems to diverge after a few iterations as shown in Figure 4.8e. Figure 4.8f shows the results of the integrated method, which though not perfect, is much better compared to the other method. (It needs to be mentioned, that like all the other experiments, we used the same initialization and the same number of harmonics to describe the boundaries in either cases.) The main reason for this improved performance is that neither region-based segmentation nor gradient-based boundary finding will actually fail as there is some information in both the gradient and the region classified image. But by themselves they have the limitations previously described and thus neither method produces desirable results. But once we combine them, the output seems to improve due to the information fusion, which relieves some of the limitations found when using the algorithms separately.

Figure 4.9 shows a similar sequence using a mid sagittal MR brain image, where the task comprises of segmenting out the corpus callosum. Once again, the expert drawing is used only for the sake of comparison. Region-based segmentation was done for three regions- the corpus callosum, the grey matter and the background (that includes the

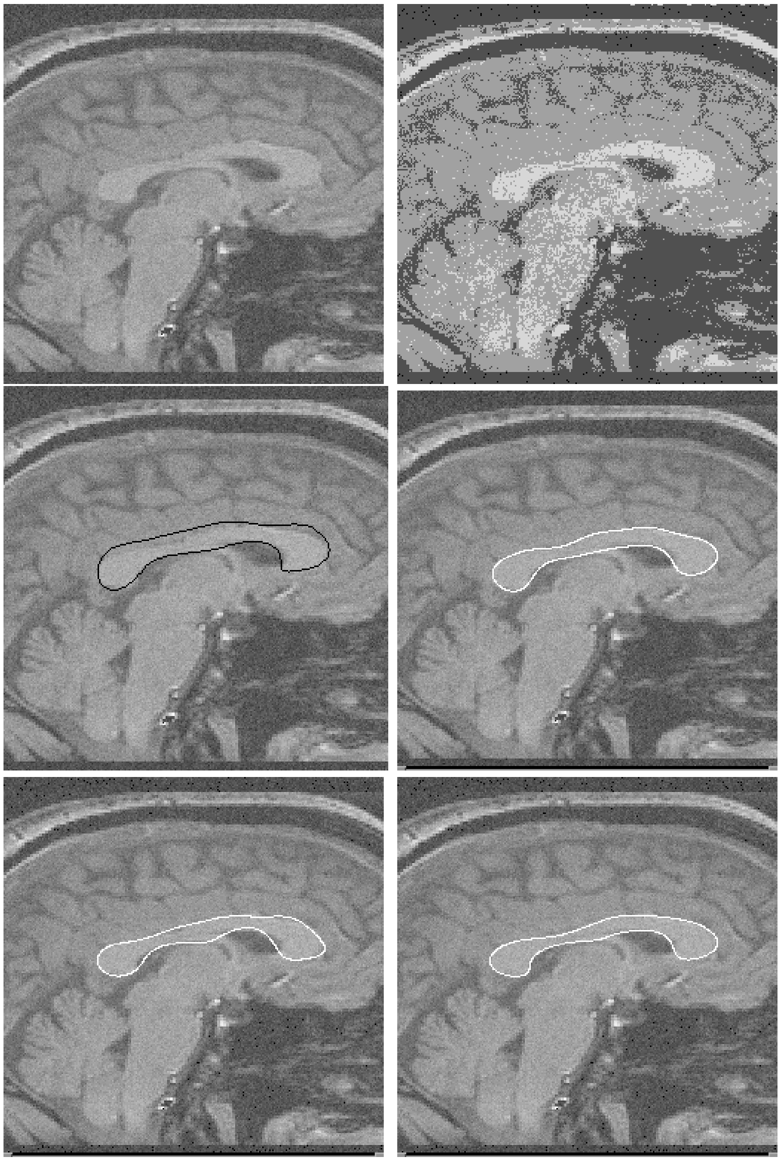


Figure 4.9: Example with an MR brain image. (a)Top,Left: Original image; (b)Top,Right: Region classified image; (c)Middle,Left: Original image with the initial contour; (d)Middle,Right: Original image with the contour of the corpus callosum drawn by an expert; (e)Bottom,Left: Output of Boundary finding using gradient information only; (f)Bottom,Right: Output of Boundary finding using an integrated approach using the same initialization.

CSF) using only the means and the variances. This is a better quality image compared to the previous application, and thus the improvement in performance is less, but even here one can easily visualize the improvement.

In Figure 4.10, the task is to outline the epicardium on an MR image of a dog heart. As in the example for the endocardium, region-based segmentation was done for three regions- endocardium, epicardium and the background using only the means and the variances (see 4.10(b)). This is a particularly difficult image because the right ventricle is not at all conspicuous and thus only an expert eye can point out the epicardial boundary as shown in Figure 4.10c. Without any constraints, the output of a normal gradient-based boundary finding algorithm is as shown in Figure 4.10e. One way of introducing constraints, where otherwise there are no image features, would be to mark out areas beyond which the boundary should not go. This can simply be done in the present framework by negating the pixels of the classified image in these forbidden areas. The forbidden region removes the right ventricle (parts of which can be differentiated in the region classified image, and essentially, the forbidden region just connects them). In this example, a region was roughly pointed out as forbidden for the epicardium by the user depending on a rough estimate of where the epicardial wall separates the right ventricle. Using this procedure, the output of the integrated approach is as shown in Figure 4.10f.

Finally, we consider a case which shows the importance of prior knowledge especially when both gradient and region information is inadequate primarily due to the nature of the image. Figure 4.11(a) shows an axial MR brain image. Our aim is to identify the left thalamus. Due to striations of white matter and due to partial volume effects, the two lobes of the thalamus cannot be differentiated using image features alone. Consequently, the prior information becomes important. Figure 4.11(c) shows the prior

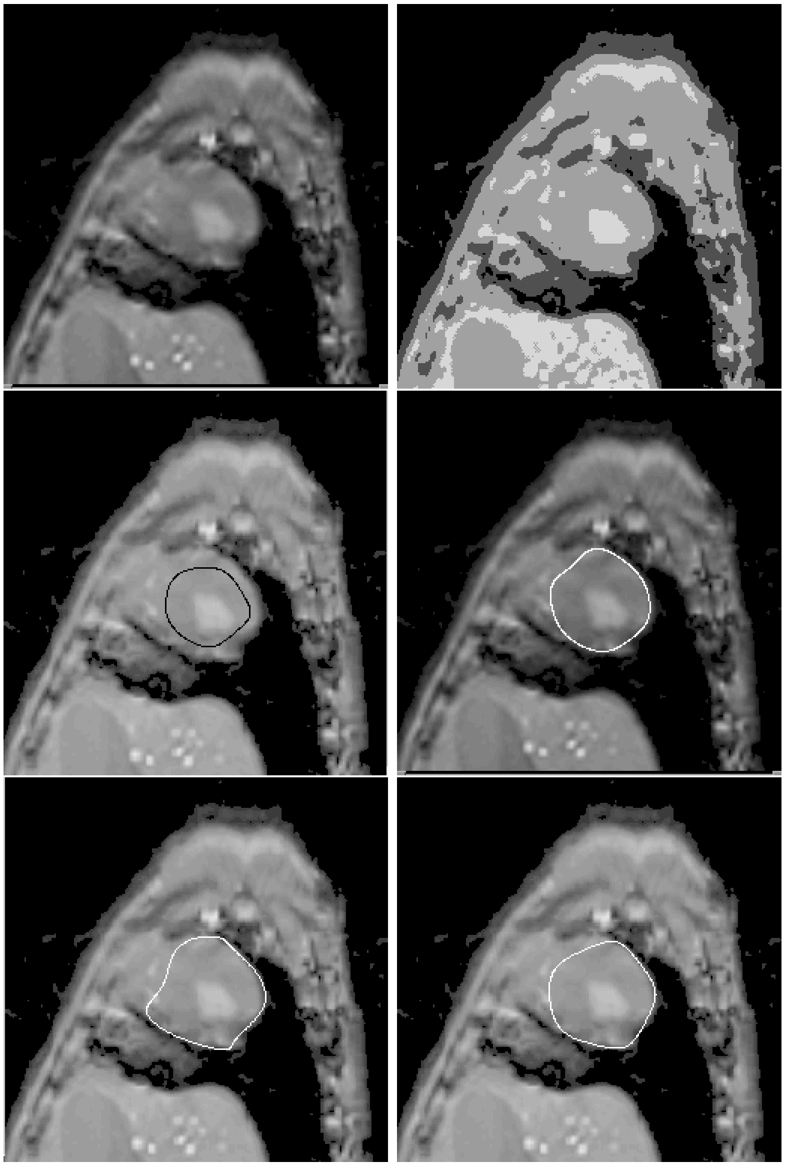


Figure 4.10: Example with a canine MR heart image where the task is to outline the epicardium; (a)Top, Left: Original image; (b)Top,Right: Region classified image; (c)Middle,Left: Initial boundary; (d)Middle,Right: Original image with the contour of the epicardium drawn by an expert; (e)Bottom,Left: Output of Boundary finding using gradient information only, without any constraints; (f)Bottom,Right: Output of Boundary finding using an integrated approach using region-based constraints and under the same initialization as in (e). As explained in the text, a region was described as forbidden at the left side of the image close to the epicardial wall separating the right ventricle, so that the final contour does not go inside the right ventricular region. Without this in the absence of any features the contour diverges.

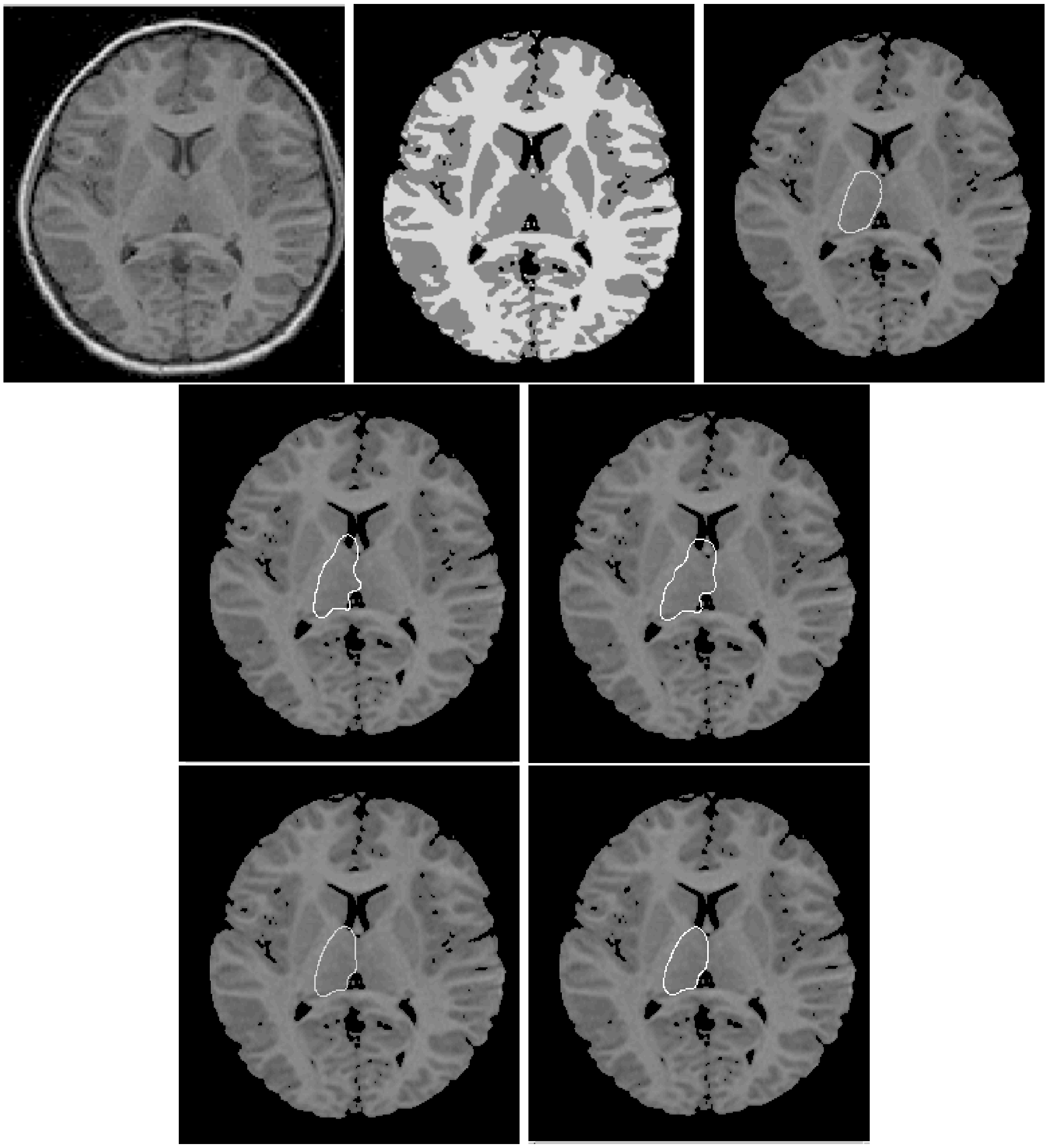


Figure 4.11: Example to show the importance of prior information. (a)Top,Left: An axial MR brain image; (b)Top,Middle: Region classified image; (c)Top,Right: Initial contour. In the last two results this was used both as the initialization and the prior; (d)Middle:Left: Output of gradient-based boundary finder without using prior information; (e)Middle,Right: Output of integrated boundary finder without using prior information; (f)Bottom,Left: Output of a gradient-based boundary finder using prior information; (g)Bottom,Right: Output of an integrated (gradient + region) boundary finder using prior information.

which was also used as the initialization. This was used only as an initialization for the results shown in Figures 4.11(d) and Figures 4.11(e) and was used both as the initialization in Figures 4.11(f) and Figures 4.11(g). Clearly, without prior information, both the gradient-only case and the integrated (gradient + region) approach diverges. However, when the prior information was used, it constrained the search space and the output is more reasonable. (Note: The skull was stripped using the integrated boundary finding approach.)

4.3.3 Reproducibility

Since there is still some human operator interaction required to use our proposed algorithm, we present results aimed at testing the reproducibility of the boundaries generated by the algorithm. We test the algorithm's reproducibility against results generated by a human operator (ie. manual tracing). We hypothesize that our approach will have a smaller variance, implying better reproducibility.

First, a bank of ten images were selected. These MR images constitute a canine heart from the apex to the base. For each one of these images, the algorithm was executed ten different times to find out the boundary of the endocardium from different initial settings as provided by an human operator. Note, that for the initialization, the operator quickly clicked a few points around the boundary. However, for manual tracings, the human operator carefully drew the boundary. The output of the algorithm consisted of the boundary of the endocardium. The boundaries were uniformly sampled to always have the same number of points. After that the mean and the variance in the contour position was calculated for each one of these images. The variance was then divided by the number of points on the contour.

For the manual tracings by a human operator, a domain expert traced carefully the boundaries for each of these images ten times and the same procedure as above was repeated. To minimize the effect of memorizing, the order of the images were randomized and the tracings were considerably spaced out in time. The results are shown in Table 1. Figure 4.12 shows the mean contours (human and algorithm generated) overlayed on

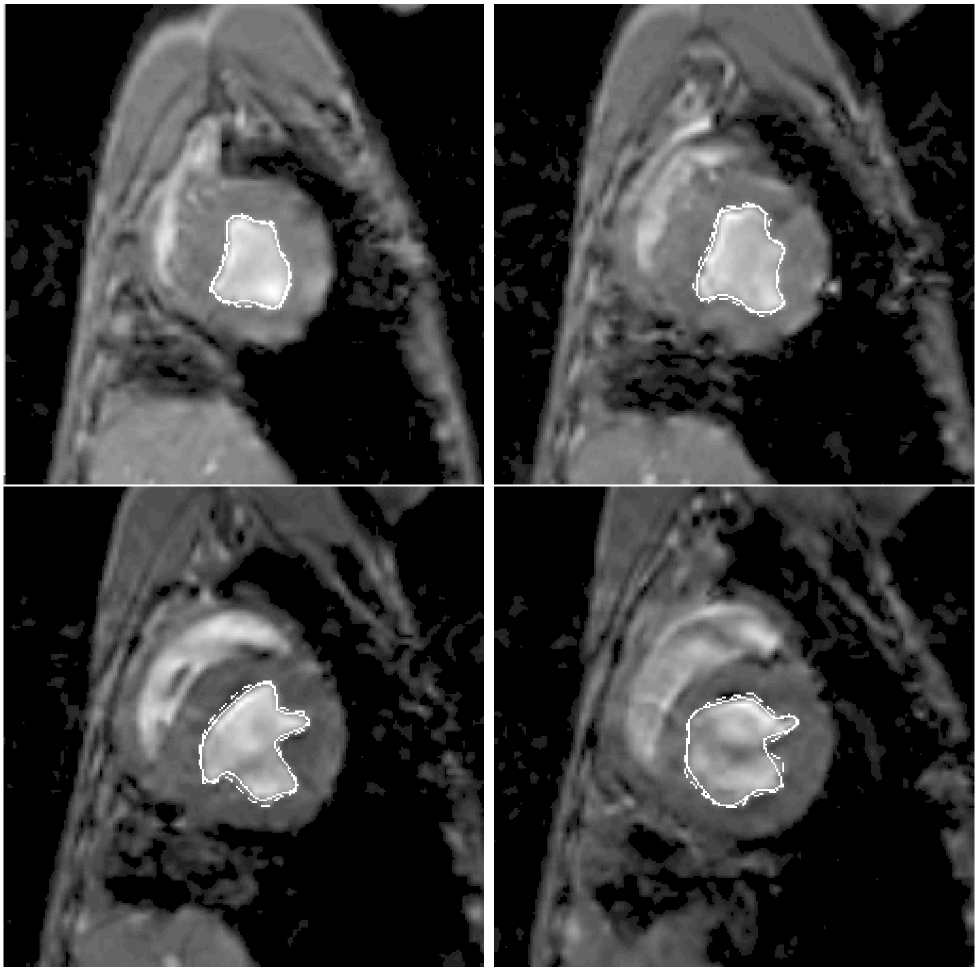


Figure 4.12: Four of the images used in the *Reproducibility* experiment with the mean contour as hand traced by an operator, and as found by the algorithm overlayed. The darker one represents the hand traced one and the brighter one the algorithmic.

four of the images, where the broken contour represents the human drawn and the solid one the algorithmic version. As we can see from the table, the variance is always at least twice as much for the human tracings when compared to the algorithmic results. This constitutes a significant deviation in reproducibility between the two methods. That this is indeed the case, is born out by a standard pairwise *T-test* on the variance data which shows that the difference between the two cases is highly significant ($p < 0.001$).

We note again that this experiment clearly shows two important things. First, as demonstrated in Figure 4.12, the results from the algorithm and the human tracings are in close agreement. Second, the result of the T-test shows that the variability in the algorithm is significantly smaller when compared to human tracings. Thus, the algorithm produces results that reasonably agree with a human expert's desired result, but provides a much more stable (i.e. less variable) estimate of the boundary location.

4.4 Conclusions

We have presented in this chapter a new technique for integrating the method of region-based segmentation into gradient-based boundary finding. This is separate from the other related works in the same way as edge detection is different from boundary finding. Our method was motivated by a Bayesian framework of maximization of the *a posteriori* probability. As the examples show, the integrated approach is more robust to both increased amounts of noise as well as increasingly displaced initialization of the initial boundary. Almost uniformly there is an improvement over the conventional gradient-based boundary finding. To prove this, we have devised a variety of experiments and the results from all of them are favorable.

Application of this method on real medical images results in noticeable improve-

Image Number	Algorithm based	Human
1	0.631	1.47
2	0.5	1.50
3	0.576	1.371
4	0.67	1.7
5	0.53	1.85
6	0.72	1.81
7	0.75	1.71
8	0.83	1.99
9	0.24	1.4
10	0.576	2.1

Table 4.1: Variations Using Human Hand drawn and automated tracings.

ment as has been shown. We are using it for clinical research purposes for outlining the endocardial and epicardial boundaries of the heart and the results are much better than what we had achieved using the purely boundary based method of Staib and Duncan [116].

Chapter 5

Multi feature Integration for Deformable Boundary Finding

5.1 Introduction

In the previous chapter, we found out that it is important to use both region and gradient information within an integrated deformable boundary finding framework to produce better segmentation results. In this chapter, our aim is to go a step further in the integration process. However, we continue with the same feature integration within the deformable boundary finding paradigm described in Chapter 1. Here, we introduce an integrated framework to handle the segmentation problem using not only gradient information, but region and curvature information as well. Further, we will also analyze the strengths and weaknesses of the different types of image-derived sources of information and under which circumstances each source of information should be given precedence over the others.

In the previous chapter we reasoned and demonstrated that low signal-to-noise ratio (SNR) and poor initialization can be better handled through the use of region information along with gradient information within the boundary finding framework. Besides the issue of low SNR, as indicated in chapter 1, another problem with the gradient-based boundary finder is that it has a tendency to smooth out sharp corners. To improve on the localization error, we propose using higher order information like curvature along with the gradient. We note here, the grey level curvature is used as an added feature that actually guides the boundary finder (towards the optimum boundary) in addition to the gradient information.

Once again, we will make use of the Bayesian framework as a guide to integrate the different sources of integration. However, integration may not lead to the desired result if any one of the sources of information is either too noisy or inaccurate. While the localization properties of derivative information like gradient and curvature are superior to those of the region-based information, the latter can handle noise better. Thus for a noisy image, more emphasis should be put on the region information. On the other hand, if the SNR in an image is not too low, a better localization can be achieved by giving more importance to the derivative information. The problem with spurious peaks is essentially similar to the low SNR problem and can be circumvented by enforcing the region information over the derivative features. The curvature information becomes important when the outline of the object curves sharply. If the image contrast is not high enough, this important feature may not be well captured by the gradient.

Thus, besides being able to combine the different image-derived features, it is also important to integrate them properly. We present experimental analysis to investigate circumstances under which particular combinations of different information sources are better than the others. Simple one-dimensional analysis is provided to back up these experimental results in the appendix (Appendix B). It also justifies the necessity of using region information along with derivative features.

5.2 Image-Derived Features

In this section, we shall briefly describe the methods used to acquire the three different data fields from the original image, all of which are used for estimating the right boundary.

5.2.1 Gradient information

The traditional way of doing edge detection or boundary finding is by making use of the gradient information. We obtain the gradient image in the traditional way of convolving the original image with the first derivative of the Gaussian kernel as in [116, 43]. Thus we get the gradient image while at the same time smoothing to some extent the effect of noise. We have already discussed this in chapters 3 and 4.

5.2.2 Curvature information

In this section, we will point out the importance of using curvature along with the gradient and the region information as suggested in the beginning of this chapter. The meaning of curvature is intuitively clear, it is a measure of the derivative of the tangent information (i.e. it measures the local deviation from the tangent line [114]). Information about the tangents to the isophote lines (lines of equal grey level intensity) can be obtained through the grey level gradient. But the information about curvature which is a second order feature cannot be obtained through the gradient, which is a first order feature.

When we want to place a boundary separating two regions in a grey level image, what we are interested in is to place the boundary in such a way that the difference between the points on either side of the boundary hits a local maximum. To do that, we must first look at the local structure of the image.

A completely continuous image can be considered as an analytic function $L(\vec{r}) = L(x, y)$, with,

$$L : \mathbf{R} \times \mathbf{R} \longrightarrow \mathbf{R} \tag{5.1}$$

Locally, we may approximate L by its Taylor series:

$$L(\vec{r} + \vec{dr}) = L(\vec{r}) + \vec{\nabla} L \cdot \vec{dr} + \frac{1}{2}\vec{dr} \cdot H[L] \cdot \vec{dr} + ... \tag{5.2}$$

with

$$\vec{\nabla} L = \left(\frac{\partial L}{\partial x}, \frac{\partial L}{\partial y} \right) = (L_x, L_y)$$

being the gradient and with

$$H[L] = \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix}$$

being the Hessian matrix. Now, since we intend to place the boundary at locations where the local pixel-wise difference in the grey level value reaches a maximum, we are interested in points, where

$$L(\vec{r} + \vec{dr}) - L(\vec{r}) = \vec{\nabla} L \cdot \vec{dr} + \frac{1}{2}\vec{dr} \cdot H[L] \cdot \vec{dr} + ... \tag{5.3}$$

is largest. The first term in the RHS of the above is clearly proportional to the gradient term and the second term is the second order term. Substituting from the Hessian, we get for the second order term,

$$\vec{dr} \cdot H[L] \cdot \vec{dr} = L_{xx}dx^2 + 2L_{xy}dxdy + L_{yy}dy^2 \tag{5.4}$$

The above is what is known as the second fundamental form for curves. Now, the curvature is proportional to the second fundamental form [114]. Thus one way of incorporating the second order information along with the gradient information would be to use the curvature. We conclude hence, that the use of curvature is important to improve the accuracy of the boundary estimation process. However, it needs to be used alongside the gradient, as it contains second order information that cannot be captured by relying on the gradient alone. This is especially true in the case of objects with sharp corners. The caution here is that, being a second order feature, it is much more sensitive to noise. Thus for an extremely noisy image this information may be misleading.

Calculation of the isophote curvatures (which approximate the curvature in the image grey level) directly from the grey level image can be obtained as in [15, 43, 64]. At every point, a local coordinate frame is assigned so that one of the axes coincides with the gradient direction and the other becomes normal to it. Calculation of the curvature then reduces simply to taking second derivative of the normal with respect to the gradient direction. This is then mapped back to the original coordinate frame. It can be shown that the expression for curvature at a location (x, y) in the image is given by [43] $\kappa = \frac{2L_xL_yL_{xy} - L_x^2L_{yy} - L_y^2L_{xx}}{(L_x^2+L_y^2)^{3/2}}$. Those points at which the gradient becomes zero (i.e. the denominator in the above goes to zero) are (must be) excluded from the curvature calculation, because the curvature is not defined at those points. In the following discussion $I_c()$ is calculated by computing κ at all the image positions.

5.2.3 Region information

Methods to acquire region information have already been described in great detail in chapters 3 and 4. We could use any of the two methods that were discussed in chapter 3. However, for most of the examples in this chapter (as in the last chapter) we used the MRF-based classification method. We note here again that the region information includes texture information as well.

5.3 Integration

Once, the data have been acquired from the original image in the form of region, gradient and curvature information, our aim here is to combine them using the maximum *a posteriori* framework. This is suitable for incorporating *a priori* shape information if available. Without the priors, it becomes similar to the maximum likelihood estimation problem.

Thus, we want to maximize $P(\vec{p}|I_g, I_c, I_s)$, where as described in previous chapters (chapters 3 and 4), $\vec{p}$ is the vector of parameters used to parameterize the contour, I_g is the gradient image, I_c is the curvature image, and I_s is the region classified image. Now,

$$\begin{aligned} P(\vec{p}|I_g, I_c, I_s) &= \frac{P(\vec{p}, I_g, I_c, I_s)}{P(I_g, I_c, I_s)} \\ &= \frac{P(I_s|I_g, I_c, \vec{p})P(\vec{p}, I_g, I_c)}{P(I_g, I_c, I_s)} \end{aligned}$$

$$\begin{aligned} &= \frac{P(I_s|I_g, I_c, \vec{p})P(I_c|I_g, \vec{p})P(I_g, \vec{p})}{P(I_g, I_c, I_s)} \\ &= \frac{P(I_s|I_g, I_c, \vec{p})P(I_c|I_g, \vec{p})P(I_g|\vec{p})P(\vec{p})}{P(I_g, I_c, I_s)} \end{aligned} \tag{5.5}$$

Thus ignoring the denominator $P(I_g, I_c, I_s)$, which is not a function of $\vec{p}$ the aim is to find:

$$\begin{aligned} \arg\max_{\vec{p}} P(\vec{p}|I_g, I_c, I_s) &\propto \arg\max_{\vec{p}} P(I_s|I_g, I_c, \vec{p})P(I_c|I_g, \vec{p})P(I_g|\vec{p})P(\vec{p}) && (5.6) \\ &\equiv \arg\max_{\vec{p}} [lnP(\vec{p}) + lnP(I_g|\vec{p}) + lnP(I_c|I_g, \vec{p}) \\ &+ lnP(I_s|I_g, I_c, \vec{p})] && (5.7) \end{aligned}$$

In equation (5.8), we have just taken the natural logarithm, which is a monotonically increasing function. Now, once again realizing that the use of derivative information can be used in the calculation of I_s, in an identical fashion as was done in Chapter 4, the final form can be written as a combination of the following matching terms:

$$\begin{aligned} \arg\max_{\vec{p}} M(\vec{p}, I_g, I_c, I_s) &= \arg\max_{\vec{p}} [M_{prior}(\vec{p}) + M_{gradient}(I_g, \vec{p}) \\ &+ M_{curvature}(I_c, \vec{p}) + M_{region}(I_s, \vec{p})] \end{aligned} \tag{5.8}$$

The first term in equation (5.8) corresponds to the prior shape term which we have already discussed in chapter 4. The second term in equation (5.8) is actually the likelihood term which depends on the gradient image and is identical to the second term in (4.6) of chapter 4 and has already been discussed. The third term is dependent on the curvature information, and can be treated in exactly the same way as the gradient term. It can be considered to give additional information for local difference in the pixel strength that may not have been captured by the gradient term. This term will be most effective at those places where the gradient information is not particularly strong, yet there are sharp changes in the curve e.g. at corner locations. Thus,

$$M_{curvature}(I_c, \vec{p}) \propto \int_{C_{\vec{p}}} I_c(x, y)dt \tag{5.9}$$

In the above, $I_c(x, y)$ is the curvature image (derived directly from the grey scale values as described before).

However, as has already been mentioned, curvature being a second order feature is highly susceptible to noise. To avoid the effects of noise one could emphasize the importance of curvature at those locations where we expect it to play an important role and de-emphasize it at other locations. One way to do it is to weigh the curvature term along the boundary by the curvature of the contour itself. Thus, where the contour curves rapidly, the curvature information is given more importance and vice-versa. Noise from the curvature image thus enters the optimization process only where the curvature has a significant value. Thus,

$$M_{curvature}(I_c, \vec{p}) \propto \int_{C_{\vec{p}}} \alpha_2(x, y)I_c(x, y)dt \tag{5.10}$$

In the above, $\alpha_2(x, y)$ is the scaled curvature of the contour (to be calculated only at the contour points).

The last term in (5.8), incorporates the region information into the boundary finding framework. It is equivalent to the last term in (4.6) of chapter 4 and is evaluated in a similar way. Hence, finally we have,

$$\begin{aligned}
\arg\max_{\vec{p}} M(\vec{p}, I_g, I_c, I_s) &= \arg\max_{\vec{p}} \left[M_{prior}(\vec{p}) + M_{gradient}(I_g, \vec{p})\right. \\
&+ \left. M_{curvature}(I_c, \vec{p}) + M_{region}(I_s, \vec{p})\right] \qquad (5.11) \\
&\equiv \max_{\vec{p}} \left[M_{prior}(\vec{p}) + K_1 \int_{C_p} \{I_g(x,y) \right. \\
&+ \alpha_2(x,y) I_c(x,y)\}\, dt \\
&+ \left. K_2 \int\int_{A_{\vec{p}}} I_s(x,y) dA \right] \qquad (5.12)
\end{aligned}$$

where K_1 and K_2 are the weighting constants which signifies the relative importance of the two terms in the above equation and $\alpha_2(x, y)$ is a scaled version of the contour curvature as previously described. It would be appropriate to restate here that while the approach is motivated by the maximization of the *a posteriori* probability method, in the final formulation, intuitively appealing definitions rather than exact probabilistic terms are used.

As in chapter 4 we could once again combine the line and area integrals using Green's theorem and the result is as follows.

$$\begin{aligned}
\arg\max_{\vec{p}} M(\vec{p}, I_g, I_c, I_s) &\equiv \arg\max_{\vec{p}} [M_{prior}(\vec{p}) \\
&+ \int_{C_{\vec{p}}} (K_1 \{I_g(x,y) \\
&+ \alpha_2(x,y) I_c(x,y)\}\, dt \\
&+ K_2\{N_s(x,y)\frac{\partial x}{\partial t} + M_s(x,y)\frac{\partial y}{\partial t}\}\,)dt]
\end{aligned}$$

$$(5.13)$$

where, $M_s(.)$ and $N_s(.)$ are as in equations (4.10) and (4.11). Thus, finally in this section, we have presented a boundary finding procedure that integrates along with the gradient, curvature and region information.

While the above gives us a model to combine the different sources of information, how much of what needs to be combined remains an issue. The quantity $\alpha_2(x(p,t), y(p,t))$ controls the relative importance of the curvature term to the gradient term. From our discussion regarding the curvature term, it is clear that under high SNR conditions it could be as important as the gradient term. Hence, it should have a maximum value of unity. The relationship between K_1 and K_2 controls the relative importance of the derivative and region information. We shall have more to say about this in the discussion and in Appendix B.

Optimization of the above is achieved using the conjugate gradient method as before.

5.4 Results

In this section, we show some results that reinforces not only our claim that integration as explained above results in a better boundary finding procedure, but also explains the circumstances that make a particular piece of information more useful than the others.

First, we shall consider the importance of using gray level curvature. As we have already mentioned, this is second order information and thus cannot be used alone to do boundary finding. It has to be used in conjunction with the gradient information. This information becomes particularly important when we are trying to trace an object that does not have a very strong contrast with the background. In other words, cases under which the gradient information is not strong enough. In such situations, especially if the object curves strongly, curvature information could be quite helpful. However, it must be ensured that the SNR of the image is not too low because in an image with low SNR, the isophote curvatures (grey-level curvature) cannot be computed reliably.

One such example is shown in Figure 5.1(a). As we can see, the contrast is low. The SNR for this image is 3.0 where the signal strength is assumed to be the difference in intensity between the target region and the background, and the noise power is given by the variance of the added zero-mean Gaussian noise. In Figure 5.1(c) the output of boundary finding using just the gradient information is shown. The darker contour shows the initialization, and the brighter boundary is the final output. Figure 5.1(d) depicts the same when we use both the gradient and the curvature information.

Clearly, the integrated method does a much better job. Figure 5.1(e) shows the outputs from the two methods overlayed on the true object. The darker boundary is the one due to the first method, where only the gradient information is used, and the brighter one corresponds to the later method. As expected, the difference is prominent where there are sharper features, i.e. at the location of the high curvature points.

In the next example, we continue with the same object but drastically reduce the SNR. For this image (shown in Figure 5.2(a)), the $SNR = \frac{1}{3}$. As we have argued previously, under such low SNR situations, unless we have a very good initialization, it is profitable to give more emphasis to the region information. This is especially true if as in this case, the size of the object is not negligible. Figure 5.2(b) shows the result of the boundary finder using only gradient information. Curvature information is not used due to the extremely low SNR. Once again, as in the previous case, the darker contour is the initialization, and the brighter contour the final version. It fails to latch on the true object boundary. On the other hand, if we use both the region and the gradient information

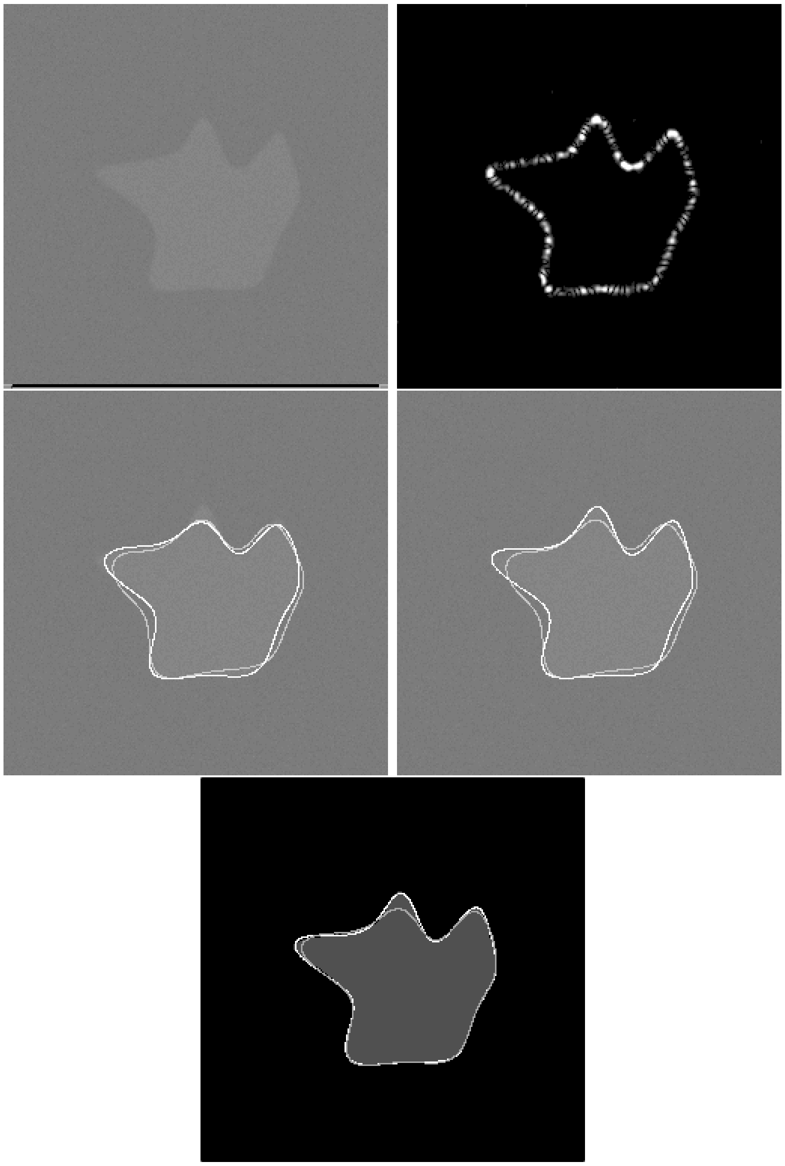


Figure 5.1: Example with a synthetic image to show the importance of integrating curvature information into the boundary finding framework along with gradient information. For this image, the $SNR = 3.0$, but the contrast is low. (a)Top,Left: The original image; (b)Top,Right: Curvature image; (c)Middle,Left: Output of boundary finding using gradient information only. The darker contour represents the initialization, the brighter one the final output. (d)Middle,Right: Output of boundary finding using gradient and curvature information. The darker contour represents the initialization, the brighter one the final output. (e)Bottom: The outputs overlayed on the template of the object out of which the noisy image was generated. The brighter contour is the one generated using both gradient and curvature information and the darker contour is the one generated by using just the gradient. Note that the one using gradient and curvature is superior to the gradient-based boundary finder (shown by the darker contour) is evident wherever the object curves sharply.

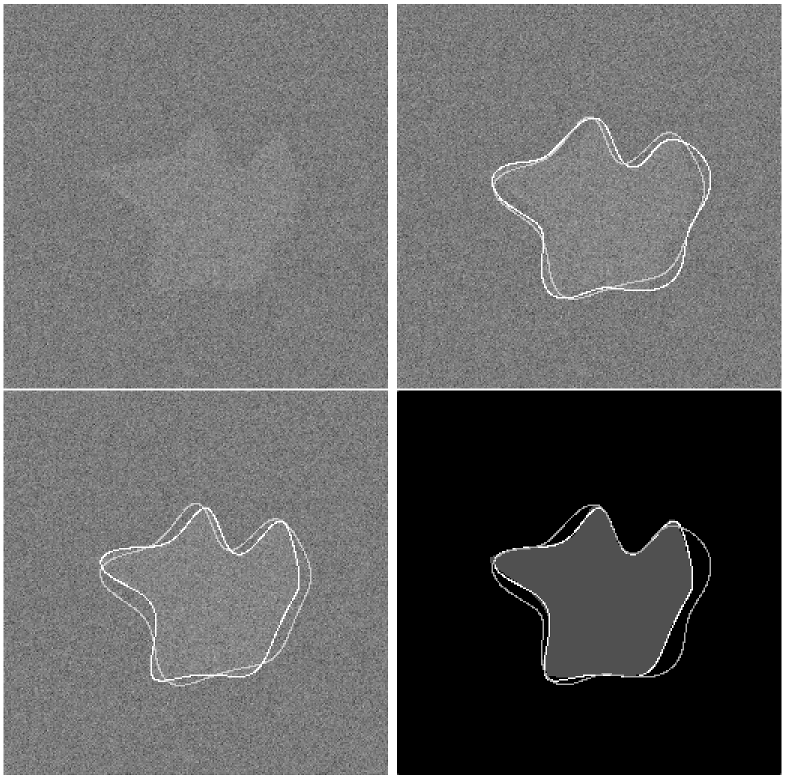


Figure 5.2: Example with a synthetic image to show the importance of integrating region information into the boundary finding framework along with gradient information. For this image, we have a very low $SNR = \frac{1}{3}$. (a)Top, Left: The original image; (b)Top, Right: Output of boundary finding using gradient information only. The darker contour represents the initialization, the brighter one the final output. (c)Bottom, Left: Output of boundary finding using gradient information. (d)Bottom, Right: The outputs overlayed on the template of the object out of which the noisy image was generated. The brighter contour is the one generated using both gradient and region information. That it is superior to the gradient-based boundary finder (shown by the darker contour) is evident.

as shown in Figure 5.2(c), we get a reasonable output. For this example we gave equal weighting to the gradient and the region term. Figure 5.2(d) shows the two outputs overlayed on top of the actual object. The brighter one is the output of the integrated version. However, if we look carefully to the brighter contours in Figures 5.2(d), and 5.1(d), we notice that the one in 5.1(d) is better localized. This is to be expected since the one in Figure 5.2(d) relies more on the region information, which when compared to gradient information has a higher localization error. This is so as in the previous image, derivative information (gradient and curvature) is used, whereas in the latter case, it is the region information that decides the final output.

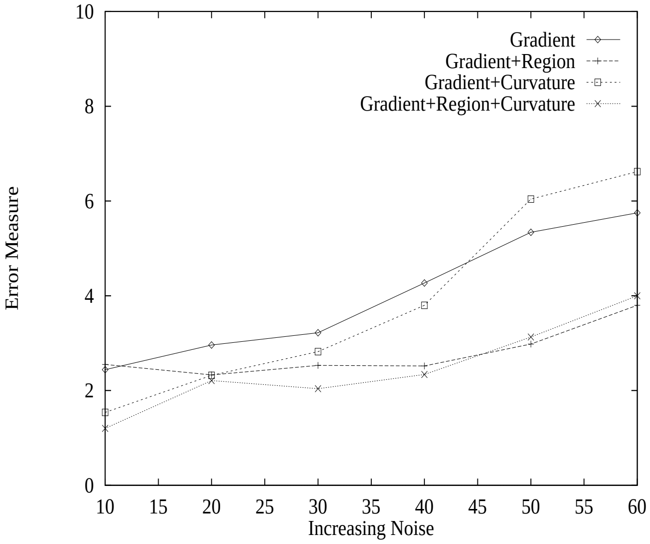


Figure 5.3: Performance under varying noise conditions. The SNR varies from 3.0 to 0.5. (Explanations are given in the results section of the text.)

To further verify the circumstances under which a particular piece of information should be used, we formulated experiments similar to [25] where under the same initialization, the distance between the final output and the true contour was compared under different circumstances and under different situations of information fusion. The x-axis for Figure 5.3 represents noise values (The SNR varies from 3.0 to 0.5). The y-axis represents the error measure which is a scaled version of the Euclidean distance (see the previous chapter and [116, 25] for details) between the final contour the algorithm produces and the actual boundary. For all these examples, once again, we gave equal weighting to the gradient and the region term.

From Figure 5.3, we observe that some form of information fusion is almost always beneficial. As we would expect, however, performance of all the methods deteriorate with increasing noise. However, when the noise is too high, the curvature information becomes extremely unreliable, resulting in a huge overall error. Clearly, the stability of the region information is obvious. Initially, the method combining gradient and region information does not perform as well as the method that combines gradient and curvature because under high SNR, the curvature and the gradient information are very reliable. So the primary source of error is the localization error which is higher for the region information. But soon, as the noise increases, the advantage that derivative information might have over the region information is more than nullified by the better noise performance of the method that combines region and gradient information. Thus for images with a low SNR, this is the preferred method. Any method that uses derivative information predominantly is likely to get trapped in a false minima under low SNR conditions. The integrated method which combines all three sources of information does better for the most part, except when the noise is too large. When the noise is too large, due to unreliable curvature information, it starts performing worse than without it i.e., the method that only combines gradient and region information and excludes the curvature information. We note here that while the above conclusions are related to the results shown, our experience with a host of other real world clinical images also conform to the above.

In the following example we consider the case of outlining the endocardium of the heart. Figure 5.4(a) shows a short axis MR image of a canine heart. From our experience this is one of the most difficult images of its type. Figure 5.4(b) shows the expected boundary as outlined by a domain expert. In Figure 5.4(c) we see the results of using just the gradient-based boundary finder and in Figure 5.4(d), the results of an integrated boundary finder using both region and curvature information along with the gradient information. In the last two images, the white contour represents the initialization, and the black one the final output. Obviously, the integrated method has a superior performance. The gradient-based method fails due to the lack of strong gradient features. Once again, in this example, we gave equal weighting to the gradient and the region term as it was not possible to choose one source of information over the other.

In the final example, we consider the case of an ultrasound image of the heart as shown in Figure 5.5(a), where the task is to outline the left ventricle. This is an ideal example where the texture and not just the pixel intensity is important. Region classification was done using texture properties and is shown in Figure 5.5(b). A gradient-based method alone is not likely to produce good results. Thus, as shown in Figure 5.5(d), using a gradient-based boundary finder does not give us desired results. On the other hand, the boundary finder in the integrated case as shown in Figure 5.5(e) does not diverge and the result looks reasonable. Here, we gave the region term three times the weight of the gradient term. The region information was derived using a texture segmentation

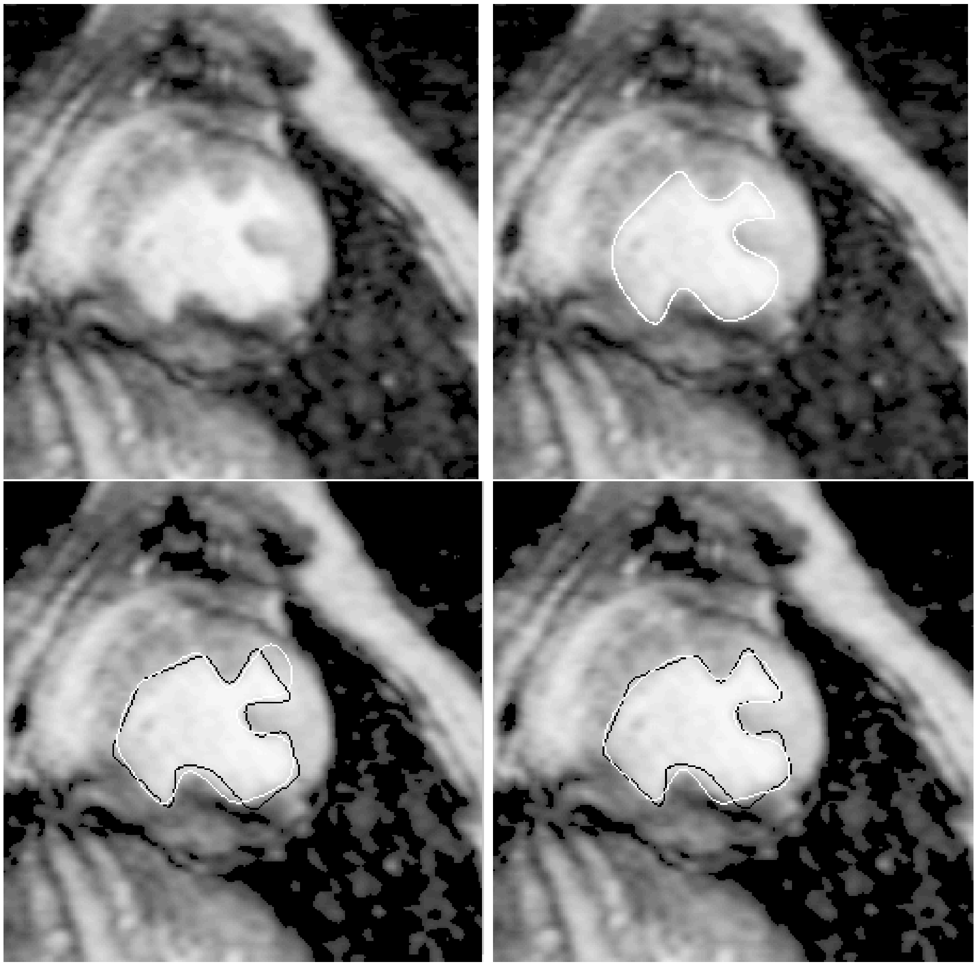


Figure 5.4: Example with a short axis MR image of a canine heart. (a)Top, Left: Original image; (b)Top, Right: Original image with the contour of the endocardium drawn by an expert; (c)Bottom, Left: Output of Boundary finding using gradient information only. It seems to be diverging; (d)Bottom, Right: Output of Boundary finding using an integrated approach using the same initialization. The output looks reasonable. In both (c) and (d), the black contour is the initialization, and the white one is the final output. The superior output in (d) is due to the use of additional features as compared to that in (c).

procedure described in chapter 3. Clearly, the boundary finder in this integrated case does not diverge and the result looks reasonable although no claim is made about quantitative accuracy.

5.5 Discussion

In the section in which we described the integration method, and again in the results section, we pointed out the significance of the observation that while information fusion

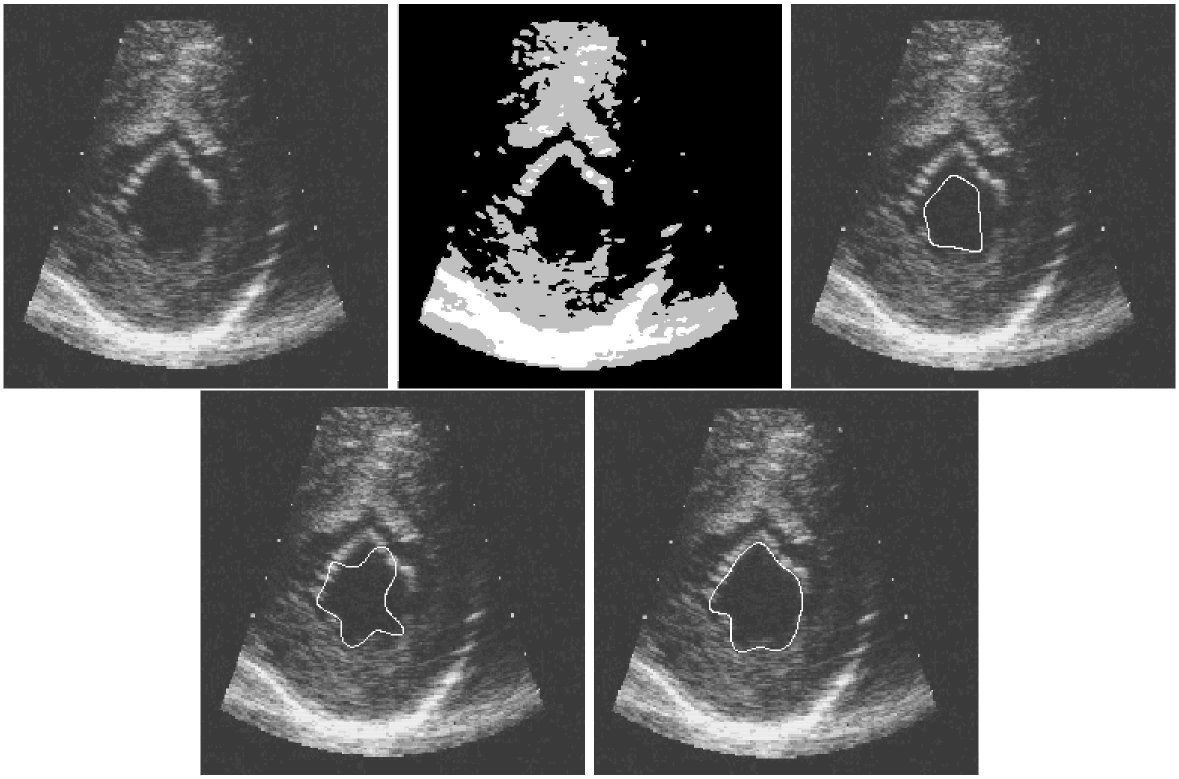


Figure 5.5: Example with an ultrasound image of the heart. (a)Top,Left: Original image; (b)Top,Middle: Region classified image; (c)Top,Right: Initial boundary; (d)Bottom,Left: Output of a gradient-based boundary finder. That it diverges is clear; (e)Bottom,Right: Output of an integrated (gradient + region) boundary finder which looks quite reasonable.

is helpful in general, it needs to be done appropriately. This is so as there might be cases (as we have already seen) where a particular source of information may be much more reliable than others and vice versa. Under such circumstances, the proper way to combine information would be to give more importance to the reliable features.

While the relation between the gradient and the curvature information is clear, the relationship between the region and the derivative information (gradient and curvature) is not so obvious.

Here we shall try to address that issue through a very brief mathematical analysis. In part motivated by earlier efforts, [21], we shall consider the effects of localization, the output signal to noise ratio, and response to spurious peaks.

To keep the analysis tractable, we will do the analysis for the 1-D equivalent of a boundary as has been done by many other researchers including those by [21] and [59]. Thus the boundary reduces to a step and the purpose of the procedure is then to estimate precisely the location of the step. Further, as in the above references, we shall model the edge as a Gaussian step and the noise as additive white Gaussian noise. Gaussian edge profile is used because most of the commonly encountered edges can be approximated by this model [21] and other kinds of profiles are either far less common or can be constructed from linear combinations of steps [59]. Further, the blurring caused by the imaging system can be reasonably approximated by the Gaussian kernel.

The details of the analysis are provided in Appendix B. However, we note the following:

1)If we look at the expressions for the localization error and the SNR for the gradient-based boundary finding (see equations (B.11) and (B.24) in appendix B), we can easily observe that for the cases where the edge blur is smaller than the smoothness ($\sigma_s \leq \sigma_c$, which constitute the reasonable choices, where as defined in the appendix, σ_s is the standard deviation of the Gaussian that is used to approximate the edge-blur, and σ_c is the standard deviation of the Gaussian smoothing kernel), both the localization error and the SNR are proportional to σ_c. Thus as we increase the amount of smoothing, for a noisy image, the output SNR improves, but the localization error increases as well.

On the other hand, for region-based boundary finding (see equations(B.20), (B.28) in appendix B), we note that there is no such tradeoff. The localization error for this method is proportional to σ_s^2. Thus, if σ_s is small and particularly if it is smaller than unity but if the noise is high, it will require a larger value of σ_c to get a reasonable value of the SNR for the gradient only case. Under such circumstances we find that the region-based method would result in a much better localization. In other words, if the step is almost perfect but the noise is high, there are circumstances, when the region-based method will have a lower localization error. But, in the general case of a non-perfect step, the localization error would be higher in the case of the region-based term.

2) The localization error gives a measure of the maximum deviation that can take place once the edge has been found, i.e. the boundary finding process has converged close to the true boundary. But the criteria that drives a process to that end, i.e. helps a process to converge is the output signal to noise ratio. If the SNR is high, in general the convergence is fast. The SNR in the gradient-based process, as seen in (B.24) is proportional to σ_c, so there is some scope of improving it, at the cost however of increased localization error.

On the other hand, the output SNR from the region-based method (B.28) is proportional to $\sqrt{L}/2$ where L is the size of the region. Thus, in general the SNR in this case would be much larger than the gradient-based method. The only exception to this would be the case when L is small. But, then these are precisely the cases when we would want the output SNR to be low, because a small L would mean a tiny region which could be due to spurious peaks and we are not interested in allowing the boundary finder to be attracted to it. Thus we observe, that the SNR is in general larger in the case of the region-based method (as used here) and it depends on the size of the region beyond the step. This is a stronger global property than what the SNR is based on in the gradient-based method.

3) Regarding the response to spurious peaks, as the equations (B.30) and (B.31) in appendix B point out, not surprisingly the conclusions are similar to that based on SNR. Since the region-based method is a more global approach, the significance of spurious peaks are smaller. This is so as in the region-based case, the response is proportional to $\alpha/\sqrt{L}$. Thus as long as the width of the spurious peak is small compared to the width of the region that we are trying to find the boundary of, the significance of the spurious peak is negligible.

Thus we might infer based on both the 1D analytical results and the experimental results that for an integrated framework, we should do the following,

(i) If the image has an high SNR, we should emphasize the gradient information over the region information to take advantage of the low localization error of the gradient information.

(ii) If the initialization is far away, or if the SNR is low, we should emphasize the region information to avoid false boundaries or divergence. In a real world example, of course it is not possible to know how far the initialization is. In that case, the confidence the user has in the initialization becomes important.

(iii) If we are interested in a particular value of the SNR, the above analysis allows us to get an optimal combination of K_1 and K_2. However, for most images, exactly quantifying the weights may be very difficult and could involve too much of preprocessing. In that case, a reasonable weighting scheme would be to use $K_1 = 1$ and $K_2 = 1/SNR$ in equation (5.13) i.e. give more weight to the region term as the noise increases.

5.6 Conclusions

We have, presented in this chapter a novel method to integrate features within the deformable boundary finding framework, namely region (which includes texture) and curvature information along with the gradient information. We would like to point out that here we used the grey level curvature in the image as an added source of information to enhance the performance of the boundary finding procedure.

Also, we have presented experimental and analytical results to point out what the benefits could be of using the different sources of information and under which circumstances which information should be given more importance.

While gradient information is in general useful, under low SNR conditions, region information is more helpful. Also, it is more robust to outliers such as spurious peaks and shallow minimas. Curvature information is particularly useful when the object has sharp corners. However, if the noise is high it should not be used as it is extremely susceptible to it. When used judiciously the integrated method improves the performance of the boundary finding procedure and this is obvious from the results presented both with the synthetic and clinical images. However, we note that while the treatment in this and the previous chapter uses both, gradient and region information, it does not exploit the coupling that exists between them in an optimal way. Also, the use of isophote curvatures has its drawbacks in the sense that sometimes in real images, the isophotes may not be well defined.

Finally, we note that one might argue using higher order features (beyond second order) within the deformable boundary finding framework. Theoretically, that would mean using more terms in the expansion given in equation (5.2) and should further improve the final solution. But in reality, realizing this gain may almost be impossible. First, we have the problem of noise. The second order derivatives are already very noise sensitive as we can expect and was borne out by the experiments. Third and higher orders are likely to be even more noise sensitive, making the whole system unstable. Next, even when the noise is low, points at which third and higher order derivatives are likely to be useful would be very few. We noted before that the second order information would make a difference only at those points where the curves turn sharply, making the first order information inadequate. But these are the corners on the curve. The corners along with the flat portions are adequate in defining most of the contours, making third and higher order information almost redundant except for curves which undergo topological changes at the corners. However, since such instances are rare and are likely to be an effect of noise rather than real physical object boundaries, use of such information might in practice not only increase the computational burden unnecessarily, but also might make the system highly noise sensitive. Consequently, we do not further extend the feature integration framework by adding further higher order information within the deformable boundary finding framework.

Chapter 6

Game Theoretic Integration

6.1 Introduction

In the last two chapters, we discussed methods for doing feature based integration within the deformable boundary finding framework. There we used additional information besides gradient to perform boundary finding. As we observed, this improved the boundary estimate. Region information was first acquired from the raw image, and was subsequently used within the boundary finding framework. But no effort was made to improve the region information itself. Effectively, it was thus a one way flow of information. The basic premise of this chapter is that we could do even better by refining the boundary and region information at the same time and feeding back these complimentary sources of information to each other. Our strategy will be to carry out the above as module integration where the image segmentation problem is modeled as an integration of two computational modules, one region-based and the other, boundary based. Integration in this case will be achieved using a game theoretic framework as already discussed in chapter 1. This consists of allowing the region and boundary modules to assume the roles of individual players who are trying to optimize their individual cost functions within a game theoretic framework. This keeps them decoupled in the sense that each module updates its own output only. The flow of information is restricted to passing only the results of the decisions between the modules allowing us to easily combine the incommensurate objectives involved. For any one of the modules, the results of the decisions of the other modules are used as priors. This makes it unnecessary to construct a giant objective function and optimize all the parameters simultaneously. The solution of the problem is given by the equilibrium achieved as a consequence of this rational decision making. We also note that this is technically more general than the single objective approach mainly because the modules are allowed to have a different view of the world, i.e. work under different probability spaces. When the probability spaces are identical, the equilibrium solution becomes equivalent to that of the single objective approach [79]. Further, the decoupling also makes the computation easier.

Module integration is expected to improve the outputs of both the modules, as improvement in one of the modules automatically means improvement for the other module as well because related information from one module is fed back to the other and vice versa. As we had already noted in the introduction, feature integration as described in the last two chapters can be considered to be a special case of module integration. This is so because in the feature integration case, optimization is carried out over a set of parameters, while another set of parameters remains unchanged. However, in the case of module integration, we optimize over the whole set of parameters corresponding to both the modules.

Another issue that is worth mentioning over here is the rationale behind our use of Bayesian strategies to motivate the work. In the game-theoretic integration framework discussed here, flow of information consists of passing the output of each one of the modules to the other modules in the form of prior information. Here, a module has no authority to change the outputs of other modules. Thus it can use the information regarding other's output to update its own output, but cannot manipulate the information it gets from others. This use of prior information clearly makes a Bayesian framework suitable. Further, the use of Bayesian strategies also gives us the flexibility to incorporate other kinds of relevant information, if available.

In this chapter, we shall first briefly mention (without details) some preliminaries of game theory that we shall use later, and then go on to discuss in detail the non-cooperative Nash game as applied to our image segmentation problem. This includes a mathematical description of the modules under consideration. We will then present the results of this novel method.

6.2 Game Theory

Game theory as a concept has its roots in decision making under conflicting and often hostile environment. The theory of finite zero-sum games, which is the most elementary type of game, dates back to Borel in the early 1920s, [16]. Borel introduced the notion of a conflicting decision situation that involves more than one decision maker and some other important concepts such as pure and mixed strategies (we shall shortly see what these are) but didn't develop a complete theory of games. The foundations of game theory as we know it today, were laid down by Von Neumann in his pioneering book with Morgenstern [93]. Other enlightening works include those due to Mckinsey [86], the two volumes edited by Kuhn and Tucker [70, 71] and others.

6.2.1 Preliminaries

In this section, we will discuss some of the keywords in game theory that will be important for our subsequent discussions.

A *game* is a set of rules and conventions for playing. A *play* is a particular possible realization of these rules. A *move* is a point in a game at which one of the players picks out an alternative from a set of alternatives. The set of alternatives that are available to a player constitute his *strategy* space. The particular alternative that the player picks from that set is considered to be his *choice, strategy or decision*.

Games can be distinguished as *one-person*, *two-person* and so on, dependent on the *number of players* (teams) that are playing the game. In an *N-person game* the rules are such that the players fall into N mutually exclusive sets in such a way that players within each set have identical interests. Every player is associated with a *payoff function* that evaluates the performance of the player based on his decision and those of his co-players. Thus, the rational choice of every player is to make his move in the strategy space in such a way that he maximizes his *payoff function*. It incorporates goals that are purely internal to the player as well as constraints imposed on him by the other players.

A further classification of games are based on the characteristic of their payoff functions. Accordingly, they can be either *zero sum* or *non-zero sum*. In a *zero sum game* what a player gains is necessarily at the expense of one or more players who incur a loss. In a *non-zero sum game* no such restriction is present. The game that we will be considering falls under this category. Games can also be grouped as either *noncooperative* or *cooperative* [96]. In a *noncooperative* game any type of collusion such as correlated strategies and side payments are forbidden. In a *cooperative* game, any such cooperation is permitted. In this work, we shall only be concerned with noncooperative games. Depending upon the number of choices that are available to the decision makers, a game can be either *finite* or *infinite*. In a *finite* game the number of choices available to the players are limited while in the *infinite* case they are unlimited.

A game is said to have reached its *equilibrium* when there is no incentive for any unilateral deviation by any one of the players.

6.2.1.1 Noncooperative Finite Games: Two-person Zero-sum

The most elementary type of games are where there are two agents that are fighting to share a fixed amount of resources. Thus necessarily, the gain of one of the players is at the expense of the other. These games can be viewed in a matrix form where the matrix coefficients describe the gains for one of the players or alternately, the loss to the other. Thus at every turn, one player tries to maximize his gain and minimize his loss. The dimension of the matrix is determined by the alternatives that are available to the players. In a pure form of the game, the equilibrium is determined by a min-max strategy which leads to a saddle point. Sometimes the game is not carried out in a pure form and is instead carried out in a *mixed* form. A mixed strategy for a player is a probability distribution on the space of his pure strategies. Saddle-point equilibrium in such cases can also be attained by a similar minimax strategy. For a more detailed discussion and an elegant proof of the min-max theorem one can look in [8, 86]. In our work, we would only be interested in pure strategies.

6.2.1.2 Noncooperative Finite Games: N-person Nonzero-sum

The main distinction between the *nonzero-sum* and the previous *zero-sum* games is that the outcome of a decision process in the present case does not necessarily dictate the verdict that what one player gains culminates in a loss to some other player.

Keeping this in mind, the two person matrix games of the previous case can be easily extended to form what can be termed as a *bimatrix game.* Accordingly, a bimatrix game is comprised of two matrices, $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$, with each pair of entries (a_{ij}, b_{ij}) denoting the outcome of the game corresponding to a particular pair of decisions made by the players. Thus, the strategies for the players $P1$ and $P2$ are defined respectively by the rows and the columns.

Stipulating that there exists no collaboration between the players and that the players make their decisions independently, the equilibrium is defined as follows:

Definition: A pair of strategies, {row i^*, column j^* } is said to constitute a *noncooperative (Nash) equilibrium solution* to a bimatrix game $(A = \{a_{ij}\}, B = \{b_{ij}\})$ if for all i and j it satisfies,

$$\begin{aligned} a_{i^*j^*} &\leq a_{ij^*} \\ b_{i^*j^*} &\leq b_{i^*,j} \end{aligned} \tag{6.1}$$

Furthermore, the pair $(a_{i^*j^*}, b_{i^*j^*})$ is known as a *noncooperative (Nash) equilibrium outcome* of the game.

Probably the most famous example of the bimatrix game is the so called *Prisoners dilemma.* It characterizes a situation in which two criminals are held on suspicion of having committed a certain crime. Since there is no direct evidence against them, their conviction depends on their confessions. If both confess, both get a sentence of 8 years, if neither confesses, they each get 2 years, and if one confesses and the other doesn't then the one that does confess is set free and the other one gets 30 years. It is easy to see that this symmetric game has a dominant strategy for both the players under which both of them confess in order to reduce their risk. Since, this is a non-cooperative game, they cannot collude. Thus, the equilibrium solution is that both prisoners get 8 years. This simple example not only illustrates the existence of a *Nash equilibrium* for this problem but also shows that it can be different from the general notion of optimum. This clearly shows that the solution obtained under a game theoretic setting may not be identical to the one obtained by optimizing a single objective function.

As in the previous case, the bimatrix games can also be extended to have mixed strategies. A very powerful result regarding such games say that *every bimatrix game has at least one Nash equilibrium solution in mixed strategies.*

The theory of the finite *N-person nonzero-sum games* is almost identical to the bimatrix games that we just discussed.

6.2.2 Noncooperative Infinite Games

The main difference between this and other types of noncooperative games is that here at least one of the players has at his disposal an infinite number of alternatives to choose from.

Here it is assumed that all the players have effectively an infinite number of strategies to choose from. The game may be either sequential or parallel. In a sequential game, the decision makers (players) follow a certain sequence in making their moves. In the parallel mode on the other hand, the players make their moves simultaneously. This difference in the mode of decision making results in different equilibria for the two cases. While in the sequential case, the game achieves what is called the *Stackelberg equilibrium* which depends upon who the leader is, in the other case, it achieves the *Nash equilibrium.*

6.2.2.1 ϵ-Equilibrium Solutions

Let us assume that there are N decision makers or players, D_1, D_2, ..., D_N where F^i, the payoff of D_i, the i^{th} player depends not only on this player's actions, but also on those of some or all of the other players. D_i 's action is denoted by p^i, which belongs to his strategy space, i.e. $p^i \in P^i$. Then, the following [8] is a precise definition of an ϵ-*equilibrium solution* in N-person games within the context of pure strategies.

Definition: For a given $\epsilon \geq 0$, a N-tuple $\{p_\epsilon^{1^*}, ..., p_\epsilon^{N^*}\}$, with $p_\epsilon^{i^*} \in P^i$, $i \in \mathcal{N}$, is called a (pure) ϵ-*Nash equilibrium solution* for an N-person nonzero-sum infinite game if

$$F^i(p_\epsilon^{1^*}, ..., p_\epsilon^{N^*}) \leq \inf_{p_\epsilon^i \in P^i} F^i(p_\epsilon^{1^*}, .., p_\epsilon^{(i-1)^*}, p^i, p_\epsilon^{(i+1)^*} .., p_\epsilon^{N^*}) + \epsilon, \quad i \in \mathcal{N} \tag{6.2}$$

For $\epsilon = 0$, one simply speaks of *equilibrium* instead of 0-*equilibrium* solution in which case, we denote the equilibrium strategy of D_i by $p_\epsilon^{i^*}$. From the viewpoint of each player, its Nash decision is locally optimal. If all the module's objectives require only individual optimality, the Nash equilibrium is a natural definition for rationality. In the following we describe the connection between rationality and Nash Equilibrium in more details.

6.2.2.2 Rationality and Nash equilibria

A pure-strategy *Nash equilibrium* solution in infinite games can be obtained as the intersection point of the reaction curves of the players. Formally we have the following definition [8].

Definition: In a two person game, let the minimum of $F^2(p^1, p^2)$ with respect to $p^2 \in P^2$, be attained for each $p^1 \in P^1$. Then, the set $R^2(p^1) \subset P^2$ defined by

$$R^2(p^1) = \{\xi \in P^2 : F^2(p^1, \xi) \leq F^2(p^1, p^2), \quad \forall p^2 \in P^2\} \tag{6.3}$$

is called the optimal response or the rational response of D_2. If $R^2(p^1)$ is a singleton for every $p^1 \in P^1$, then it is called the reaction curve or reaction function of D_2 and is denoted by $l_2(p^1)$. The reaction set and curve of D_1 are similarly defined (simply by interchanging the indices). In Figure 6.1, the *constant level* or *iso-cost* curves of $F^1(.)$ and $F^2(.)$ have been drawn for a specific game with $P^1 = P^2 = \mathcal{R}$. For fixed p^1, say $p^1 = \bar{p}^1$, the best D_2 can do is to minimize F^2 along the line $p^1 = \bar{p}^1$. Assuming that this minimization problem

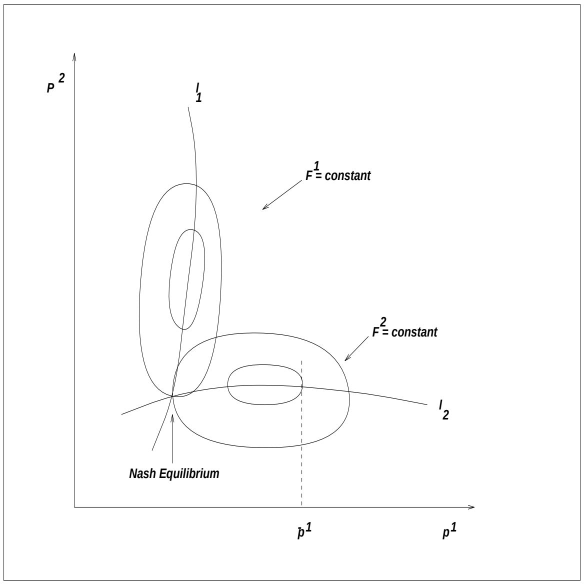


Figure 6.1: Constant level curves for $F^1(.)$ and $F^2(.)$ and the corresponding reaction curves (l_1 and l_2) of D_1 and D_2 respectively.

admits an unique solution, the optimal response of D_2 is determined in the figure as the point where the line $p^1 = \bar{p}^1$ is tangent to an *iso-cost* curve $F^2(.) = constant$. For each different $\bar{p}^1$, a different unique optimal response can be found for D_2, and the collection of all these points form the reaction curve of D_2 indicated by the curve l_2 in the figure. The reaction curve of D_1 is similarly constructed: it is the collection of all points (p^1, p^2), where horizontal lines are tangent to the *iso-cost* curves of F^1, and is indicated by l_1 in the figure. By definition, the Nash solution must lie on both reaction curves, and thus if these curves meet at only one point, the Nash solution exists and is unique.

With this background, in the following sections, we shall first characterize the two-person game for our segmentation problem and then describe the actual modules.

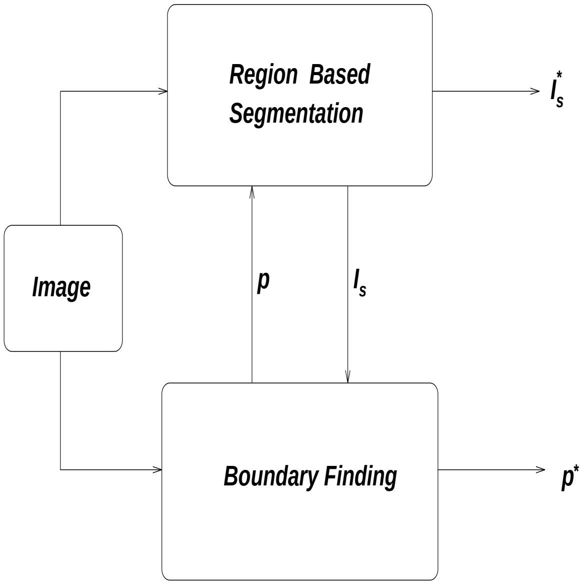


Figure 6.2: Flow diagram for game-theoretic integration of region-based segmentation and boundary finding. The outputs of each of the modules, I_s for the region module and p for the boundary module are fed back to each other after every decision making step. The game stops when none of the modules can improve their positions unilaterally. The final solution pair, (I_s^*, p^*) constitutes the Nash Equilibrium

6.3 Game theoretic integration of Segmentation modules

Our goal here is to develop a fully bi-directional framework for integrating boundary finding and region-based segmentation as shown in Figure 6.2. This would lead to a system where the two modules would operate in parallel so that at each step the output of each module gets updated using information from the outputs of both the modules from the previous iteration and the data itself. Thus as the game progresses, both the modules improve their positions through mutual information sharing. Even though the modules produce related outputs, they often produce separate complimentary information. For example, in an image we might be interested in the shape and precise location of some structures, while for some other structures (which are not well located) it might be important to quantify them i.e. to find the total activity. Clearly, neither boundary finding, nor region-based segmentation by themselves are going to be able to provide us with all the necessary answers because while region-based segmentation is not likely to give us precise information regarding shape and location, boundary finding may not be feasible if some of the structures are not well located. We have already seen before in chapters 4 and 5 how the use of region information refines the boundary estimate. Use of boundary information can also be useful if for example there is an overlap in the region types of one of the poorly located structures which needs to be quantified with that which can be well isolated using boundary finding. We will show such an example later. However, even if we are interested in either of the modules, it is necessary to improve the solution of the other module because, as the goals are related, an improved solution for one module automatically means a better solution for the other one too. Thus, if for instance, we are interested in the final boundary finding output (see Figure 6.2) it is important to improve the region module too (which might be necessary for other purposes) as the output of that module is used to find the boundary. A clinical example of this could be to delineate the endocardial boundary of the heart (as in Figure 6.5) to analyze its shape, but maintain the hypothesis that the region interior (the left ventricular (LV) blood pool) should be roughly homogeneous. On the other hand, sometimes the region information could be more important. For analyzing the size of a cancerous growth, we might want to calculate the area of a tumor due to certain Magnetic Resonance Image signal properties but preserve a rough, physician guided shape.

In the following, we pose the image segmentation problem as a 2-person, non-zero sum, non-cooperative game.

6.3.1 Nash equilibrium for the Two-player Segmentation problem

The game is played out by the two segmentation modules, who assume the role of the players. Thus, $\mathcal{N} = \{1, 2\}$ is the player set, P^1 and P^2 the strategy spaces of the decision makers D_1 and D_2 respectively and

$F^1 \; : P^1 \times P^2 \longrightarrow \; R$

is the cost function for decision maker 1 and,

$F^2 \; : P^1 \times P^2 \longrightarrow \; R$

is the cost function for decision maker 2. Decision maker 1 corresponds to the region module and Decision maker 2 is the boundary module.

The normal form of the game is described by $\{F^1, F^2\}$ where it is assumed that P^1 and P^2 are the appropriate Hilbert spaces and we have the following definition of Nash equilibrium from (6.2)

Definition: A pair of strategies $(\bar{p}^1 \in P^1, \bar{p}^2 \in P^2)$ constitutes a *Nash Equilibrium* solution if $\forall p^1, p^2$

$$\begin{aligned} F^1(\bar{p}^1, \bar{p}^2) &\leq F^1(p^1, \bar{p}^2) && (6.4) \\ F^2(\bar{p}^1, \bar{p}^2) &\leq F^2(\bar{p}^1, p^2) && (6.5) \end{aligned}$$

Definition: If we consider k to be the time (i.e. iteration) index, then a *parallel decision making* game theoretic algorithm can be viewed as [79]:

$$\begin{aligned} p^1_{k+1} &= \arg\min_{p^1 \in P^1} F^1(p^1, p^2_k) && (6.6) \\ p^2_{k+1} &= \arg\min_{p^2 \in P^2} F^2(p^1_k, p^2) && (6.7) \end{aligned}$$

where $p^1_0 \in P^1$ and $p^2_0 \in P^2$ is the initial point.

As of now, we have assumed arbitrary cost function structures for the payoff functions. But for the segmentation problem, we shall assume that the cost functions have the following structure:

$$\begin{aligned} F^1(p^1, p^2) &= f_1(p^1) + \alpha f_{21}(p^1, p^2) \\ F^2(p^1, p^2) &= f_2(p^2) + \beta f_{12}(p^1, p^2) && (6.8) \end{aligned}$$

where α and β are some scaling constants.

We also consider the following constraints on F^1 and F^2.

(i) F^i is bounded in $p^i \in P^i$.

(ii) F^i is continuously second order differentiable in $p^i \in P^i$.

(iii) $\exists$ a closed neighborhood $u^i \subseteq P^i$ such that F^i is strongly convex in U^i. For the above assumptions, we have the following which can be considered to be a corollary to a more general theorem in [79] but is more relevant to our image analysis problem:

Theorem

For the given above structure of $F^1(p^1, p^2)$ and $F^2(p^1, p^2)$ there exists a locally stable Nash Equilibrium solution, i.e. for any $p^1 \in U^1 \subseteq P^1$ and $p^2 \in U^2 \subseteq P^2$, the sequence of rational choices generated by the parallel decision making process converges and the limit point is a Nash equilibrium solution if α and β satisfy the following condition:

$$\begin{aligned} &\left\| \left[\left(\alpha^{-1} \frac{\partial^2}{\partial p^1 \partial p^1} f_1(p^1) + \frac{\partial^2}{\partial p^1 \partial p^1} f_{21}(p^1, p^2) \right)^{-1} \left(\frac{\partial^2}{\partial p^1 \partial p^2} f_{21}(p^1, p^2) \right) \right] \right. \\ &\left. * \left[\left(\beta^{-1} \frac{\partial^2}{\partial p^2 \partial p^2} f_2(p^2) + \frac{\partial^2}{\partial p^2 \partial p^2} f_{12}(p^1, p^2) \right)^{-1} \left(\frac{\partial^2}{\partial p^2 \partial p^1} f_{12}(p^1, p^2) \right) \right] \right\| < 1 \end{aligned} \tag{6.9}$$

We present the proof below, which follows similar reasoning as those in [79]. The first few steps are similar and are given here for continuity. However, the bulk of it is different because we are looking at a different set of requirements, suitable for our image analysis problem.

Proof

By strong convexity and continuous second order Freschet differentiability, the operators $\frac{\partial^2}{\partial p^1 \partial p^1} F^1(p^1, p^2)$and $\frac{\partial^2}{\partial p^2 \partial p^2} F^2(p^1, p^2)$exist, are continuous and are strongly positive in U^1 and U^2. Hence, $(\frac{\partial^2}{\partial p^1 \partial p^1} F^1(p^1, p^2))^{-1}$ and $(\frac{\partial^2}{\partial p^2 \partial p^2} F^2(p^1, p^2))^{-1}$ exist and thus by the implicit function theorem on Banach spaces (Hilbert spaces) [60] $\exists\, L_1$ and L_2 such that

$$L_1 \;:\; U^2 \longrightarrow U^1 \tag{6.10}$$

$$L_2 \; : \; U^1 \longrightarrow U^2 \tag{6.11}$$

both continuously locally Freschet differentiable such that

$$\left.\frac{\partial}{\partial p^1} F^1(p^1, p^2)\right|_{p^1 = L_1(p^2)} = 0 \tag{6.12}$$

$$\left.\frac{\partial}{\partial p^2} F^2(p^1, p^2)\right|_{p^2 = L_2(p^1)} = 0 \tag{6.13}$$

in the neighborhood of some given pair, $(\tilde{p}^1, \tilde{p}^2) \in U^1 \times U^2 \subseteq P^1 \times P^2$. Since, this is true for any such pair and L_1, L_2 are unique in each case due to the strong convexity, the parallel decision making model is equivalent to

$$p^1_{k+1} = L_1(p^2_k) \tag{6.14}$$

$$p^2_{k+1} = L_2(p^1_k) \tag{6.15}$$

$\forall (p^1_0, p^2_0) \in U^1 \times U^2 \subseteq P^1 \times P^2$. Hence, for any $\bar{p}^1 \in U^1$, $\exists (p^1, p^2)$, such that

$$p^1 = L_1(p^2) = L_1(L_2(\bar{p}^1)) \tag{6.16}$$

Let's denote,

$$T \,:\, U^1 \longrightarrow U^1 \tag{6.17}$$

ie. $T = L_1 L_2$ is a self mapping. Hence,

$$\frac{\partial}{\partial p^1} T(p^1)\Big|_{\bar{p}^1} = \dot{L_1}(p^2)\dot{L_2}(\bar{p}^1) \tag{6.18}$$

or

$$\dot{T}(p^1)\Big|_{\bar{p}^1} = \dot{L_1}(p^2)\dot{L_2}(\bar{p}^1) \tag{6.19}$$

Now, we are interested in that value of p^1 that satisfies

$$p^1 \; = \; \arg\min F^1(p^1, p^2) \tag{6.20}$$

Hence,

$$\frac{\partial}{\partial p^1} F^1(p^1, p^2) = 0 \tag{6.21}$$

Thus, for that p^1,

$$\frac{\partial}{\partial p^2} \left[\frac{\partial}{\partial p^1} F^1(p^1, p^2) \right] = 0 \tag{6.22}$$

or, changing the order the differentiation,

$$\frac{\partial}{\partial p^1} \left[\frac{\partial}{\partial p^2} F^1(p^1, p^2) \right] = 0 \tag{6.23}$$

or,

$$\frac{\partial}{\partial p^1}\left[\frac{\partial}{\partial p^2}\left\{f_1(p^1) + \alpha f_{21}(p^1, p^2)\right\}\right] = 0 \tag{6.24}$$

or, noting that $p^1 = L_1(p^2)$,

$$\frac{\partial}{\partial p^1}\left[\frac{\partial}{\partial p^1}f_1(p^1)\frac{\partial p^1}{\partial p^2} \;+\; \alpha\frac{\partial}{\partial p^1}f_{21}(p^1, p^2)\frac{\partial p^1}{\partial p^2} \;+\; \alpha\frac{\partial}{\partial p^2}f_{21}(p^1, p^2)\right] = 0 \tag{6.25}$$

or,

$$\frac{\partial}{\partial p^1}\left[\frac{\partial}{\partial p^1}f_1(p^1)\,\dot{L_1}(p^2) \;+\; \alpha\frac{\partial}{\partial p^1}f_{21}(p^1, p^2)\,\dot{L_1}(p^2) \;+\; \alpha\frac{\partial}{\partial p^2}f_{21}(p^1, p^2)\right] = 0 \tag{6.26}$$

or,

$$\left[\frac{\partial^2}{\partial p^1 \partial p^1}f_1(p^1)\dot{L_1}(p^2) + \alpha\frac{\partial^2}{\partial p^1 \partial p^1}f_{21}(p^1, p^2)\dot{L_1}(p^2) + \alpha\frac{\partial^2}{\partial p^1 \partial p^2}f_{21}(p^1, p^2)\right] \;=\; 0 \tag{6.27}$$

Thus,

$$\begin{aligned} \dot{L_1}(p^2) &= -\left(\frac{\partial^2}{\partial p^1 \partial p^1}f_1(p^1) + \alpha\frac{\partial^2}{\partial p^1 \partial p^1}f_{21}(p^1, p^2)\right)^{-1} \alpha\left(\frac{\partial^2}{\partial p^1 \partial p^2}f_{21}(p^1, p^2)\right) \\ &= -\left(\alpha^{-1}\frac{\partial^2}{\partial p^1 \partial p^1}f_1(p^1) + \frac{\partial^2}{\partial p^1 \partial p^1}f_{21}(p^1, p^2)\right)^{-1} \left(\frac{\partial^2}{\partial p^1 \partial p^2}f_{21}(p^1, p^2)\right) \end{aligned} \tag{6.28}$$

Similarly, it can be shown,

$$\dot{L_2}(p^1) = -\left(\beta^{-1}\frac{\partial^2}{\partial p^2 \partial p^2} f_2(p^2) + \frac{\partial^2}{\partial p^2 \partial p^2} f_{12}(p^1, p^2)\right)^{-1} \left(\frac{\partial^2}{\partial p^2 \partial p^1} f_{12}(p^1, p^2)\right) \tag{6.29}$$

Substituting the above in the expression for T, we get,

$$\begin{aligned} \dot{T}(p^1) &= \dot{L_1}(p^2)\dot{L_2}(\bar{p}^1) && (6.30)\\ &= \left[\left(\alpha^{-1}\frac{\partial^2}{\partial p^1 \partial p^1} f_1(p^1) + \frac{\partial^2}{\partial p^1 \partial p^1} f_{21}(p^1, p^2)\right)^{-1} \left(\frac{\partial^2}{\partial p^1 \partial p^2} f_{21}(p^1, p^2)\right)\right] \\ &\quad \left[\left(\beta^{-1}\frac{\partial^2}{\partial p^2 \partial p^2} f_2(p^2) + \frac{\partial^2}{\partial p^2 \partial p^2} f_{12}(p^1, p^2)\right)^{-1} \left(\frac{\partial^2}{\partial p^2 \partial p^1} f_{12}(p^1, p^2)\right)\right] && (6.31) \end{aligned}$$

Now, by the use of the condition stated in (6.9) we get,

$$\left|\left|\dot{T}(p^1)\right|\right| < 1 \tag{6.32}$$

Thus for any p_1^1 and $p_2^1 \in U^1 \subseteq P^1$, using the Mean Value Theorem, we get

$$\begin{aligned} \left|\left|T(p_1^1) - T(p_2^1)\right|\right| &\leq \left|\left|p_1^1 - p_2^1\right|\right| \times \sup_{0<\theta<1} \left|\left|\dot{T}(p_2^1 + \theta(p_1^1 - p_2^1))\right|\right| && (6.33)\\ &< \left|\left|p_1^1 - p_2^1\right|\right| && (6.34) \end{aligned}$$

The second line in the above equation is obtained via the use of equation (6.32). This proves that

$$\left|\left|L_1L_2(p_1^1) - L_1L_2(p_2^1)\right|\right| < \left|\left|p_2^1 - p_2^1\right|\right| \tag{6.35}$$

Hence, by the above equation, we have shown that the operator

$$L_1L_2 \;\; : U^1 \; \longrightarrow U^1 \; \subseteq \; P^1 \tag{6.36}$$

is a contraction mapping.

Similarly, it can be shown that, the operator,

$$L_2L_1 \;\; : U^2 \; \longrightarrow U^2 \; \subseteq \; P^2 \tag{6.37}$$

is also a contraction mapping.

This proves that the parallel decision making model converges. The convergence is achieved by minimizing at each step the cost function of each one of the players keeping the other's output fixed. Consequently, the convergence point is by definition also the Nash Equilibrium. Further, it also shows that for problems having a cost function structure defined in equation (6.8) Nash equilibrium exists, provided the inequality constraint given in the theorem is satisfied. Consequently, it proves that an appropriate choice of the coupling coefficients can assure the existence and stability of the Nash equilibrium. However, we note that since it is an inequality constraint, it need not be an unique value. Further, we can also make some simplification to the requirement of (6.9) to shed more light on the role of the constants α and β. To do that, we proceed as follows:

Let $A = \frac{\partial^2}{\partial p^1 \partial p^1} f_1(p^1)$, $B = \frac{\partial^2}{\partial p^1 \partial p^1} f_{21}(p^1, p^2)$, $C = \frac{\partial^2}{\partial p^1 \partial p^2} f_{21}(p^1, p^2)$, $D = \frac{\partial^2}{\partial p^2 \partial p^2} f_2(p^2)$, $E = \frac{\partial^2}{\partial p^2 \partial p^2} f_{12}(p^1, p^2)$ and $G = \frac{\partial^2}{\partial p^2 \partial p^1} f_{12}(p^1, p^2)$. Then the condition in (6.9) can be restated as

$$||[(\alpha^{-1}A + B)^{-1}C][(\beta^{-1}D + E)^{-1}G]|| < 1 \tag{6.38}$$

If we assume that the coefficients α and β are small, and in particular if the norm of $\alpha A^{-1}B$ and $\beta D^{-1}G$ are small compared to the identity matrix (meaning that the objective function of the two modules are dominated by their own terms and not by that of the influence from the other module), then we have

$$||\alpha\beta|| \, ||A^{-1}CD^{-1}G|| < 1 \tag{6.39}$$

Since we want the above to be true for all values in the neighborhood i.e. $\forall(p^1, p^2) \in U^1 \times U^2 \subseteq P^1 \times P^2$, let

$$\gamma = \sup_{(p^1, p^2) \in U^1 \times U^2} ||A^{-1}CD^{-1}G|| \tag{6.40}$$

Hence, an approximate relation between α and β is given by

$$||\alpha\beta|| < \gamma^{-1} \tag{6.41}$$

The above suggests a relationship that the two coupling coefficients should follow to guarantee the existence of Nash Equilibria. One way to choose appropriate values for α and β would be to fix either one of them to a reasonable value and then let the constraint

(6.41) dictate the selection of the other one. Note that the above also assumes that A^{-1} and D^{-1} exists, which means that the modules by themselves are also convex in the vicinity of the Nash Equilibria.

6.4 Interacting Modules

As already mentioned, the modules that we consider here are the two types of segmentation procedure (see Figure 6.2), one region-based and the other boundary based. Even though the method can be extended to work for other more general image models, for simplicity of analysis, we shall assume that the image can be modeled as

$$y(i,j) = x(i,j) + n(i,j); \quad 1 \le i \le M; \quad 1 \le j \le N \tag{6.42}$$

where $n(i,j)$ corresponds to additive white Gaussian noise (AWGN). Now, as for $x(i,j)$, we make the further assumption that it can be modeled as a collection of homogeneous regions of uniform or slowly varying intensities. As noted in the previous chapters, this assumption merely states that the intra-region variability is smaller than the inter-region variability, something that can be safely assumed to be true for many image segmentation applications. Here, we describe mathematically the two interacting modules. Without any interaction, the modules have already been discussed in Chapter 2. The interaction is represented by an extra term in the cost function and we shall show that the form of the cost functions will represent that of the parallel decision theoretic model that we had described earlier in (6.8).

Both the modules will be posed in a Bayesian framework for reasons mentioned in section 6.1 of this chapter.

6.4.1 Region based segmentation influenced by boundary finding

We have already described the region-based segmentation method. Initially the requirement was to develop a region module given the raw image and the prior knowledge which was modeled using a MRF which emphasized smoothness. Now, besides the input image and the smoothness constraint, we also have as a prior, the output of the boundary finder, the parameterized contour $\vec{p}$. Thus the aim is to find $\arg\max_{\mathbf{x}} P(\mathbf{x}|Y, \vec{p})$, where as already defined, Y is the raw image data and $\mathbf{x}$ is the segmented image. Now,

$$\begin{aligned} P(\mathbf{x}|Y,\vec{p}) &= \frac{P(\mathbf{x},Y,\vec{p})}{P(Y,\vec{p})} \\ &\propto \frac{P(Y|\mathbf{x})P(\vec{p}|\mathbf{x})P(\mathbf{x})}{P(Y,\vec{p})} \\ &= \frac{P(\mathbf{x}|Y)P(\vec{p}|\mathbf{x})P(Y)}{P(Y,\vec{p})} \qquad (6.43) \end{aligned}$$

In the second step of the above equation, we have ignored the dependence of p on Y because p is obtained as a prior for this module and is not modified within this module. Since we are concerned with maximization w.r.t. $\mathbf{x}$, we have,

$$\arg\max_{\mathbf{x}} P(\mathbf{x}|Y,\vec{p}) \equiv \arg\max_{\mathbf{x}} P(\mathbf{x}|Y)P(\vec{p}|\mathbf{x}) \qquad (6.44)$$

The first term in the above equation is derived from the image data and the previous output of this module, whereas the second term represents the influence of the previous output of the boundary finding module. Now, taking the logarithm in the previous equation, we get,

$$\arg\max_{\mathbf{x}} lnP(\mathbf{x}|Y,\vec{p}) \equiv \arg\max_{\mathbf{x}} lnP(\mathbf{x}|Y) \ + \ \arg\max_{\mathbf{x}} lnP(\vec{p}|\mathbf{x}) \qquad (6.45)$$

As we can recognize, the first term in the above equation (6.45) is the same as that in the stand alone module in (3.31). The second term,however corresponds to the information derived from the boundary finding module.

Even though this procedure can be extended to images with more than two regions, for the time being, we shall continue with the assumption that there are only two regions, one of which represents the target object. Thus the contour vector $\vec{p}$ represents the boundary of that object. Now, let $A_{\vec{p}}$ correspond to the points that lie inside the contour and $\bar{A}_{\vec{p}}$ those points that lie on the background. Thus, $A_{\vec{p}} \bigcup \bar{A}_{\vec{p}} = \{(i,j);\ 1 \leq i \leq M,\ 1 \leq j \leq N\}$ ie. the whole image consists of those two parts only.

A very simple model to represent this information is thus given by,

$$lnP(p|\mathbf{x}) \propto \sum_{(i,j)\in A_{\vec{p}}} (x_{i,j} - u_{i,j})^2 + \sum_{(i,j)\in \bar{A}_{\vec{p}}} (x_{i,j} - v_{i,j})^2 \qquad (6.46)$$

where $u_{i,j}$ represents the information regarding the intensity of those points that lies inside the contour and $v_{i,j}$ of those that lie outside. For most simple images with uniform regions, they could actually represent just the expected value of the pixels at those points, something that can easily be computed *a priori* [83]. We explained this in chapter 3. If we assume that the image can be segmented to simple uniform regions, then of course we simply have $u = u_{i,j} = \sum_{(i,j)\in A_{\vec{p}}} x_{i,j}$ and $v = v_{i,j} = \sum_{(i,j)\in \bar{A}_{\vec{p}}} x_{i,j}$. However, more complex modeling of the intensity information, (e.g. in the case of a texture image) can also be done. Thus, finally the region-based module including the interacting term is mathematically equivalent to the minimization of the following:

$$\min_{\mathbf{x}} E \;=\; \min_{\mathbf{x}} \left[\sum_{i,j} [y_{i,j} - x_{i,j}]^2 + \lambda^2 \left(\sum_{i,j} (x_{i,j} - x_{i-1,j})^2 + \sum_{i,j} (x_{i,j} - x_{i,j+1})^2 \right) \right]$$

$$\begin{aligned} &+ \quad \alpha \left[\sum_{(i,j)\in A_p} (x_{i,j} - u_{i,j})^2 + \sum_{(i,j)\in \bar{A_p}} (x_{i,j} - v_{i,j})^2 \right] \qquad (6.47) \\ &= \quad \boxed{\min_{\mathbf{x}} [f_1(\mathbf{x}) + \alpha f_{21}(\mathbf{x}, p)]} \end{aligned}$$

We note that the above poses the problem in the format of the parallel decision making structure of (6.8) as shown in equation (6.47), with the first term of the objective function $f_1(\mathbf{x})$ representing the contribution of the region-based module by itself (which is identical to the expression in (3.34) and the other term $f_{21}(\mathbf{x}, \vec{p})$ representing the interaction term which uses the latest available output $\vec{p}$ of the boundary module (via the definitions $u_{i,j}$ and $v_{i,j}$.

6.4.2 Boundary finding influenced by region-based segmentation

The aim of this module is to estimate the boundary using besides gradient information, region information as well. Once again, the entire problem will be posed in a Bayesian framework (for reasons already mentioned in section 6.1) as we shall shortly see. Thus, we want to maximize $P(\vec{p}|I_g, I_s)$, where as described in the previous section, $\vec{p}$ is the vector of parameters used to parameterize the contour and I_g represents the gradient image and I_s (obtained from $\mathbf{x}$ of the region module) is the region segmented image.

We shall not describe mathematically this module as it has already been discussed in chapter 4. The relevant equation (4.12) that we carry over from that development is:

$$\begin{aligned} \max_{\vec{p}} M(\vec{p}, I_g, I_s) \quad &= \quad \max_{\vec{p}} [M_{prior}(\vec{p}) \\ &+ \quad \int_{C_{\vec{p}}} (K_1 I_g(x,y) \\ &+ \quad K_2\{N_s(x,y)\frac{\partial x}{\partial t} + M_s(x,y)\frac{\partial y}{\partial t}\})dt] \qquad (6.48) \end{aligned}$$

We now rewrite this equation as follows:

$$\begin{aligned}
\max_{\vec{p}} M(I_g, I_s, \vec{p}) &= \max_{\vec{p}} [M_{prior}(\vec{p}) \\
&+ \int_{C_{\vec{p}}} I_g(x(\vec{p}, t), y(\vec{p}, t))dt\} \\
&+ \beta \int_{C_{\vec{p}}} \{N_s(x, y)\, \frac{\partial x}{\partial t} + M_s(x, y)\, \frac{\partial y}{\partial t}\}\, dt] \qquad (6.49) \\
&= \boxed{\max_{\vec{p}}[f_2(\vec{p}) + \beta f_{12}(I_s, \vec{p})]}
\end{aligned}$$

Here, K_1, K_2 and β are constants. We first fix $K_1 = 1$ and then we have $\beta = K_2$. The functions $N_s()$ and $M_s()$ has been described in equations (4.10) and (4.11). As before, the intuition behind using $f_{12}(.)$ is that if we could identify the target region by positive values, and the other by negative values, the above area integral is maximum when the boundary correctly encircles the target region. For the simple two region problem, after the segmentation has been done, we could linearly translate the intensity values to satisfy the above requirement. Here, that would just mean subtracting out the mean of the means of the two characteristic regions. In other words, for this case, $I_s(i, j) = x(i, j) - m$, where (i, j) represents the pixel location and m is the mean of the means of the two regions. For a more sophisticated image model a better representation for m could be used.

We note that the construct of the objective function is once again consistent with the game theoretic framework of (6.8) as shown in equation (6.49) where $f_2(p)$ is the contribution of the gradient-based boundary finder and $f_{12}(I_s, p)$ is the interaction term that uses the latest available output from the region module. This along with the theorem stated and proved in section 6.3.1 assures us that a right choice of coupling coefficients will allow the problem to converge to a Nash equilibrium.

6.5 Results

Experiments were carried out both with synthetic and real biomedical images to verify the performance of game theoretic integration.

6.5.1 Synthetic Images

The first part of the experiments involved synthetic images with known amounts of noise and with objects of known shape.

Comparisons were made first between the outputs generated using game theoretic integration and the corresponding outputs obtained without using information fusion. The equations (6.47) and (6.49) jointly describe the outputs of the game-theoretic integration where equation (6.47) gives the region output and equation (6.49) the boundary output under the integrated framework. When the coupling coefficients α and β are set to zero, we have the stand alone modules.

In the first experiment, we created a synthetic image that has one target object in the center surrounded by an uniform background. We used the same template that was used in the previous chapters. Figure 6.3(a) represents a noisy (with added Gaussian noise) version of the above mentioned image. The signal to noise ratio (SNR) for this image is 0.67. For this image we applied the region-based method and the gradient-based boundary finding without information sharing as described before and also the game-theoretic procedure with information sharing as has already been discussed. Figure 6.3(b) shows the output of the region-based method when it is used by itself. Figure 6.3(c) represents the output of the gradient-based boundary finder. For this figure and in Figure 6.3(e), the darker contour represents the initialization and the brighter contour the final boundary found. Figure 6.3(d) represents the output of the region-based segmentation

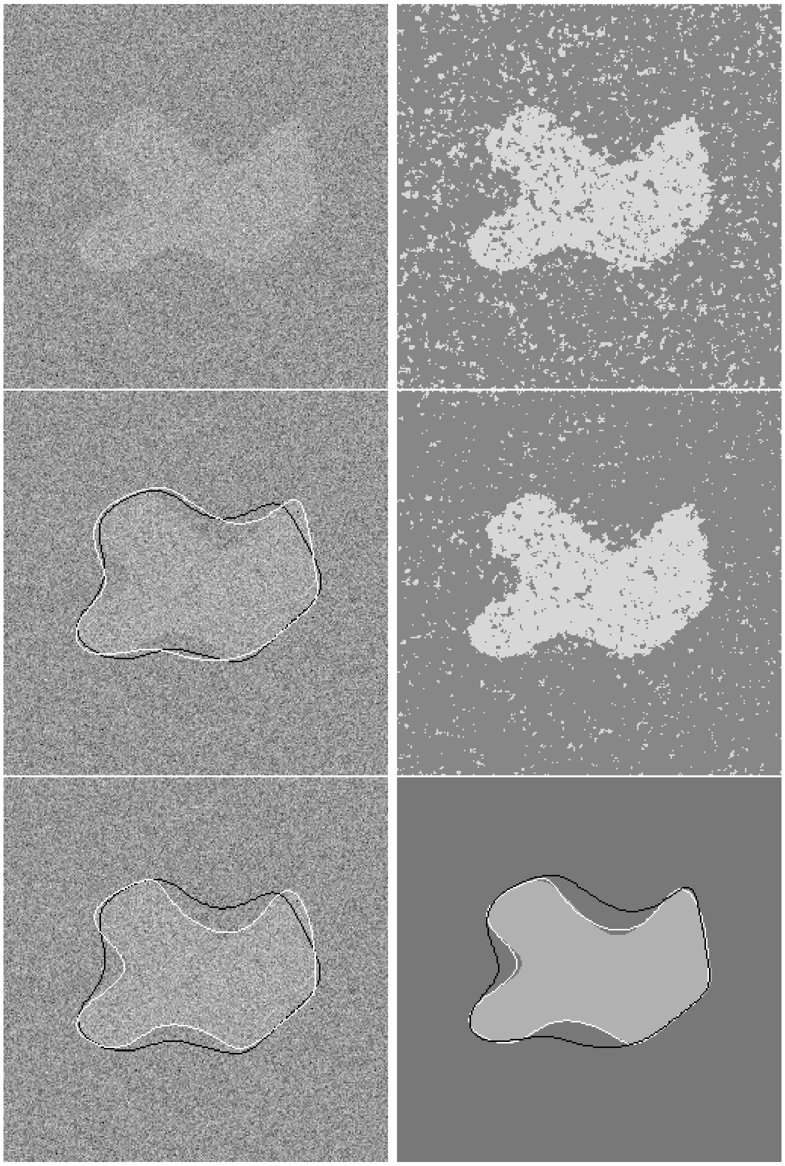


Figure 6.3: Example with a synthetic image to show the effects of game theoretic integration. (a)Top,Left: Original image with $SNR = 0.67$. (b)Top,Right: Output of region-based segmentation without using game-theoretic integration. (c)Middle,Left: Output of boundary finding without using game-theory; The dark contour represents the initialization, and the brighter one is the final output. (d)Middle,Right: Output of region-based segmentation using game-theoretic integration. (e)Bottom,Left: Output of boundary finding using game-theory; The dark contour represents the initialization, and the brighter one is the final output. (f)Bottom,Right: The final boundary outputs generated by the two methods overlayed on top of the template. The dark contour represents the output of the gradient-based boundary-finder without using game-theory, and the brighter one is the final output using game-theoretic integration.

module under the game theoretic framework. Figure 6.3(e) on the other hand, corresponds to the output of the boundary finding module under the game theoretic framework. As we would expect, the output in Figure 6.3(d) is much better than that in Figure 6.3(b) where there is no information fusion. Similarly, the final contour output in Figure 6.3(e) with information fusion is much better than that in Figure 6.3(c) without integration. In Figure 6.3(f) we overlay the boundary outputs on top of the true image so that the difference in performance can easily be compared. The darker contour represents the output without information fusion and the brighter one with it. The output in Figure 6.3(c) using just gradient-based boundary finding shows divergence.

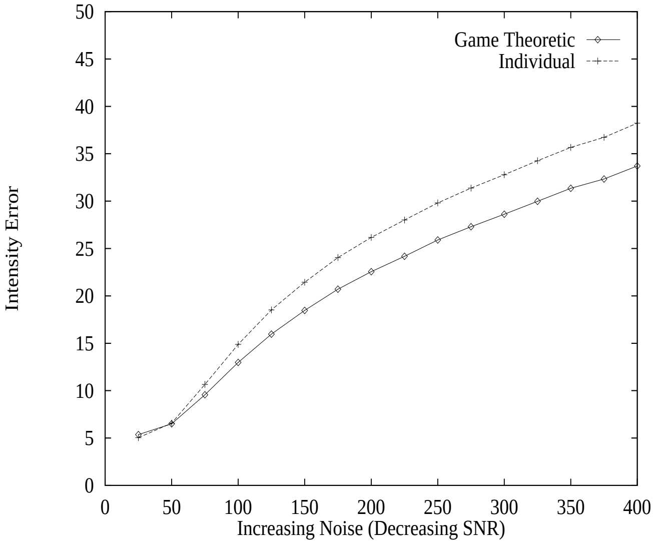


Figure 6.4: Comparison of region module with and without game-theoretic information sharing under increasing noise conditions. Clearly, the error is larger without information sharing than what we get using game-theoretic integration.

Next, we try to analyze the effects of integration for both the modules under

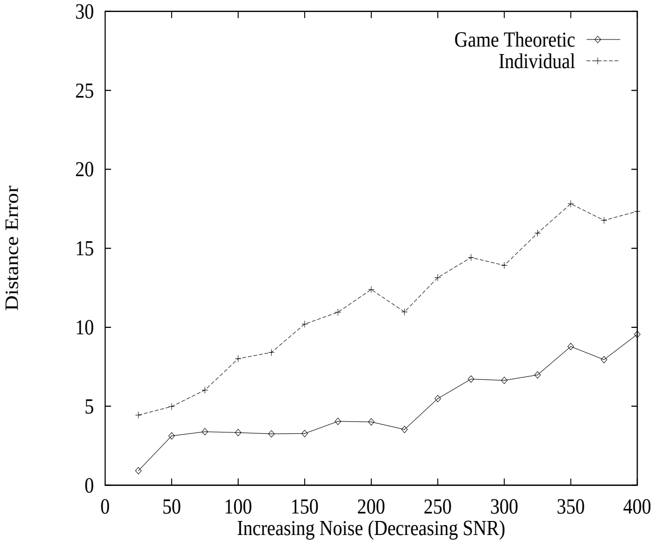


Figure 6.5: Comparison of boundary finding with and without game-theoretic information sharing. Clearly, the error is larger without information sharing than what we get using game-theoretic integration.

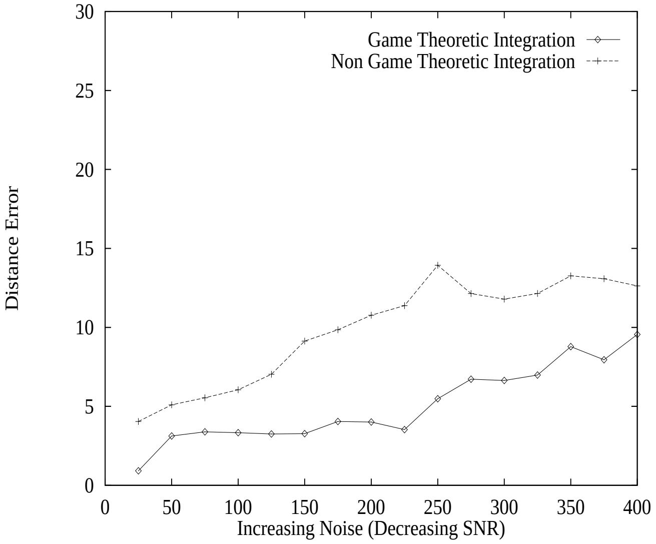


Figure 6.6: Plot showing performance of boundary module with and without region update. Clearly, the performance is better when we use game-theoretic integration where the region module is also updated.

increasing amounts of noise. In Figure 6.4 we compare the output of stand alone region-based segmentation ($\alpha = 0$) with the same output obtained using game-theoretic integration ($\alpha \neq 0$) as in equation (6.47).

In Figure 6.5 we compare the output of stand alone gradient-based boundary finder ($\beta = 0$) with the same output obtained using game-theoretic integration ($\beta \neq 0$) as in equation (6.49). We use the same template as in Figure 6.3. The contrast difference between the background and the target is 200.0 and the x-axis in the Figures 6.4 and 6.5 correspond to the standard deviation σ for the white noise added. Both the experiments had noise varying from $\sigma = 25$ to $\sigma = 400$. In other words, the SNR varies from 8.0 to 0.5. As for the y-axis, in Figure 6.4, it represents the average per-pixel intensity error, and in Figure 6.5, it represents a scaled version of the average Euclidean distance between the true contour and the final one obtained. In the last chapter we described the details of this. For every point in the above two plots, the experiments were repeated ten times with the same initialization. One needs to do this because even though the noise distribution remains the same, the pixel values of the noise could be different. Clearly, the results are superior for both the modules when there is game-theoretic information fusion. We also compared the game-theoretic output with integration achieved in the manner described in Chapter 4, where the boundary module used the information from the region module, but the region module wasn't updated. We argued that this feature integration framework can be considered to be a special case of the more general module integration framework considered here. The result of this comparison is shown in Figure 6.6. As expected, the result of the game-theoretic method is superior to the integrated method without region update.

The next part of the experiments using synthetic images involved comparisons of the game-theoretic method with a single objective function approach that simply adds together the objective functions involving the region and the boundary processes. As in [18, 17], a single objective function involving both the region and boundary process was created by merely adding the functions in the equations (6.47) and (6.49). Optimization was achieved using gradient descent. Figure 6.7 shows the boundary output for the same image and the same SNR=0.67. We can see that the game theoretic output is superior to the output obtained using the single objective function approach.

Figure 6.8 shows a noise comparison of the game-theoretic approach with the single objective function method. Once again, we use the same template as in Figure 6.3. The contrast difference between the background and the target is 200.0 and the x-axis in figure 6.8 correspond to the standard deviation σ for the white noise added which varies from $\sigma = 25$ to $\sigma = 400$, i.e. the SNR varies from 8.0 to 0.5. Clearly, the game-theoretic method produces better results. Further comparing the figures 6.7 and 6.8 with the corresponding figures 6.3 and 6.5 reveal that the result of the single objective approach is not significantly better than the individual modules, primarily due to the non-optimal combination to form the single objective function in the way described before. However this does not mean that an optimal solution is not possible. But what it does indicate is that integration using a single objective function is much more complicated, at least in the present case. We note that this is similar to the conclusion achieved in [18]. The biggest practical advantage of the game-theoretic approach is that not only does it produce superior results, the computational costs are also much lighter as compared to the single-objective approach. We further did an analysis to see how sensitive the outputs of the two integration methods were to the weighting coefficients. The same template and the same initialization as in the previous experiments were chosen. Once

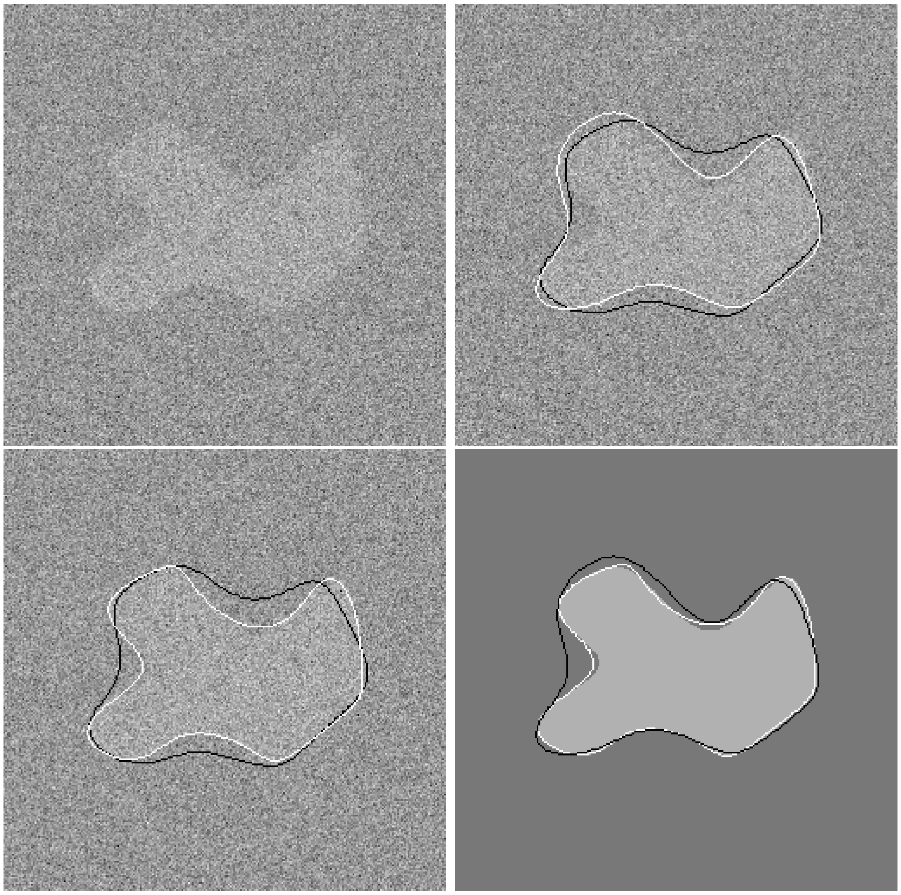


Figure 6.7: Comparison of game theoretic integration with the single objective function approach. (a)Top,Left: Original image with $SNR = 0.67$. (b)Top,Right: Output using a single-objective function approach. The dark contour represents the initialization, and the brighter one is the final output. (c)Bottom,Left: Output using game-theory; The dark contour represents the initialization which is the same as in (b), and the brighter one is the final output. (d)Bottom,Right: The final boundary outputs generated by the two methods overlayed on top of the template. The dark contour represents the output of the single-objective function approach and the brighter contour represents the game theoretic output.

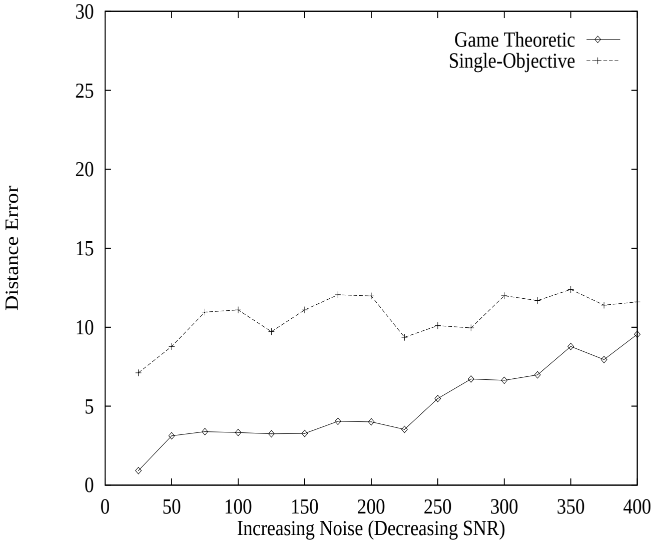


Figure 6.8: Comparison of the game-theoretic approach with the single-objective function approach under varying amounts of noise.

again an $SNR = 0.67$ was selected. First, the value of α was varied, while the value of β was held constant at $\beta = 0.4$. The result is shown in Figure 6.9(a). Next, the value of β was varied, while the value of α was held constant at $\alpha = 0.2$. The result is shown in Figure 6.9(b). While both the methods are somewhat sensitive, the game theoretic method generally provides the better result. With low values of α and β, the difference between the two methods is smaller. This is to be expected because then the coupling is really low. However, for larger values of α and β, the game-theoretic method deteriorates, probably because of the non-existence of Nash equilibrium resulting in the fact that if one of the modules make a wrong move, both the modules are effected by it considerably.

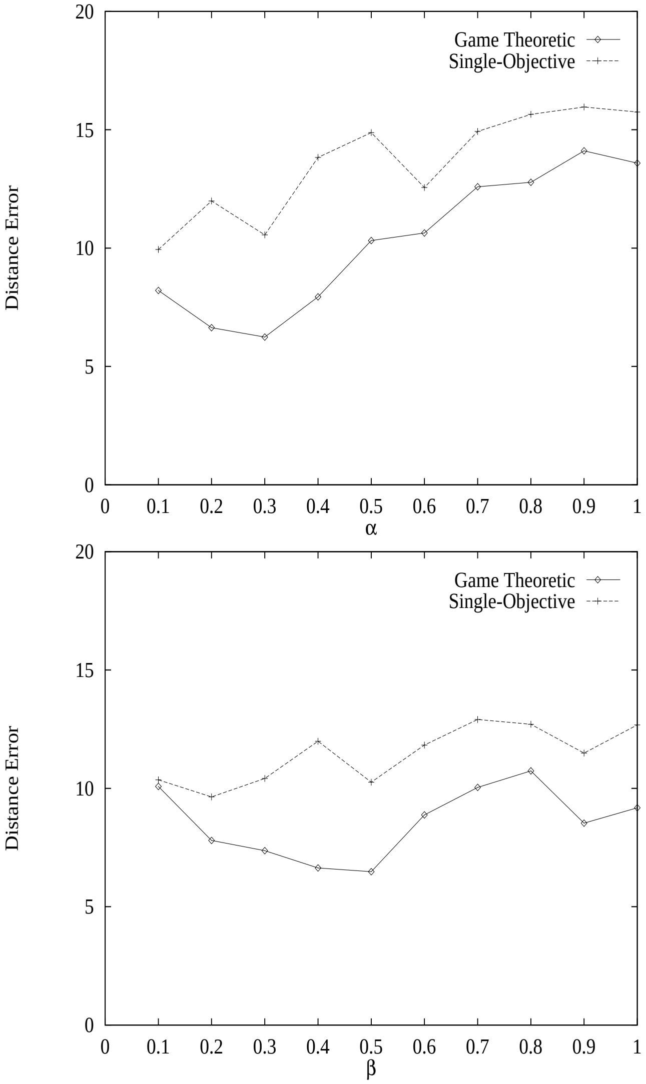


Figure 6.9: Sensitivity comparison of the game-theoretic approach with the single-objective function approach. The y-axis gives the distance error between the true contour position and the final output found as the coupling coefficients were varied. (a)Top: Plot showing the variation with α as β was held fixed at $\beta = 0.4$. (b)Bottom: Plot showing the variation with β as α was held fixed at $\alpha = 0.2$.

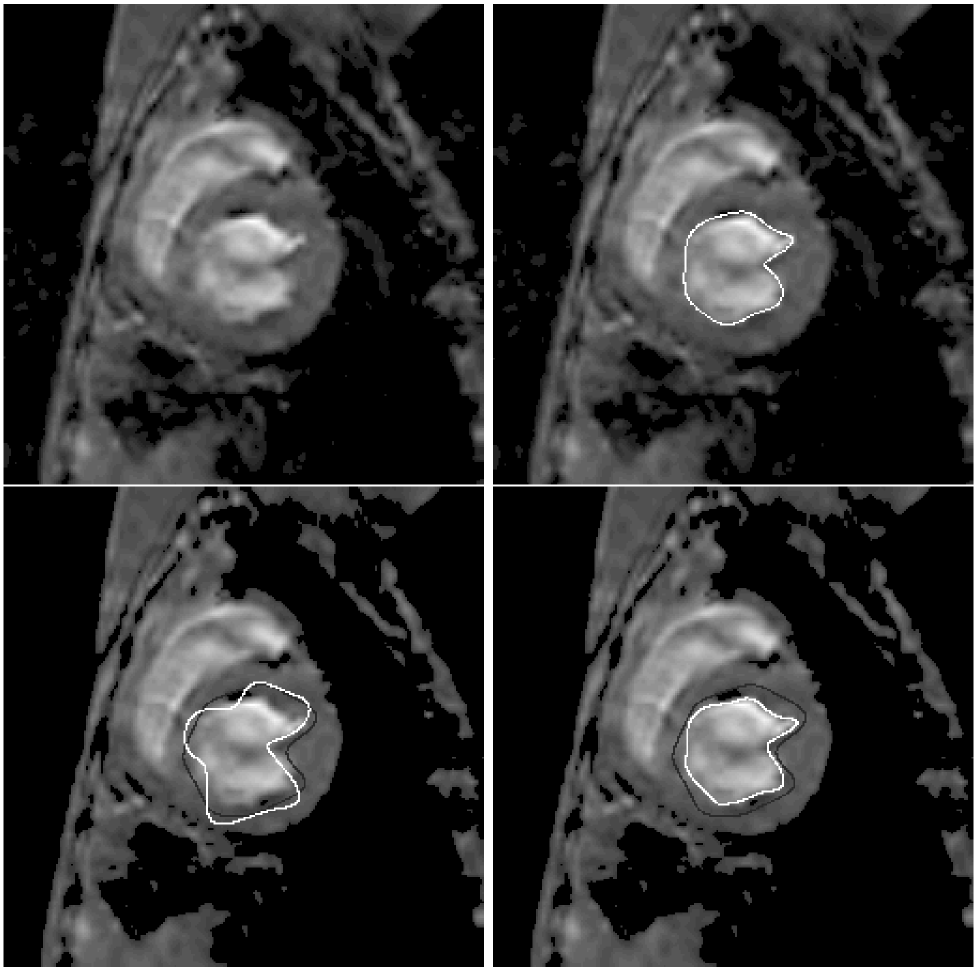


Figure 6.10: Example with a short axis MR image of a canine heart. (a)Top,Left: Original image; (b)Top,Right: Boundary of the endocardium hand-traced by a domain expert; (c)Bottom,Left: Output of gradient-based boundary finding without using game theoretic integration. The dark contour represents the initialization, and the brighter one is the final output. (d)Bottom,Right: Output of gradient-based boundary finding using game theoretic integration. The dark contour represents the initialization, and the brighter one is the final output.

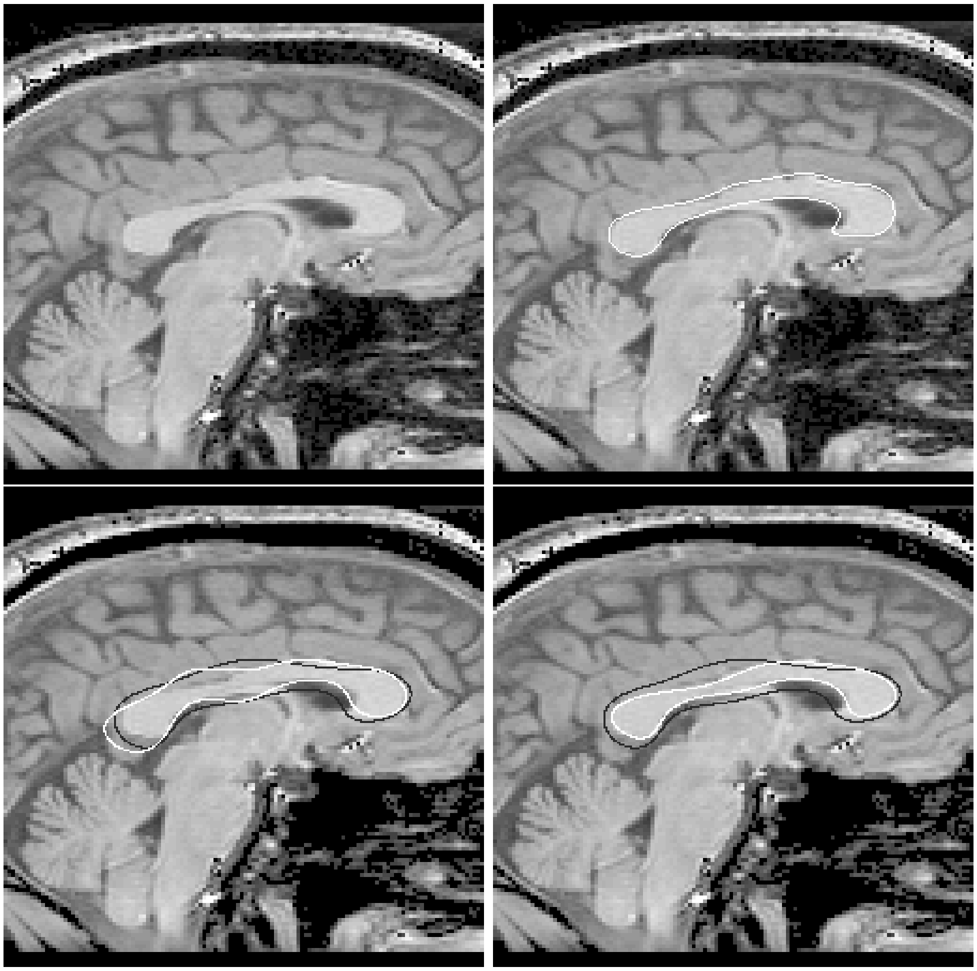


Figure 6.11: Example with an MR image of the brain. (a)Top,Left: Original image; (b)Top,Right: Boundary of the corpus callosum hand-traced by a domain expert; (c)Bottom,Left: Output of gradient-based boundary finding without using game theoretic integration. The dark contour represents the initialization, and the brighter one is the final output. (d)Bottom,Right: Output of gradient-based boundary finding using game theoretic integration. The dark contour represents the initialization, and the brighter one is the final output.

6.5.2 Real Images

The next part of the experiment utilizes clinical images. Figure 6.10(a) shows the short axis MR image of a canine heart taken from one of our clinical studies, the task being to be able to segment out the endocardium. It is one of the more difficult images of its type. Figure 6.10(b) shows the hand traced result by a domain expert. It is used for the sake of comparison. Figure 6.10(c) shows the result of the stand-alone boundary finder. In Figure 6.10(d), we show the output of the boundary finder under game theoretic integration with the same initialization. The superiority of the final output in this case is obvious. Both in Figures 6.10(c) and 6.10(d), the darker contour represents the initialization, and the brighter contour represents the final output. The reason for the failure of the stand-alone boundary finder is a combination of fuzzy edges and poor initialization.

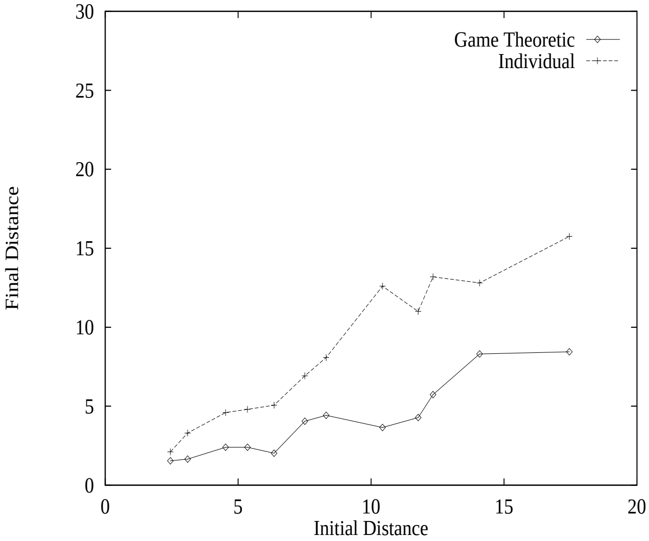


Figure 6.12: Plot showing performance under different initializations for a heart image

To further investigate this, we compared the outputs of the two methods (i.e. the stand-alone boundary finder and the game-theoretic method) under different initializations against the expert-driven boundary and the result is shown in Figure 6.12. As we can see, there is almost no difference in the performance between the two methods when the initialization is close to the true contour. But as the initialization moves further away, the integrated method outperforms the stand-alone module.

Next, we investigate the situation with a MR image of the brain shown in Figure 6.11(a), where the aim is to segment the corpus callosum. Figures 6.11(c) and 6.11(d) represents the output using just a gradient-based boundary finder and the game-theoretic boundary finder respectively.

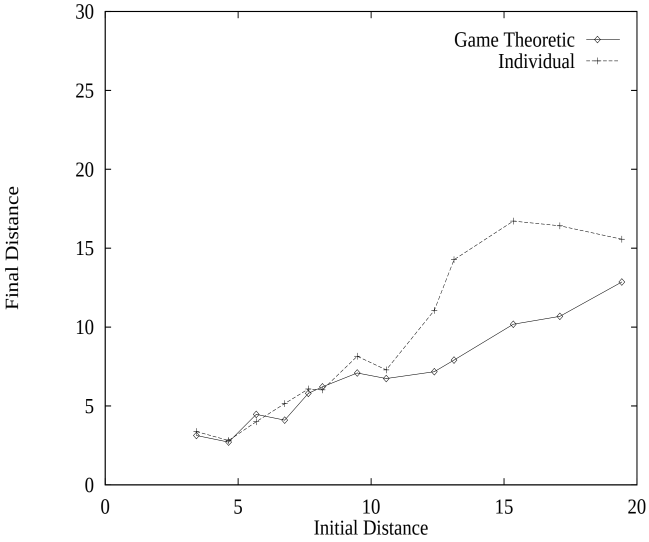


Figure 6.13: Plot showing performance under different initializations for a brain image

As in the previous case, we compared the outputs under different initializations against the expert-driven boundary and the result is shown in Figure 6.13. Once again, the performance of the two methods are almost identical when the initialization is close to the true contour. But as the initialization moves further away, the integrated method outperforms the stand-alone module.

While in the heart image, noisy and fuzzy edges coupled with poor initialization seem to be the problem for the stand-alone module, in the brain image the issue is primarily the poor initialization. However, the game theoretic outputs in both the cases are satisfactory. This clearly demonstrates the robustness of the game theoretic method.

Next, we show the same heart example shown previously, where our aim is to not only find out the endocardial boundary, but also to find good region estimates for the blood pool, muscles etc. Since it is not possible to extract all the small substructures via boundary finding, region information is considered important. However, there is a considerable overlap between the intensity levels resulting in erroneous results if only region-based segmentation without using integration is done. We can see the output in Figure 6.14(d). However, if we do a game-theoretic integration, the output of the region module gives a much more reliable estimate as shown in Figure 6.14(e). The better output can be explained by the fact that the algorithm uses the information that it is unlikely to find blood pools outside the two ventricles and that within each blood pool muscles cannot exist. The region process gets this information from the boundary module. Without integration, the region method has no way of using this information. Further, it also effectively isolates the regions of stray misclassified pixels of high intensity in a more effective way. Local smoothness is not as effective under such circumstances. This shows that the integrated algorithm has a natural way of handling image consistencies.

Finally, we show two examples, where once again we consider a case where the region information is of primary importance. Figure 6.15(a) shows a Nuclear Medicine Single Photon planar radioisotope image (histogram equalized) of the kidney taken from a clinical study performed to study the quantitation of renal function [139]. Because of the poor quality of the image it is very difficult to do quantification using stand-alone region-based segmentation. Results using region-based segmentation alone is shown in figure 6.15(d), which clearly is inaccurate. On the other hand, the region output of the

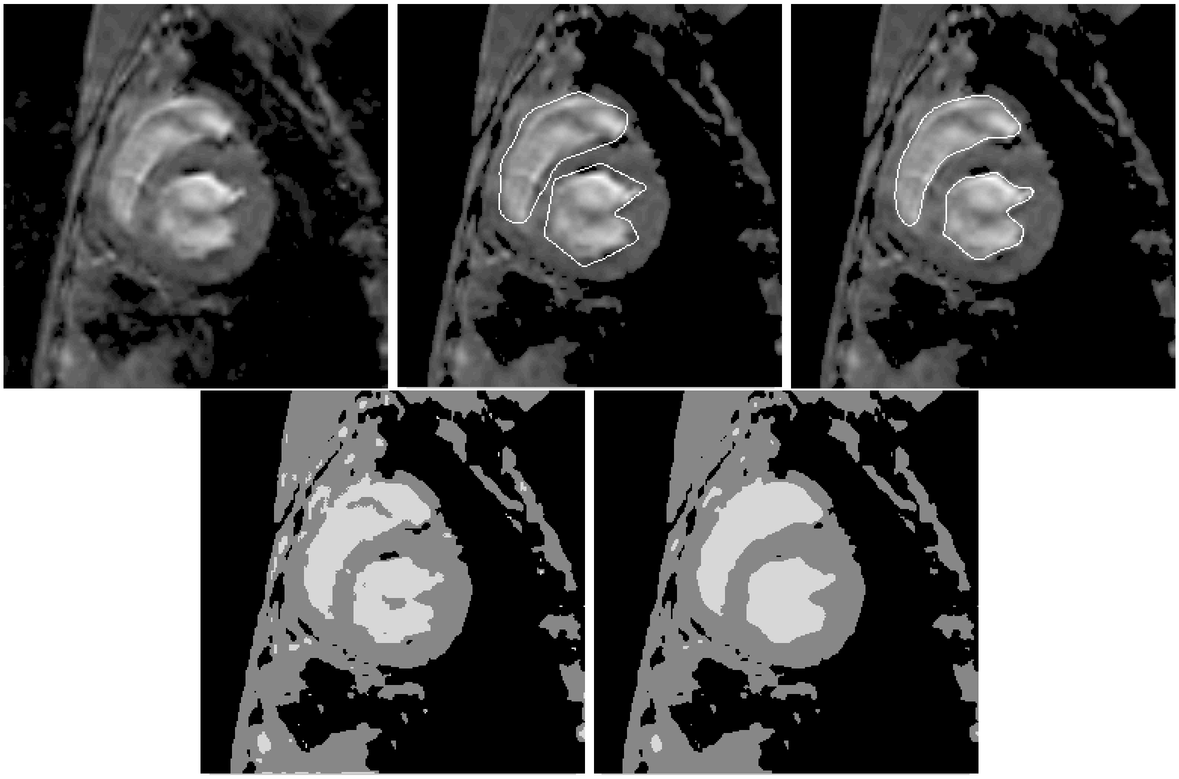


Figure 6.14: Example to demonstrate the use of region information in an MR heart image. (a)Top,Left: Original image; (b)Top,Middle: Initial boundary for the right and the left ventricle; (c)Top,Right: Final boundaries for the right and left ventricles; (d)Bottom,Left: Region-based segmentation output without integration; (e)Bottom,Right: Region-output with game-theoretic integration;

integrated game-theoretic method shown in figure 6.15(e) looks much more reasonable. Once again, the improvement in the performance can be attributed to the sharing of information that is possible through the game-theoretic method. Thus the region process can use the information that it is not looking for radioisotope activation outside the kidney. Another such example is shown in Figure 6.16 for a patient who has only one functional kidney. This example is more difficult than the other Nuclear Medicine image example. However, we note once again, that the integrated method produces a much

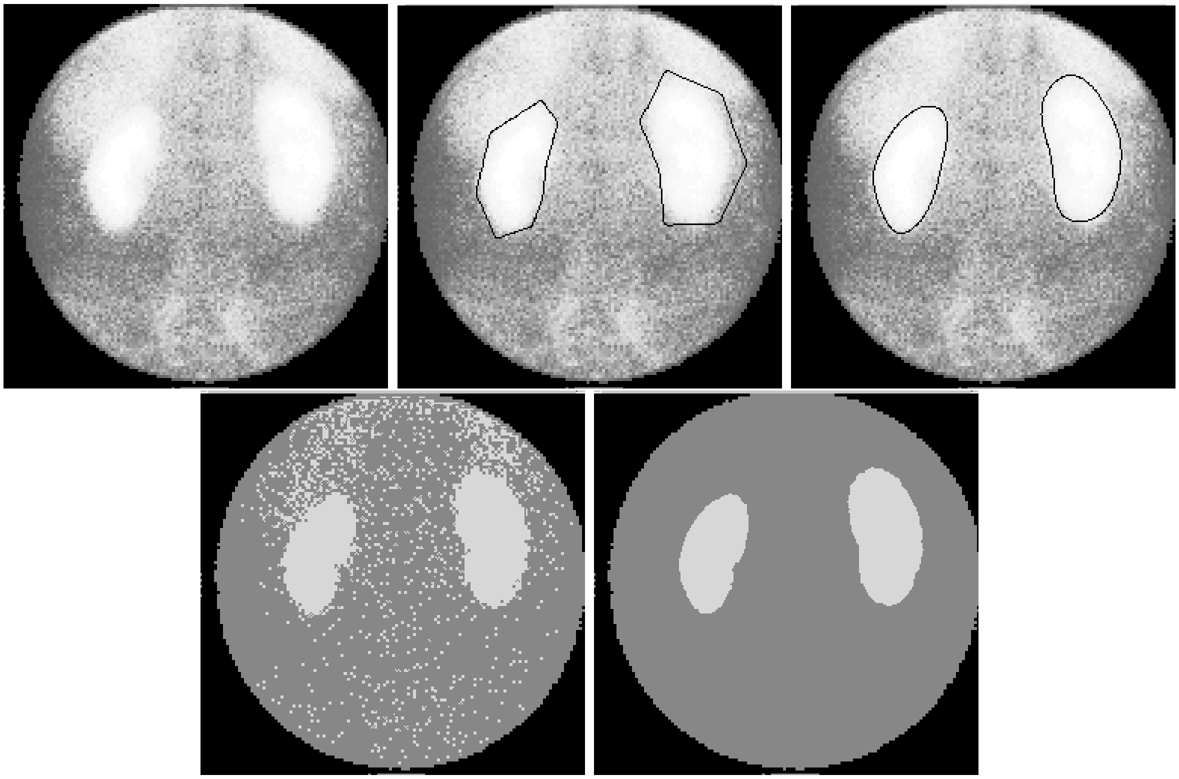


Figure 6.15: Example to demonstrate the use of region information in a nuclear medicine image. (a)Top,Left: Original image; (b)Top,Middle: Initial boundary for the two kidneys; (c)Top,Right: Final boundaries; (d)Bottom,Left: Region-based segmentation output without integration; (e)Bottom,Right: Region-output with game-theoretic integration;

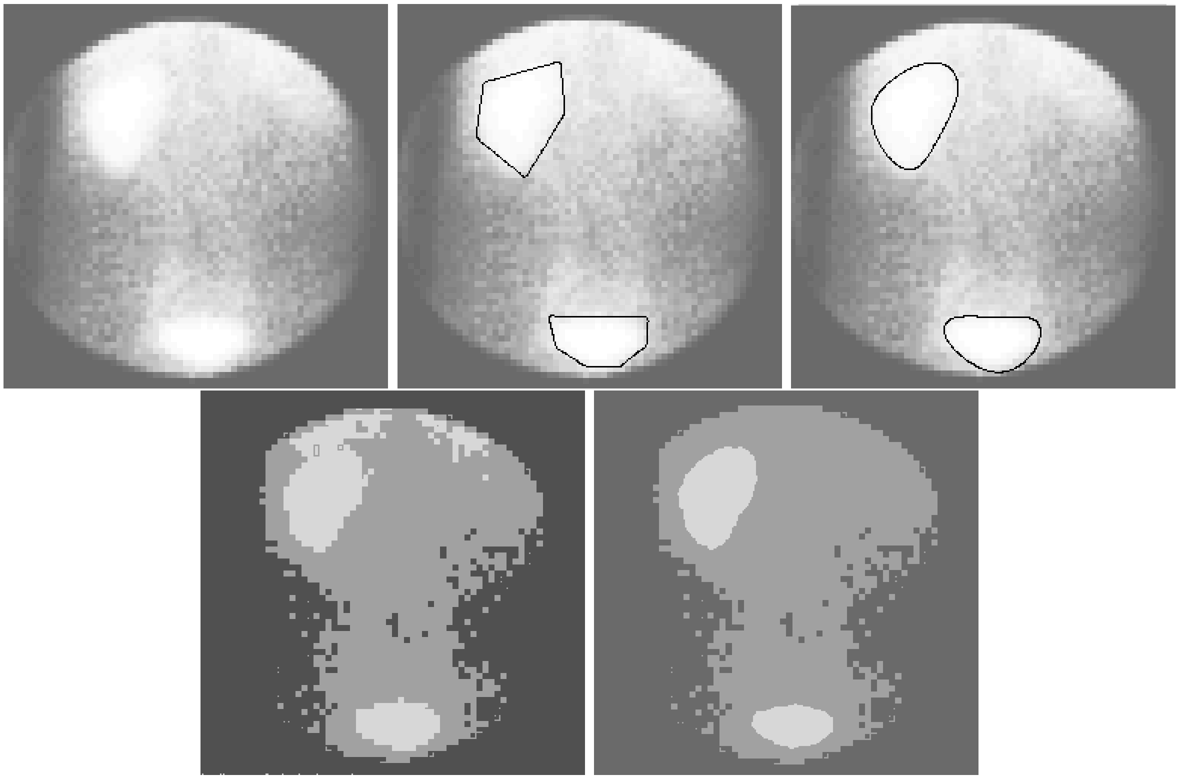


Figure 6.16: Example to demonstrate the use of region information in a second nuclear medicine image. (a)Top,Left: Original image; (b)Top,Middle: Initial boundary for the kidney and the bladder; (c)Top,Right: Final boundaries; (d)Bottom,Left: Region-based segmentation output without integration; (e)Bottom,Right: Region-output with game-theoretic integration;

better region estimate.

6.6 Discussion

6.6.1 Existence of Nash Equilibrium

In the above we described a game-theoretic method to integrate boundary finding and region-based segmentation, whereby both the modules improve their position through mutual sharing of information. We also stated and proved a theorem for the existence of Nash equilibrium for the types of objective functions that we had for the segmentation problem. It turned out that the requirements were in effect an inequality constraint on the coupling coefficients α and β in equation (6.8). The results show a definite improvement due to the integration of the modules. But direct verification of the requirements of the theorem (6.9) could be non-trivial. However, as shown in Appendix C, it is possible to argue the existence of the Nash equilibrium. There, we have shown the existence for the equivalent 1D problem and have argued extensions to 2D.

6.6.2 Convergence to Nash equilibrium

While for simple problems it might be possible to mathematically choose right values for α and β in (6.8), often for complicated problems it is almost impossible. Fortunately however, they are inequality constraints rather than equalities. Thus the exact values are not important. But even then, finding a reasonable value may not be trivial. However, for most of the cases, a reasonable guess gives satisfactory results. Further, the simplified formulation given in (6.41) gives a much simpler relation between the coupling coefficients, which could be used as well. It effectively provides an upper bound for the coupling coefficients. But then by no means it guarantees convergence to the actual desired Nash Equilibrium. Similar to the case with multiple optima, multiple Nash Equilibria might exist, and we may very well converge to the ones that are not desirable. However, since we are likely to give a reasonable initialization to the problem, we may expect the process to converge to the appropriate one.

6.6.3 Moving towards Nash Equilibrium

Once we have defined the existence of an equilibrium, or rather a point of convergence, the next issue is an algorithm to achieve it. Here, we do parallel gradient descent on the corresponding objective functions when the players have their turns. It was shown in [79, 17] that this process of decision making is acceptable and the algorithm indeed converges to the Nash equilibrium. Thus at each turn rather than reaching the optima of the cost functions, the players (modules) take steps in the right direction, i.e. go in the direction in which their cost changes in the desired manner. For the region module that means making a move to decrease the energy defined in (6.47) and for the boundary module, that means making a move to increase the objective function given in (6.49). The starting point for the boundary module is the user-provided initialization and for the region module it is the output of a simple thresholding procedure.

6.7 Conclusions

We have presented in this chapter a new technique for integrating region-based segmentation and gradient-based boundary finding using the game theoretic framework. We have posed this as a non-zero sum two-person non-cooperative game where the cost functions of both the modules are motivated using the Bayesian theory of maximization of the *a posteriori* probability. The main advantage of using the game-theoretic integration is that it can bring together the incommensurate region and boundary methods that operate in different probability spaces into a common information sharing framework of rational decision making. This leads to an improved output for both the modules without extensive computational overload. As the examples show, the integrated approach is much more robust to both noise and poor initialization. Application of this method on synthetic and natural images results in noticeable improvement as has been shown.

Chapter 7

Extensions to 3D

So far all the work that we have done, and all the examples that we have considered, involved two dimensional images. However, the ideas that we have presented in this work can be extended to three dimensional images as well. In this chapter, we extend the idea of integrating region information within the surface finding framework (the counterpart of which for 2D images was described in chapter 4) to 3D images.

7.1 Introduction

Three dimensional image analysis is important especially in the medical imaging domain due to the wide availability and use of three dimensional images from such modalities as magnetic resonance imaging (MRI), computed tomography (CT), single photon emission computed tomography (SPECT) [42]. In most of these cases, the analysis is comprised of precisely identifying and quantifying structures and abnormalities. Often, it is standard practice is to treat the 3D image as a stack of 2D images (see for example [36]), thereby reducing it essentially to a 2D image analysis problem. While successful in many cases, the problem with such methods is that they tend to either oversimplify or ignore altogether the true 3D properties of the structures under consideration. Thus, it is important to use 3D image analysis methods for these images. Our interest here lies in finding the surface boundaries of relevant structures. However, as in the 2D case, boundary finding using only local information in 3D images is problematic due to the effects of noise, poor contrast, unfavorable viewing conditions, presence of other objects in the near vicinity etc. In addition, some of the methods that are applicable in 2D can no longer be used for 3D images. For example, pixel search methods that follow an optimal path through the two dimensional images cannot naturally be extended to three dimensions because the voxels in a surface have no such ordering. Hough transform methods [5] can be used, but for three dimensional images it is very expensive both in terms of storage and computational costs. To overcome these problems, once again, as in the 2D case, the use of whole boundary methods have been advocated. This allows us to augment imperfect image data with shape information provided by a geometric model [115, 118, 112]. The main idea is to form over-constrained estimates that use a few parameters to describe a large number of points.

We adopt the Fourier parameterization of [115, 118] primarily because we consider it suitable for the class of problems that we are looking at and because of its flexibility of using prior information easily, when such information becomes available. Also, it is a natural extension to our adopted 2D boundary parameterization. Besides the Fourier representation [115, 118] that we describe below after [118] other approaches to three dimensional parametric modeling include generalized cylinders [103], superquadrics [112], hyperquadrics [72], finite element methods [32, 34, 35], etc. Some of this was discussed in the background section in chapter 2.

7.2 Fourier surface representations

A surface in three dimensions can be represented by three coordinate functions of two surface parameters as $\mathbf{x}(u, v) = (x(u, v), y(u, v), z(u, v))$, where u and v are the free parameters that very over the surface. Since there are two free parameters, a function of two parameters is necessary to describe a surface. The Fourier representation of [115, 118] uses the following basis:

$$\begin{aligned} \phi \quad = \quad & \{1, \cos mu, \sin mu, \cos lv, \sin lv, \cos mu \cos lv, \sin mu \cos lv, \\ & \cos mu \sin lv, \sin mu \cos lv, ... \quad (m = 1, 2, ...; \; l = 1, 2, ...)\} \end{aligned} \tag{7.1}$$

The functions $x(.,.), y(.,.)$ and $z(.,.)$ are composed as weighted sums of the elements of the above basis and has the following form:

$$\begin{aligned} f(u, v) = \sum_{m=0}^{K} \sum_{l=0}^{K} \lambda_{m,l}[\quad & a_{m,l} \cos mu \cos lv + b_{m,l} \sin mu \cos lv \\ + \quad & c_{m,l} \cos mu \sin lv + d_{m,l} \sin mu \sin lv] \end{aligned} \tag{7.2}$$

where,

$$\lambda_{m,l} \quad = \quad \begin{cases} 1 & for\ m = 0,\ l = 0 \\ 2 & for\ m > 0,\ l = 0\ or\ m = 0,\ l > 0 \\ 4 & for\ m > 0,\ l > 0 \end{cases}$$

The series is truncated at K, i.e. only a finite number of harmonics are used. This again is similar to the case in 2D, and is done to constrain the space of functions. Taken together, the coefficients form the parameter vector,

$$\vec{p} = (a_x, b_x, c_x, d_x, a_y, b_y, c_y, d_y, a_z, b_z, c_z, d_z) \tag{7.3}$$

The four basic classes of surfaces in three dimensions are tori (closed tubes), open surfaces (with one edge), tubes (open surfaces with two edges) and closed surfaces.

The torus, which is periodic in both the surface variables is formed with the entire basis in equation (7.1). Closed surfaces, which we are most interested in, is given by the representation:

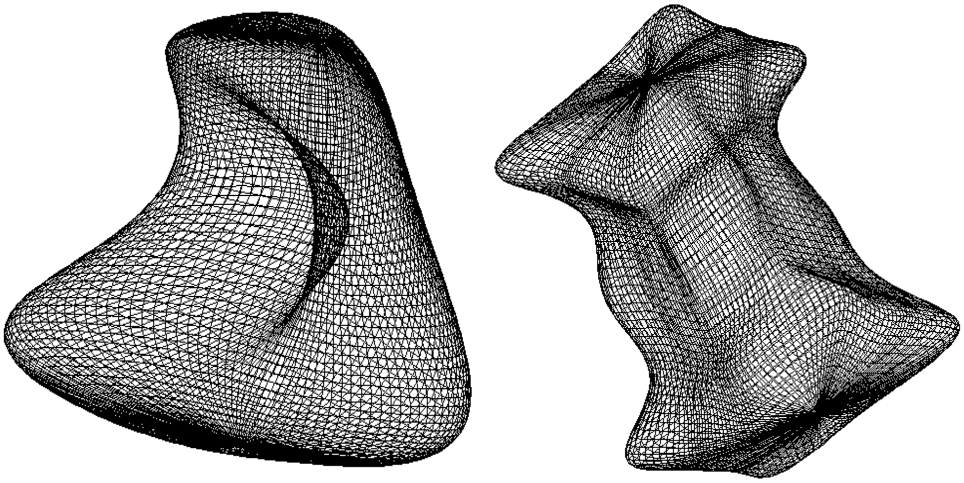


Figure 7.1: Two closed surface examples using upto four and eight harmonics (taken from [118])

$$\phi_{closed} = \{1, \sin lv, \cos mu \sin lv, \sin mu \sin lv, ...\} \tag{7.4}$$

which forces the functions to be constants at $v = 0, \pi, 2\pi$. But this forces the ends to be together as well. The ends need to be separated by adding a weighted term to each coordinate of the form $sin(v - \pi/2)$ resulting in three more additional shape parameters. Two closed surfaces shown in Figure 7.1 demonstrates the capability of the above mentioned parameterization. We note here that it is also possible to represent open surfaces and tubes by this parameterization, details of which can be found in [118].

7.3 Region information

As in the 2D case, the idea here is to classify the image into a number of regions or classes. Thus for each voxel in the image, we need to decide or estimate which class it belongs to. We have described two methods to do region based segmentation in chapter 3 and any of those could be used here. However, we choose to use the region classification method that was described in detail in chapter 3, the final equation for which is given by equation (3.54). Note that the ICM method for optimization was used here. That equation continues to hold and the procedure remains the same in 3D. Thus every pixel is classified to belong to a particular class depending on its actual pixel characteristics (i.e. the grey level value) and those of its neighbors. This neighborhood information is again modeled using a MRF model. A first order neighborhood system has been used which has six neighbors (2 neighbors along the three axes) instead of the four that we had in the 2D case.

7.4 Integrated surface finding objective function

In this section, we shall define the objective function, optimizing which would result in the estimated surface parameters. The development is similar to the 2D version in section

4.2 of chapter 4.

As in the 2D case, the input to the problem consists of the actual image I and the region classified image I_s, which is obtained from I after passing it through a region based segmentation step as discussed above. We shall also assume that the interior of the boundary that we are trying to find belongs to a single region in I_s. Let us note here that similar to the 2D case, all that this assumption requires is that the intra region variability should be smaller than the inter region variability. Further relaxations to this can be attained in a similar way as was achieved in the 2D case (see chapter 4 for details). The traditional surface finding problem does not use the original image directly. Being a gradient based approach, it uses instead the gradient image I_g. As in the Staib and Duncan [118, 115] approach, we shall use the magnitude of the gradient vector at each voxel location. I_g can be obtained from I either by convolving the input image I with the derivative (taken in the three directions) of a Gaussian kernel and then computing I_g to be the magnitude of the above resulting vector image. (Alternatively, one can obtain I_g from I by first convolving with a Gaussian to smooth the effects of noise followed by taking a finite difference approximation to the partial derivatives in the three directions and then calculating the magnitude of the gradient vector at each voxel location.) Thus finally, the input to the system is the gradient image I_g and the region classified image I_s.

Parallel to our development for the 2D case, the above surface estimation problem using gradient and region homogeneity information can be posed in the maximum *a posteriori* framework. This is suitable for incorporating *a priori* shape information if available.

Our objective is to maximize $P(\vec{p}|I_g, I_s)$, where as already mentioned, $\vec{p}$ is the vector of parameters used to parameterize the surface. This development is identical to the one in 2D case described in chapter 4, and the relevant equation, that we carry over here is as follows:

$$\arg\max_{\vec{p}} M(\vec{p}, I_g, I_s) \quad \equiv \quad \arg\max_{\vec{p}} [M_{prior}(\vec{p}) \, + \, M_{gradient}(I_g, \vec{p}) \, + \, M_{region}(I_s, \vec{p})] \tag{7.5}$$

The first term in equation (7.5) corresponds to the prior shape term. When it is non-uniform, it biases the model towards a particular range of shapes. However, since there might be other objects in the image, we would always need an initial estimate of the surface to start the optimization process. The information fusion that we present in this case increases the reliability of the surface finding procedure under increased uncertainty in the initial boundary placement. Experimental results to validate this claim will be provided in the results section.

The second term is the gradient likelihood term. It is a measure of the likelihood of the surface being the true object boundary given the gradient image. At each point on the surface, the strength of the boundary is evaluated by the magnitude of the gradient at that particular voxel, given by the gradient image. Thus the likelihood of the whole surface being the true boundary becomes proportional to the sum of the magnitude of the gradients at all the points that lie on the surface boundary. If we assume that the noise can be approximated by a zero mean Gaussian, and further assume that the voxels on the boundary are independent, then we may express the above term in the probability expression as the following area integral (see Staib and Duncan [115, 118] for further details) where σ_n^2 is the variance of the underlying noise process:

$$M_{gradient}(I_g, \vec{p}) \propto \int\int_{A_{\vec{p}}} I_g(x, y, z) dA \tag{7.6}$$

where the area element on the surface is given by:

$$dA = |\mathbf{x}_u \times \mathbf{x}_v| dudv \tag{7.7}$$

The third term in equation (7.5) is analogous to the corresponding term in (4.6) and is responsible for incorporating the region information into the surface finding framework. Again, the notion is that we would expect the bounding surface to include a homogeneous region inside it. We note that the comments made before regarding the homogeneity assumption are valid here as well. For simplicity's sake, if we assume that we are dealing with an image where the target object is surrounded by a single background, we could as in the previous chapters, assign positive values to the interior of the object and negative values outside. However, as explained before in chapter 4, this assumption of a single target object in an uniform background is not necessary. Once we have associated positive values with the target object and negative values with points that lie outside, a volume integral that sums up all the points inside the surface is taken. Clearly, this integral would be a maximum when the bounding surface is optimally placed over the object. Thus the third term in (7.5) is given by:

$$M_{region}(I_s, \vec{p}) \propto \int\int\int_{V_{\vec{p}}} I_s(x, y, z) dV \tag{7.8}$$

Hence finally we have,

$$\begin{aligned}\arg\max_{\vec{p}} M(\vec{p}, I_g, I_s) &= \max_{\vec{p}} [M_{prior}(\vec{p}) + M_{gradient}(I_g, \vec{p}) + M_{region}(I_s, \vec{p})] \\ &\equiv \max_{\vec{p}} \Bigg[M_{prior}(\vec{p}) + K_1 \int\int_{A_{\vec{p}}} I_g(x,y,z)dA \\ &+ K_2 \int\int\int_{V_{\vec{p}}} I_s(x,y,z)dV\Bigg] \qquad (7.9)\end{aligned}$$

where K_1 and K_2 are the weighting constants which signifies the relative importance of the two terms in the above equation.

Of the last two terms in (7.9), one is an area integral and the other is a volume integral. In general, computing an area integral is much less expensive compared to a volume integral. Thus we would save a lot of computation, especially when we carry out an iterative optimization procedure, if we could convert the volume integral to to an area integral which we have to compute anyway, as the second term which is present even in the original surface finding method involves computing an area integral. The above can be done using Gauss' divergence theorem [9] which states

Gausses' Divergence Theorem: If W is a simple region in $\mathcal{R}^3$ and if $(\partial W, N)$ be the positively oriented boundary of W, then for any C^1 vector field $\mathbf{F} : W \subseteq \mathcal{R}^3 \longrightarrow \mathcal{R}^3$

$$\int\int \mathbf{F} \cdot dA = \int\int\int div\mathbf{F}\, dV \qquad (7.10)$$

For our problem we let,

$$\mathbf{F}_x(x,y,z) = \frac{1}{3}\int_0^x I_s(\alpha, y, z)d\alpha \qquad (7.11)$$

$$\mathbf{F}_y(x,y,z) \;=\; \frac{1}{3}\int_0^y I_s(x,\beta,z)d\beta \tag{7.12}$$

$$\mathbf{F}_z(x,y,z) \;=\; \frac{1}{3}\int_0^z I_s(x,y,\gamma)d\gamma \tag{7.13}$$

where $\mathbf{F} = (\mathbf{F}_x, \mathbf{F}_y, \mathbf{F}_z)$. We note that the definition of $F()$ is done in such a way that the C^1 continuity requirement in the statement of the above theorem is met. Under the above assumptions, we have,

$$\begin{aligned}
\int\!\!\int\!\!\int_{V_{\vec{p}}} I_s(x,y,z)dV &= \int\!\!\int \mathbf{F}\cdot dA && (7.14)\\
&= \int\!\!\int_{A_{\vec{p}}} \mathbf{F}\cdot(\mathbf{x}_u \times \mathbf{x}_v)dudv\\
&= 3\int\!\!\int_{A_{\vec{p}}} \mathbf{F}_x(y_u z_v - z_u y_v)dudv\\
&= 3\int\!\!\int_{A_{\vec{p}}} \mathbf{F}_y(z_u x_v - x_u z_v)dudv\\
&= 3\int\!\!\int_{A_{\vec{p}}} \mathbf{F}_z(x_u y_v - y_u x_v)dudv\\
&= \int\!\!\int_{A_{\vec{p}}} [\mathbf{F}_x(y_u z_v - z_u y_v) + \mathbf{F}_y(z_u x_v - x_u z_v)\\
&\qquad + \mathbf{F}_z(x_u y_v - y_u x_v)]dudv && (7.15)
\end{aligned}$$

Substituting the above in (7.9) we finally get,

$$\begin{aligned}
\max_{\vec{p}} M(I_g, I_s, \vec{p}) &= \max_{\vec{p}} [M_{prior}(\vec{p})\\
&+ K_1 \int\!\!\int_{A_{\vec{p}}} I_g(x,y,z)dA\\
&+ K_2 \int\!\!\int \mathbf{F}\cdot dA] && (7.16)
\end{aligned}$$

Let us note here that the calculation of $\mathbf{F}_x()$, $\mathbf{F}_y()$, and $\mathbf{F}_z()$ are done only once at the start of the optimization process. Also, we note that these calculations merely involve summing up the values of the voxels in the image I_s in the three directions. Further their derivatives, which we need during the optimization process results in the values of the image I_s itself. We mention this to point out that the use of the extra information hardly introduces any extra computational burden.

In the above, we have presented a surface finding procedure that introduces a prior term that incorporates information that we might obtain from region based segmentation. Further, use of Gauss's divergence theorem allows us to reduce the whole problem to computing surface integrals only rather than both surface and volume integral.

7.5 Evaluation and Optimization

The objective function in equation (7.16), can be evaluated by numerical integration. The gradient of the objective is necessary for optimization. The derivative of the objective is given by,

$$\begin{aligned}\frac{\partial M}{\partial p_x} &= \frac{\partial M_{prior}(\vec{p})}{\partial p_x} + K_1 \int\int_{A_{\vec{p}}} \left[I_g(x,y,z) \frac{\partial}{\partial p_x} |\mathbf{x}_u \times \mathbf{x}_v| \right. \\ &+ \left. \frac{\partial I_g(x,y,z)}{\partial x} \frac{\partial x(\vec{p},u,v)}{\partial p_x} |\mathbf{x}_u \times \mathbf{x}_v| \right] dudv \\ &+ 3K_2 \int\int_{A_{\vec{p}}} I_s(x,y,z)(y_u z_v - z_u y_v) \frac{\partial x(\vec{p},u,v)}{\partial p_x} dudv \qquad (7.17)\end{aligned}$$

and similarly for y and z. This expression can also be evaluated by numerical integration. Expressions like $\frac{\partial I_g(x,y,z)}{\partial x}$ can be obtained using discrete derivative calculation. Other expressions like $\frac{\partial x(\vec{p},u,v)}{\partial p_x}$ and $\mathbf{x}_u$ and $\mathbf{x}_v$ can be obtained analytically from equations (7.1) and (7.2). The derivatives of the prior terms can be obtained in exactly the same way as in the 2D case.

Optimization is achieved using conjugate gradient method which is a local optimization method. For surface finding even local maximization involves a lot of computation. Thus to avoid even further computational burden, global optimization methods were not considered at the cost however of not being able to guarantee global convergence. However, since the method is initialized close to the actual location, global optimization methods may not be required.

7.6 Results

Experiments were carried out both with synthetic and clinical images to verify the performance of the above mentioned method. However to evaluate it, we need a method to calculate the error between two surfaces expressed parametrically. We do this using the same definition (given below) as the one used in [118]. The error is defined as the average distance between each point on the estimated surface and the closest point on the true surface. That is, the error between surfaces S and $\hat{S}$ is defined as

$$e(S,\hat{S}) = \frac{\int_{(u,v)\in\hat{S}} \min_{(u',v')\in S} |S(u',v') - \hat{S}(u,v)| dA}{\int_{(u,v)\in\hat{S}} dA} \tag{7.18}$$

This can be computed discretely by first taking a distance transform of a binary volume representing the true surface [118]. This is then correlated with the binary volume representing the estimated surface, which gives the minimum distance between the estimated and the true surface. This is then normalized by the area of the estimated surface.

We first used a synthetic example to evaluate the algorithm developed. This is useful because for this case we had exact knowledge of the true surface boundary. Comparisons of the integrated method was done against the traditional gradient based surface finding approach.

Figure 7.2 shows a simple synthetic example of a closed surface with added Gaussian noise. The SNR for this image was 1.6. The initial surface was roughly placed on the target object. Clearly, we can observe that the combined method performed better. It can be observed from the wireframe diagrams that the surface finder diverges under these noise conditions when using gradient information alone which is not the case for the integrated method.

Figure 7.3 shows a comparison of the two methods under increasing noise conditions. Here, the X-axis corresponds to the noise level given by the standard deviation of the noise used. The Y-axis gives a measure of the distance between the estimated surface and the true one using equation (7.18). Once again, it is obvious that the integrated method is superior under high noise conditions. This upholds our claim that the proposed integrated method is robust to noise which is even more dramatic in 3D.

Now, as for the initialization, Figure 7.4 shows the performance when the vertical shift was varied from the true position, keeping the initialization for the other parameters fixed. It shows that the integrated method has a larger capture region. In other words, the integrated method converges to the desired target object, which is much further off when compared to the gradient-only case. Thus the the region within which the initialization should be is larger for the integrated algorithm.

In Figure 7.5 we try out the proposed algorithm on a three dimensional cardiac image of a dog's heart obtained using the Dynamic Spatial Reconstructor (DSR). The

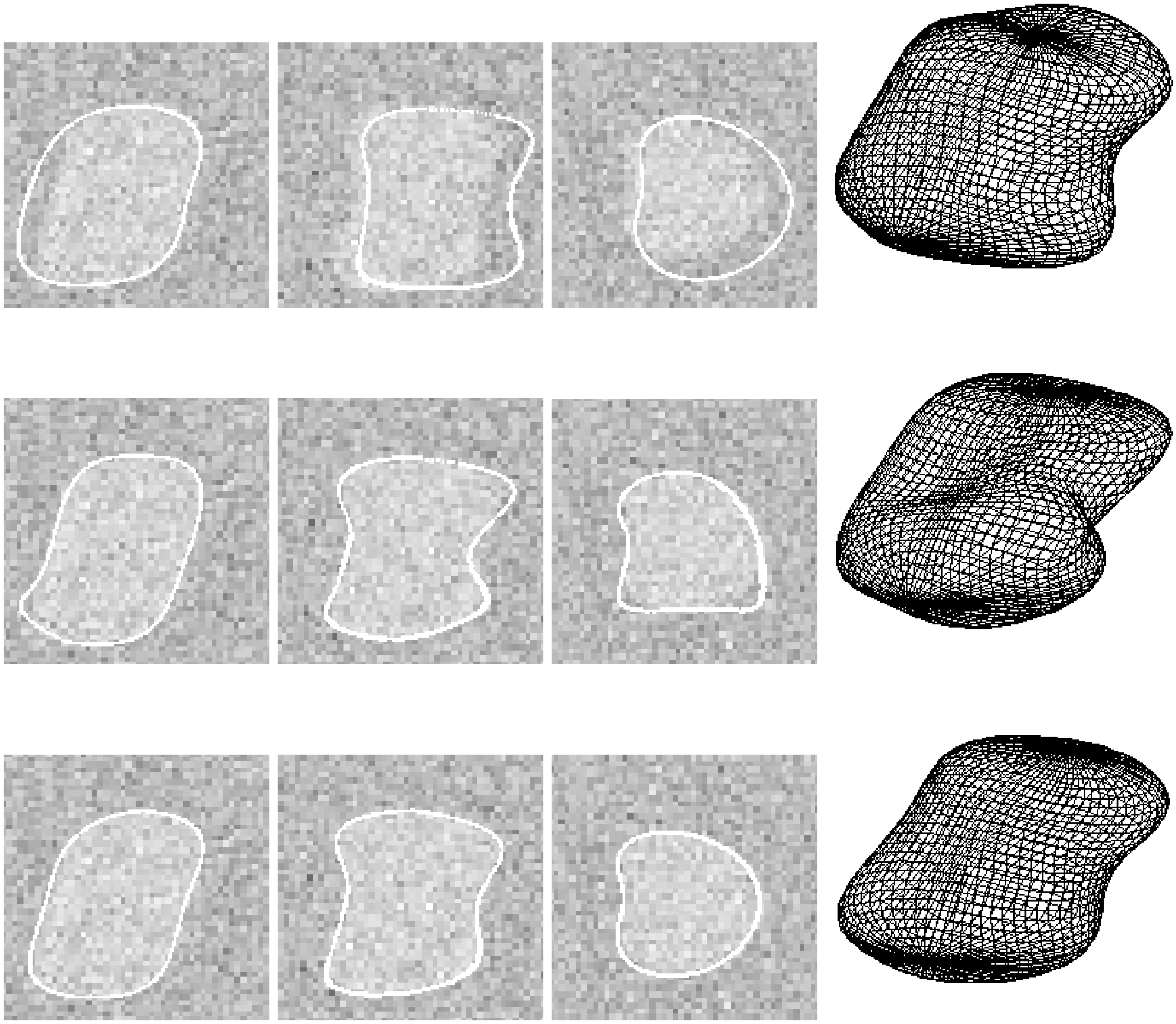


Figure 7.2: Surface finding for a synthetic image with and without region information. (a)Top: Three perpendicular slices through the 3D image ($48 \times 48 \times 48$) are shown with the initial surface along with and the wireframe. (b)Middle: The same slices through the same 3D image are shown with the surface obtained using only the gradient information and the corresponding wireframe. (c)Bottom: The same slices through the same 3D image are shown with the surface obtained using both the gradient and the region information and the corresponding wireframe.

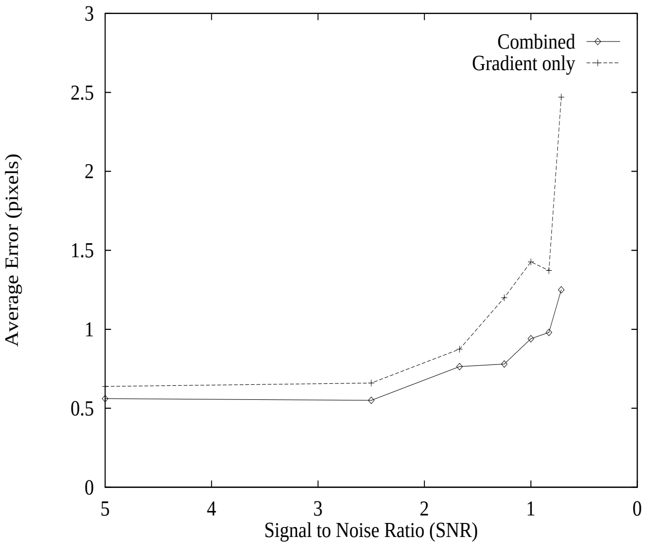


Figure 7.3: Noise performance of the surface finder with and without region information. Clearly, the combined method is superior

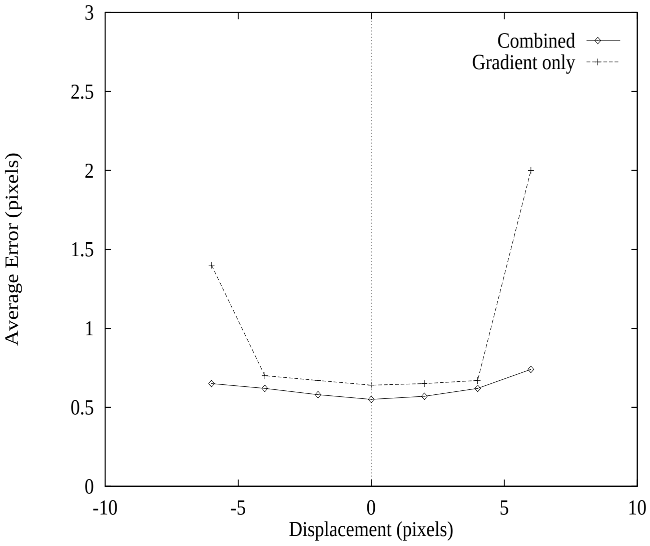


Figure 7.4: Performance of the surface finder with and without region information under different starting positions. This was varied by shifting the initialization vertically. Clearly, the combined method is superior

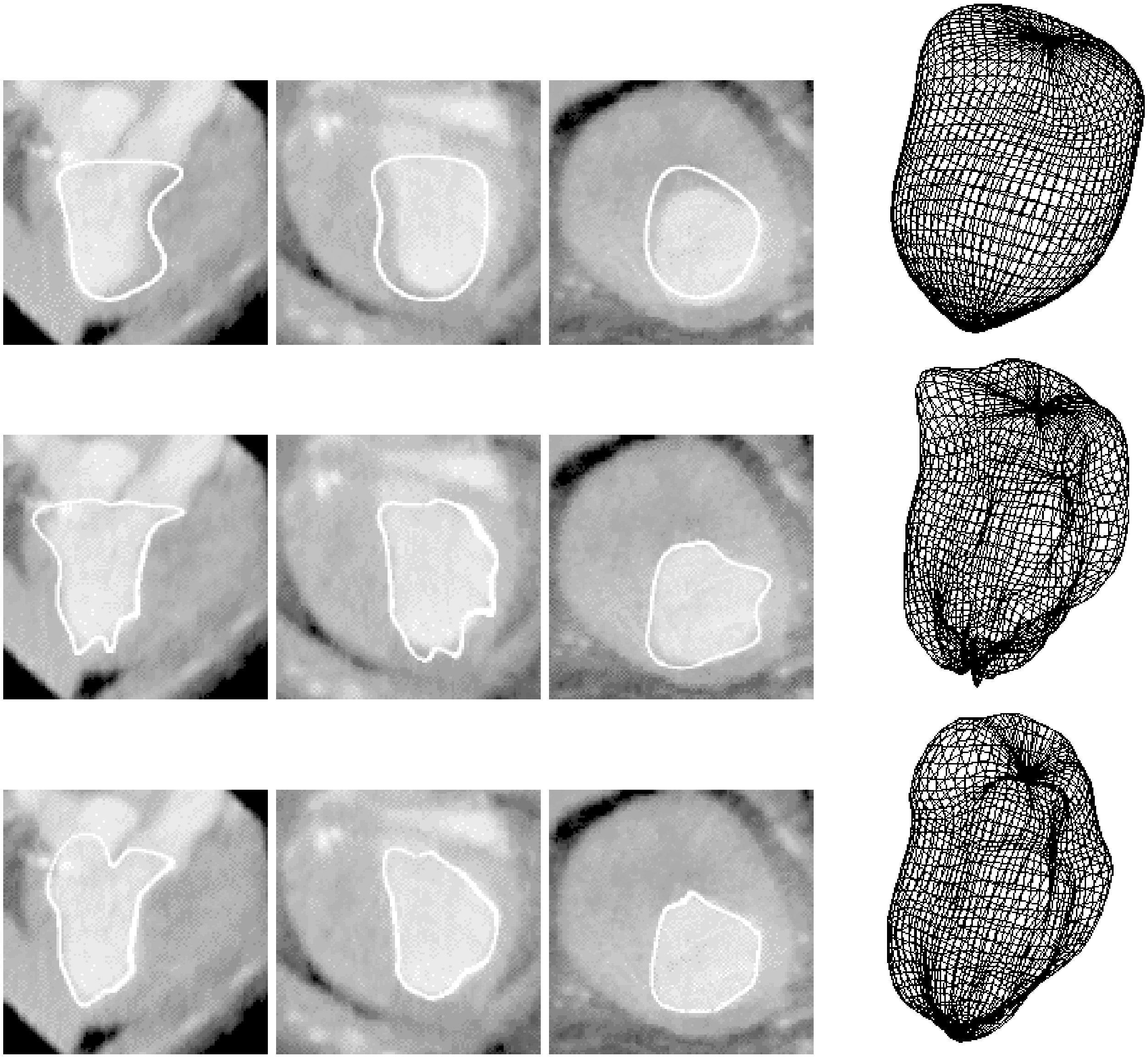


Figure 7.5: Surface finding for a DSR image of a canine heart with and without region information. (a)Top: Three perpendicular slices through the 3D image (98×100×110) are shown with the initial surface and the wireframe. (b)Middle: The same slices through the same 3D image are shown with the surface obtained using only the gradient information and the corresponding wireframe. (c)Bottom: The same slices through the same 3D image are shown with the surface obtained using both the gradient and the region information and the corresponding wireframe.

DSR is 3D imaging device based on high speed x-ray computed tomography [105]. An expert cardiologist can observe that the combined method performs better. One can see that, by noting that the integrated method does a better job of capturing the blood pool (the bright area) in the slices shown.

7.7 Conclusions

We have presented in this chapter an integrated method for doing surface finding using both region and gradient information. As the examples show, the integrated approach is more robust to both increased amounts of noise as well as increasingly displaced initialization of the initial boundary. Almost uniformly there is an improvement over the conventional gradient based boundary finding. To prove this, we have devised a variety of experiments and the results from all of them are favorable. Application of this method on real medical images results in noticeable improvement as shown.

Chapter 8

Conclusion

This thesis has developed a systematic approach to the problem of integrated methods for image segmentation. The failure of any single image segmentation method using a single image-derived source of information for a wide range of images with a variety of image content has generated considerable interest for integrated methods in recent years. This thesis has provided a step in that direction. It develops and portrays the integration problem from two inter-related viewpoints, *feature integration* and *module integration*. Feature integration within the boundary finding framework was achieved by introducing region and curvature information in addition to gradient information within the deformable boundary finding framework. This vastly improved the performance of the boundary finder when compared to the traditional approach of using gradient information alone. While the use of region information made the process significantly robust to noise and poor initialization, the use of curvature information bolstered its localization accuracy. Then we considered the more general case of module integration where computational modules representing the region and the boundary processes were integrated within a game theoretic framework. Thus unlike in the feature integration case, both the region and boundary processes were simultaneously improved. While being able to produce better results using an integrated method constitutes a significant contribution, the way it was achieved is also important. Often in computer vision, integration is considered synonymous with optimizing a global single objective function. This is not only inadequate on certain occasions, but also may be impossible or at least computationally very expensive due to the incommensurate nature of the objectives involved. We circumvent this by using a novel game theoretic framework where the incommensurability is not considered to be a problem. The proposed integrated method yields significantly better results which has been proved by examples involving both synthetic and real image data.

However, there remain areas of potential improvements. While experimental results pointed out that an integrated method results in an improvement, further theoretical analysis can be done to bolster that argument through the development of theoretical bounds as is often done in the signal processing literature. More rigorous theoretical analysis can also be performed regarding the existence and uniqueness of Nash equilibrium for the segmentation problem considered. Another issue is regarding the use of game theoretic Nash equilibrium as opposed to the optima of a global objective function. While it is true that it might be difficult to frame many problems like the present one within the global objective framework, it is not impossible. Thus the question remains as to which is likely to lead us to a better solution. It has been shown that these two different methodologies may result in different answers [17], but the superiority of one over the other hasn't yet been clearly established. It is possible that one might want to further investigate in that direction. On a more related theme, it is possible to extend the whole methodology to three dimensions. This thesis showed preliminary results that indicated that it most likely will generate improvements. We are in the process of doing further research in that direction.

Appendix A

Gradient Calculation

In Chapter 4, we developed an integrated method for doing deformable boundary finding using region information in addition to gradient. Here, we show how the gradient of the objective function developed can be calculated. We have from equation (4.9)

$$M(\vec{p}, I_g, I_s) = \left[M_{prior}(\vec{p}) + \int_{C_{\vec{p}}} K_1 I_g(x,y)dt + K_2 \int\int_{A_{\vec{p}}} I_s(x,y)dA \right] \tag{A.1}$$

Taking the gradient of the above with respect to $\vec{p}$, we get,

$$\nabla_{\vec{p}} M(\vec{p}, I_g, I_s) = \frac{\partial}{\partial \vec{p}} M_{prior}(\vec{p}) + K_1 \int_{C_{\vec{p}}} \frac{\partial}{\partial \vec{p}} I_g(x,y)dt + K_2 \frac{\partial}{\partial \vec{p}} \int\int_{A_{\vec{p}}} I_s(x,y)dA \tag{A.2}$$

We shall make the simplifying assumption that dt is not a function of $\vec{p}$. For Gaussian prior distributions as in (3.12) the first term on the RHS of (A.2) is

$$\frac{\partial}{\partial p_i} M_{prior}(\vec{p}) = -\frac{p_i - m_i}{2\sigma_i^2} \tag{A.3}$$

Now, for the second term, we have

$$\int_{C_{\vec{p}}} \frac{\partial}{\partial \vec{p}} I_g(x,y)dt = \int_{C_{\vec{p}}} \left[\frac{\partial I_g(x,y)}{\partial x} \frac{\partial x}{\partial \vec{p}} + \frac{\partial I_g(x,y)}{\partial y} \frac{\partial y}{\partial \vec{p}} \right] dt \tag{A.4}$$

Before going to the third term in the RHS of (A.2), we observe the following from Green's theorem:

$$\begin{aligned} \int\int_{A_{\vec{p}}} I_s(x,y)dA &= \frac{1}{2} \int_{C_{\vec{p}}} \left[N_s(x,y) \frac{\partial x}{\partial t} + M_s(x,y) \frac{\partial y}{\partial t} \right] dt \\ &= \int_{C_{\vec{p}}} N_s(x,y) \frac{\partial x}{\partial t} dt = \int_{C_{\vec{p}}} M_s(x,y) \frac{\partial y}{\partial t} dt \end{aligned} \tag{A.5}$$

where $M(.)$ and $N(.)$ is as defined in equations (4.11) and (4.10).

We also observe that $\vec{p} = (\vec{p}_x, \vec{p}_y)$ where

$$\vec{p}_x = (a_0, a_1, b_1, a_2, b_2, ...) \tag{A.6}$$

and

$$\vec{p}_y = (c_0, c_1, d_1, c_2, d_2, ...) \tag{A.7}$$

Thus from the observations in (A.5), (A.6) and (A.7), we get

$$\begin{aligned} \frac{\partial}{\partial \vec{p}_x} \int\int_{A_{\vec{p}}} I_s(x,y)dA &= \frac{\partial}{\partial \vec{p}_x} \int_{C_{\vec{p}}} M_s(x,y) \frac{\partial y}{\partial t} dt = \int_{C_{\vec{p}}} \left(\frac{\partial}{\partial \vec{p}_x} M_s(x,y) \right) \frac{\partial y}{\partial t} dt \\ &= \int_{C_{\vec{p}}} I_s(x,y) \frac{\partial x}{\partial \vec{p}_x} \frac{\partial y}{\partial t} dt \end{aligned} \tag{A.8}$$

and

$$\begin{aligned} \frac{\partial}{\partial \vec{p}_y} \int\int_{A_{\vec{p}}} I_s(x,y) dA &= \frac{\partial}{\partial \vec{p}_y} \int_{C_{\vec{p}}} N_s(x,y) \frac{\partial x}{\partial t} dt = \int_{C_{\vec{p}}} \left(\frac{\partial}{\partial \vec{p}_y} N_s(x,y) \right) \frac{\partial x}{\partial t} dt \\ &= -\int_{C_{\vec{p}}} I_s(x,y) \frac{\partial y}{\partial \vec{p}_y} \frac{\partial x}{\partial t} dt \qquad \text{(A.9)} \end{aligned}$$

The partials of $I_g(.)$, the gradient image can be obtained by using a central divided difference approximation. As for the partials of x and y with respect to $\vec{p}$, we can calculate them from equation (3.4) as:

$$\begin{array}{rl} \frac{\partial x(t)}{\partial a_0} = 1 & \frac{\partial y(t)}{\partial a_0} = 0 \\ \frac{\partial x(t)}{\partial c_0} = 0 & \frac{\partial y(t)}{\partial c_0} = 1 \\ \frac{\partial x(t)}{\partial a_k} = cos(kt) & \frac{\partial y(t)}{\partial a_k} = 0 \\ \frac{\partial x(t)}{\partial b_k} = sin(kt) & \frac{\partial y(t)}{\partial b_k} = 0 \\ \frac{\partial x(t)}{\partial c_k} = 0 & \frac{\partial y(t)}{\partial c_k} = cos(kt) \\ \frac{\partial x(t)}{\partial d_k} = 0 & \frac{\partial y(t)}{\partial d_k} = sin(kt) \end{array}$$

Appendix B

Performance analysis

As mentioned in the Discussion section of chapter 5, for the one dimensional analysis we shall model the received edge as:

$$r(x) = I\Phi(\frac{x-l}{\sigma_s}) + n(x) \tag{B.1}$$

where I is the intensity of the edge, l is the location of it, and σ_s is the amount of blur and Φ is the step response of a Gaussian. We assume that L is the extent of the image (here, the 1-D signal), i.e. $x \in [0, L]$. The noise is distributed as:

$$n(x) \sim N(0, \sigma_n^2)$$

B.1 Localization

In this section assuming the model chosen above, we shall analyze the localization errors for boundary finding using just the gradient information and using just the region information.

B.1.1 Gradient based

This is the equivalent to first convolving the image with a linear filter and then finding the maxima. This is exactly the way the gradient information is used in a lot of boundary finding and edge detection methods that include for example [21, 59, 116, 62]. It has been concluded that the best filter comprises of the Gaussian kernel. Since we are interested in finding the maxima of the changes we convolve with a Gaussian and then take derivatives. Thus mathematically this is equivalent to:

$$\arg\max_x [\dot{\phi}_{\sigma_c}(x) * r(x)] \tag{B.2}$$

where

$$\dot{\phi}_{\sigma_c}(x) = \frac{-x}{\sqrt{2\pi}\sigma_c^3} e^{-\frac{x^2}{2\sigma_c^2}}$$

is the first derivative of the Gaussian and σ_c controls the smoothness. We shall assume that the integration window is of width $2T_x$ around the point of observation. Now, finding the maxima of the derivative as in above is also equivalent to solving for

$$\ddot{\phi}_{\sigma_c}(x) * r(x) = 0 \tag{B.3}$$

or,

$$\ddot{\phi}_{\sigma_c}(x) * [s(x) + n(x)] = 0 \tag{B.4}$$

where $s(x)$ is the edge in the received signal.

We are trying to analyze the localization error $\hat{l} - l$ of a linear operator (see also [21]) since convolution is a linear operation. Let H represent the convolution filter in the last equation. Thus it becomes:

$$Hs(\hat{l}) \; + \; Hn(\hat{l}) \; = \; 0 \tag{B.5}$$

Our interest is in finding the localization error $\hat{l} - l$ when the assumption is that the true edge is located at l. However, without loss of generality, we could choose $l = 0$ and then $\hat{l}$ becomes the localization error. By expanding $Hs(.)$ about the origin, in a Taylor series we get,

$$Hs(0) \; + \; \hat{l}\, H_x s(0) \; + \; O(\hat{l}^2) \; + Hn(0) = 0 \tag{B.6}$$

In the above, we have replaced $Hn(\hat{l})$ by $Hn(0)$ using the assumption that $n(.)$ is a uniform noise field. H_x represents $\frac{\partial}{\partial x} H$. Neglecting the effect of the second order terms and solving the above gives:

$$\hat{l} \;\; = \;\; -\frac{Hn(0)}{H_x s(0)} \tag{B.7}$$

$$\;\; = \;\; -\frac{\int_{-T_x}^{T_x} \phi''_{\sigma_c}(z) n(z) dz}{\int_{-T_x}^{T_x} \phi'''_{\sigma_c}(z) I \Phi(\frac{z}{\sigma_s}) dz} \tag{B.8}$$

We can now obtain the mean and the variance as:

$$E(\hat{l}) = 0 \tag{B.9}$$

$$Var(\hat{l}) = E(\hat{l}^2) = -\frac{\sigma_n^2 \int_{-T_x}^{T_x} (\phi''_{\sigma_c}(z))^2 dz}{[\int_{-T_x}^{T_x} \phi'''_{\sigma_c}(z) I \Phi(\frac{z}{\sigma_s}) dz]^2} \tag{B.10}$$

Using integration by parts and assuming that $T_x >> max(\sigma_c, \sigma_s)$ (the integrating window for convolution is much larger than the standard deviation of the smoothness) it can easily be shown that

$$Var(\hat{l}) = \frac{\sigma_n^2}{I^2} \frac{3\sqrt{\pi}}{4\sigma_c^5} (\sigma_s^2 + \sigma_c^2)^3 \tag{B.11}$$

B.1.2 Region based

The simplest equivalent of region based classification can be posed as a hypothesis testing. As a matter of fact we use a slightly more complicated version of the same approach in real 2D images as we have already seen. For the simple 1D equivalent of an edge, we basically have two hypothesis. One where there is no signal, ie. before it encounters the step, and the one after it where the mean signal strength is I. One can then construct the following likelihood ratio [125] test and if it is greater than unity, it is assigned to the class with signal and to the other class otherwise. Note that this is basically what the image classification process does. (The smoothness effect is ignored here.)

$$\lambda(R) \quad = \quad \frac{exp(-(r-I)^2/2\sigma_n^2)|_{r=s+n}}{exp(-r^2/2\sigma_n^2)|_{r=n}} \tag{B.12}$$

It can easily be shown that this is equivalent to the following test:

$$2r - I > 0 \;\; or \;\; r > I/2 \tag{B.13}$$

Thus, simply the two classes could be characterized by the positivity of the above. Hence, all that is positive belongs to one class and all that is negative belongs to the other. It is easy to see that for the step edge shown, points after the step edge has values $+I/2$ and points before that have value, $-I/2$. At the boundary the values change gradually. As $\sigma_s \to 0$ the only values are $I/2$ and $-I/2$. We shall use this output itself. For a perfect noiseless image this in itself corresponds to region based classification.

Since we are looking to place the boundary to differentiate homogeneous regions, the boundary should be placed such that the following is satisfied.

$$\arg\max_{l} \int_{l}^{L} (r(x) - I/2)dx \tag{B.14}$$

where l is the precise location of the image, and L is the limit point of the $1D$ image. Taking derivatives, the location of the boundary can be given by:

$$r(l) - I/2 = 0 \tag{B.15}$$

or,

$$I\Phi(l/\sigma_s) + n(l) - I/2 = 0 \tag{B.16}$$

Once again as in the previous case, we may assume without loss of generality $l = 0$ and then expand it around the origin. Ignoring the higher order terms, we get,

$$I\Phi(0) + \hat{l}I\phi_{\sigma_s}(0) + n(0) - I/2 = 0 \tag{B.17}$$

Again we have assumed uniformity of the noise field. Thus we finally get the estimate which is also the localization error,

$$\begin{aligned} \hat{l} &= -\frac{n(0)}{I\phi_{\sigma_s}(0)} \\ &= -\sqrt{2\pi}\sigma_s \frac{n(0)}{I} \end{aligned} \tag{B.18}$$

It is easy to calculate the mean and the variance for this expression given by:

$$E(\hat{l}) = 0 \tag{B.19}$$

and

$$Var(\hat{l}) = 2\pi\sigma_s^2 \frac{\sigma_n^2}{I^2} \tag{B.20}$$

B.2 Signal to noise ratio

In this section our aim is to calculate the SNR at the location of the edge or boundary in this simple 1D image. We first start with the method that uses gradient only and then later on go to do the same for the region based one. We will once again assume the same model that we started with along with the assumption that the true step is located at the origin.

B.2.1 Gradient based

From equation(B.2) the signal power at the location of the boundary is given by:

$$S = \left[\int_{-T_x}^{T_x} \phi_{\sigma_c}^{'}(z) I \Phi(\frac{z}{\sigma_s}) dz \right]^2 \tag{B.21}$$

Once again, using the approximation, $T_x >> max(\sigma_c, \sigma_s)$, it can be shown that

$$S = \frac{I^2}{2\pi(\sigma_c^2 + \sigma_s^2)} \tag{B.22}$$

and the noise power is given by:

$$\begin{aligned} N &= \sigma_n^2 \int_{-T_x}^{T_x} [\phi_{\sigma_c}^{'}(x)]^2 dx \\ &= \frac{\sigma_n^2}{4\sqrt{\pi}\sigma_c^3} \end{aligned} \tag{B.23}$$

Hence, the SNR is given by:

$$SNR = \sqrt{\frac{S}{N}} = \frac{I}{\sigma_n} \frac{\sqrt{2}}{\pi^{\frac{1}{4}}} \frac{\sigma_c^{\frac{3}{2}}}{(\sigma_c^2 + \sigma_s^2)^{\frac{1}{2}}} \tag{B.24}$$

The above gives an expression of the SNR at the location of the edge.

B.2.2 Region based

From equation(B.14) the signal power at the location of the boundary is given by:

$$S = \left[\int_0^L (I\Phi(\frac{x}{\sigma_s}) - \frac{I}{2}) dx \right]^2 \tag{B.25}$$

It can be shown that assuming $L >> \sigma_s$ is true, the above integration reduces to:

$$S = I^2 \left(\frac{L}{2} - \frac{\sigma_s}{\sqrt{2\pi}} \right)^2 \tag{B.26}$$

The noise power is given by:

$$N = \sigma_n^2 L \tag{B.27}$$

Thus, the SNR is given by:

$$SNR = \sqrt{\frac{S}{N}} = \frac{I}{\sigma_n} \left(\frac{\sqrt{L}}{2} - \frac{\sigma_s}{\sqrt{2\pi L}} \right) \approx \frac{I\sqrt{L}}{2\sigma_n} \tag{B.28}$$

Thus the SNR not only depends on the local characteristics, it also depends on L, which is the width of the region we are trying to find the boundary of. This is actually a more global characteristic.

B.3 Response to spurious peaks

A very simple way to introduce spurious peaks in our simple signal model is:

$$r(x) = i_1 \Phi(\frac{x - x_1}{\sigma_p}) + i_2 \Phi(\frac{x - x_1 - \alpha}{\sigma_p}) + I\Phi(\frac{x - l}{\sigma_s}) + n(x) \tag{B.29}$$

Thus there is a fluctuation, the signal has an overall increase of value i_1 at x_1 and then decreases by i_2 at the location $x_1 + \alpha$. If $i_1 = i_2 = i$, this represents a small peak of width equal to α.

Now, if $\alpha > \sigma_c$, the increase in SNR at the location $x = x_1$ is given by:

$$SNR\ increase\ at\, x_1 = \frac{i}{\sigma_n}\frac{\sqrt{2}}{\pi^{\frac{1}{4}}}\frac{\sigma_c^{\frac{3}{2}}}{(\sigma_c^2+\sigma_s^2)^{\frac{1}{2}}} \tag{B.30}$$

On the other hand the increase in SNR in the other case, due to region based method alone is given by:

$$SNR\ increase\ at\, x_1 = \frac{i}{\sigma_n}\frac{\alpha}{\sqrt{L}} \tag{B.31}$$

Appendix C

Existence of the Nash equilibrium for the Segmentation problem

As mentioned in the Discussions section of Chapter 6, this appendix discusses the issues regarding the existence of the Nash Equilibrium for the image segmentation problem. The cost functions have the structure defined in (6.8) and the modules have the cost function given in (6.47) and (6.49). We shall consider the modules one by one. As we will see, under some restrictions, for the region module, the conditions for the existence of the Nash equilibrium are satisfied. However, for the boundary module, for the general 2D case verifying them is extremely difficult. For the equivalent 1-D case, we prove that the requirements are met. We then try to extend those arguments qualitatively to the 2D case, without actually proving them.

C.1 Region module

The objective function for this module is given by the equation (6.47).

Now, as for the boundedness requirement, since we are dealing in discrete images with finite intensity values, it is always met. Also since all the terms in the equation are positive semidefinite (being quadratic), the minimum value is zero and the maximum value is limited by the maximum level over which the intensity function can fluctuate.

We can also see that all the terms in the above function are quadratic in the different elements i.e. $x_{i,j}$, thus they are indeed continuously differentiable with respect to each one of those variables.

The same characteristic ensures strong convexity. Hence, all requirements of the theorem are met.

C.2 Boundary module

The objective function for this module is given by the equation (6.49). This module is mathematically much more complicated than the previous model. So what we shall do is that we will do the analysis for the equivalent 1-D model, and then try to extend the arguments to a 2D case as has been done for example in [21].
Let us assume the $1-D$ input is given by:

$$r(x) = s(x) + n(x) \tag{C.1}$$

where $s(x)$ and $n(x)$ represent the signal and the noise component of the input, where as usual, we shall assume zero mean additive white Gaussian noise. This represents the cross section in the x-direction, for any fixed y. We note here that here x corresponds to the location of the edge and is thus equivalent to the existence of a boundary in a 2D image. The intensity values are modeled by the signal level $r()$. If we are in the vicinity of an edge and the X-axis is aligned in the direction perpendicular to the edge, as in [59], $r(x)$ can be approximated by a Gaussian step edge profile:

$$r(x) = I\Phi\left(\frac{x-l}{\sigma_s}\right) + n(x) \tag{C.2}$$

where

$$\Phi\left(\frac{x}{\sigma}\right) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x} exp\left(-z^2/2\sigma^2\right)dz \tag{C.3}$$

I is the intensity of the edge, l is the location of the edge and σ_s is the blur of the Gaussian edge.

For the 1-D problem the question reduces to estimating l, the location of the edge. If $x \in (-L, L)$ and if $i(x)$ represents the segmented image (in the region sense) then the equivalent of (6.49) is given by:

$$\max_x \left[\phi'_{\sigma_c}(x) * r(x) \; + \; \alpha \int_x^L i(z)dz\right] \tag{C.4}$$

where, ϕ_{σ_c} is the Gaussian smoothing kernel.

$$\phi_{\sigma_c}(x) = \frac{1}{\sqrt{2\pi}\sigma_c} exp(-x^2/2\sigma_c^2) \tag{C.5}$$

The first term in the above equation represents the gradient term, which looks for the maxima of the gradient. As is normally the case, here also, we convolve with the gradient of the Gaussian and look for the maxima of the resultant output.

Taking derivatives is in general not well defined, but convolution with a Gaussian makes it well defined [128]. This assures that the first term is continuously second order differentiable.

As for the other requirement, we need to show concavity rather than convexity since this is a case of maximization rather than than minimization. If we assume that the width of the smoothing kernel large enough, i.e. if we assume that the limits of integration in the convolution $(-T_x, T_x)$ are large enough, then we may ignore the effects of noise in the first term as we had assumed that the noise is zero-mean. Without loss of generality we shall also assume that the edge we are looking for is located at $x = 0$, i.e. $l = 0$. Now, if we take the second derivative, of the first term in (C.4) we get due to convolution with the Gaussian,

$$\phi_{\sigma_c}^{'''}(x) * r(x) = \int_{-T_x}^{T_x} \phi_{\sigma_c}^{'''}(y) I \Phi(-y/\sigma_s) dy \tag{C.6}$$

Using integration by parts, and assuming that $T_x \gg max(\sigma_c, \sigma_s)$, it can be shown that

$$\phi_{\sigma_c}^{'''}(x) * r(x) = -\frac{1}{\sqrt{2\pi}(\sigma_c^2 + \sigma_s^2)^{\frac{3}{2}}} \tag{C.7}$$

Since the right hand side of the above equation is assuredly negative, the concavity condition is met.

For the second term in (C.4) we can assume that $i(x)$ the output of the region based procedure is given by:

$$i(x) = s(x) + m(x) = I\Phi\left(\frac{x}{\sigma_s}\right) + m(x) \tag{C.8}$$

where, $m(x)$ is a derived process from the original AWGN, $n(x)$. If we ignore the effect of the residual noise process, $m(x)$, then once again the second term in (C.4) is continuously differentiable and the second derivative is given by:

$$-I\phi_{\sigma_s}(x) = -\frac{I}{\sqrt{2\pi}\sigma_s} exp(-x^2/2\sigma_s^2) < 0 \tag{C.9}$$

Hence, this term is also concave. Since, the sum of two concave functions is a concave function as well,we see that in the simple 1-D case, the requirements of the existence of a Nash equilibrium are met.

In the 1-D case, the problem was reduced to basically finding the location of the step edge. However, in two dimensions, both the location as well as the orientation is important. Since the whole contour is represented as a collection of points, it is equivalent to locate the best edge around the initial placement of the contour. If all the other points remain constant, locating the position of one particular point can actually be reduced to the 1-D approximation already discussed. This is so as we can approximately always detect the orientation of the edges in the same way as done in [21] using a directional operator, and then search in that direction. As long as the points on which the search is performed are not too far away from the optimum locations, and if they are not too close to each other then the objective function in (6.49) can be broken down to an array of smaller functions similar to the 1-D problem discussed. Since, the 1-D problem was shown to be strictly concave we can expect a similar behavior from the actual 2D function (6.49). However, we must emphasize that the above is not intended to be a proof. It only indicates a possibility of existence and uniqueness of the Nash equilibrium, which is true for the 1-D case.

Bibliography

[1] L. Alvarez, P.L. Lions, and J.M. Morel. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal of Numerical Analysis*, 29:845–866, 1992.

[2] A.A. Amini, S. Tehrani, and T.Weymouth. Using dynamic programming to minimize the energy of active contours in the presence of snakes. *Proceedings of the International Conference on Computer Vision*, 1988.

[3] A.A. Amini, T.E. Weymouth, and R.C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:855–867, September 1990.

[4] J. Babaud, A. Witkin, M. Baudin, and R. Duda. Uniqueness of the gaussian kernal for scale space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 1986.

[5] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice Hall, Englewood Cliffs, 1982.

[6] R. Bartels, J. Beatty, and B. Barsky. *An introduction to splines for use in computer graphics and geometric modeling.* Morgan Kaufman, Los Altos, CA 94022, 1987.

[7] T. Basar. An equilibrium theory for multiperson decision making with multiple probabilistic models. *IEEE Transactions on Automatic Control*, 30:118–132, 1985.

[8] T. Basar and G.J. Olsder. *Dynamic noncooperative game theory.* Academic Press, 1982.

[9] P. Baxandall and Hans Liebeck. *Vector Calculus.* Oxford University Press, 1986.

[10] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer Verlag, 1985.

[11] J. Besag. Spatial interaction and statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–225, 1974.

[12] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48:259–302, 1986.

[13] A. Blake and A. Yuille. *Active Vision.* MIT Press, Cambridge, MA, 1992.

[14] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, MA, 1987.

[15] J. Blom. *Topological and Geometrical Aspects of Image Structure.* PhD thesis, University of Utercht, Holland, 1992.

[16] E. Borel. The theory of play and integral equations with skew symmetrical kernels. On games that involve chance and skill of the players. On systems of linear forms of skew symmetric determinants and the general theory of play. *Econometrika*, 21:97–117, 1953. Trans. by L. J. Savage.

[17] H.I. Bozma. *Decentralized Integration in Modular Systems Using a Game-Theoretic Framework.* PhD thesis, Yale University, 1992.

[18] H.I. Bozma and J.S. Duncan. A game-theoretic approach to integration of modules. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1074–1086, 1994.

[19] M. E. Brummer, R. M. Mersereau, R. L. Eisner, and R. Lewine. Automatic detection of brain contours in MRI data sets. In A. Colchester and D. Hawkes, editors, *Information Processing in Medical Imaging*, pages 188–204. Springer-Verlag, Berlin, 1991.

[20] P.J. Burt, T.H. Hong, and A. Rosenfeld. Segmentation and estimation of region properties through co-operative hierarchical computation. *IEEE Transactions on System, Man and Cybernetics*, 11:802–809, 1981.

[21] J.F. Canny. A computational approach to edge-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[22] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, to appear.

[23] A. Chakraborty and J.S. Duncan. Integration of boundary finding and region-based segmentation using game theory. *XIVth International Conference on Information Processing in Medical Imaging*, pages 189–200, 1995.

[24] A. Chakraborty, L.H. Staib, and J.S. Duncan. Deformable boundary finding influenced by region homogeneity. *Proc. Computer Vision and Pattern Recognition*, pages 624–627, 1994.

[25] A. Chakraborty, L.H. Staib, and J.S. Duncan. An integrated approach to boundary finding in medical images. *Proc. IEEE Workshop on Biomedical Image Analysis*, pages 13–22, 1994.

[26] A. Chakraborty, L.H. Staib, and J.S. Duncan. Deformable boundary finding in medical images by integrating gradient and region information. *IEEE Transactions on Medical Imaging*, submitted.

[27] A. Chakraborty, M. Worring, and J.S. Duncan. On multi-feature integration for deformable boundary finding. *Proceedings of the International Conference on Computer Vision*, pages 846–851, 1995.

[28] R. Chellappa. *Two dimensional discrete Gaussian Markov random field models for image processing.* Progress in Pattern Recognition 2, L.N. Kanal and A. Rosenfeld Ed., 1985.

[29] C.C. Chu and J.K. Agarwal. The integration of image segmentation maps using region and edge information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1241–1252, 1993.

[30] H. E. Cline, W. E. Lorensen, R. Kikinis, and F. Jolesz. Three-dimensional segmentation of MR images of the head using probability and connectivity. *Journal of Computer Assisted Tomography*, 14(6):1037–1045, Nov./Dec. 1990.

[31] F.S. Cohen and D.B. Cooper. Simple parallel hierarchical and relaxation algorithms for segmentating noncausal markovian random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:195–219, 1987.

[32] I. Cohen, L.D. Cohen, and N. Ayache. Using deformable structures to segment 3D images and infer differential structures. *Computer Vision, Graphics and Image Processing: Image Understanding*, 13:242–263, 1992.

[33] Laurent D. Cohen. On active contour models and baloons. *Computer Vision, Graphics and Image Processing: Image Understanding*, 53:211–218, 1991.

[34] Laurent D. Cohen and Issac Cohen. Finite element methods for active contour models and baloons for 2d and 3d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15, 1993.

[35] L.D. Cohen, E. Bardinet, and N. Ayache. Surface reconstruction using active contour models. *Proc. SPIE-93 Conference on Geometric Methods in Computer Vision*, 1993.

[36] T. Cootes, A. Hill, C. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. In H. H. Barrett and A. F. Gmitro, editors, *Information Processing in Medical Imaging*, pages 33–47. LNCS 687, Springer-Verlag, Berlin, 1993.

[37] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models - Their training and application. *Computer Vision, Graphics and Image Processing: Image Understanding*, 61:38–59, 1995.

[38] A. Dempster, N.M. Laird, and D.B. Rubin. Maximim likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[39] H. Derin and W.S. Cole. Segmentation of textured images using gibb's random field. *Computer Vision, Graphics and Image Processing*, 35:72–98, 1986.

[40] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using gibb's random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:39–55, 1987.

[41] H. Derin, H. Elliott, R. Cristi, and D. Geman. Bayes smoothing algorithm for segmentation of binary images modeled by markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:707–720, 1984.

[42] P.J. Ell and B.L. Holman. *Computed Emission Tomography.* Oxford University Press, Oxford, 1982.

[43] L. Florack. Scale and differential structure of images. *Image and Vision Computing*, 10:376–388, 1992.

[44] D.S. Fritsch, S.M. Pizer, B.S. Morse, D.H. Eberly, and A. Liu. The multiscale medial axis and its application in image registration. *Pattern Recognition Letters*, 15:445–452, 1994.

[45] K.S. Fu and J.K. Mui. A survey of image segmentation. *Pattern Recognition*, 13:3–16, 1981.

[46] Davi Geiger and Alan Yuille. A common framework for image segmentation. *International Journal of Computer Vision*, 6:227–243, 1991.

[47] D. Geman and S. Geman. Stochastic relaxation, gibbs distribution and bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[48] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:609–628, 1990.

[49] G. Gerig, O. Kubler, R. Kikinis, and F.A Jolesz. Nonlinear anisotropic filtering of mri data. *IEEE Transactions on Medical Imaging*, 11:221–232, 1992.

[50] G. Gerig, J. Martin, R. Kikinis, O. Kubler, M. Shenton, and F. A. Jolesz. Automating segmentation of dual-echo MR head data. In A. Colchester and D. Hawkes, editors, *Information Processing in Medical Imaging*, pages 175–185. Springer-Verlag, Berlin, 1991.

[51] C.R. Giardina and F.P. Kuhl. Accuracy of curve approximation by harmonically related vectors with elliptic loci. *Computer Graphics and Image Processing*, 6:277–185, 1977.

[52] W.E.L. Grimson. Recognition of object families using parametrized models. *Proceedings of the International Conference on Computer Vision*, pages 93–101, 1987.

[53] J.F. Haddon and J.F. Boyce. Image segmentation by unifying region and boundary information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:929–948, 1990.

[54] F.R. Hansen and H. Elliott. Image segmentation using simple markov random field models. *Computer Graphics and Image Processing*, 20:101–132, 1982.

[55] R.M. Haralick and L.G. Shapiro. Survey, image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.

[56] Y.C. Ho. Team decision theory and information structures. *Proc. IEEE*, 68:644–654, 1980.

[57] J.J. Hopfield. Neurons with graded response have collective properties like those of two-state neurons. *Proceedings, National Academy of Sciences, USA*, 81:3088–3092, 1984.

[58] A. Hummel, B. Kimia, and S. Zucker. Deblurring gaussian blur. *Computer Vision, Graphics and Image Processing*, 38:66–80, 1987.

[59] R. Kakarala and A.O. Hero. On achievable accuracy in edge localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:777–781, 1992.

[60] L.V. Kantorovich and G.P. Akilov. *Functional Analysis.* Permagon Press, 1982.

[61] R.L. Kashyap and R. Chellappa. Estimation and choice of neighbors in spatial interaction models of images. *IEEE Transactions on Information Theory*, 29:60–72, 1983.

[62] M. Kass, A. Witkin, and D. Terzopoulus. Snakes: Active contour models. *International Journal of Computer Vision*, 1:312–331, 1988.

[63] S. Kichenassamy, P. Olver, A. Kumar, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. *Proceedings of the International Conference on Computer Vision*, pages 810–815, 1995.

[64] L. Kitchen and A. Rosenfeld. Grey level corner detection. *Pattern Recognition Letters*, 1:95–102, 1982.

[65] J. Kittler and J. Illingworth. On threshold selection using clustering criteria. *IEEE Transactions on System, Man and Cybernetics*, 15:652–655, 1985.

[66] C. Koch, A. Yuille, and C. Marroquin. Analog neuronal networks in early vision. *Proc. Natl. Academy of Science*, 83:4263–4267, 1985.

[67] J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[68] R. Kohler. A segmentation based on thresholding. *Computer Vision, Graphics and Image Processing*, 15:319–338, 1981.

[69] F.P. Kuhl and C.R. Giardina. Elliptic fourier features of a closed contour. *Computer Graphics and Image Processing*, 18:236–258, 1982.

[70] H.W. Kuhn and A.W. Tucker eds. *Contributions to the theory of games, vol 2, Annals of Mathematics Studies.* Princeton University Press, Princeton, New Jersey, 1950.

[71] H.W. Kuhn and A.W. Tucker eds. *Contributions to the theory of games, vol 2, Annals of Mathematics Studies.* Princeton University Press, Princeton, New Jersey, 1953.

[72] S. Kumar, S. Han, D. Goldgof, and K. Bowyer. On recovering hyperquadrics from range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:1079–1083, 1995.

[73] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of gibbs random fields using simulated annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:799–813, 1989.

[74] K. Lange, M. Bahn, and R. Little. A theoretical study of some maximum likelihood algorithms for emission and transmission tomography. *IEEE Transactions on Medical Imaging*, 6:106–114, 1987.

[75] R. Leahy, T. Herbet, and R. Lee. Application of markov random fields in medical imaging. *Information Processing in Medical Imaging*, pages 1–14, 1989.

[76] Y.G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.

[77] M.D. Levine and A. Nazif. An optimal set of image segmentation rules. *Pattern Recognition Letters*, 1:417–422, 1984.

[78] M.D. Levine and A.M. Nazif. Dynamic measurement of computer generated image segmentation. *IEEE Transactions on System, Man and Cybernetics*, 7:155–164, 1985.

[79] S. Li and T. Basar. Distributive algorithms for the computation of noncooperative equilibria. *Automatica*, 23:523–533, 1987.

[80] A. Liu, S. Pizer, D. Eberly, B. Morse, J. Rosenman, E. Chaney, E. Bullitt, and V. Carrasco. Volume registration using the 3D Core. *Visualization Biomed. Computing 1994, Proc. SPIE 2359*, pages 217–225, 1994.

[81] R.C. Luo and M.G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on System, Man and Cybernetics*, 19, 1989.

[82] R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:158–175, 1995.

[83] B.S. Manjunath and R. Chellappa. Unsupervised texture segmentation using markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:478–482, 1991.

[84] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Royal Society, London*, B207:187–217, 1980.

[85] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. *Proceedings of the International Conference on Computer Vision*, pages 840–845, 1995.

[86] J.C.C. Mckinsey. *Introduction to the theory of games*. Mc-Graw Hill, New York, 1952.

[87] S. Menet, P. Saint-Marc, and G. Medioni. B-snakes: Implementation and application to stereo. *Proc. DARPA Image Understanding Workshop*, pages 720–726, 1990.

[88] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculation by fast computing machines. *J. Chemical Physics*, 21:1087–1091, 1953.

[89] U. Montaneri. On the optimal detection of curves in noisy pictures. *Communications ACM*, 14:335–345, 1971.

[90] D. Mumford. Bayesian rationale for the variational formulation. In B. H. Romeny, editor, *Geometry Driven Diffusion in Computer Vision*, pages 135–146. Kluwer, 1994.

[91] D. Mumford and J. Shah. Boundary detection by minimizing functionals. *Proceedings, IEEE Conf. Computer Vision and Pattern Recognition*, page 22, 1985.

[92] Y. Nakagawa and A. Rosenfeld. Some experiments on variable thresholding. *Pattern Recognition*, 11:191–204, 1979.

[93] J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior.* Princeton University Press, Princeton, New Jersey, 1947.

[94] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[95] N. Otsu. A threshold selection method from grey level histograms. *IEEE Transactions on System, Man and Cybernetics*, 9:62–66, 1979.

[96] G. Owen. *Game Theory.* Academic Press, 1982.

[97] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 1993.

[98] T.N. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Transactions on Signal Processing*, pages 901–914, 1992.

[99] T. Pavlidis and Y. Liow. Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:225–233, 1990.

[100] A.P. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4:107–126, 1990.

[101] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffussion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[102] E. Persoon and K.S. Fu. Shape description using fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:388–397, 1986.

[103] K. Rao and R. Nevatia. Computing volume descriptions from sparse 3d data. *International Journal of Computer Vision*, 2:33–50, 1988.

[104] S. P. Raya. Low-level segmentation of 3-D magnetic resonance brain images - A rule-based system. *IEEE Transactions on Medical Imaging*, 9(3):327–337, September 1990.

[105] R.A. Robb. High speed three-dimensional x-ray computed tomography. *Proc. IEEE*, 71:308–319, 1983.

[106] B. Romeny. *Geometry Driven Diffusion in Computer Vision.* Kluwer, 1994.

[107] A. Rosenfeld and A. Kak. *Digital Picture Processing, 2nd ed.* Academic Press, New York, 2nd edition, 1982.

[108] P.K. Sahoo, S. Soltani, A.K.C. Wong, and Y.C. Chen. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.

[109] J.A. Sethian. A review of recent numerical algorithms for hypersurfaces moving with curvature dependent flows. *Journal of Differential Geometry*, 31:131–161, 1989.

[110] J.A. Sethian and J. Strain. Crystal growth and dendritic solidification. *Journal of Differential Geometry*, 98:231–253, 1992.

[111] B.S. Manjunath T. Simchony and R. Chellappa. Stochastic and deterministic networks for texture segmentation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:1039–1049, 1990.

[112] Franc Solina and Ruzena Bajcsy. Recovery of parametric models from range images: The case for superquadratics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:131–147, 1990.

[113] F. Spitzer. Markov random fields and gibb's ensambles. *American Mathematical Monthly*, 78:142–154, 1971.

[114] M. Spivak. *A Comprehensive Introduction to Differential Geometry.* Publish or Perish, Berkley, CA, 1970.

[115] L.H. Staib. *Parametrically Deformable Contour Models for Image Analysis.* PhD thesis, Yale University, 1990.

[116] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:161–175, 1992.

[117] L.H. Staib and J.S. Duncan. Deformable fourier models for surface finding in 3d images. *Proc. Conference on Visualization in Biomedical Imaging (R.A. Robb ed.)*, pages 90–104, 1992.

[118] L.H. Staib and J.S. Duncan. Model-based deformable surface finding for medical images. *IEEE Transactions on Medical Imaging*, submitted.

[119] G. Szekely, A. Keleman, C. Brechbuhler, and G. Gerig. Segmentation of 3D objects from MRI volume data using constrained elastic deformations of flexible surface models. In N. Ayache, editor, *Lecture Notes in Computer Science: First International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine.* Springer-Verlag, Nice, France, 1995.

[120] T. Taxt, P.J. Flynn, and A.K. Jain. Segmentation of document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1322–1329, 1989.

[121] H. Tek and B.B. Kimia. Image segmentation by reaction diffusion bubbles. *Proceedings of the International Conference on Computer Vision*, pages 156–162, 1995.

[122] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics and Image Processing*, 24:52–96, 1983.

[123] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:413–424, 1986.

[124] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry seeking models for 3d object reconstruction. *International Journal of Computer Vision*, 1:211–221, 1988.

[125] H.L. Van Trees. *Detection, Estimation and Modulation Theory, Vol. 1.* Wiley, New York, 1968.

[126] S.S. Trivedi, G.T. Herman, and J.K. Udupa. Segmentation into three classes using gradients. *IEEE Transactions on Medical Imaging*, 5:116–119, 1986.

[127] B.C. Vemuri, A. Radisavljevic, and C.M. Leonard. Multiresolution stochastic 3d shape models for image registration. *Proc. Information Processing in Medical Imaging*, pages 62–76, 1993.

[128] V.Torre and T.A. Poggio. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:147–163, 1986.

[129] B. Widrow. The rubber mask technique - i and ii. *Pattern Recognition*, 5:175–211, 1973.

[130] A.P. Witkin. Scale-space filtering. *Proceedings, International Joint Conference on Artificial Intelligence*, pages 1019–1022, 1983.

[131] A.C.K. Wong and P.K. Sahoo. A grey level threshold selection method on maximum entropy principle. *IEEE Transactions on System, Man and Cybernetics*, 19:866–871, 1989.

[132] M. Worring, A.W.M. Smeulders, L.H. Staib, and J.S. Duncan. Parametrized feasible boundaries in gradient vector fields. *Information Processing in Medical Imaging*, pages 48–61, 1993.

[133] S.D. Yanowitz and A.M. Bruckstein. A new method for image segmentation. *Computer Vision, Graphics and Image Processing*, 46:82–95, 1989.

[134] A. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction in faces using deformable templates. *International Journal of Computer Vision*, 8:99–111, 1992.

[135] A. Yuille and T. Poggio. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 1986.

[136] A.L. Yuille, D.S. Cohen, and P.W. Hallinan. Feature extraction in faces using deformable templates. *Proceedings, IEEE Conf. Computer Vision and Pattern Recognition*, pages 104–109, 1989.

[137] J. Zhu and J.A. Sethian. Projection methods coupled to level set interface techniques. *Journal of Computational Physics*, 102:128–138, 1992.

[138] S.C. Zhu, T.S. Lee, and A.L. Yuille. Region competition: Unifying snakes, region growing and bayes/MDL for multi-band image segmentation. *Proceedings of the International Conference on Computer Vision*, pages 416–423, 1995.

[139] I.G. Zubal and V.J. Caride. The Technetium-99m-DPTA renal uptake- plasma volume product: A quantitative estimation of glomerular filtration rate. *Journal of Nuclear Medicine*, 33:1712–1716, 1992.