Image Segmentation by Energy and Related Functional Minimization Methods

Eric H. Mason
M.S. in Mathematics, December, 1976, University of Arkansas
B.S. in Electrical Engineering, May, 1975, University of Arkansas

A Dissertation Submitted to The Graduate School at the
University of Missouri – St. Louis
in partial fulfillment of the requirements for the degree
Doctor of Philosophy in Applied Mathematics

July 2006

Advisory Committee

Charles Chui, Ph.D.
Chairperson

Wenjie He, Ph.D.

Haiyan Cai, Ph.D.

Henry Kang, Ph.D.

Abstract

Effective and efficient methods for partitioning a digital image into image segments, called "image segmentation," have a wide range of applications that include pattern recognition, classification, editing, rendering, and compressed data for image search. In general, image segments are described by their geometry and similarity measures that identify them. For example, the well-known optimization model proposed and studied in depth by David Mumford and Jayant Shah is based on an L_2 total energy functional that consists of three terms that govern the geometry of the image segments, the image fidelity (or closeness to the observed image), and the prior (or image smoothness).

Recent work in the field of image restoration suggests that a more suitable choice for the fidelity measure is, perhaps, the l_1 norm. This thesis explores that idea applied to the study of image segmentation along the line of the Mumford and Shah optimization model, but eliminating the need of variational calculus and regularization schemes to derive the approximating Euler-Lagrange equations.

The main contribution of this thesis is a formulation of the problem that avoids the need for the calculus of variation. The energy functional represents a global property of an image. It turns out to be possible, however, to predict how localized changes to the segmentation will affect its value. This has been shown previously in the case of the L_2 norm, but no similar method is available for other norms. The method described here solves the problem for the l_1 norm, and suggests how it would apply to other forms of the fidelity measure.

Existing methods rely on a fixed initial condition. This can lead to an algorithm finding local instead of global optimizations. The solution given here shows how to specify the initial condition based on the content of the image and avoid finding local minima.

Preface

This thesis describes a new kind of mathematical image segmentation. The ideas first began to form in a course taught by Charles Chui, the chairman of my advisory committee. He introduced us to the methods first proposed by Perona and Malik that use a non-isotropic form of the heat equation to enhance images and to the energy functional developed by Mumford and Shah for segmentation. The latter was, to me, a remarkable formulation. The essence of the problem could be expressed in a single minimization. The course was also my first serious exposure to the calculus of variation. It seemed to be a shame to take the elegant formulation and construct the complicated partial differential equations that can, in some situations, lead to a characterization of a solution.

Nevertheless, the topic seemed worth pursuing as a dissertation topic. That research led to the level set formulation put forth by Chan and Vese. They were able to cast the problem in a form that leads to automatic solution in a very practical sense. The calculus of variation and the partial differential equations remain, but the solution is manageable. It is not possible to study their work without finding references to the total variation model for image restoration proposed by Rudin, Osher, and Fatimi. That eventually leads to the recent work in the use of different norms for the fidelity term and to the fast algorithm proposed by Song and Chan that addresses the minimization problem by working directly with the energy functional. The combination of these two ideas is the theme of this thesis. Is it possible to apply the notion of working directly with the energy functional to different norms? In particular, using an integer lattice as the domain of the image, is it possible to measure and minimize the absolute deviation from a centroid instead of with a square error term, and to demonstrate a practical algorithm? The question is answered here in the affirmative.

Acknowledgments

First, I thank my advisor, Charles Chui, who stayed with me even when it looked as if no progress would ever be made, and Wenjie He, also on my committee, who was so helpful in checking the details and in organizing the work. I acknowledge the assistance from my employer of twenty eight years, The Boeing Company. I hope their support through the “Learning Together Program” will prove to be a good investment. At Boeing I have many friends and colleagues working on the F/A-18 program, in Phantom Works, and in the process improvement and software tools groups. They must surely have thought that this task would never be completed. I thank them for their patience and confidence that there would be an end.

Finally, without the encouragement of my family this thesis would never have become a reality. My children Jennifer, Timothy, Benjamin, Elizabeth, and Lydia have been steadfast supporters. Above all others, I thank my wife of over thirty years, Gayle. Without her love and understanding this would still be nothing more than a distant dream.

Contents

Appendices

List of Figures

Chapter 1

Introduction

The field of image segmentation deals with the problem of identifying and isolating those portions of an image that possess like properties. At one level the problem can seem almost trivial because it is a natural part of our ability to recognize specific objects in a scene. However, as the content of an image increases in complexity and possesses more and more detail, the problem becomes more complex. For example, even in noise free images, precise boundaries of a region can be occluded by the interiors of other regions.

The applications of image segmentation typically are to improve the appearance of an image whose quality has been degraded by noise or damage and to identify those parts of the image that may be of additional interest. The first of these is part of the field of image restoration or noise removal. This was one of the early objectives of image processing and is the source of the tools that are widely used today in commercially available image processing packages.

One common application of segmented images is in the field of medicine where the aim is to identify anomalous portions that may have clinical significance. There is also the potential of using information about a particular portion of the image to characterize it and call it to the attention of the diagnostician examining the image. The first step in any such process would be segmenting the image into areas that may require further attention.

Another use has both military and security applications. As with medical images the first step is to segment the image. The segments can then be examined for identification of the content. The aim is to categorize the object perhaps as friend or foe or even to identify a particular individual person or structure from a security video.

Finally, there are potential applications in motion detection. Given a segmented image from a video frame it is possible to compare it with subsequent frames and discover which portions have moved relative to the background.

It is because of this very large potential application space that image segmentation is such an active area of research. There is a much to be gained by advancing the state of the art. Also, many classical mathematical tools are applicable to the field and can be readily adapted and exploited for the purpose. The basic model of an image is simple yet it can be easily molded to different needs.

An image is defined to be a mapping

$$u\colon \Omega \to \mathcal{I} \tag{1.1}$$

The assumptions made about the nature of u, Ω, and $\mathcal{I}$ depend on the application and the theorems to be established. The most common situation is that Ω is a bounded, simply connected subset of $\mathbb{R}^2$, and that $\mathcal{I}$ is a bounded subset of the non-negative reals, $\mathbb{R}^+$. The function u is at least assumed to be measurable. The elements of $\mathcal{I}$ typically represent the intensity of the image at each pixel location. Ω represents the geometry of the object on which the image is rendered. This could include a computer screen or a printed medium.

The discrete form of an image is a mapping on an integer lattice. In practice this is the usual case and the domain Ω is a rectangle.

$$u\colon \{1, 2, \ldots, W\} \times \{1, 2, \ldots, L\} \to \mathbb{R} \tag{1.2}$$

In this form the symbols W and L represent the width and length of the rectangle. For black and white images the range represents the gray scale level at each location (i, j). The level is often represented by a subset of non-negative integers, $\mathbb{Z}^+$. For example, in most of the common image formats in use today $u(i, j) \in \{0, 1, \ldots, 255\}$. Thus, the following definition suffices for many theoretical purposes and for almost all practical representations of a black and white image.

$$u\colon \{1, 2, \ldots, W\} \times \{1, 2, \ldots, L\} \to \mathbb{Z}^+ \tag{1.3}$$

It is this form that is most commonly employed in this thesis. The only modification needed to equation (1.3) to include color images is to increase the dimension of the range of u. Three dimensions would cover the common case of an image with red, blue, and green intensity levels. Using the continuous domain model the formal definition of a segmentation of an image is as follows:

Definition 1. *Let u_0 be a real valued observed image defined on a bounded, open, simply connected subset, Ω, of $\mathbb{R}^2$. A segmentation of u_0 is a finite collection of open subsets of Ω, $R_1, R_2, \ldots, R_n$, and a function u with the following properties:*

1. *The boundary ∂R_l of each R_l has finite length.*

2. *Using $\overline{S}$ to denote the closure of the set S,*

$$\overline{\Omega} = \overline{R_1} \cup \overline{R_2} \cup \cdots \cup \overline{R_n}.$$

3. *$u \colon \Omega \to \mathbb{R}$*

The function u is called the approximating function of the segmentation. Each of the the sets R_l is called a segmentation component.

One method of segmentation that has been studied extensively is the use of an energy functional that is minimized by a good segmentation. This idea began with the well known paper by David Mumford and Jayant Shah [41]. They proposed using the following functional:

$$E(u, K) = \mu^2 \int_\Omega (u - u_0)^2 \, dxdy + \int_{\Omega \setminus K} |\nabla u|^2 \, dxdy + \nu |K|. \tag{1.4}$$

Here u_0 us the observed image, u is a test function that approximates the original image, K is a closed subset of Ω that represents the edges, and $|K|$ is the length of the edge set. The coefficients μ and ν are positive weighting parameters. In terms of definition 1 u is the approximating function and

$$K = (\partial R_1 \cup \partial R_2 \ldots \partial R_n) \setminus \partial\Omega.$$

The general idea with this method is to find a finite number of regions in the image where u varies smoothly and to make the boundaries of the segmentation smooth. The first term in equation 1.4 measures how closely the function u approximates the observed image. For that reason it is called the fidelity term. The second term measures the smoothness of the function u in the regions bounded by the edges. The third term measures the smoothness of the edges by their total length, so it is called the length term. Intuitively, then, a solution to the problem

$$E(u^*, K^*) = \inf_{u,K} E(u, K)$$

is a function u^* that closely approximates the observed image and a collection of regions bordered by K^* with the property that in each region the function u^* has no abrupt changes. It is, however, possible that the value of u^* could have discontinuities at the boundaries of the regions.

Much has been written about both explicit solutions of this problem and about algorithms available to solve it numerically. In the complete formulation it is very difficult to study without making some specific assumptions about the result. In fact, in their paper Mumford and Shah only conjecture that the problem is well posed in the case where u_0 is assumed to be continuous. A problem is said to be well posed if (1) a solution exists, (2) the solution is unique, and (3) the solution depends continuously on the data. The term is applied to differential equations, so in this context it asks that the Euler-Lagrange equation derived from euqation (1.4) be well posed.

The complete statment of this conjecture is, "E has a minimum in the set of all pairs (u, K), u differentiable in each region, K a finite set of points joined by C^1 arcs." There are cases where a solution is known. For example, De Pauw and Smets show in [46] that there are solutions for some cases when Ω is the disk. However, this conjecture remains unsolved.

Mumford and Shah also demonstrate that, in some sense, all three terms in equation 1.4 are necessary. They do this by showing that if any one is removed without some additional restrictions on u the infimum is zero. The argument is brief but instructive. They write:

> Dropping any of these three items, $\inf E = 0$:
>
> without the first, take $u = 0$, $K = \emptyset$;
>
> without the second, take $u = u_0$, $K = \emptyset$;
>
> without the third, take K to be a fine grid of N horizontal and vertical lines, $R_i = N^2$ small squares, $u =$ average of u_0 on each R_i.
>
> The presence of all three terms makes E interesting.

An assumption often made is that the image is constant, or at least has small variation, in each of the regions of the segmentation. In that case the second term vanishes and the segmentation becomes sets of constant intensity. The set K is defined to be the boundary of the regions where u is constant. This implies that the admissible functions, u, are those whose range is a finite set $\{c_1, c_2, \ldots, c_n\}$. It also follows that there will be only a finite number of components in the segmentation which are $R_i = \{x \in \Omega \colon u(x) = c_i\}$. Thus, the requirement in the definition of a segmentation that there be only a finite number of segmentation components is redundant in this case. This is the so called piece-wise constant Mumford Shah model. They use the symbol E_0 for this functional and demonstrate that it is the natural limit of E as $\mu \to 0$. Furthermore, they show that this problem is well posed for contunuos u_0. It is this piece-wise model and a closely related functional that are the subject of this thesis. At first glance this formulation may seem to be at odds with the necessity of all three terms in equation 1.4. The difference is that here the edge set cannot be arbitrarily chosen. It depends on the choice of u. A complete description of the existence theorem can be found in chapter 2. There is another interesting observation about the functional E_0. It completely separates two problems. The fidelity term measures how well u approximates u_0 and the length term is solely about geometry.

In their 2001 paper, Chan and Vese [14] propose an algorithm that uses the level set method developed by Osher and others. They use a regularization of the Heaviside function to write the Euler-Lagrange equation for the energy functional 1.4 with the piece-wise constant assumption. Their original paper only addresses the two-phase model, which assumes that there are only two distinct intensity levels of interest. In a later paper [8] they show how the method can be extended to multi-phase problems when the number of phases is a power of 2.

In [51] and later in [50], Song and Chan give an improvement to the Chan Vese algorithm that eliminates the Euler-Lagrange equations. This avoids the need for any regularization of the Heaviside function because it deals with the energy functional directly. The method also eliminates the somewhat troublesome long processing time of the Chan Vese algorithm as well as the problem of beginning with an assumption of a Lipshitz level set function then having its derivative appear in the Euler-Lagrange equation.

Song and Chan's method is based on the idea that at each pixel it is possible to easily compute the change in the energy functional value if the pixel is moved from one partition to the other. This results in a large improvement in the time required to arrive at a solution. Song and Chan show that the algorithm will converge in just a few sweeps through the image.

In a 2004 paper, Chan and Esedoglu [9] studied the possibility of replacing the fidelity term in equation (1.4) with the L_1 norm. The paper actually focuses on the closely related problem of image restoration, but in a special case the two problems are the same. They suggest that there may be advantages to using that norm instead of the square error term in the original model. However, this change to the model eliminates the solution path of the calculus of variation since the absolute value function in the fidelity term is not differentiable. They show how it is possible to approximate a solution by using a regularization of the fidelity term. In [10] the authors study the use of this norm in the case of binary images and the two-phase piece-wise Mumford Shah model. A binary image takes on only the two values 0 and 1, so in equation (1.1) $\mathcal{I} = \{0, 1\}$. This modification changes the functional to

$$E(u) = \mu \int_{\Omega} |u - u_0| \, dxdy + \int_{\Omega \setminus K} |\nabla u|^2 \, dxdy + \nu |K|. \tag{1.5}$$

In this form the fidelity L_1 norm replaces the L_2 norm and the coefficient is no longer squared to make the units of the parameter the same.

One of the main results in this thesis will show how the problem of using the l_1 norm can be solved directly without the use of regularization. The general idea is as follows: Another name for the fidelity term with absolute value instead of square error comes from statistics. The quantity is known as the absolute deviation. It is well known that the term can be minimized not with the mean, as is the case with a square error, but with the median of the sample.

Song and Chan's method uses a well known recurrence relation for the mean and variance to compute the change in energy at each pixel. No such recurrence relation is readily available for the median and absolute deviation. Such a method is given in chapter 3. It makes it possible to predict the change in energy from a histogram of each segment of the image. Thus, by investing a small amount of memory and time in constructing the histograms it becomes a simple matter to both compute the change in energy and, if necessary move a given pixel from one partition to the other.

Furthermore, theorem 4 in chapter 3 shows that it is also possible to predict a priori what the minimum value of the fidelity term will be and what the values of the range of u should must be to obtain that minimum. This leads to a method of determining an initial condition for the solution of the problem and for determining reasonable values for the parameters μ and ν.

The outline of the thesis is as follows:

Chapter 2 provides some background on the segmentation problem and a description of other approaches.

Chapter 3 develops the basic theory needed to apply the l_1 norm as a fidelity term.

Chapter 4 applies the results of chapter 3 to formulate a solution using the l_1 norm and develops the details of a practical algorithm.

Chapter 5 shows some experimental results.

Chapter 6 draws conclusions and makes suggestions for future work.

Appendix A and appendix B present the computer code used to produce the experimental results.

Appendix C contains some descriptions of the source of the images used and acknowledgments where appropriate

The notation used throughout this thesis is standard. One detail that should be mentioned is the use of the symbols l_2, L_2, l_1, and L_1. When the context involves a continuos image domain the symbols L_1 and L_2 appear. When the context involves an image defined on an integer lattice the symbols l_1 and l_2 are used to indicate that the counting measure is being applied instead of Lebesgue measure.

Chapter 2

Background

This chapter gives some general background on the image segmentation problem as formulated by Mumford and Shah. The focus is on the piece-wise constant model for which there is a reasonable mathematical theory. Next there is a description of two algorithms proposed to solve the problem in the special case where the segmentation is assumed to have only two phases. That is, the range of the approximating image function has a cardinality of two. Following that is an explanation of what the possible benefits are to replacing the usual fidelity term with the L_1 norm. Next is an introduction to the ideas that lead to an algorithm using that norm and a description of the theory available to show the existence of solutions. The chapter concludes with a brief description of other segmentation methods.

2.1 The piece-wise constant model of Mumford and Shah

In [41] Mumford and Shah prove the following theorem:

Theorem 1. *Let Ω be an open rectangle in $\mathbb{R}^2$, and let u_0 be continuous on $\Omega \cup \partial\Omega$. For every one dimensional $K \subset \Omega$ such that $K \cup \partial\Omega$ is made up of a finite number of C^1 arcs meeting only at end points and for all locally constant u on $\Omega \setminus K$ let*

$$E_0(u, K) = \iint\limits_{\Omega} (u - u_0)^2 \, dxdy + \lambda \cdot Length(K). \tag{2.1}$$

Then there exists a u and a K that minimize E_0.

Proof. see [41], pages 613–629. □

In [40] Morel and Solimini prove a more general version of this theorem. Their proof only requires that u_0 be measurable. The theorem as stated here is sufficient for practical applications. Before proving this theorem Mumford and Shah make the following important remark:

> If K has an arc, γ, which is surrounded by a single component of $\Omega \setminus K$ we can reduce E_0 just by removing γ. Hence, we may as well assume in our original formulation that the boundary of each component consists of piece-wise C^1 loops which mutually intersect in only finitely many points.

This situation is shown in figure 2.1. The arc $\overset{\frown}{df}$ is completely surrounded by region R_2. Therefore, if it is removed from the set K the value of the fidelity term is

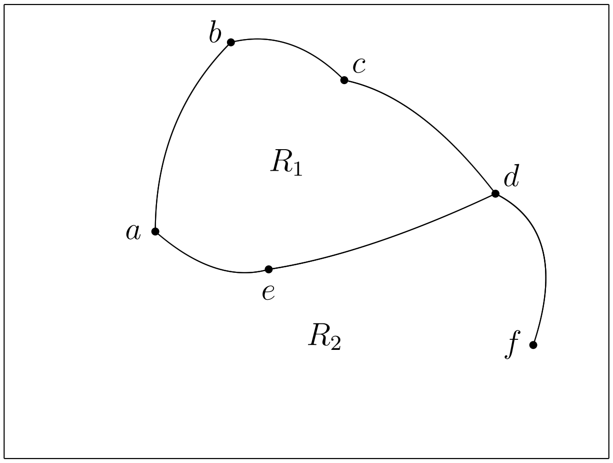


Figure 2.1: A two-component segmentations with an extra arc

unaffected while the total length of K decreases. Since the objective is to find the minimum of E_0 nothing is lost by removing $\widehat{df}$.

Figure 2.2 shows a three component segmentation with the boundaries formulated as piece-wise C^1 loops. The boundary between R_1 and R_2 is formed by the two arcs $\widehat{ae}$ and $\widehat{ed}$. Morel and Solimini [40] call this kind of segmentation *1-normal*. More specifically they write,

> A segmentation will be called 1-normal if it is made of a finite number of rectifiable Jordan curves, meeting each other and $\partial\Omega$ only at their tips and if each Jordan curve separates two different regions.

It is these 1-normal segmentations with which this thesis is primarily concerned.

The examples of boundaries shown thus far have all been interior. That is, they do not intersect with the boundary of the image. The theorem does allow for this possibility. Notice that it requires that $K \subset \Omega$ so that the length of K does not include the boundary of the image. This is as it should be because including it would

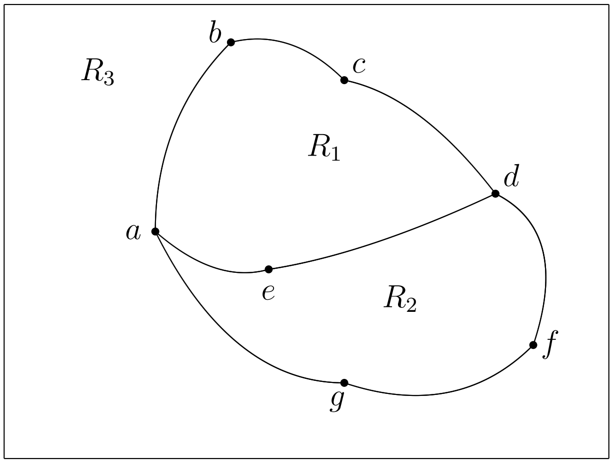


Figure 2.2: A three-component segmentations with piece-wise C^1 loops forming the boundaries of the components

have the undesired result of adding a constant to the length term. It is possible, however, for an arc to terminate at the boundary because it is $K \cup \partial\Omega$ that is made up of C^1 arcs. This arrangement is shown in figure 2.3. The boundaries R_3 and R_4 include the arcs $\widehat{ah}$ and $\widehat{gi}$.

In the literature about this kind of minimization problem one frequently encounters statements about the energy functionals and convexity. This section concludes with a discussion of that property and its role in proving the existence of solutions.

A linear functional $E\colon L \to \mathbb{R}$ on a linear space L is said to be convex if for any $x, y \in L$ and $t \in (0, 1)$

$$E(tx + (1-t)y) \leq tE(x) + (1-t)E(y).$$

E is called strictly convex if this inequality is strict when $x \neq y$. A subset M of L is

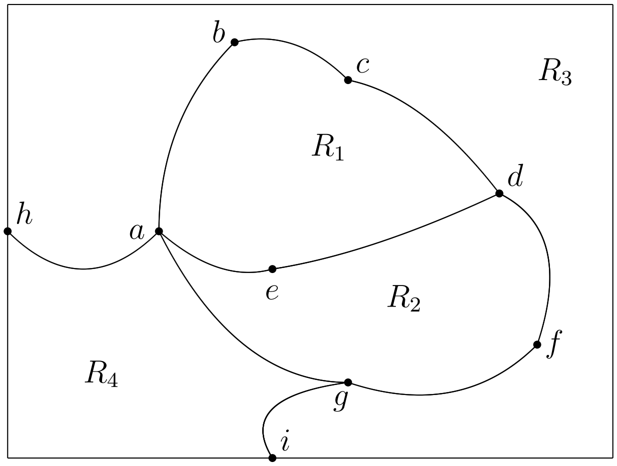


Figure 2.3: A four-component segmentation showing arcs ending at the image boundary, $\partial\Omega$

said to be convex if

$$x \in M \text{ and } y \in M \Rightarrow tx + (1-t)y \in M$$

In [54] the following fundamental theorem is proved.

Theorem 2. *Let M be a convex subset of a linear space L and let $E \colon L \to \mathbb{R}$ be convex. If $x^* \in M$ is a local minimizer of E, i.e., there is a neighborhood U of x^* for which $E(x^*) \leq E(x)$ for each $x \in U$, then x^* is also a global minimizer of E. If E is strictly convex then there is at most one global minimizer.*

The difficulty with applying this theorem to minimizing equation (2.1) is that the energy E_0 is not even linear. The length term depends only on the geometry of the segmentation. That means that if u is the approximating function and $K(u)$ is the implied boundary then

$$\text{Length}\,(K(u)) = \text{Length}\,(K(\alpha u)) \text{ for any nonzero } \alpha.$$

Thus the straightforward path of applying theorem 2 is not available.

2.2 The two-phase algorithm of Chan and Vese

While theorem 1 assures the existence of solutions to the minimization of E_0 it provides no hint at how they are to be found. This chapter now turns to a description of two algorithms described in the literature for a two phase segmentation. Before introducing the ideas a description of the background that leads to the algorithms is in order. The methods are inspired to a large extent by the use of the level set method developed by Osher an Sethian and first published in [44]. A more recent overview is given in [43]. Besides image analysis, level sets have been applied to computational fluid dynamics and crystal growth.

In image processing the idea of a level set is used as at tool to define the regions of an image. Using the notation of equation (1.1) on page 2, assume that Ω is a bounded and simply connected subset of the plane. Define a real valued function ϕ that is at least assumed to satisfy the Lipshitz condition on Ω. The set $C = \{(x, y) : \phi(x, y) = 0\}$ is called the zero level set of ϕ. An example of a level set function is the signed

Euclidean distance to a given curve. That is, if C is a given closed curve in Ω then

$$\phi(x,y) = \begin{cases} +\inf_{(w,z)\in C}(\sqrt{(x-w)^2+(y-z)^2}) & \text{if } (x,y) \text{ is inside } C \\ -\inf_{(w,z)\in C}(\sqrt{(x-w)^2+(y-z)^2}) & \text{if } (x,y) \text{ is outside } C \\ 0 & \text{if } (x,y)\in C. \end{cases} \tag{2.2}$$

If, for example, C is the unit circle $\phi(x,y)$ is a cone with vertex one unit above the origin. A precise definition of *inside* and *outside* is obtained by letting ω be an open subset of Ω with boundary $C = \partial\omega$ of finite length. A point (x,y) is inside C if $(x,y)\in\omega$, and outside C if $(x,y)\in\Omega\setminus\bar{\omega}$. There is no requirement that ω be connected, so C can be the union of several disjoint Jordan curves. The surface defined in equation (2.2) satisfies a Lipshitz condition everywhere as required of a level set function. Chan and Vese used a level set function in [14] to give an algorithm for the 2-phase piece-wise constant Mumford-Shah model.

The idea with the Chan-Vese algorithm is to restrict the function u in equation (1.4) on page 4 to take on only two values. That is, $u(x,y) = c_1$ inside C and c_2 outside C. This reduces the energy functional to

$$\begin{aligned} E(c_1,c_2,C) = & \int_{inside\,C}(u_0(x,y)-c_1)^2\,dxdy \\ & + \int_{outside\,C}(u_0(x,y)-c_2)^2\,dxdy \\ & + \lambda\cdot\text{Length}(C). \end{aligned} \tag{2.3}$$

Notice that in this formulation there is only the parameter λ on the length term. Of course, this makes the energy a function of just the curve C, the constants c_1 and c_2, and the parameter λ. In their paper Chan and Vese also include a term for the area inside C. Moreover, they use separate coefficients for each of the terms. They then point out that in all their numerical results they ignored the area term and used the same coefficient for both the fidelity terms. In all that follows that same approach is taken and only the coefficient on the length term is considered. This is more in keeping with the original formulation of the energy functional which had a single parameter on the fidelity term. This is a nice formulation of the Mumford-Shah functional. It reduces it to a form that lends itself well to the level set method. Instead of the curve C use the level set function ϕ which, as noted above, also defines a curve as its zero level set. The level set function, ϕ, is used in place of the independent variable C to write equation (2.3) as

$$\begin{aligned} E(c_1, c_2, \phi) = & \int_{\phi(x,y)>0} (u_0 - c_1)^2 \, dxdy \\ & + \int_{\phi(x,y)<0} (u_0 - c_2)^2 \, dxdy \\ & + \lambda \cdot \int_\Omega |\nabla H(\phi(x,y))| \, dxdy \\ = & \int_\Omega (u_0 - c_1)^2 H(\phi(x,y)) \, dxdy \\ & + \int_\Omega (u_0 - c_2)^2 (1 - H(\phi(x,y))) \, dxdy \\ & + \lambda \cdot \int_\Omega |\nabla H(\phi(x,y))| \, dxdy. \end{aligned}$$

Here $H(\cdot)$ is the Heaviside function

$$H(z) = \begin{cases} 0, & \text{if } z < 0 \\ \\ 1, & \text{if } z \geq 0 \end{cases}$$

and is used in place of the characteristic function of the interior, ω, of the curve C. That

$$\text{Length}(C) = \int_{\Omega} |\nabla H(\phi(x,y))| \, dxdy$$

follows from the identity proved in [23]

$$\text{Perimiter}(\omega) = \int_{\Omega} |\nabla 1_{\omega}(x,y)| \, dxdy,$$

where $1_{\omega}(x,y)$ is the characteristic function of ω. Now, using standard methods like those described in [56], it is possible to write the Euler-Lagrange equation that gives the minimal energy for fixed c_1 and c_2,

$$\delta(\phi(x,y)) \left[\lambda \, \text{div} \left(\frac{\nabla \phi(x,y)}{|\nabla \phi(x,y)|} \right) - (u_0(x,y) - c_1)^2 - (u_0(x,y) - c_2)^2 \right] = 0, \qquad (2.4)$$

with a Neumann boundary condition of zero. The symbol $\delta(\phi)$ represents the derivative of the Heaviside function.

As Mumford and Shah point out in [41] it is immediate that for fixed C the value of c_1 and c_2 that minimizes the energy is just the mean of u_0 in the regions to which they apply because then the length term in equation (2.3) is just a constant.

Assuming that u_0 permits differentiation under the integral, setting partials to zero to find the extreme points gives

$$\begin{aligned} 0 &= \frac{\partial E}{\partial c_1} = \int_{inside\, C} 2(-c_1)(u_0 - c_1)\, dxdy \\ 0 &= \frac{\partial E}{\partial c_2} = \int_{outside\, C} 2(-c_2)(u_0 - c_2)\, dxdy \end{aligned}$$

which can be solved for c_1 and c_2 showing that equation (2.3) is minimized when

$$\begin{aligned} c_1 &= \int_{inside\, C} u_0\, dxdy \div \int_{inside\, C} dxdy \\ c_2 &= \int_{outside\, C} u_0\, dxdy \div \int_{outside\, C} dxdy \end{aligned} \tag{2.5}$$

Chan and Vese use a regularization of the Heaviside function to develop a numerical scheme for solving the Euler-Lagrange equation (2.4). The steps of the algorithm are as follows:

1. Choose an initial value of the curve C and compute the corresponding level set function ϕ as defined in equation (2.2).

2. Compute $c_1(\phi)$ and $c_2(\phi)$ by finding the mean value of u_0 inside C, where $\phi > 0$, and outside C, where $\phi < 0$.

3. Solve the Euler-Lagrange equation (2.4) with fixed c_1 and c_2 to get a new ϕ.

4. Reinitialize ϕ to be the signed distance to C as in equation (2.2)

5. Check for a stationary solution that indicates convergence,.

This is the algorithm as it was proposed in [14]. It is often cited in the literature and is commonly known as the two phase Chan-Vese algorithm for the solution of the Mumford-Shah model or simply the Chan-Vese algorithm. It was later expanded in [12] and earlier in [13] and [8] to include the case of 2^n phases by incorporating n level set functions. Improvements and modifications are still an active area of study. See, for example, [5] and [26], and [22]. A novel approach to the problem appeared shortly after the publication of the Chan-Vese algorithm. A description of that approach is the next topic.

2.3 The two-phase algorithm of Song and Chan

The second algorithm avoids the use of the calculus of variation. Song and Chan observed in [50] that it was possible to develop an algorithm to minimize the Chan and Vese functional that did not involve the calculus of variation. This is an example of a greedy algorithm as described in [19]. Greedy algorithms seek to find a solution to a global optimization problem by, at each step, making what seems to be the best choice. To be greedy the method of choice at each step may depend on previous steps but not on future ones. The algorithm described here solves the two phase segmentation using the Mumford-Shah model. That is, it solves the same problem as does the Chan-Vese algorithm but with a very different method. The idea follows from two observations about the level set solutions.

1. The level set function, ϕ, in the Chan-Vese algorithm serves primarily to partition the domain of the image. The actual nature of the function is of little

consequence. With that in mind, it should suffice to assume that it takes on only two values, say -1 and 1. This completely eliminates the need to maintain the shape of ϕ as the Euclidean distance to the zero level set.

2. For a given pixel it is possible to predict how the energy will change if the pixel is moved from one partition to another.

It is observation 2 that makes the greedy algorithm possible. To understand the concept recall the well known recursive method of computing the mean of a data sample. If $X = \{x_1, x_2, \ldots, x_N\}$ with mean $\overline{x}$ and if a new element x_{N+1} is added to form a new set $X' = X \cup \{x_{N+1}\}$ then the mean of X' is

$$\overline{x'} = \frac{N\overline{x} + x_{N+1}}{N+1}. \tag{2.6}$$

Similarly, a rather tedious but straight forward calculation shows that if

$$S_N = \sum_{i=1}^{N} (x_i - \overline{x})^2 \text{ then } S_{N+1} = S_N + \frac{N}{N+1}(x_{N+1} - \overline{x})^2. \tag{2.7}$$

Thus it is possible to compute a new value of S_N when an element is added to X without computing a new mean. Equations (2.6) and (2.7) can both be rearranged to compute the result when an element of X is removed. It is the combination of adding and removing an element that Song and Chan exploit.

Replace the integrals in equation (2.3) on page 17 with summations on an integer lattice and write, using observation 1,

$$\begin{aligned} E(c_1, c_2, C) = & \sum_{\phi(i,j)=1} (u_0(i,j) - c_1)^2 \\ & + \sum_{\phi(i,j)=-1} (u_0(i,j) - c_2)^2 \\ & + \lambda \cdot \text{Length}(C). \end{aligned} \tag{2.8}$$

The two summations in (2.8) are in precisely the form of (2.7) because, as shown in equation (2.5) on page 20, once C is fixed they are minimized by the mean pixel value in the partition. Because, as will be shown in section 3.2, the length is also a local property a simple calculation using equation (2.7) makes it possible to determine how the energy will change if a pixel is moved from one partition to the other. With this background the description of the algorithm is as follows:

1. Pick an initial partition of the image, ϕ_0, and compute the corresponding c_1 and c_2.

2. Examine each pixel of the image and determine how the energy will change if the pixel is moved to the other partition.

3. If the energy will decrease then move the pixel and recompute c_1 and c_2 using equation (2.6).

4. Repeat until an entire pass through the image moves no pixels from one partition to the other.

A complete listing of MATLAB code for this is presented in appendix A.

It is the nature of greedy algorithms that there is no guarantee that they will find a global optimum. The result here is that the segmentation found will not necessarily be the best possible. It is the price paid for the large increase in speed.[1] On the one hand this may appear to be a serious flaw. However, as Chan and Vese note, the same thing is possible with their algorithm. They use a special regularization of the Heaviside function that seems to avoid finding local instead of global minima. It would seem evident that a regularization with compact support would more closely represent equation (2.4). Chan and Vese note that this does not seem to be the case. Instead they report that the following regularization is more likely to avoid local minima

$$H_\epsilon(z) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{z}{\epsilon}\right)\right). \tag{2.9}$$

They hypothesize that, because the support is not compact, it can more reliably act on all the level curves of ϕ. Nevertheless, there is no theory available that guarantees a global optimization even with the use of equation (2.9). Other methods have been proposed to deal with the problem of finding local minima. For example, in [59] the authors artificially set the length to a large negative number when it is zero. The hope is that the policy will help a solution to escape from a local minimum. They claim some empirical success, but still no theory is available to avoid the situation. It will be shown in chapter 4 that is possible, with a sound theoretical basis, to predict a priori that there is the potential of a local minimum interfering with a global solution.

[1] Song and Chan claim that a factor of ten increase in speed is possible.

2.4 Image restoration and the L_1 norm

Thus far, all the models have involved a fidelity term of the form

$$F_2(u) = \iint\limits_{\Omega} (u - u_0)^2 \, dxdy = \|u - u_0\|_2^2.$$

It is the purpose of this thesis to explore how this can be replaced with the fidelity term

$$F_1(u) = \iint\limits_{\Omega} |u - u_0| \, dxdy = \|u - u_0\|_1.$$

The notion comes from recent research in the closely related problem of image restoration. The goal of image restoration is to remove the artifacts of noise or possible damage to the original image. The first is commonly called image denoising while the second is known as image inpainting.

Many methods have been proposed for image denoising. The simplest is a low pass filter. This technique convolves the original image with a smoothing kernel to create a new image with a reduced noise level. That is,

$$\begin{aligned} u^*(x, y) &= g(x, y) * u_0(x, y) \\ &= \iint\limits_{\Omega} u(s, t) g(x - s, y - t) \, dsdt \end{aligned} \tag{2.10}$$

The convolution kernel g is typically a smooth bump shaped function such as a

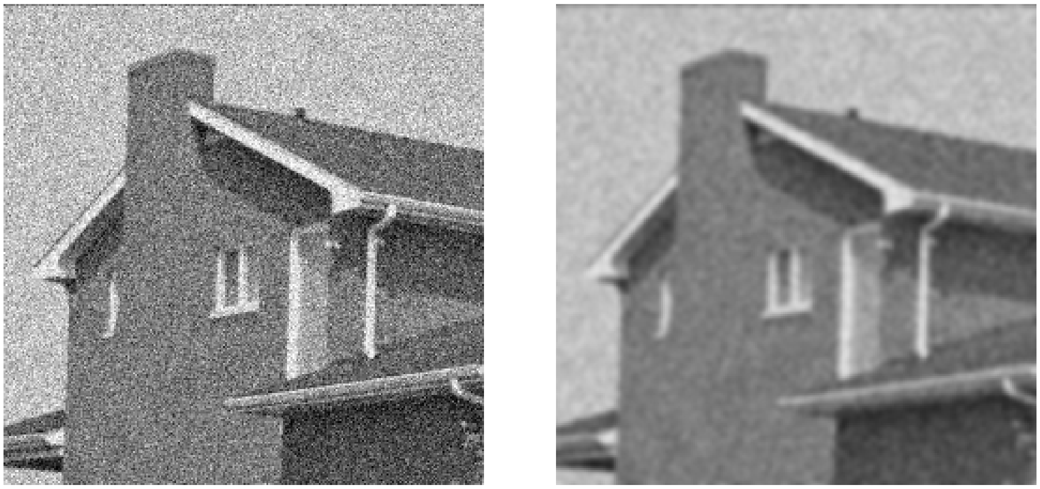


Figure 2.4: Left: A noisy image. Right: Filtered with a Gaussian kernel

Gaussian normalized to have unit volume,

$$g(s,t) = \frac{1}{2\pi\ \sigma^2} \exp\left(\frac{s^2+t^2}{2\sigma^2}\right).$$

While this is often useful in conjunction with other methods it has the undesired effect of blurring the edges and other sharp features in an image. This can be seen in figure 2.4. The noise is reduced, but the edges are blurred as well. This shortcoming leads to the study of non-linear methods of image denoising.

A recent example is proposed by Garnett and others in [27]. They introduce the Rank-Ordered Absolute Differences statistic which is a measure of how close the intensity of a pixel is to its most similar neighbors. Their method successfully removes both Gaussian and impulse noise.

Another method is described in detail in [2] and also in chapter 2 of [40]. These are called anisotropic diffusion methods and are variations of the idea, originally proposed by Perona and Malik in [47], to replace the heat equation with a nonlinear diffusion equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= \operatorname{div}\left(f\left(|\nabla u|\right)\nabla u\right) \\ u(0) &= u_0. \end{aligned} \tag{2.11}$$

The function f in equation (2.11) is smooth, non-increasing, defined on $\mathbb{R}^+$, and has the following additional properties:

$$f(0) = 1; \quad f(s) \geq 0; \quad \lim_{s\to\infty} f(s) = 0.$$

It replaces the usual constant in the isotropic heat equation. Where u is smooth it permits diffusion but limits it when the gradient of u is large near an edge. Thus, it permits noise to be smoothed without disturbing the edges and avoids the blurring caused by equation (2.10). This formulation is an improvement, but there is a subtle difficulty with equation (2.11). It lacks a fidelity term. That means there is nothing to force the solution to remain close to u_0. Instead, a stopping time for the evolution is used that is based on the variance of the noise. An improvement to this method is the total variation method of Rudin, Osher, and Fatimi [48] commonly called the ROF model.

The original formulation of the ROF model is also a minimization problem.

$$\text{Minimize} \iint\limits_{\Omega} \sqrt{u_x^2 + u_y^2}\; dxdy$$

$$\text{Subject to the constraints} \iint\limits_{\Omega} u\; dxdy = \iint\limits_{\Omega} u_0\; dxdy$$

$$\text{and} \iint\limits_{\Omega} \frac{1}{2}(u - u_0)^2\; dxdy = \sigma^2$$

which can be written in an unconstrained form as,[2]

$$\text{Minimize} \quad \iint\limits_{\Omega} (u - u_0)^2\; dxdy + \lambda \iint\limits_{\Omega} \sqrt{u_x^2 + u_y^2}\; dxdy \tag{2.12}$$

This is a satisfying formulation of the noise elimination problem. By way of the constraints it includes a reference to the original image thereby eliminating the objections of the Perona-Malik theory. The first constraint maintains the average value of the image. The second asks that denoised image u approximate the original within the variance, σ^2, of the noise. Rudin, Osher, and Fatimi took this as a given, but there are now methods for estimating the variance (e.g., see [32]) from the observed image, u_0. The term to be minimized in equation (2.12) is called the total variation of u. Notice that is essentially an L_1 norm of derivatives. As stated it requires that u be differentiable, but a more general definition can be found in [23] that removes this restriction and is the basis for the space of functions of bounded variation on Ω.

A great deal has been written about the ROF model, see [7] for example which

[2] All this really means is that the minimization problem is written in a form to use Lagrange multipliers for the solution.

is a summary of recent developments in the field. As noted in that paper there is good reason to consider replacing the fidelity term with the L_1 norm. The question of different norms for the fidelity term is also found in [45] and [20]. Under this formulation the ROF model can be written in unconstrained form as the following problem:[3]

$$\text{Minimize} \quad \iint_{\Omega} |u - u_0| \, dxdy + \lambda \int_{\Omega} |\nabla u| \tag{2.13}$$

where the total variation term has been written in the more customary form, see [23]. Regarding the benefits of this change of the fidelity term the authors of [7] write

> First and foremost, the scaling between the two terms of (2.13) is different from the one in the original ROF model (2.12), and leads to contrast invariance: If u is the solution of (2.13) with u_0 as the given image, then cu is the solution of (2.13) with cu_0 as the given image.

In addition to the contrast invariance property, the authors note that this form leaves a large class of images unmodified. The definitive papers on the idea of using the L_1 norm with the ROF model are [9] and [10]. They list these advantages to using model (2.13) in place of model (2.12):

> 1. The regularization imposed on solutions by the L_1 model is more geometric. By "more geometric" we mean that the regularization process has less dependence on the contrast of image features than on their shapes. Indeed, as some of our analytical results show, the

[3] There is a slight difference between the way the ROF model is written here and how it usually appears in the literature. Here the parameter λ is the coefficient of the total variation term. It is often shown elsewhere as the coefficient of the fidelity term. Chapter 3 will show that when the parameter is with the regularization term it more closely resembles a scale.

L_1 model almost decouples the level sets of the given image from each other and treats them independently of their associated level (grayscale value).

2. As distinct from the standard model, small features in the image maintain their contrast even as the fidelity parameter is lowered, maintaining good contrast until they suddenly disappear.

3. An unexpected consequence of the modification is that it suggests a data driven scale selection technique: It seems possible to identify certain critical values of the parameter λ at which features at the corresponding scale go through a discontinuous change.

The first observation above is of particular interest as segmentation is more of a geometric problem than it is a noise removal problem. Therefore, it is natural to ask, having studied both the Mumford-Shah model for segmentation and the modified ROF model of equation (2.13) how the same idea might apply.

The goal then, is to minimize

$$E_0^{(1)}(u, K) = \iint_{\Omega} |u - u_0| \, dxdy + \lambda \cdot \text{Length}(K), \tag{2.14}$$

again with u restricted to piece-wise constant functions. Notice the use of the symbol $E_0^{(1)}(u, K)$ to distinguish this energy from (2.1). An immediate consequence of this new model is that the claim that for fixed K the value of u in each region is the mean is no longer true. In fact, the differentiation that leads to equation (2.5) is not even possible. The existence of solutions to the problem of non-smooth fidelity terms was studied extensively by Nikolova in [42]. The theory developed there applies to the ROF model where the regularization term is directly a function of the approximating image u. In the case of the Mumford-Shah model regularization comes from the length of the set K which is an independent variable. The subject of using absolute deviations in place of least squares method is described in great detail in [4]. The question of the L_1 norm is also addressed in [35] and [36]. However, if the domain is assumed to be an integer lattice as in equation (1.2) on page 3 the results of chapter 3 show how the minimum can be obtained. Moreover, it will be possible to solve the problem with no regularization of the L_1 norm. All the methods typically in use approximate the L_1 norm with

$$\|u\|_1 = \iint\limits_{\Omega} \sqrt{u^2 + \epsilon}\, dxdy$$

where ϵ is some small constant. That makes it possible to write an Euler-Lagrange equation. Thus any algorithm will be similar in many respects to the Chan-Vese method described above. Another possibility would be to follow a strategy similar to that of Song and Chan and develop a greedy algorithm for the solution of (2.14). It is this algorithm that will be developed in detail in the following chapters.

2.5 Other segmentation methods

Before proceeding to further theoretical development a description of some other methods of image segmentation is appropriate. The most basic methods are those that seek to isolate the significant edges in an image. They are based on the notion that an edge is defined by rapidly changes in the intensity level. They have the advantage that they are simple to implement and require little processing time. These generally fall into three categories.

The first type is based on the magnitude of the gradient of the image. Methods of this type approximate the first partials in both the horizontal and vertical directions with a first difference equation. Various methods are in use. The most basic is, using an integer lattice for the image domain,

$$\begin{aligned} u_x &= \frac{\partial u}{\partial x}(i,j) \approx u(i+1,j) - u(i,j) \\ u_y &= \frac{\partial u}{\partial y}(i,j) \approx u(i,j+1) - u(i,j). \end{aligned}$$

The magnitude of the gradient at the point (i, j) is then

$$|\nabla u(i,j)| = \sqrt{\left(\frac{\partial u}{\partial x}(i,j)\right)^2 + \left(\frac{\partial u}{\partial y}(i,j)\right)^2}.$$

This quantity is then compared with some predefined threshold to determine if a particular point does or does not belong to an edge. The second type is similar, but instead of computing the gradient magnitude at each point they look for zero crossings of the Laplacian, $\nabla^2 u = u_{xx}+u_{yy}$. These methods require a second difference equation. Because of their dependence on first and second differences both gradient magnitude and Laplacian based methods are sensitive to noise in the image in that they frequently identify pixels corrupted by noise as an edge.

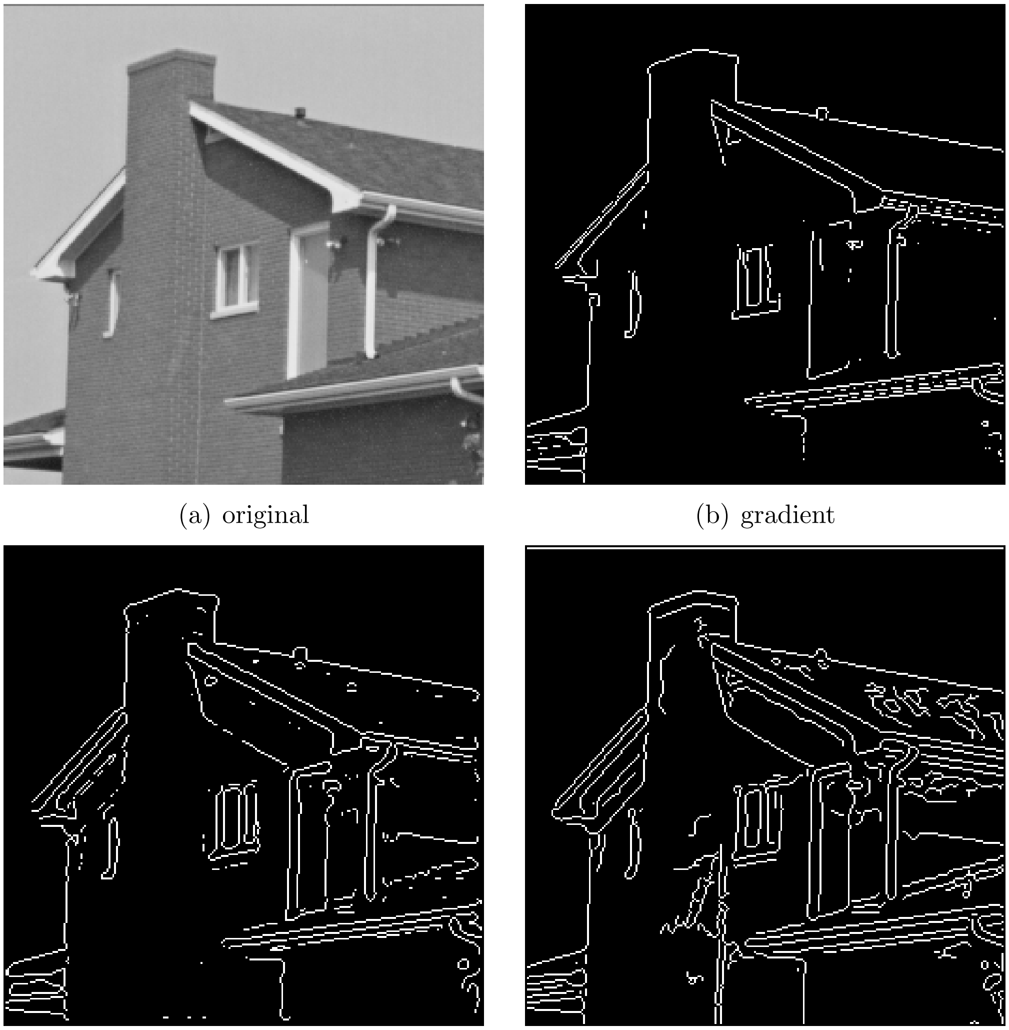
To address this problem Canny [6] proposed an approach that reduces the sensitivity to noise. He does a thorough analysis of the problem. His method is based on two criteria. In the introduction to [6] Canny writes:

> . . . there are common criteria relevant to edge detector performance. The first and most obvious is low error rate. It is important that edges that occur in the image should not be missed and that there be no spurious responses. In all the above cases, system performance will be hampered by edge detector errors. The second criterion is that the edge points be well localized. That is, the distance between the points marked by the detector and the "center" of the true edge should be minimized. This is particularly true of stereo and shape from motion, where small disparities are measured between left and right image or between images produced at slightly different times.

Canny's edge detector is in some sense optimal because it attempts to maximize a measure of these two criteria. Figure 2.5 shows how each method performs on the same image.[4] It is clear from the results the the concept of just detecting edges is of limited value in segmentation. Exactly what is called an edge depends strongly on the method used. Even in an essentially noise free image the results are vastly different. Also, just detecting edges gives no immediate answer to the question of where the similar regions lie.

Another common method of segmentation is based on textures. With textures

[4] All of these edge images were create with version 7 of MATLAB by the *edge* function using method 'Sobel', 'log', and 'Canny' with the default thresholds.



(a) original

(b) gradient

(c) Laplacian

(d) Canny

Figure 2.5: Edge detector examples

the pixels from a gray scale image are transformed to a different domain before the segmentation is begun. This has been studied in [33], [53], [57], and elsewhere. Textures are regions of the image characterized, not by similar intensity levels, but by similar correlated spatial variation. Typically the textures are obtained by applying a Gabor type filter to the image. Such filters are described in [15]. They are a special class of short time Fourier transforms. In the time domain they take the form

$$(G_\phi f)(b,\xi) = \int_{-\infty}^{\infty} f(t)e^{-j\xi t}\phi(t-b)\,dt$$

where $\phi(\cdot)$ is a Gaussian windowing function. In any region, whose minimal size is governed by the uncertainty principle, this transform exposes the frequency content of the signal $f(t)$. The same thing applies to spatial frequency when a two dimensional image is used with an appropriate window function. The values of the transform then represent the textures of the image and segmentation methods can be applied to the transformed image.

Other recent developments include the statistical approaches of [24], [30], [55], [52]. Also, segmentation of hand written documents by a scale space approach is studied in [38], and in [16] an edge detection scheme is combined with quadrilateral approximations to the edges to identify the boundaries. In [29] the authors apply graph partitioning to segmentation. It is evident that there is an extensive literature on the subject of segmentation and a variety of methods are available. With this background the focus now turns to the energy functional (2.14) and methods of solution.

Chapter 3

l_1 Theory

The rest of this thesis focuses on the following problem: Given an observed image u_0 defined in some region Ω find a piecewise constant image u that minimizes

$$E_1(u) = \iint\limits_{\Omega} |u - u_0| \, dxdy + \lambda \cdot |K(u)| = \|u - u_0\|_1 + \lambda \cdot |K(u)| \tag{3.1}$$

which is essentially the same problem described in equation (2.14) on page 30. The image u is restricted to have a finite range set. The quantity $|K(u)|$ represents the length of the boundary between the regions defined by u. This chapter contains a section devoted to each of the two terms in (3.1).

3.1 Minimizing the l_1 Fidelity Term

As noted in chapter 1 there are advantages to being able to predict the value of the fidelity term in equation 3.1. This section is a departure from image segmentation to prove a general theorem that properly belongs in the field of discrete mathematics. It can be used to efficiently find the minimum possible value of the fidelity term when using the l_1 norm. Parts of the theorem proved here are new as are the application to practical uses of a non-smooth norm. It gives a necessary condition that can be used to find the smallest possible value of the fidelity term once the number of different phases in the solution is chosen. The essence of the theorem is this: Consider the quantity

$$S = \sum_{i \in I_1} |x_i - c_1| + \sum_{i \in I_2} |x_i - c_2|$$

where I_1 and I_2 are a partition of some finite index set I and $\{x_i : i \in I\} \subset \mathbb{R}$. If the value of S is to be minimum then c_1 and c_2 must be medians of $Y = \{x_i : i \in I_1\}$ and $Z = \{x_i : i \in I_2\}$ respectively. Moreover, I_1 and I_2 may be chosen so that the largest element of Y does not exceed the smallest element of Z, and it must be that Y and Z are split by $\frac{1}{2}(c_1 + c_2)$. That is, $\max(Y) \leq \frac{1}{2}(c_1 + c_2) \leq \min(Z)$.[1] This result provides an initial condition for algorithms that seek to approximate a set of real numbers by two constants, and suggests a method of dividing a statistical sample into two closely related groups. A generalization of the theorem addresses the case where the number of subsets in the partition is arbitrary. The description and proof of the theorem begins with definitions of important terms.

Definition 2. *Let $X = \{x_1, x_2, \ldots, x_N\}$ be a nonempty finite set of real numbers. A partition of X is a collection of pairwise disjoint nonempty subsets of X, $X_1, X_2, \ldots, X_n$, such that $\cup_{j=1}^{n} X_j = X$. If X is ordered so that $x_1 \leq x_2 \leq \cdots \leq x_N$*

[1] To be completely correct this statement requires a more general use of the word median. This more general definition will follow.

a partition is called ordered if it is of the form

$$X_1 = \{x_1, \ldots, x_{s_1}\}$$

$$X_2 = \{x_{1+s_1}, \ldots, x_{s_2}\}$$

$$\vdots$$

$$X_n = \{x_{1+s_{n-1}}, \ldots, x_N\}.$$

Call two elements X_i and X_j of an ordered partition adjacent if $j = i+1$ or $i = j+1$.

Definition 3. *Let $X = \{x_1, x_2, \ldots, x_N\}$*[2] *be any finite set of real numbers with $x_1 \leq x_2 \leq \cdots \leq x_N$ and $N = 2m+p$ where $p \in \{0, 1\}$. The right median, M_X^+, is the value of x_{m+1}. The left median, M_X^-, is the value of x_{m+p}.*

When N is odd the left and right median are equal and coincide with the usual definition of the median. When N is even the mean of the left and right median is the usual definition of the median. In either case $M_X^- \leq M_X^+$.

Lemma 1. *If the set X is as described in definition 3 then the quantity*

$$D_X^c = \sum_{i=1}^{N} |x_i - c|$$

is minimized by any c satisfying $M_X^- \leq c \leq M_X^+$.[3]

[2] This definition does not require that the elements be unique. In some settings this kind of object is known as a bag. There is no need for this here because the unique elements that form the set are the symbols x_1, x_2, …, and x_N.

[3] Notice that this lemma is a slight generalization of the well known fact from statistics that the absolute deviation, sometimes called the mean deviation, is minimized when c is the median. For example, see [60].

Proof. First observe that it must be that $c \leq x_N$ because if $c > x_N$ then

$$D_X^c = \sum_{i=1}^{N}(c - x_i) > \sum_{i=1}^{N}(x_N - x_i).$$

Likewise, if $c < x_1$ then

$$D_X^c = \sum_{i=1}^{N}(x_i - c) > \sum_{i=1}^{N}(x_i - x_1).$$

Therefore it must be that $x_1 \leq c \leq x_N$, so consider c and c' with $x_n \leq c \leq c' < x_{n+1}$ and $1 \leq n \leq N - 1$. In this case

$$\begin{aligned} D_X^{c'} - D_X^c &= \sum_{i=1}^{n}(c' - x_i) + \sum_{i=n+1}^{N}(x_i - c') \\ &\quad - \big(\sum_{i=1}^{n}(c - x_i) + \sum_{i=n+1}^{N}(x_i - c)\big) \\ &= n(c' - c) - (N - n)(c' - c) \\ &= (2n - N)(c' - c). \end{aligned}$$

This shows that, because $c' - c \geq 0$, D_X^c is non-increasing when $n \leq m$ and non-decreasing otherwise. If N is even D_X^c is is constant whenever $x_m \leq c \leq x_{m+1}$ because $2m - N = 0$. if N is odd $2m - N < 0$ and $2(m + 1) - N > 0$ so D_X^c is non-increasing when $n < m + 1$ and non-decreasing when $n \geq m + 1$. In either case the minimum occurs between the left and right median as required. □

Definition 4. *The minimum value of D_X^c given by lemma 1 is called the total deviation of X and is represented by the symbol D_X.*

The following lemma collects some basic facts about the total deviation that are useful as computational tools and to simplify subsequent proofs. The purpose is to determine the value of the total deviation when arbitrary elements are either added to or removed from the set X. This lemma can be used to give a recursive definition of the total deviation. All that is needed is to add an algorithm to compute new values of the left and right median. Such an algorithm need only adjust the index of the medians after an item is removed or added to the set X. The operation requires no computation at all. Only the logical operations are necessary.

Lemma 2. *If the set X is as described in definition 3 then,*

1. *The total deviation may be written as*

$$D_X = \sum_{i=m+1+p}^{N} x_i - \sum_{i=1}^{m} x_i$$

2. *If a new element, say y, is added to X to form a new set $X' = X \cup \{y\}$ then $D_{X'} \geq D_X$ and*

$$D_{X'} - D_X = \begin{cases} y - M_X^+ & \text{if } y \geq M_X^+; \\ M_X^- - y & \text{if } y \leq M_X^-; \\ 0 & \text{otherwise.} \end{cases}$$

3. *If an element, say x_k, is removed from X to form a new set $X' = X \setminus \{x_k\}$ then*

$D_{X'} \leq D_X$ *and* [4]

$$D_{X'} - D_X = \begin{cases} x_k - M_X^+ & \text{if } x_k \leq M_X^-; \\ M_X^- - x_k & \text{if } x_k \geq M_X^+; \\ 0 & \text{otherwise.} \end{cases}$$

Proof. The sections of the proof follow the statement of the lemma.

1. If N is even then by lemma 1

$$D_X = \sum_{i=1}^{N} |x_i - c|$$

for any c satisfying $x_m = M_X^- \leq c \leq M_X^+ = x_{m+1}$, so

$$D_X = \sum_{i=m+1}^{N} (x_i - c) + \sum_{i=1}^{m} (c - x_i) = \sum_{i=m+1}^{N} x_i - \sum_{i=1}^{m} x_i.$$

If N is odd $c = x_{m+1}$, so

$$D_X = \sum_{i=m+1}^{N} (x_i - x_{m+1}) + \sum_{i=1}^{m} (x_{m+1} - x_i) = \sum_{i=m+2}^{N} x_i - \sum_{i=1}^{m} x_i.$$

2. There are six cases to consider. All use part (1) of the lemma.

[4] Notice the difference between this and the result when an element is added to X. The effect is the difference from the most distant median rather than from the nearest.

N **odd,** $y \leq M_X^-$**:**

$$\begin{aligned} D_{X'} &= \sum_{i=m+1}^{N} x_i - (y + \sum_{i=1}^{m} x_i) \\ &= x_{m+1} - y + \sum_{i=m+2}^{N} x_i - \sum_{i=1}^{m} x_i \\ &= M_X^- - y + D_X \end{aligned}$$

N **odd,** $y \geq M_X^+$**:**

$$\begin{aligned} D_{X'} &= y + \sum_{i=m+2}^{N} x_i - \sum_{i=1}^{m+1} x_i \\ &= y - x_{m+1} + \sum_{i=m+2}^{N} x_i - \sum_{i=1}^{m} x_i \\ &= y - M_X^+ + D_X \end{aligned}$$

N **odd,** $M_X^- \leq y \leq M_X^+$**:** Note that $y = x_{m+1}$ because $x_{m+1} = M_X^- \leq y \leq M_X^+ = x_{m+1}$.

Therefore, $D_{X'} = x_{m+1} + \sum_{i=m+2}^{N} x_i - \sum_{i=1}^{m+1} x_i = \sum_{i=m+2}^{N} x_i - \sum_{i=1}^{m} x_i = D_X$.

N **even,** $y \geq M_X^+$**:**

$$\begin{aligned} D_{X'} &= y + \sum_{i=m+2}^{N} x_i - \sum_{i=1}^{m} x_i \\ &= y - x_{m+1} + \sum_{i=m+1}^{N} x_i - \sum_{i=1}^{m} x_i \\ &= y - x_{m+1} + D_X = y - M_X^+ + D_X \end{aligned}$$

N **even,** $y \leq M_X^-$**:**

$$\begin{aligned} D_{X'} &= \sum_{i=m+1}^{N} x_i - (y + \sum_{i=1}^{m-1} x_i) \\ &= \sum_{i=m+1}^{N} x_i - (y - x_m + \sum_{i=1}^{m} x_i) \\ &= M_X^- - y + D_X \end{aligned}$$

N **even,** $M_X^- \leq y \leq M_X^+$**:** In this case y becomes the median of X' so

$$D_{X'} = \sum_{i=m+1}^{N} x_i - \sum_{i=1}^{m} x_i = D_X$$

3. Again there are 6 cases to consider

N **odd,** $x_k \leq M_X^-$**:**

$$\begin{aligned} D_{X'} &= \sum_{i=m+2}^{N} x_i - \sum_{\substack{i=1\\ i\neq k}}^{m+1} x_i \\ &= \sum_{i=m+2}^{N} x_i - (x_{m+1} - x_k + \sum_{i=1}^{m} x_i) \\ &= x_k - x_{m+1} + D_X = x_k - M_X^+ + D_X \end{aligned}$$

N **odd,** $x_k \geq M_X^+$**:**

$$\begin{aligned} D_{X'} &= \sum_{\substack{i=m+1\\ i\neq k}}^{N} x_i - \sum_{i=1}^{m} x_i \\ &= \sum_{i=m+2}^{N} x_i + x_{m+1} - x_k - \sum_{i=1}^{m} x_i \\ &= x_{m+1} - x_k + D_X = M_X^- - x_k + D_X \end{aligned}$$

N **odd,** $M_X^- \leq x_k \leq M_X^+$**:** (i.e., $x_k = x_{m+1}$)

$$D_{X'} = \sum_{i=m+2}^{N} x_i - \sum_{i=1}^{m} x_i = D_X$$

N **even,** $x_k \leq M_X^-$**:**

$$\begin{aligned} D_{X'} &= \sum_{i=m+2}^{N} x_i - \sum_{\substack{i=1 \\ i\neq k}}^{m} x_i \\ &= \sum_{i=m+1}^{N} x_i - x_{m+1} - (\sum_{i=1}^{m} x_i - x_k) \\ &= x_k - x_{m+1} + D_X = x_k - M_X^+ + D_X \end{aligned}$$

N**even,** $x_k \geq M_X^+$**:**

$$\begin{aligned} D_{X'} &= \sum_{\substack{i=m+1 \\ i\neq k}}^{N} x_i - \sum_{i=1}^{m-1} x_i \\ &= \sum_{i=m+1}^{N} x_i - x_k - (\sum_{i=1}^{m} x_i - x_m) \\ &= x_m - x_k + D_X = M_X^- - x_k + D_X \end{aligned}$$

N **even,** $M_X^- \leq x_k \leq M_X^+$**:** There is nothing to prove here because the case $M_X^- = x_m < x_k < x_{m+1} = M_X^+$ cannot occur.

This completes the proof of lemma 2. □

The above proof is tedious, but straight forward. Much of the difficulty in the proof is the need to deal with the different cases depending on whether N is even or odd. An important feature of lemma 2 is that it eliminates the need to consider the parity of N when studying the properties of the total deviation. Lemmas 1 and 2 together demonstrate that the usual choice of the median for the case when N is even (i.e., the mean of the left median and right median) is somewhat arbitrary. If the absolute deviation is used as a measure of the spread of a distribution any convex combination of x_m and x_{m+1} yields exactly the same absolute deviation. Lemma 2 plays a prominent role in the proof of theorem 3.

Theorem 3. *Suppose that the set X described in definition 3 is partitioned into two subsets Y and Z. Let $S = D_Y + D_Z$. If S is the minimum value over all such partitions then Y and Z can be written as $Y = \{x_1, x_2, \ldots, x_s\}$ and $Z = \{x_{s+1}, x_{s+2}, \ldots, x_N\}$ with $1 \leq s < N$ and $x_s \leq \frac{1}{2}(M_Y^+ + M_Z^-) \leq x_{s+1}$.*[5]

Proof. Let Z and Y be a partition of X that yield the minimum value of S. Such a partition exists because X is a finite set. To distinguish between the elements write $Y = \{y_1, y_2, \ldots, y_r\}$ and $Z = \{z_1, z_2, \ldots, z_s\}$. It is safe to assume, without loss of generality, that $M_Y^- \leq M_Z^-$ for if $M_Y^- > M_Z^-$ then it is also true that $M_Z^- \leq M_Y^-$ and the role of Y and Z can be reversed in all that follows.

Now, suppose that $z_1 \leq M_Y^-$, and consider the effect on S if z_1 is moved to Y.[6] By lemma 2(2) D_Y will increase[7] by $M_Y^- - z_1$ and, by lemma 2(3), D_Z will decrease by $M_Z^+ - z_1$. The net change in S will then be

$$M_Y^- - z_1 - (M_Z^+ - z_1) = M_Y^- - M_Z^+ \leq 0 \text{ because } M_Y^- \leq M_Z^- \leq M_Z^+.$$

It follows that the smallest element of Z can be moved to Y until $M_Y^- \leq z_1$. In fact,

[5] The partition need not be unique as simple examples demonstrate.

[6] The word *moved* as used here is taken to mean the entire process of removing an element from one set, adding it to the other, ordering both and adjusting the indexes.

[7] The words *increase* and *decrease* are used in a weak sense. e.g., increase can also mean remains unchanged.

this can continue until $M_Y^+ \leq z_1$ because, by lemma 2(2), if $M_Y^- \leq z_1 \leq M_Y^+$ then if z_1 is moved to Y there will be no change in D_Y and D_Z can only decrease.

Next, suppose that $y_r \geq M_Z^+$. If y_r is moved to Z then D_Z will increase by $y_r - M_Z^+$ while D_Y will decrease by $y_r - M_Y^-$. The net change in S is

$$y_r - M_Z^+ - (y_r - M_Y^-) = M_Y^- - M_Z^+ \leq 0.$$

So again it follows that the largest element of Y can be moved to Z until $y_r \leq M_Z^+$ and that, in fact, this can continue until $y_r \leq M_Z^-$. Together, these arguments lead to the conclusion that Y and Z can be arranged so that $M_Y^+ \leq z_1$ and $y_r \leq M_Z^-$ and still maintain the minimum value of S.

Now, if $z_1 < y_r$ then y_r can be moved to Z while z_1 is moved to Y because the medians will be unchanged, but both sets will give up an extreme element (i.e., largest or smallest) in exchange for an element closer to the median. This process can continue until $y_r \leq z_1$ which establishes the first conclusion of the theorem.

To prove that the partition must be split as claimed suppose that $y_r > \frac{1}{2}(M_Z^- + M_Y^+)$. If y_r is moved to Z then D_Y decreases by $y_r - M_Y^-$ and D_Z increases by $M_Z^- - y_r$. The net change in S is

$$\begin{aligned} M_Z^- - y_r - (y_r - M_Y^-) &= M_Z^- + M_Y^- - 2y_r \\ &\leq M_Z^- + M_Y^+ - 2y_r \\ &= 2(\frac{1}{2}(M_Z^- + M_Y^+) - y_r) < 0 \end{aligned}$$

contradicting the assumption that S is the minimum. Similarly, if $z_1 < \frac{1}{2}(M_Z^- + M_Y^+)$ and z_1 is moved to Y, D_Z will decrease by $M_Z^+ - z_1$ and D_Y will increase by $z_1 - M_Y^+$. The net change in S is

$$\begin{aligned} z_1 - M_Y^+ - (M_Z^+ - z_1) &= 2z_1 - (M_Z^+ + M_Y^+) \\ &\leq 2z_1 - (M_Z^- + M_Y^+) \\ &= 2(z_1 - \frac{1}{2}(M_Z^- + M_Y^+)) < 0 \end{aligned}$$

again contradicting the minimum nature of S. Taking these together it follows that $y_r \leq \frac{1}{2}(M_Z^- + M_Y^+) \leq z_1$ which completes the proof. □

Theorem 3 shows that, in order to minimize the quantity S, the subsets Y and Z cannot be arbitrarily intertwined. Only partitions of the form $Y_s = \{x_1, x_2, \ldots, x_s\}$, $Z_s = \{x_{s+1}, x_{s+2}, \ldots, x_N\}$ are of interest. Any such partition can be identified with the index s, so define the map

$$S\colon \{0, 1, 2, \ldots, N\} \to \mathbb{R}^+ \text{ by } S(s) = D_{Y_s} + D_{Z_s} \tag{3.2}$$

with the understanding that if $s = 0$ then $Y = \emptyset$, and if $s = N$ then $Z = \emptyset$. That is, s is the number of elements in the set Y_s. With this definition it follows that $S(0) = S(N) = D_X$. Furthermore, for $s \in \{1, 2, \ldots, N-1\}$

$$S(s) = D_{Y_s} + D_{Z_s} \leq D_X = S(0) = S(N). \tag{3.3}$$

That this is the case can be seen by applying lemma 1 and definition 4.

$$\begin{aligned} S(s) &= D_{Y_s} + D_{Z_s} \\ &\leq \sum_{i=1}^{s} |x_i - M_X^+| + \sum_{i=s+1}^{N} |x_i - M_X^+| \text{ (by lemma 1)} \\ &= \sum_{i=1}^{N} |x_i - M_X^+| = S(0) = S(N). \end{aligned}$$

It would be helpful if it were true that $S(s)$ is convex for then the minimum value would occur for some unique range of the argument s. This would give a stopping point in an algorithm used to find the minimum value of $S(s)$. However, this is not the case as the example $X = \{1, 2, 2, 3\}$[8] demonstrates. The definition of $S(s)$ combined with lemma 2 does suggest an efficient way of computing all the values of $S(s)$. First compute the value of $S(0)$. The rest of the values can be computed recursively. The generalization of Theorem 3 includes partitions with more than two subsets of X.

Theorem 4. *Let P_n be the set of all partitions of X of size n, and let $P_n^* \subset P_n$ be the set of all ordered partitions of X of size n. For $p \in P_n$ let $S_p = D_{X_1} + D_{X_2} + \cdots + D_{X_n}$.*

1. *If $S_n = \min_{p \in P_n}(S_p)$ and $S_n^* = \min_{p \in P_n^*}(S_p)$ then $S_n^* = S_n$.*

2. *If p^* is an ordered partition satisfying $S_{p^*} = S_n$ then for any pair of adjacent elements of p^*, say X_i and X_{i+1},*

$$\max(X_i) \leq \frac{1}{2}(M_{X_i}^+ + M_{X_{i+1}}^-) \leq \min(X_{i+1}). \tag{3.4}$$

[8] i.e., $N = 4$ with $x_1 = 1$, $x_2 = x_3 = 2$, and $x_4 = 3$ as in definition 3.

Proof. First note that (2) will follow immediately once (1) is established. For if $p^* \in P^*$ is a partition satisfying $S_{p^*} = S$ and X_i, X_{i+1} are adjacent elements of p^* then inequality (3.4) must be satisfied. Otherwise it would contradict theorem 3 applied to the set $X_i \cup X_{i+1}$.

It only remains then to prove (1). The method of proof is by induction. The case $n = 2$ is established by theorem 3. The induction hypothesis is that for a partition of size $n-1$ an ordered partition can be found that yields S_{n-1}.[9]

Let p be a partition of size n that yields S_n. If necessary, rename the elements of p in such a way that $\max(X_n) \geq \max(X_j)$ for all $j < n$. Now apply the induction hypothesis to $Y = X_1 \cup X_2 \cup \cdots \cup X_{n-1}$ to find an ordered partition $X_1^*, X_2^*, \ldots, X_{n-1}^*$ of Y. If $\max(X_{n-1}) \leq \min(X_n)$ nothing remains to prove because then $X_1^*, X_2^*, \ldots, X_{n-1}^*, X_n$ is an ordered partition of X that yields S_n. Therefore, suppose that for some X_j^*, $\min(X_n) < \max(X_j^*)$. Apply theorem 3 to the set $X_j^* \cup X_n$ to produce the ordered partition X_j^{**}, X_n^* noting that it remains true, by the arguments in the proof of theorem 3, that $\max(X_j^{**}) \leq \min(X_{j+1}^*)$. Now apply the same argument to X_{j+1}^*, X_{j+2}^* and so on until the elements of the partition are exhausted. The result is an ordered partition of size n that yields S_n.

This shows that the induction hypothesis implies that (1) holds for partitions of size n and completes the proof. □

Now observe that equation (3.2) can be extended to include partitions of arbitrary size. The mapping then becomes a function of the $n-1$ integers $s_1, s_2, \ldots, s_{n-1}$

[9] As used here and elsewhere in the proof the phrase "yields S" when applied to a partition p means, of course, that $S_p = S$.

described in definition 2. While the number of possible arguments of $S(s_1, s_2, \dots s_{n-1})$ could still be a very large number it is extremely small compared to the total number of partitions of X.

The number of partitions of size n of a set with N elements is the Stirling number of the second kind, $\mathcal{S}_N^{(n)}$.[10] If $N = 25$ and $n = 3$ then $\mathcal{S}_{25}^{(3)} = 141{,}197{,}991{,}025$.[11] Instead of this large number of partitions only the ordered partitions are of interest. A straight forward calculation shows that there are only 276 such partitions, a reduction of almost 9 orders of magnitude. This example illustrates the benefits of theorem 4 when it is used for very large sets of data. More generally, if n is fixed and $N = n + j$ for $j = 0, 1, \dots$ write,

$$\begin{aligned}\mathcal{S}_N^{(n)} = \mathcal{S}_{n+j}^{(n)} = \frac{1}{n!}\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}k^{n+j} \le \frac{1}{n!}\sum_{k=0}^{n}\binom{n}{k}k^{n+j} \\ \le \frac{1}{n!}\sum_{k=0}^{n}\binom{n}{k}n^{n+j} = \frac{n^{n+j}}{n!}2^n = \frac{2^n}{n!}n^N,\end{aligned}$$

which shows that $\mathcal{S}_N^{(n)}$ is $O(n^N)$ because the coefficient of n^N in the last expression is constant for fixed n. On the other hand, the number of possible arguments in $S(s_1, s_2, \dots s_{n-1})$ is no more than $N(N-1)\dots(N-n+2)$. This expression is $O(N^{n-1})$. Using $|P|$ and $|P^*|$ to denote the cardinality of the sets P and P^* described in theorem 4 this analysis can be summarized by writing

$$|P| = O(n^N) \text{ and } |P^*| = O(N^{n-1}). \tag{3.5}$$

[10] There seems to be no standard notation for Stirling numbers of the second kind. This is the symbol introduced in [1] page 822.

[11] Ibid, page 835.

Therefore, once n is fixed the value of S_n and a partition yielding that value can be found in polynomial time instead of exponential time.

Theorem 4 leads to a form of non-hierarchical clustering. There is a vast literature on this subject. See, for example, [34], [39], [58], and [21]. The goal of clustering is to discover which subsets of a data sample are closely related. Clearly this theorem aids in that process if one uses D_X as the measure of closeness. The most common method employed is Minimum Sum of Squares Clustering (MSSC) which uses a Euclidian measure instead of the absolute value. In the clustering literature the l_1 norm is referred to as the taxicab or Manhattan metric. The name comes from the analogy of distance along streets in a city laid out in a grid. Theorem 4 may seem to be at odds with the results noted in [3] and proved in [28] and [37] that clustering is NP-complete. However, those papers address the more general problem of clustering that uses multi-dimensional values of the data set. The result proved here speaks only to data sets with a single real dimension.

In [37] Krivanek actually proves a theorem in hierarchical clustering, but cites Gonzalez [28] as having proved a similar result for non-hierarchical clustering which is more closely related to theorem 4. Gonzalez proves that, in the case where the objects in the set have two or more dimensions, the problem of finding a minimal partition is NP-complete. In that paper Gonzalez states that Fisher showed in [25] that, in one dimension, the problem can be solved in polynomial time. Fisher's result is similar to theorem 4.

Fisher really says nothing about polynomial time. The concept of NP-complete and polynomial time did not appear until 1971 in Cook's famous paper [18]. Fisher's paper is from 1958. Fisher proves that part 1 of theorem 4 holds if one uses the squared distance as a measure of closeness. Instead of the total deviation of definition 4 he uses,

$$D = \sum_{i=1}^{N} (x_i - \overline{x})^2, \text{ where } \overline{x} \text{ is the mean.}$$

He uses a combinatorial argument similar to 3.5 to show that the number of partitions that must be searched for a minimum is substantially reduced. Fisher uses the word *contiguous* instead of *ordered* to describe partitions that do not overlap, and he speaks of *transferring* elements instead of *moving* them. Other than these differences in terminology his argument is very much like the proof of theorem 3. Of course, his calculation of the change in D when an element is transferred is not the same because the measures of closeness differ. There is no result in Fisher's paper that is analogous to part 2 of theorem 4.

In [31] Hwang proves the following even more general result:

Theorem 5. *If a distance function is used to measure closeness in each partition then an optimal partition is an ordered partition. In the terminology of theorem 4, if* $S_p = D_{X_1} + D_{X_2} + \cdots + D_{X_n}$ *where* $D_{X_i} = \sum_{x_j \in X_i} d(x_j, c_i)$ *and* c_i *is the centroid with respect to the distance function* d *then* $S_n = S_n^*$.[12]

Part 1 of theorem 3 is a clearly a special case of theorem 5. Thus the problem of minimizing the fidelity term for any norm one chooses can be reduced to a search of just the ordered partitions.

This section concludes with one final observation. Theorem 4 shows how to find

[12] The centroid is defined to be any c_i that minimizes $\sum_{x_j \in X_i} d(x_j, c_i)$. As lemma 1 shows the centroid need not be unique.

the partitions with the minimum value of S_p. It is natural to ask what can be said about partitions with the maximum value. The following lemma gives a partial answer to the question.

Lemma 3. *The set of all values of S_p is bounded by the total deviation of X. That is,*

$$\textit{For each } p \in P_n,\ S_p \leq D_X. \tag{3.6}$$

Proof. The proof is similar to the derivation of equation 3.3. If p is any partition of X in P_n then

$$\begin{aligned} S_p &= D_{X_1} + D_{X_2} + \cdots + D_{X_n} \\ &= \sum_{x_i \in X_1} |x_i - M^+_{X_1}| + \sum_{x_i \in X_2} |x_i - M^+_{X_2}| \\ &\quad + \cdots + \sum_{x_i \in X_n} |x_i - M^+_{X_n}|\, , \text{ by definition,} \\ &\leq \sum_{x_i \in X_1} |x_i - M^+_{X}| + \sum_{x_i \in X_2} |x_i - M^+_{X}| \\ &\quad + \cdots + \sum_{x_i \in X_n} |x_i - M^+_{X}|\, , \text{ by lemma 1,} \\ &= D_X \, , \text{ by definition.} \end{aligned}$$

□

This means that the range of the map defined in 3.2 and its natural extension to partitions of arbitrary size includes both the minimum value of S_p and the maximum.

With these results complete it is now possible to see how to apply them to image segmentation. Lemma 1 is analogous to equation (2.5) on page 20. It says that once an edge set K is known the fidelity term is minimized, not by the mean in each region, but by the median. Lemma 2 is analogous to Song and Chan's observation 2 on page 22. A very simple calculation can predict the change in the fidelity if a pixel is moved.

There is even more available now. Recall that the first step in both algorithms described in chapter 2 was to pick an initial condition. Theorem 4 provides a way to pick an initial condition that will minimize the l_1 fidelity term. The function S of equation (3.2) will prove useful, along with lemma 3, in predicting where the local minima are and how to avoid picking an initial condition that finds just a poor optimization. These observations all apply not just to the two phase models, but to any number of phases. To complete the development all that remains is a definition of the length term in equation (2.14) on page 30. That is the subject of the next section.

3.2 Discretizing the length

In chaper 2 the length term of equation 1.4 on page 4 was discussed in some detail with no method given of how it is to be calculated. Of course, one could just observe that the C^1 arcs of theorem 1 are rectifiable and the length of K is just the sum of all the finite number of arcs making up the boundary. This statement answers the mathematical question, but it does little to advance the development of a practical algorithm. This section describes a discretization of the length term which is only moderately difficult to implement and very easy to understand. The method is similar to the edge sets described in [11]. That method associates an *edge element* or "edgel" with each adjacent pixel pair. This discretization, on the other hand, associates with each rectangle bounded by four pixels its contribution to the length of the segmentation boundary K. Thus, if the image is defined on an integer lattice of the (x, y) plane given by $x \in \{1, 2, \ldots, W\}$ and $y \in \{1, 2, \ldots, L\}$ there are $(W-1)(L-1)$ rectangles. Each of these contributes to the length of K in a way that is dependent only on the segmentation componentss in which the corner pixels lie. This makes it just a local property which can be computed quickly. Also, dealing with the rectangles instead of the pixels simplifies dealing with the image boundary because even along the edges of the image the rectangles are well defined.

Figure 3.1 helps to define how the length term is computed. The point C is the center of a typical region. The four points labeled $p_{i,j}, p_{i+1,j}, p_{i,j+1}$, and $p_{i+1,j+1}$ are the four pixels at the corners of the rectangle. Each rectangle contains four possible length segments. Let $R(p_{i,j})$ be the segmentation component to which pixel $p_{i,j}$ is assigned. A length segment is present if two adjacent pixels are in different regions. The segment is assumed to be perpendicular to the line containing the two pixels and to connect that line to the center of the rectangle, C. The length of the each segment is defined to be $\frac{1}{2}$ the spatial spacing of the pixels. Assuming the spatial distance between the pixels is 1, each segment is assigned a length of $\frac{1}{2}$. If, for example, $R(p_{i,j}) \neq R(p_{i+1,j})$, segment $\overline{CD}$ is assigned the value of $\frac{1}{2}$. In all that follows it is assumed that the spatial distance between the pixels is 1 and is the same in both x and y. This need not be the case. An example is an aerial photograph taken at an oblique angle where the ground separation distance of the pixels depends on the

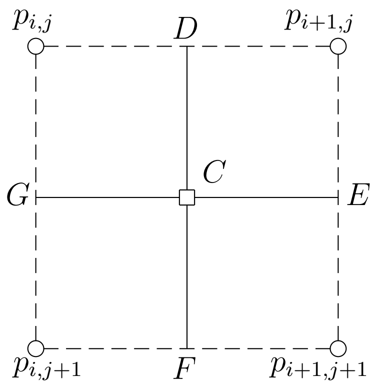


Figure 3.1: Typical rectangle used to calculate the length

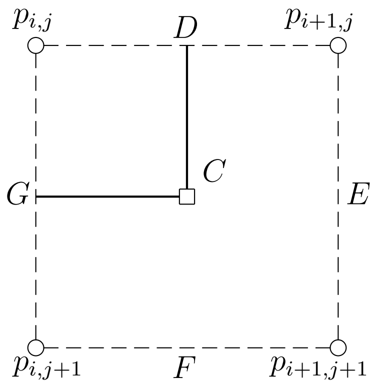
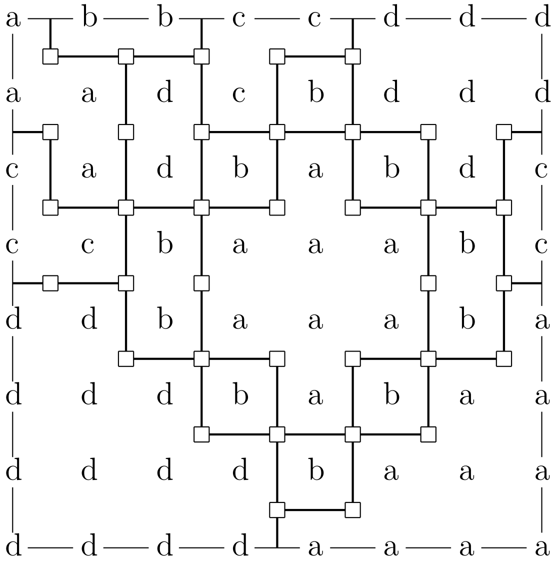


Figure 3.2: An example of a *turn*

geometry when the image was captured. The definition given in this section applies equally well to that case. Care must be taken to give proper attention to the meaning of the length segments.

Figure 3.2 shows the case where $p_{i,j}$ is in a different segmentation component than the other three corner pixels of the rectangle. That is, $R(p_{i,j}) \neq R(p_{i+1,j})$ and $R(p_{i,j}) \neq R(p_{i,j+1})$ but $R(p_{i+1,j}) = R(p_{i+1,j+1}) = R(p_{i,j+1})$. This region contributes 1 to the total length of K.

Figure 3.3 shows an example of small image 8 pixels in height and 8 pixels in width. The labels on the pixels are either a, b, c, or d and indicate the segmentation



a	b	b	c	c	d	d	d
a	a	d	c	b	d	d	d
c	a	d	b	a	b	d	c
c	c	b	a	a	a	b	c
d	d	b	a	a	a	b	a
d	d	d	b	a	b	a	a
d	d	d	d	b	a	a	a
d	d	d	d	a	a	a	a

Figure 3.3: A small image with the entire length term shown

component to which each pixel has been assigned. As in figures 3.1 and 3.2 the small square in the center of each rectangle is the point at which each segment is joined. To reduce clutter in the diagram, if a rectangle contains no length segments the small square in the center is omitted. The heavy lines in the interior regions all count as one unit of the length of the border. Each of these terminates on both ends in the center point of a rectangle. The heavy segments around the boundary of the entire image each count as $\frac{1}{2}$ units. The thin lines forming the boundary are not counted in the length. Thus in the example of figure 3.3 there are 45 segments with length 1 in the interior and 8 segments with length $\frac{1}{2}$ along the image boundary for a total of 49 units of length. The example contains instances of all the possible configurations of the length segments in a rectangle.

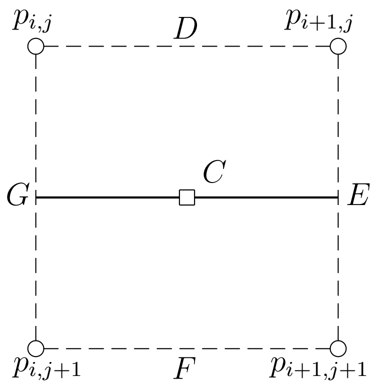


Figure 3.4: An example of a *continuation*

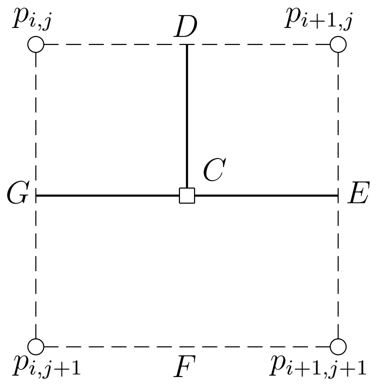


Figure 3.5: An example of a *T-junction*

There are five possible configurations of the boundary inside a given region. One of these is shown in figure 3.2 and is called a turn. If all four pixels at the corners of a region are in the same segmentation region there is no contribution to the boundary. The other three possibilities are shown in figures 3.4, 3.5, 3.6, and 3.7. In [11] these are called a continuation, a T-junction, a mosaic, and a termination, respectively.

The configuration shown in figure 3.5 can be realized when, using the notation of figure 3.3, $R(p_{i,j}) = a$, $R(p_{i+1,j}) = b$, and $R(p_{i,j+1}) = R(p_{i+1,j+1}) = c$.

The configuration shown in figure 3.6 can be realized by grouping each pair diagonal pixels in some component and grouping the other diagonal pair in a different

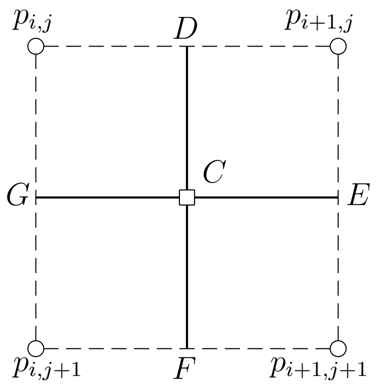


Figure 3.6: An example of a *mosaic*

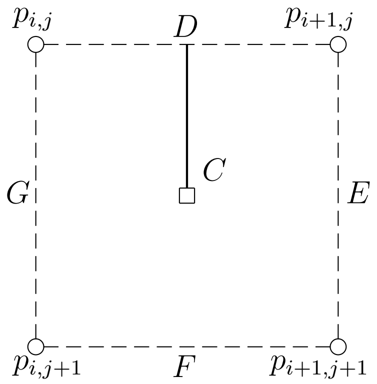


Figure 3.7: An example of a *termination*

component. The same result is obtained if all four pixels are in different components.

The configuration show in figure 3.7, a termination, is not possible with this definition. That this is so can be seen by observing that if, as in figure 3.7, $R(p_{i,j}) \neq R(p_{i+1,j})$ and also $R(p_{i+1,j}) = R(p_{i+1,j+1}) = R(p_{i,j+1}) = R(p_{i,j})$ then surely $R(p_{i,j}) = R(p_{i+1,j})$ which is a contradiction. The exclusion of this kind of region is consistent with the comments following theorem 1 on page 12. Recall that if an arc is surrounded by a single region it can be removed and decrease the total energy of the segmentation. The same comment applies to this definition of the length.

The following definitions, motivated by the preceding discussion, give a formal meaning to the length associated with a given segmentation.

Definition 5. *Let u_0 be an image defined on an integer lattice in $\mathbb{R} \times \mathbb{R}$*

$$u_0 \colon \{1, 2, \ldots, W\} \times \{1, 2, \ldots, L\} \to \mathbb{R},$$

and let $\mathcal{A} = \{r_1, r_2, \ldots, r_n\}$ be a finite set. Call the elements of $\mathcal{A}$ component identifiers.[13] *A segmentation of u_0 is a map, R, on the same integer lattice that associates each pixel with a component identifier, i.e.,*

$$R \colon \{1, 2, \ldots, W\} \times \{1, 2, \ldots, L\} \to \mathcal{A},$$

along with a real valued approximating function u defined on the same integer lattice.

Definition 6. *Let R be a segmentation of an image u. The length of the edge set of R is defined as*

$$\operatorname{len}(R) = \frac{1}{2} \sum_{i=1}^{W-1} \sum_{j=1}^{L-1} \ell_{i,j} \text{ where,}$$

mapping the the result of a logical expression from "true" or "false" to 0 or 1,[14]

$$\begin{aligned} \ell_{i,j} = \quad & [R(i,j) = R(i+1,j)] + [R(i+1,j) = R(i+1,j+1)] \\ & + [R(i+1,j+1) = R(i,j+1)] + [R(i,j+1) = R(i,j)]. \end{aligned}$$

[13] Nothing is implied about the nature of the component identifiers. They could be real numbers, integers, or, as in figure 3.3, the symbols a, b, c, and d. The elements of $\mathcal{A}$ should be viewed as pure symbols.

[14] This is the opposite of the values typically assigned to logical expressions by programming languages. Using this mapping here avoids negative logical expressions

Some remarks about definitions 5 and 6 are in order.

1. The possible values of $\ell_{i,j}$ depends on the cardinality of the set $\mathcal{A}$, denoted by $|\mathcal{A}|$. There must be at least 3 elements for all values to be possible. If $|\mathcal{A}| = 2$ the T-junction of figure 3.5 cannot be realized. Also, as shown above, the termination is never possible. Therefore, with a two phase segmentation the possible values are 2 for a continuation and 4 for a mosaic.

2. If $|\mathcal{A}| = 4$ all values of $\ell_{i,j}$ are possible except for 1 because that would be a termination. In fact, if $|\mathcal{A}| = 3$ all values are possible, but the mosaic borders two like components that meet at a single corner. An example of that can be seen in figure 3.3. The observation that only 4 components are needed to obtain each possible length is an intuitive explanation of the claim made in [13] and again in [12] that the four color map theorem implies that $|\mathcal{A}| = 4$ suffices for piece-wise constant segmentation.

3. Once the mapping R which defines the segmentation is known the length follows. That means that the energy functional E_0 is a function of R alone because the fidelity term can be minimized separately. Moreover, a segmentation can be viewed as a partition of the image. This observation hints at the utility of the results in section 3.1 when they are applied to segmentation. It is also essentially the same as using a level set function to define the boundaries of the segmentation components.

4. The edge set defined in [11] adds a bit more to the definition. There the orientation of the edges is a random variable. In this definition all the edges are

assumed to be oriented in either the horizontal or vertical direction.

5. This definition is a special case of the C^1 arcs described in theorem 1 on page 12. Each of the edge segments, see for example figure 3.3, are C^1 arcs and the image can easily be extended to a continuous function by bilinear interpolation.

This section has described the length term as it is used in the subsequent segmentation algorithms. The definition is general enough to apply to any segmentation that identifies regions and partitions the image into disjoint sets. The next chapter describes how the fidelity term is included to complete the discretization of the functional defined by (3.1).

Before proceeding there is an interesting observation that can be made about the length and fidelity terms. The primary difference between (3.1) and the original Mumford-Shah model is the choice of the norm used in the fidelity term. The length term as defined here is also an l_1 norm in the following sense. If $x = (x_1, x_2)$ and $y = (y_1, y_2)$ are points in the image domain Ω that lie on a boundary between two regions then as defined here the length of the portion of the boundary between them is $\|x - y\|_1$. It is the distance according to the taxi-cab metric instead of a Euclidean distance. Thus both the fidelity term and the length term make use of the l_1 norm as an essential part of the definition.

Chapter 4

Application

This chapter formalizes the statement of the segmentation problem introduced in chapter 3. The first section describes how the length and the fidelity terms are combined into a discrete version of the model. The second section is the description of the algorithm used to obtain a solution.

4.1 Formalization of the problem

Only two steps remain to arrive at a complete discretization of the minimization problem (2.14) on page 30. The first is to settle on a model of the image domain and range. For that the most practical choice is the rectangular integer lattice with an integer range as defined in equation (1.3) on page 3. It is exactly the right model to use for all the images considered in this thesis. The second step is to define the l_1 fidelity term. Having settled on an integer range for the image model there is a natural way to proceed. As noted in chapter 1 the range of the image function is a small finite set of integers, typically 0 to 255. The domain, on the other hand, can literally have millions of elements. Digital cameras are widely available today with several million pixels. Therefore, many elements in the domain will likely map to the same element in the range. Recall from footnote 2 on page 38 that the results of chapter 3 do not require that the elements of the sets in question be unique. A simple way to deal with this is with a histogram.

Histograms play a prominent role in image analysis. They can be used to enhance the contrast of an image and to give a general characteristic of the range set. Using the notation of chapter 1, if $u(i,j) \in \mathcal{I}$ then the histogram, h, of u is

$$h\colon \mathcal{I} \to \mathbb{Z}^+ \text{ with } h(k) = \left|\{(i,j) : u(i,j) = k\}\right|. \tag{4.1}$$

That is, $h(k)$ counts the number of points in the image domain that are mapped to the value k. There is a natural mapping from the histogram of the image to its total deviation. Let the set X of definition 2 on page 37 contain $h(k)$ copies of the element k of $\mathcal{I}$. The total deviation of that set will be represented by the symbol D_h. The median of that set will be represented by the symbol M_h. To make the median well defined a single value must be chosen. The traditional definition will suffice so, using the notation of chapter 3, let $M_h = \frac{1}{2}(M_X^+ + M_X^-)$.

All the notation needed is now in place and can be summarized in the following definition of the fidelity term:

Definition 7. *Let R be a segmentation of an image u_0 as defined in definition 5 on page 61. For each element, r_l, of the set of component identifiers, $\mathcal{A}$, let h_l be the* *histogram of that segmentation component. That is, if* $R_l = \{(i,j) : R(i,j) = r_l\}$ *then* $h_l(k) = |\{(i,j) \in R_l : u_0(i,j) = k\}|$*. The* l_1 *fidelity of* R *is defined to be,*

$$\text{fid}(R) = \sum_{l=1}^{n} D_{h_l}$$

Now the optimal segmentation problem can be stated as follows:

$$\text{Given } u_0 \text{ minimize } \mathcal{E}(R) = \text{fid}(R) + \lambda \cdot \text{len}(R) \text{ over all segmentations of } u_0. \tag{4.2}$$

Thus, the quantity $\mathcal{E}(R)$ is a function that associates with each segmentation (or, equivalently, with each partition of the image domain of size n) an energy (or cost) that is to be minimized. The fact that the energy is a function of the segmentation alone is analogous to the level set formulation of the Euler-Lagrange equation, equation (2.4) on page 19, of the two phase Chan-Vese algorithm. Recall that in that formulation the two constants c_1 and c_2 can be computed for a given level set function ϕ. The same is true with this formulation except that the constants are the medians instead of the means.

Suppose that R^* is an optimal segmentation, that is $\mathcal{E}(R^*)$ minimizes (4.2). Then the corresponding piece-wise constant image u^* that solves the original optimization problem of (2.14) on page 30 is given by

$$u^*(i,j) = M_{h_l} \text{ where } R^*(i,j) = r_l. \tag{4.3}$$

In words this says that once an optimal segmentation is known the approximating piece-wise constant image u is defined by the median of each segmentation component. The edge set K is, of course, defined by the segmentation R^*. The problem has thus been reduced to finding the optimal R^*. To clarify the significance of this formulation some observations are in order.

1. All the sets involved are finite. Therefore, the existence of at least one R^* is assured. There is no reason to believe there is a unique solution.

2. There are a large number of possible segmentations with n components. As stated in chapter 3 there are $\mathcal{S}_N^{(n)}$ where $N = W \times L$ and $\mathcal{S}_N^{(n)}$ is a Stirling number of the second kind. This grows exponentially with N.

3. Even if the problem could be reduced to a search of just the ordered partitions, which seems unlikely, the processing involved with computing $\mathcal{E}(R)$ for each is prohibitive.

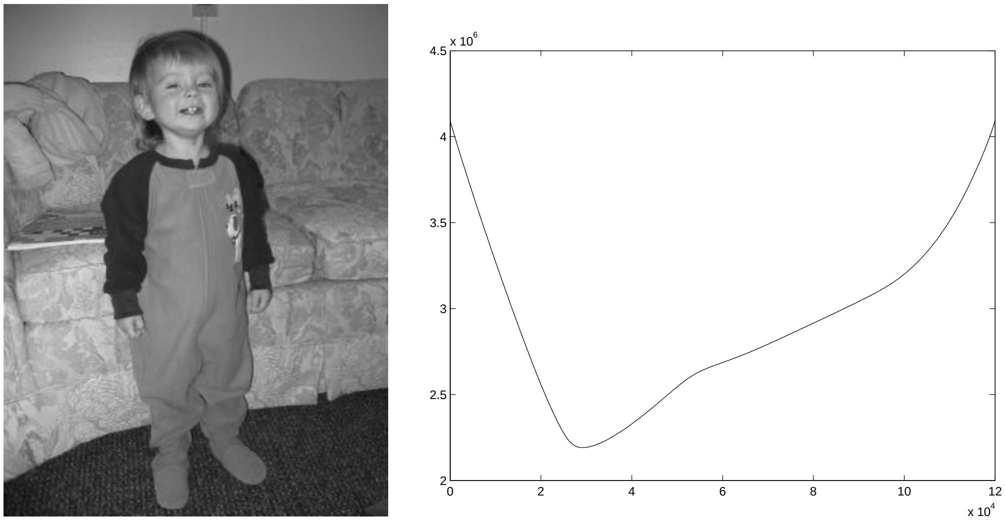
4. This formulation is valid for any number of phases which is defined by the number of segmentation component identifiers in the set $\mathcal{A}$.

5. While this focuses on the l_1 fidelity of definition 7 it is easily modified for any norm.

6. For a given h_l the corresponding set X could be very large and to compute the medians and total deviations directly requires that X be sorted. This would be impractical. On the other hand the cardinality of the co-domain $\mathcal{I}$ of the observed image u_0 is very small. This means that the domain of h_l is very small, so any practical algorithm should work directly with the histograms.

There is now available some information to assist with the issue of global minima. It is the function S of equation (3.2) on page 48. Lacking a more descriptive name the graph of S will henceforth be called the S curve of the image. Figure 4.1 is the S curve of a simple photograph. The abscissa of the graph is the index defining the point where two partitions are split. That is, it is the parameter s in equation (3.2). The shape of the curve turns out to be very useful. Notice that in this example there is a clear global minimum and no hint of local minima. Compare that with the similar graph of figure 4.2. In the second curve there is a global minimum, but there is also a clear local minimum. An example in chapter 5 will show that this image is an instance of how local versus global minima affect segmentation results. This is an important point. The S curve provides at least two pieces of information that have not been available before. First, if the curve has a single minimum and is convex as in figure 4.1 the minimum is the best choice for an initial condition. All that will remain is to move spurious pixels between segmentation components until (4.2) is sufficiently small. If on the other hand the image has local minima as in figure 4.2 the initial condition should be chosen from the among them as the one with the smallest total energy. This makes it possible to choose a priori an initial condition which will converge to a global minimum instead of a local minimum.

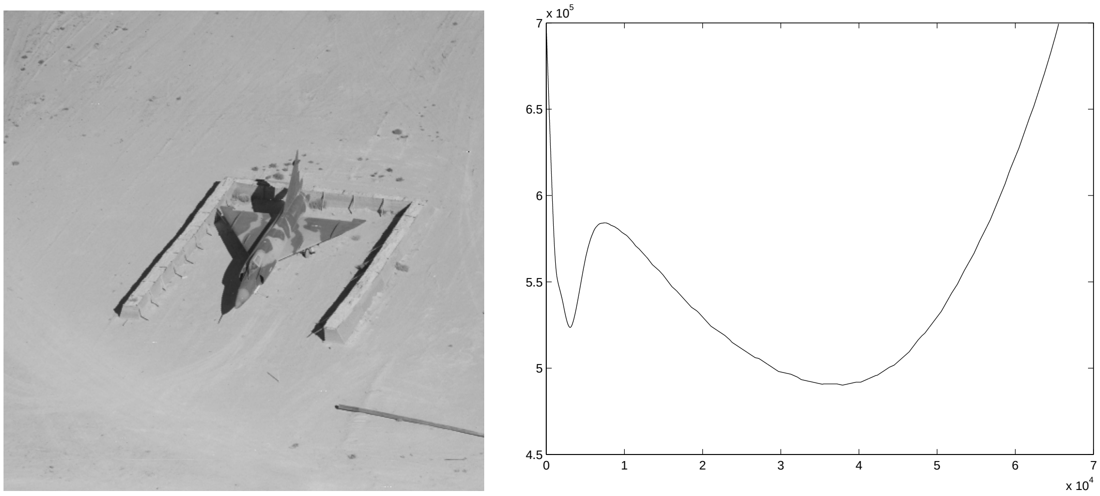
The parameter λ in (4.2) is called the scale parameter. A complete description of the concept is given in [40]. The general idea is that as λ increases the approximating image approaches a constant while the length approaches zero. In the continuous model of equation (3.1) on page 36 this is clear. However, any discrete formulation introduces limitations. In this case, one of the limitations has to do with the $\text{len}(R)$



(a) image

(b) S curve

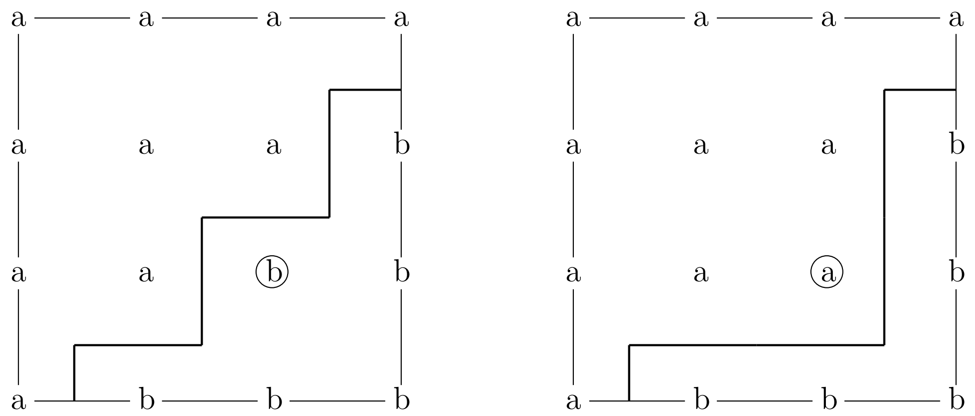
Figure 4.1: S curve of the ‘jammies’ image



(a) image

(b) S curve

Figure 4.2: S curve of the ‘airplane’ image



(a) initial segmentation

(b) with a single pixel changed

Figure 4.3: The local effect of a change in length

term of the discrete form (4.2). As noted previously the length term is a form of the Manhattan metric. One of the consequences of this is that the shortest path between two points in not unique. This can be seen by any simple example. Figure 4.3 shows the effect of a change in the length when the segmentation of a given pixel is changed. Notice that the length of the boundary between segmentation components ‘a’ and ‘b’ is the same in both cases. Thus if the value of λ is so large that changes in fidelity have a negligible affect on the total energy changes in length are not always observable. On the other hand, if the value of λ is too small the $\text{fid}(R)$ term in (4.2) will dominate and the segmentation will just be the initial condition. These two facts imply that there is a range of values $\lambda_{\text{min}} \leq \lambda \leq \lambda_{\text{max}}$ that are a reasonable choice for the scale parameter. Before showing how those can be computed a description of the algorithm is needed. That is the subject of the next section. Some detailed knowledge of the steps involved can be used to select a range of values.

4.2 The associated algorithm and data structures

The basic steps of the algorithm are as follows:

1. Construct the histogram of the entire image and use it to determine the best initial condition for the fidelity term.

2. Use the results obtained in step 1 to define an initial segmentation R and compute an initial value of $\mathcal{E}(R)$ as defined in (4.2).

3. Visit each pixel of the image and compute the change in $\mathcal{E}(R)$ if the pixel were to be moved to each of the other segmentation components. If it is possible to decrease the energy move the pixel to the component yielding the most benefit.

4. Repeat step 3 until $\mathcal{E}(R)$ is unchanged.

Notice the similarity between this algorithm and the algorithm of Song and Chan described in section 2.3. Neither would be practical were it not for the simplicity of the computation in step 3 above and in step 3 on page 23 of Song and Chan's method.

It is clear that the histograms of each segmentation component play a pivotal role. Since the key statistic involved is the median one would be tempted to use some of the data structures designed for use with searching. Such methods have long been available. See for example [49] and [17] in which the concept of a median split tree is studied. These are search trees designed to be balanced to minimize search time. Insertion times, on the other hand can be large. The requirements for the data structures needed with this algorithm are best derived by considering the operations that will be required. Assume that a data structure called 'hist' has been defined and it will be used to hold the histogram of a segmentation component as describe in definition 7. Here is a list of the operations needed:

1. `hist = InitH(X)`; which initializes the data structure based on the input data set '`X`.' The data set is the pixel values of a segmentation component.

2. `x = AdeltaS(hist,v)`; which returns the change in the total deviation of the histogram if the value '`v`' were added. This is an application of Lemma 2 on page 40, so the value will always be positive.

3. `x = RdeltaS(hist,v)`; which returns the change in the total deviation of the histogram if the value '`v`' were removed. This is another application of lemma 2, but in this case the result will always be negative.

4. `hist1=AddV(hist,v)`; which adds to value '`v`' to the histogram and creates the new histogram '`hist1`.'

5. `hist1=RemoveV(hist,v)`; which removes value '`v`' to the histogram and creates the new histogram '`hist1`.'

These operations lead to the following data structure for a histogram `H`:

```
H.hist - the histogram of the values in X
H.chist - the cumulative values of H.hist
H.m - the floor of N/2
H.p - is either 0 or 1 so that N=2*H.m + H.p
H.MXM = X(m+p) when X is sorted in ascending order
H.MXP = X(m+1) with 1 as the first index
H.TD = the total deviation from the median of X
```

The elements `H.MXM` and `H.MXP` correspond to the left and right medians defined in definition 3 on page 38. Because the image model assumes a range of 0–255 for the pixel values the indexes for the elements `H.hist` and `H.chist` are likewise 0–255. The element `H.TD` is the total deviation as defined in definition 4 on page 39. The elements `H.m` and `H.p` define the number of pixels N in the histogram by $N = 2m + p$ also as in definition 3. All the operations defined above are based on lemma 2 on 40. The values of the elements are only changed by the operations `AddV` and `RemoveV`. This fairly complex data structure is only required for the fidelity term. There is no analogy for the length term because it is, by definition 6 on page 61, strictly a local property. The potential change can be computed by examining the adjacent pixels only. A formal statement of the algorithm using the notation of this chapter can now be given.

```
$n$ = the desired number of segmentation components.
$[r_1, r_2, \ldots, r_n]$ = the set of component identifiers.
$u_0$ = the observed image.
$H_0 = \mathrm{InitH}(u_0)$.
$[c_1, c_2, \ldots, c_{n-1}]$ = split points for best fidelity.
$R$ = the segmentation defined by $[c_1, c_2, \ldots, c_{n-1}]$.
$[H_1, H_2, \ldots, H_n]$ = histograms of initial segmentation components.
repeat
  for each pixel $p$
    $r_p = R(p)$
    for each $r_l \in \mathcal{A} \setminus \{r_p\}$
      $R' = R$ with $R(p)$ changed to $r_l$
      $D_l = \mathcal{E}(R') - \mathcal{E}(R)$
    $D_{\min} = \min(D_l)$ taken from those just computed.
    $l_{\min}$ = index $l$ corresponding to minimum $D_l$.
    if $D_{\min} < 0$ set $R(p) = r_{l_{\min}}$
until no pixels are moved.
```

A complete listing of MATLAB code for this is presented in appendix B. The line above where the change in energy is computed uses the `AdeltaS` and `RdeltaS`. The line where the segmentation is actually changed uses the operations `RemoveV` and `AddV`. The change in length only involves the nearest neighbors to the pixel p. That is, the pixels directly above, below, right, and left of p. The diagonal pixels do not play a role because of the way the $\mathrm{len}(R)$ term is defined. The boundary rows and columns can be managed by reflection or by dealing with them explicitly.

Now consider the choice of the parameter λ. It should at least be large enough so the smallest change in length will dominate a noisy pixel. That is, a pixel that differs significantly in value from its neighbors. Suppose then, that moving such a pixel causes a change of 128 in fidelity. This would be the case if a pixel of 255 were moved from the component with the predominantly bright areas to the component with the darker areas. The most the length can change is 4, so to make the $\mathrm{fid}(R)$ and $\mathrm{len}(R)$ terms carry the same weight a value of $\lambda \sim 32$ is the correct order of magnitude. Similarly, if the length change is at least 1 then a value of $\lambda \sim 200$. It is reasonable then just based on these heuristics to set the range of admissible vales of λ to $10 \leq \lambda \leq 250$. Empirical results in the next chapter will demonstrate that for typical images the results are quite insensitive to the value chosen for λ provided that it falls within this range. This is true even when significant amounts of simulated noise are added to an image.

Chapter 5

Experimental results

This chapter presents the result of running the algorithm described in section 4.2 against several test images. Each of the examples exhibits some important feature. All the processing was done with MATLAB running on an Apple Macintosh PowerBook laptop computer running Mac OS version 10.3.9. The execution time depends on the size of the image, of course, but in no case is the execution time prohibitive.

Figure 5.1 shows the results of the l_1 norm applied to the Mumford-Shah model as described in chapter 4. Parts (c) and (d) of the figure show the edge set and segmentation components with a two phase solution. That it, it is analogous to the two phase solution of Chan and Vese. The bar graph in part (b) shows the value of the iteration after each pass through the entire image. Notice how rapidly the algorithm converges. It has essentially found the segmentation after just 4 iterations. This rate of convergence is typical of the images studied.

Figure 5.2 shows the initial condition obtained by applying theorem 3 on 46. In other words, this is what the segmentation looks like after step 2 on page 71. The segmentation problem has almost been solved before any iterations are completed that will reduce the total energy. This clearly shows the benefits of the theory developed in chapter 3. This behavior is typical of images with convex S curves which is shown for this image in figure 5.3. There is a clear global minimum in the S curve of part (a). Part (b) shows the final approximating image. This means that the two gray scales shown in the image are the medians of the two segmentation components as described by equation (4.3) on page 66. There is an important observation to be made about part (b). It shows that the approximating image u is not of much use by itself. While it does divide the image showing the gray scales adds little to the information. The main purpose of computing it is to provide an anchor that ensures the problem does not stray too far from the original image.

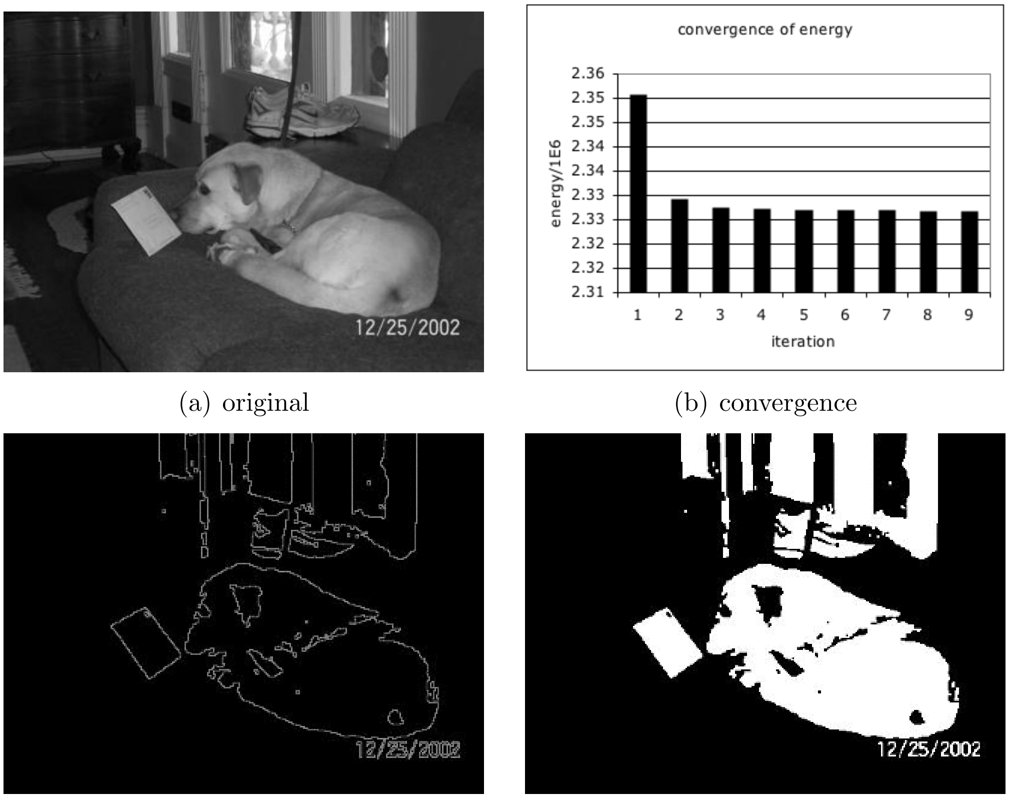
Figure 5.4 shows the behavior of the edge set as the algorithm converges. Notice how some of the smaller features visible in the first iteration slowly vanish and disappear. This is the affect of the length term. Also note that the edge set shown in iteration 4 is virtually identical to the final segmentation shown in figure 5.3. Figure 5.5 shows the effect of different scale factors with the Lena Image. The difference between the edge sets is difficult to perceive showing the insensitivity to the scale parameter λ. There is, however, an apparent difference when the to images are observed with a blink comparator. A blink comparator is a tool used to make subtle differences in images more visible. It is useful in astronomy when one observes images of the same part of the sky taken at different times. Objects such as planets will stand out from the fixed star background. There are also applications that focus on changes in an otherwise static scene. Objects that have moved even slightly become quite visible. The two images in part (a) and part (c) of figure 5.5 exhibit this behavior when viewed with a previewer that makes it possible to switch quickly between the two images when they are overlaid.

Figure 5.6 is perhaps the most significant example of this chapter. It shows what can happen when the wrong choice of initial condition is made. The original image is the airplane image of figure 4.2 on page 69. Recall that the S curve of that image shows a clear local and a clear global minimum. The segmentation shown in figure 5.6 could hardly be called a good segmentation. It is typical of the results obtained when an algorithm is stuck in a local minimum. As the authors state in [10], "The result is blatantly false." In this case it is the result of picking for an initial condition the point with the smallest fidelity term. While this usually seems to be the best choice it clearly is not with this image. The problem is that when the total energy of equation 4.2 on page 66 is considered the global minimum fidelity is not the minimum energy. In fact, the initial value of the energy is substantially less when the larger value of the local fidelity is chosen for the initial condition. Compare figure 5.6 with figure 5.7 in which the second best fidelity minimum was chosen as the initial condition. In this case the segmentation is clearly correct. Notice that by computing to total energy at just the two local minima of the S curve the problem of seeking a local minimum instead of a global minimum is completely avoided.

Figure 5.8 shows the results on a noisy synthetic image. The value of the pixels represent a two dimensional Gaussian. The darker portion near the center represents the larger values. The point of this example is to show that simple edge methods have limitations when no clear edge is available. None of the edge detection methods described in section 2.5 produce results anything like those in figure 5.8. Canny's method produces concentric circles, but all the others produce no meaningful results. Part (d) of figure 5.8 shows the results obtained with the L_2 norm in the Mumford-Shah model using the algorithm of Song and Chan. The result is similar, but notice the flattening in the north west portion of the circle. It turns out that that algorithm never really converges with this example. If the iterations are allowed to continue indefinitely the circle begins to loose its shape altogether. The algorithm described in this thesis, on the other hand, terminates normally producing the result in part (c).

Figure 5.9 shows the initial condition obtained when the number of phases is chosen to be 4. Such an image is, in a very real sense, the best possible representation of the original with only 4 gray scales. Notice the very clear edge between the child's right leg and the couch. This edge is not even visible when the image is segmented with only 2 phases. On the other hand, the fading light along the wall in the background becomes different segments. It is clear then that, as this example shows, it is advisable to proceed with caution when increasing the number of segmentation components. There is no clear rule that says what the correct number ought to be.

The final example addresses the issue of performance with noise. Figure 5.10 shows an image with additive noise. The peak signal to noise ratio is 22db. The result should be compared with figure 5.5. Some residuals are clearly visible in figure 5.10 that are not present in gure 5.5. This is to be expected considering the high level of noise. The basic segmentation is still the same, however. Not shown is the result using the L_2 norm. That algorithm does not even converge with this image. That the l_1 norm should be less sensitive to noise was one of the claims made in chapter 1.



(a) original

(b) convergence

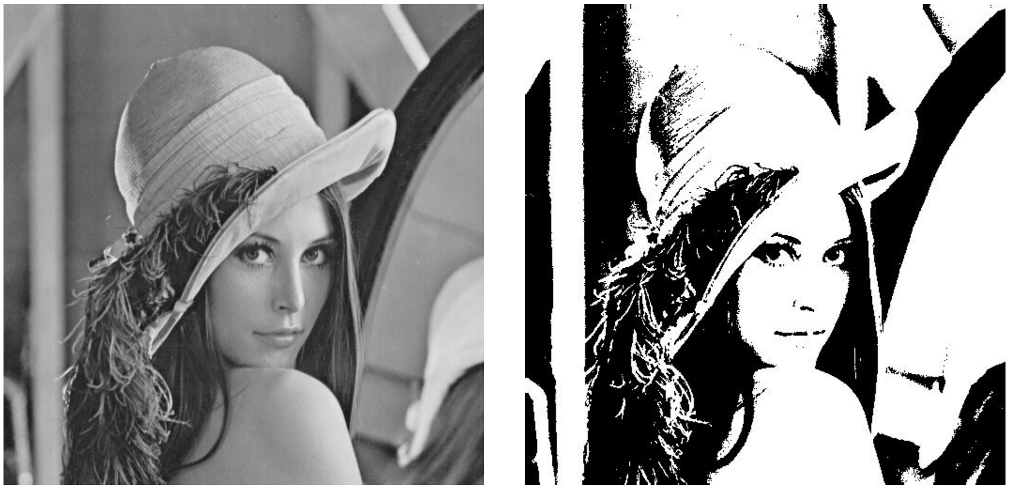
(c) edge set

(d) regions

Figure 5.1: Results from the ‘dog’ image

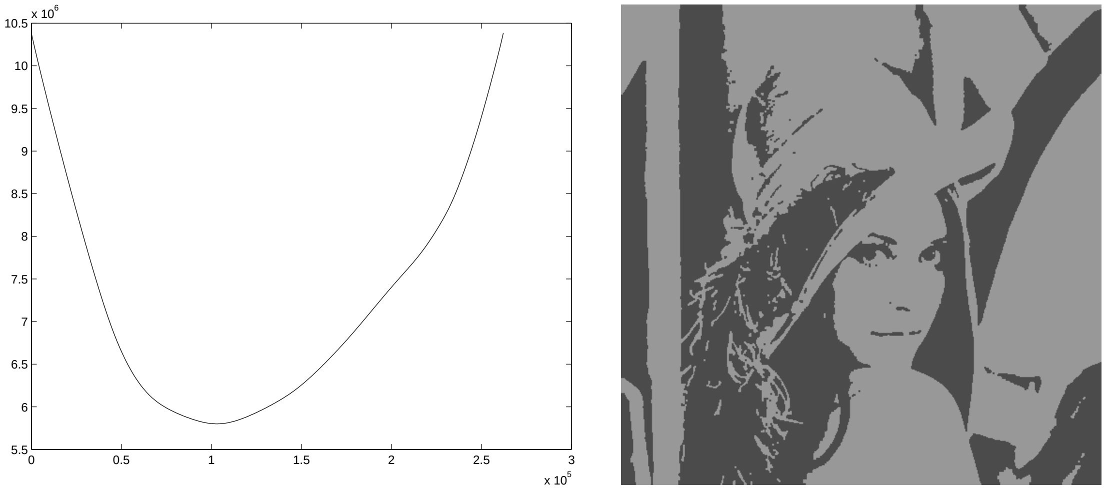
Both this example and the result shown in figure 5.8 support that assertion.

The results presented here are intended to be representative of the issues likely to be encountered. They are not an exhaustive test, but do indicate that this method is fairly robust. Two points are especially important. As expected it is possible to converge to an incorrect solution that is a local minimum. However, there is a way to pick an initial condition that avoids the problem. Also, at least in the presence of Gaussian noise, the performance is superior to the model using a square error fidelity term.



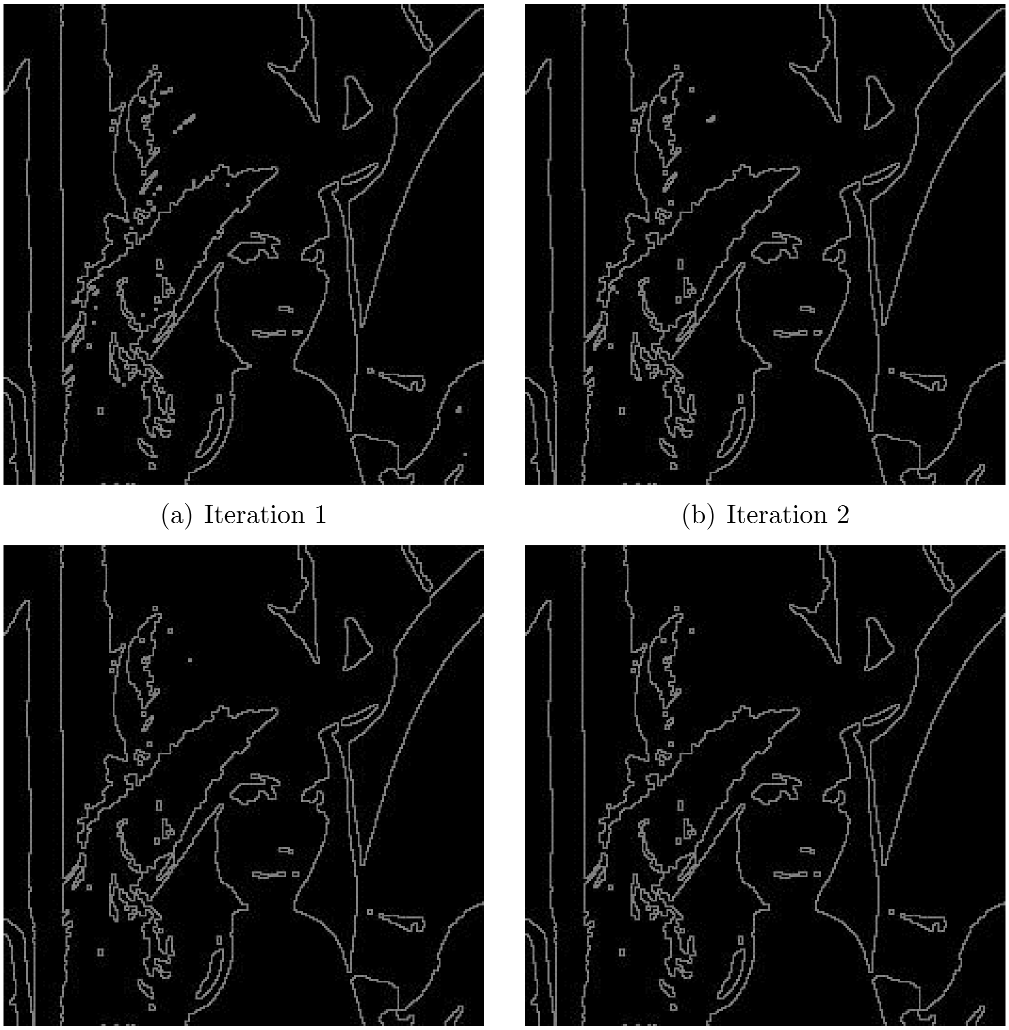
(a) original (b) Initial l_1 condition

Figure 5.2: Initial Condition with minimum fidelity



(a) S curve (b) final gray levels

Figure 5.3: Lena image S curve and final gray scales



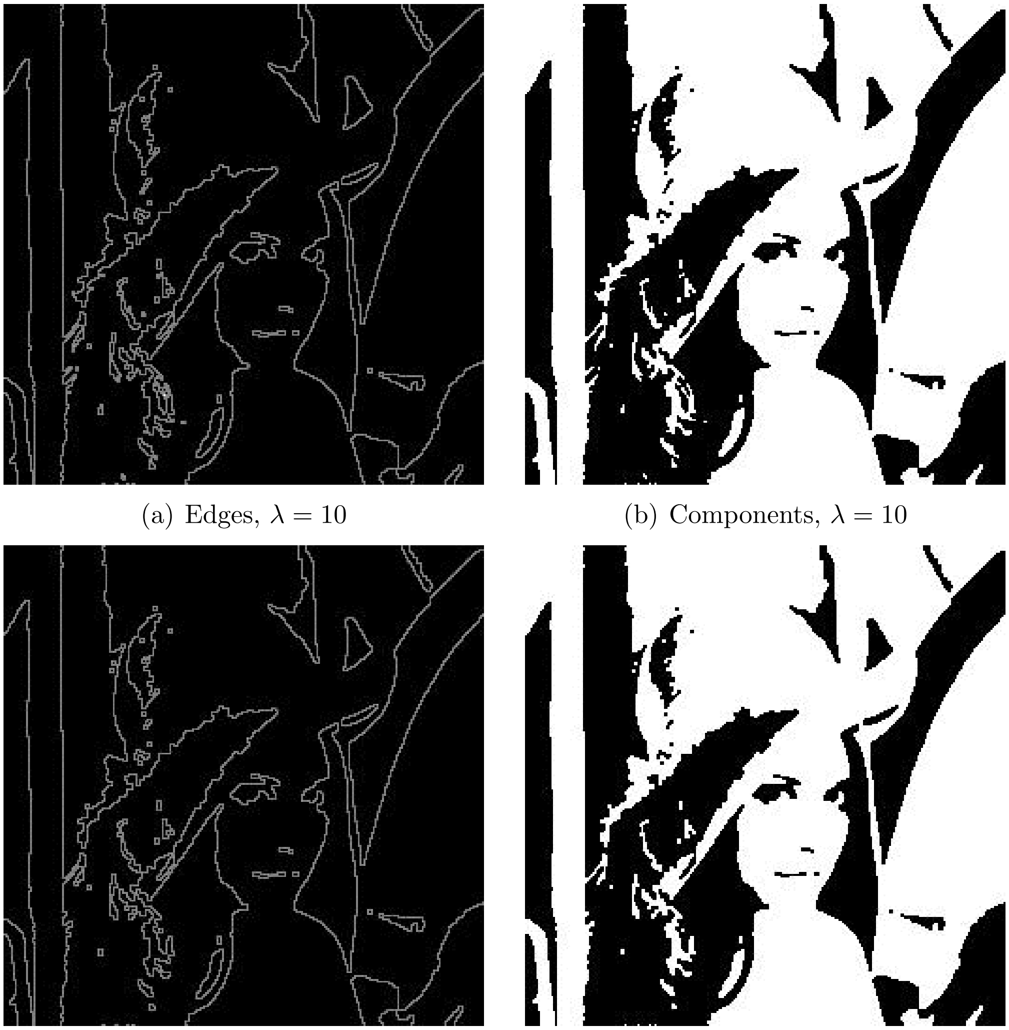
(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

(d) Iteration 4

Figure 5.4: Convergence with the Lena Image



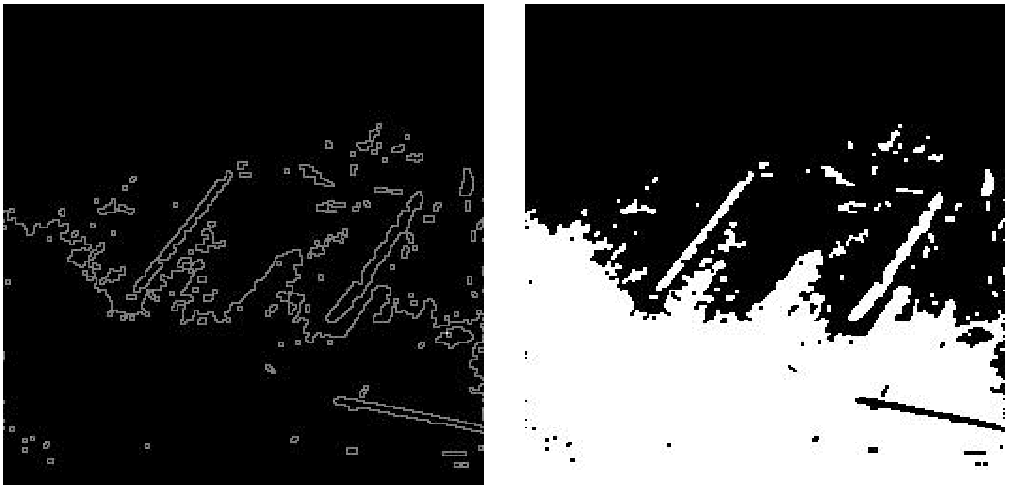
(a) Edges, $\lambda = 10$

(b) Components, $\lambda = 10$

(c) Edges, $\lambda = 40$

(d) Components, $\lambda = 40$

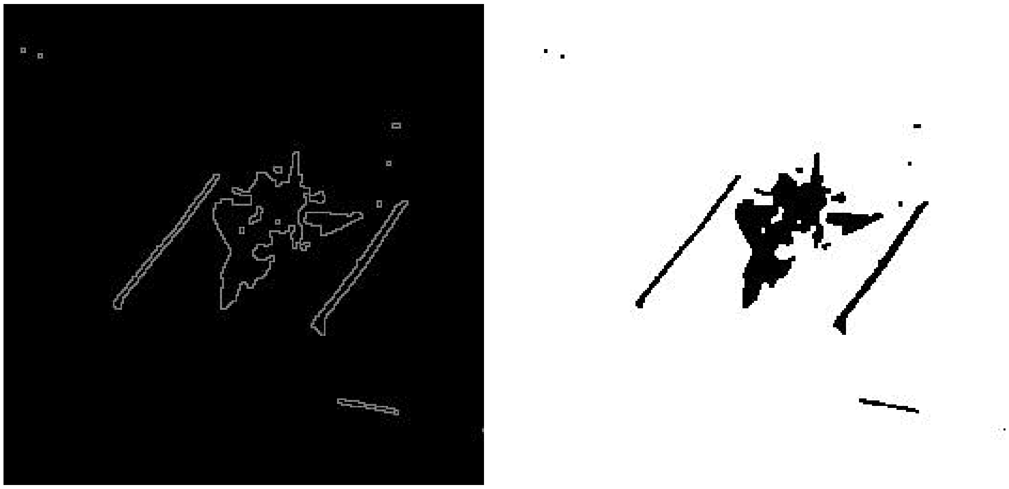
Figure 5.5: Comparison of final results with different scales



(a) edges

(b) Segmentation Components

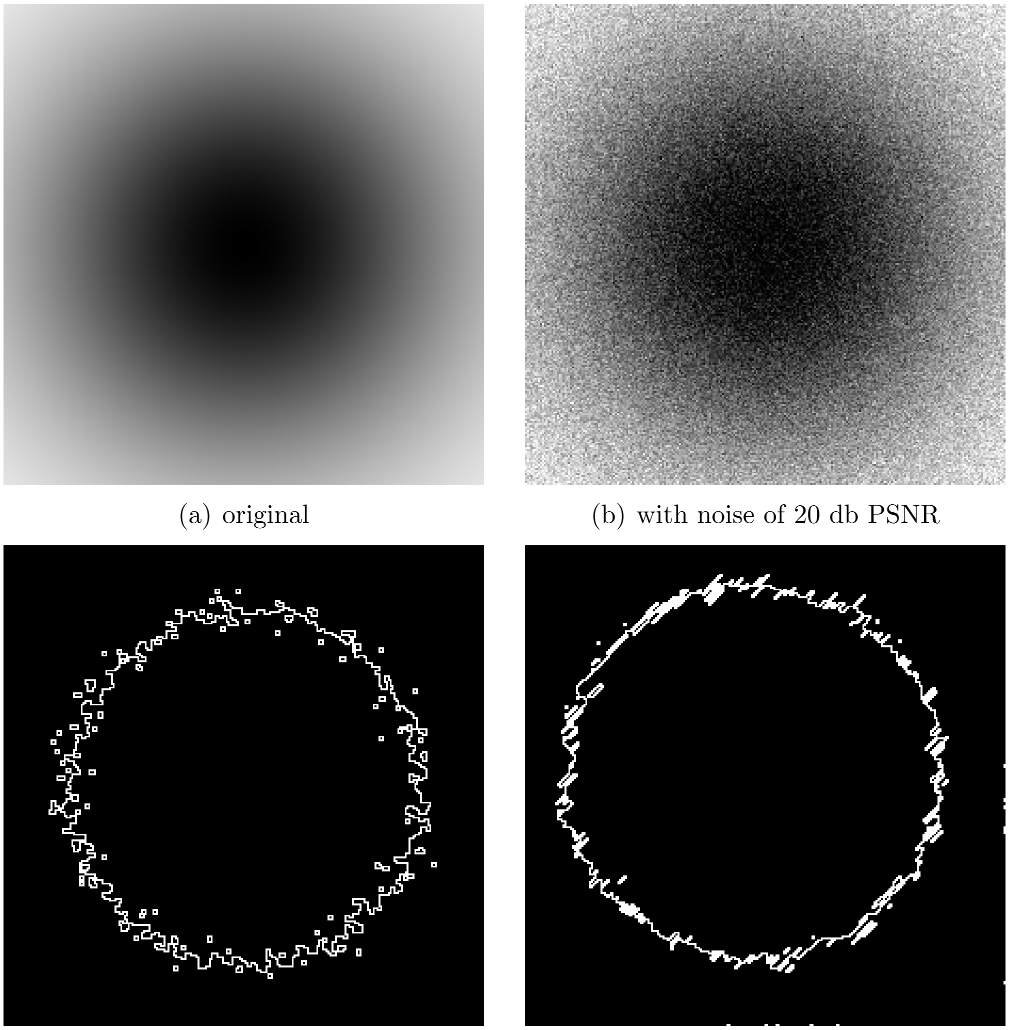
Figure 5.6: The effect of local minimums (best fidelity)



(a) edges

(b) Segmentation Components

Figure 5.7: The effect of local minimums (second best fidelity)



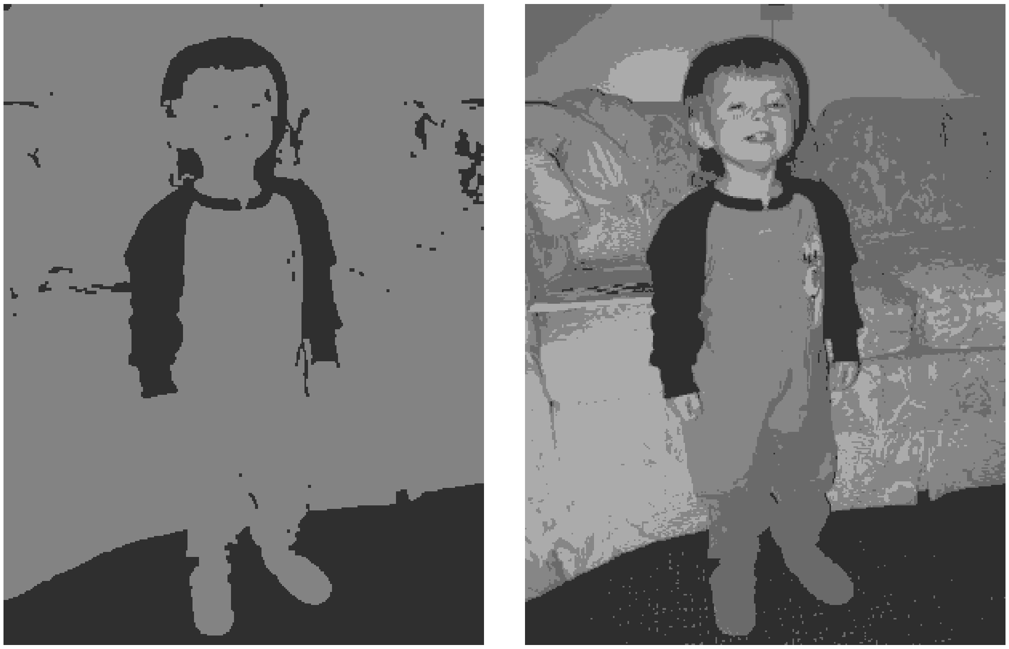
(a) original

(b) with noise of 20 db PSNR

(c) edges with l_1

(d) edges with L_2

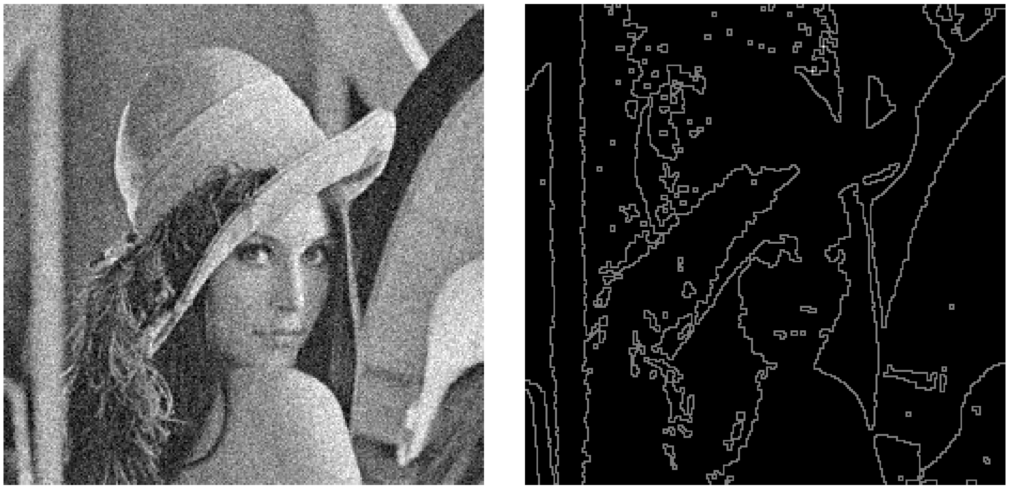
Figure 5.8: Segmentation with a synthetic Gaussian spot



(a) 2 phase

(b) 4 phase

Figure 5.9: A four phase example



(a) noisy image

(b) boundary set

Figure 5.10: A real image with additive noise

Chapter 6

Conclusion

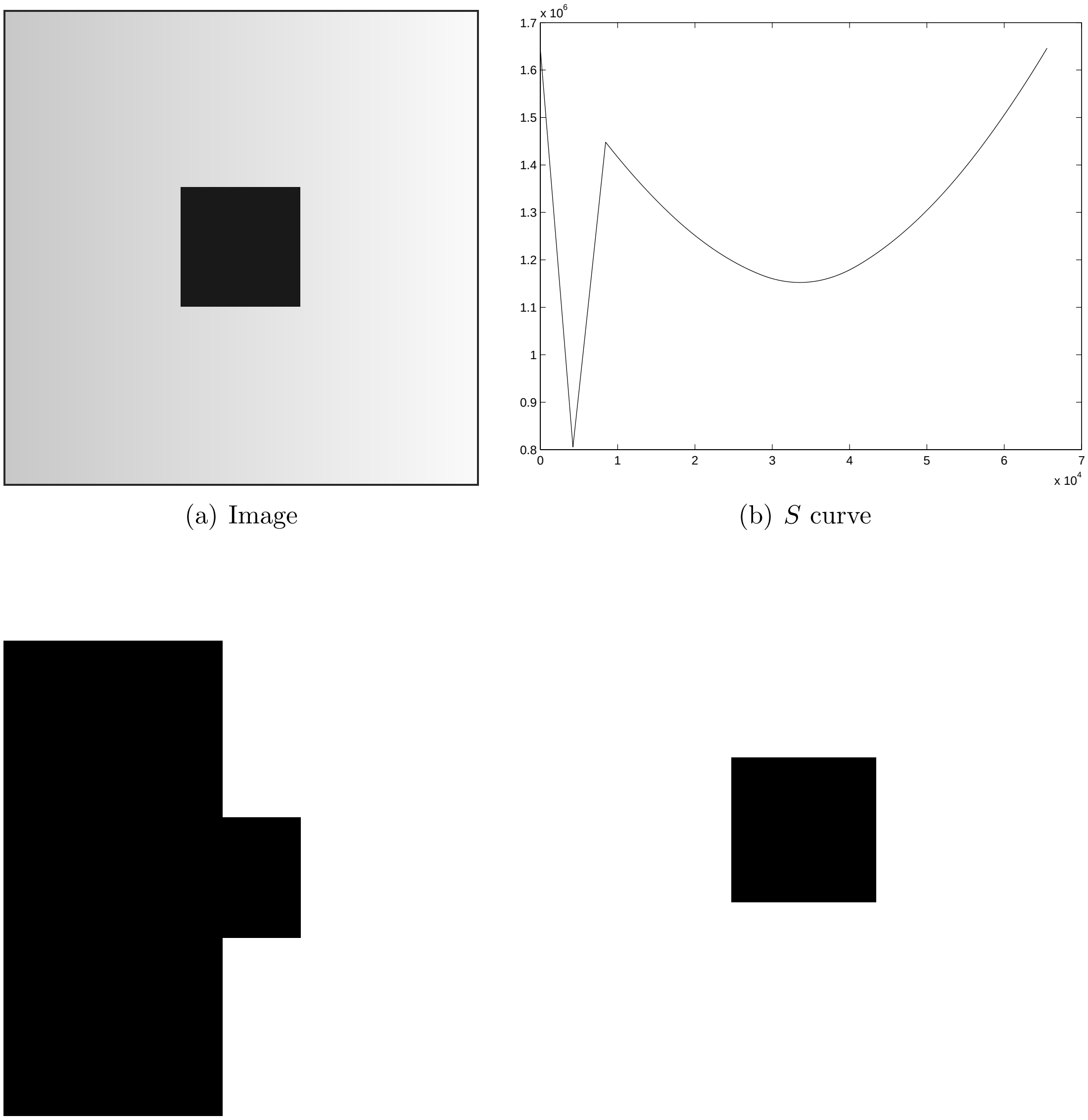
This thesis has developed a new model for image segmentation. The starting point was the energy functional first studied by Mumford and Shah. The particular class of that model used was the piece-wise constant version in which the approximating image takes on only a small number of values. The regions where the value is constant defines the segmentation components and the length is derived from their boundaries.

6.1 Summary and additional examples

The fundamental change made to the model in this thesis is the introduction of the l_1 fidelity term and a precise definition of the length. Moreover, it was shown how to predict a priori the minimum possible value of the fidelity term with relatively simple calculations. This is then used as the initial condition in a greedy algorithm that reduces the total energy. This strategy bypasses the need for the calculus of variation and regularization that would otherwise be required with traditional partial differential equation schemes. The method has been shown to converge consistently and to produce excellent results. While not immune to local minima it was shown how to use a characteristic curve generated directly from the image to predict them. Several examples of performance with both synthetic and natural images were presented, and all the code used is included in an appendix. Most of the examples used the so called two-phase model meaning that the approximating function takes on only two distinct values. However, the method can be extended to an arbitrary number of phases. This chapter presents some additional examples and proposes areas for future work.

Recall that figure 4.2 on page 69 presented an image in which the algorithm could fall into local solution. It is natural to ask what about the image causes this. To that end a synthetic image with similar S curve would be helpful. Such an image is shown in figure 6.1. Notice that the curve is similar to the S curve of the natural image shown in figure 4.2. The salient features are the dark square in the middle of the image and the slowly changing light background. The dark square is analogous to the airplane in figure 4.2. The background simulates the desert floor surrounding the plane. The small gradient in the background appears to be caused by the grazing angle at which the image was captured. The S curve of the synthetic image bears a close resemblance to that of the natural image. The two local minima shown in part (b) correspond to one incorrect and one correct segmentation (the square and the background). The false segmentation is analogous to figure 5.6 on page 83. It divides the image into left and right components using the dark square as part of the boundary. These are shown in part (c) and part (d) of figure 6.1.

A final example is shown in figure 6.2 which is a beautiful photograph of the



(a) Image

(b) S curve

(c) wrong segmentation

(d) correct segmentation

Figure 6.1: The S curve of a synthetic image

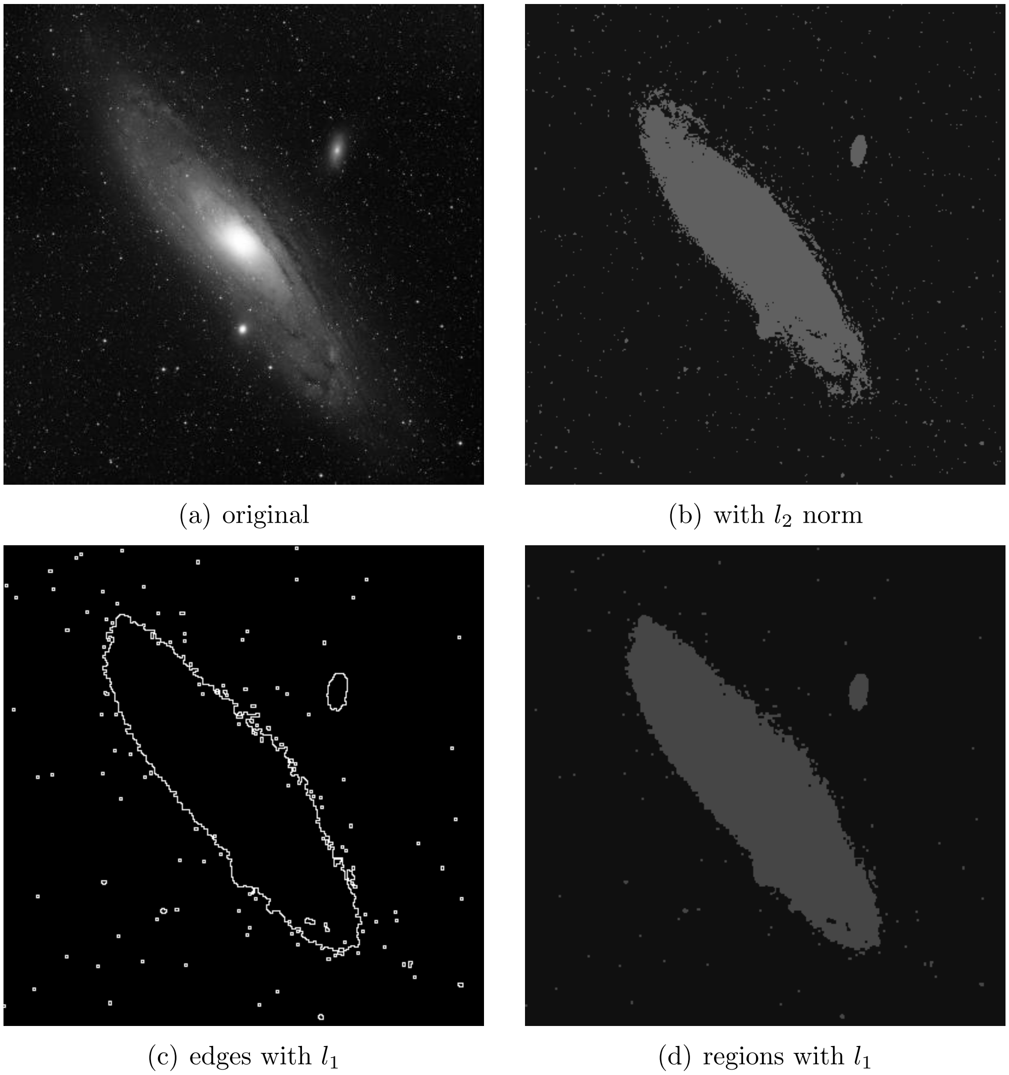
Andromeda galaxy. Part (c) and part (d) show the results using the l_1 norm. Except for a few of the bright stars in the background it essentially captures the two main features—the Andromeda galaxy and the smaller galaxy in the upper right portion of the image. While the L_2 method captures the same features the border between the regions is not nearly as well defined. This is another example of the relative insensitivity to noise. In this image little or noise is present, but it contains more clutter in the background.

In summary the advantages of the algorithm developed are:

1. There is a way to determine the initial condition from the image itself.
2. Using a characteristic of the image it is possible to predict and avoid local minimum solutions.
3. It uses a simple but effective definition of the length of the boundary.
4. It is reasonably insensitive to a noisy image.
5. The greedy algorithm employed works directly with the energy functional and avoids differential equations.

6.2 Future work

This section concludes the thesis by proposing ideas for further research. The function $S(s)$ (referred to as S curves) defined in equation (3.2) on page 48 has been shown to be helpful in analyzing the image. It almost seems to be a fundamental characteristic



(a) original

(b) with l_2 norm

(c) edges with l_1

(d) regions with l_1

Figure 6.2: l_2 and l_1 comparison with the M31 galaxy

of the image similar to the histogram. A more detailed study of this function would be useful. Lemma 3 on page 54 shows that its range completely contains the possible values of the fidelity term. Since it is so useful, better methods of computing it need to be developed. Even when just the histogram is used to compute points it tends to take more processing time than would be expected. An improvement would be to develop a version of Lemma 2 that could be applied when a number of elements with the same value are transfered from one partition element to the other.

The model becomes much more complicated when more than two phase segmentations are involved. This is clear from the pseudo-code shown in section 4.2. In the inner loop the possibility of moving a pixel to each of the other components has to be explored. Each individual test is fast, but there are a great number of them. There should be opportunities for parallelization here that would be worth exploring.

Both of the previous paragraphs only suggest improvements to the algorithm presented here. There is also the interesting possibility of applying different norms to the fidelity term and of easing the restriction to piecewise constant approximating images. Theorem 5 shows that the same ideas that have been applied here to the l_1 norm would be applicable to any norm used for the fidelity term. The only caveat would be the need for an analogy to lemma 2. Without that a similar algorithm would be impractical. However, all the comments regarding the selection of an initial condition apply. Some authors have addressed the question of approximating images that are other than piece-wise constant. Examples are [59] that also adopts an integer lattice model, and [5] where the approximating image is allowed to very linearly. There is progress being made in this area, but much remains to be done. One of the benefits of this kind of approach is that soft edges that fade into the background can be more precisely detected. An example is shown in figure 5.1 on page 79. The dog's back fades slowly into the fabric of the couch. As can easily be seen from part (c) of the figure the edge is not detected even though it is clearly visible to a human observer.

Finally, the question of how many phases are required for a good segmentation is unanswered. The field badly needs an answer to this question that can be derived from the image. This thesis has shown how it is possible in the case of the l_1 norm to detect and avoid local minimum traps, but there remains the question of how to select the value of the parameter n in definition 5.

The bibliography of this thesis shows that this is still an active area of research and a great deal remains to be learned. It is the author's hope that others will be able to build on the ideas described here and further advance this useful and interesting field of study.

Appendix A

Code for Song and Chan's algorithm

This is the MATLAB code used to generate the examples of the two phase Mumford-Shah model. It requires the image processing toolbox. It was processed with student version 7 of MATLAB. This code was written in its entirety by the author of this thesis. It is based on the description of the algorithm given in Bing Song's dissertation [50]. Every effort has been made to exactly reproduce the algorithm as it is explained there. However, no code is provided in [50] so misinterpretations are possible.

```
% An implementation of the fast algorithm of Song and Chan
% This script assumes that the image exits
% in a variable 'img' of type double.

% weighting parameter used in Song's thesis
lambda=1;
mu=0.045;

height=size(img,1);
width=size(img,2);

% initialize level set function (a large circle)
center=size(img)*.5;
radius=min(center)-4;
phi=zeros(size(img));
for y=1:height,
  for x=1:width,
    if norm([y,x]-center)<radius
      phi(y,x)=1;
    end
  end
end
```

```
% initial values of parameters
n=sum(sum(phi));
m=height*width-n;

c0=sum(sum((1-phi).*img))/m;
c1=sum(sum(phi.*img))/n;

% first visit each pixel just reducing the fidelity term.
for y=1:height,
  for x=1:width,
    % compute change in fidelity and modify phi if necessary
    u=img(y,x);
    if phi(y,x)==1
      m1=m+1;
      n1=n-1;
      DE=(m*((c0-u)^2)/m1) - (n*((c1-u)^2)/n1);
      if DE<0
        phi(y,x)=0;
        c0=(m*c0+u)/m1;
        c1=(n*c1-u)/n1;
        m=m1;
        n=n1;
      end
    else
      m1=m-1;
      n1=n+1;
      DE=(n*((c1-u)^2)/n1) - (m*((c0-u)^2)/m1);
      if DE<0
        phi(y,x)=1;
        c0=(m*c0-u)/m1;
        c1=(n*c1+u)/n1;
        m=m1;
        n=n1;
      end
    end
  end
end
imshow(Overlay(img,phi));

% Now include the length term.
num_changes=1;
lookup=[sqrt(2) 1 0 1 sqrt(2)];
aj=[0 0];
while num_changes~=0,
  num_changes=0;
  for i=1:height-1,
    for j=1:width-1,
      % compute the change in length
      pij=phi(i,j);
      pijn=1-pij;
      aj(1)=phi(i,j+1);
      aj(2)=phi(i+1,j);
      nowl=lookup(3+sum(aj-pij));
      newl=lookup(3+sum(aj-pijn));
      DL=newl-nowl;
      % now compute the change in fidelity
      u=img(i,j);
      if pij==1
        m1=m+1;
        n1=n-1;
        DF=(m*((c0-u)^2)/m1) - (n*((c1-u)^2)/n1);
        if (mu*DF+lambda*DL)<0
          phi(i,j)=0;
          c0=(m*c0+u)/m1;
          c1=(n*c1-u)/n1;
          m=m1;
          n=n1;
          num_changes=num_changes+1;
        end
      else
        m1=m-1;
        n1=n+1;
        DF=(n*((c1-u)^2)/n1) - (m*((c0-u)^2)/m1);
        if (mu*DF+lambda*DL)<0
          phi(i,j)=1;
          c0=(m*c0-u)/m1;
          c1=(n*c1+u)/n1;
          m=m1;
          n=n1;
          num_changes=num_changes+1;
        end
      end
    end
  end % for i=1:height-1;
  figure;
  imshow(Overlay(img,phi));
end % while num_changes~=0

% set up two phase image
imgs=uint8(img);
```

```
C=[uint8(c0+.5) uint8(c1+.5)];
for i=1:height,
  for j=1:width,
    imgs(i,j)=C(1+phi(i,j));
  end
end
figure;
imshow(imgs);
```

The function *Overlay* shown below is used to display the original image with an overlay of the edge set. It is used both here and in the implementation of the L_1 version in appendix B.

```
function ovy = Overlay(im, phi)

  Dims=size(phi);
  temp( :, :, 1 ) = im;
  temp( :, :, 2 ) = im;
  temp( :, :, 3 ) = im;

  phix=circshift(phi,[0;-1]);
  phix(:,Dims(2))=phi(:,Dims(2));

  phiy=circshift(phi,[-1;0]);
  phiy(Dims(1),:)=phi(Dims(1),:);

  dx=phix-phi;
  dy=phiy-phi;
  Len=255*sqrt(dx.^2 + dy.^2);
  temp(:,:,1) = temp(:,:,1)+Len;
  for i=1:Dims(1),
    for j=1:Dims(2),
      if Len(i,j)~=0,
        temp(i,j,2)=0;
        temp(i,j,3)=0;
      end
    end
  end
  ovy=uint8(temp);
%end Overlay
```

Appendix B

Code for Mumford-Shah model with l_1 norm

This appendix contains the MATLAB code used to produce the examples of segmentation with the l_1 norm used in place of the l_2 norm in the Mumford-Shah model. First, there is a short script that produces the S curves described in chapter 3. The plots shown there are of the array S.

```
% script assumes image is in a variable named img
X=double(sort(reshape(img,[size(img,1)*size(img,2),1])));
N=size(X,1);
S=X(1:N-1);
Dy=0;
ym=0;
yp=1;
zm=floor((N-1)/2);
zp=(N-1) - 2*zm;
Dz=sum(X((1+zm+1+zp):N))-sum(X((1+1):(1+zm)));
Dx=Dz+X(1+zm)-X(1);
Smin=Dz;
for n=1:N-1,
  S(n)=Dy+Dz;
  if S(n)<Smin,
    Smin=S(n);
    Minn=n;
  end
  Myp=X(ym+1);
  Mzp=X(n+zm+1);
  Dy=Dy+X(n+1)-Myp;
  Dz=Dz+X(n+1)-Mzp;
  if yp==0,
    yp=1;
  else
```

```
    yp=0;
    ym=ym+1;
  end
  if zp==1,
    zp=0;
  else
    zp=1;
    zm=zm-1;
  end
end
```

The following script is the code for two phase segmentation examples. The key elements of the greedy algorithm is set apart by comments in the code. It is the ability to quickly predict what will happen to the energy if the pixel is moved that really makes the algorithm practical.

```
% 2 phase segmentation
% script assumes a uint8 image named img exists

width=size(img,2);
height=size(img,1);

H=InitH(reshape(img,width*height,1));
MD=TotDev2(H);
if MD(1)>1
  error(’multiple minimums’);
else
  Fmin=MD(3); % minimum fidelity term
  MD=uint8(MD(2)); % split point
end
Hs=SplitH(H,MD);
Hs0=Hs;
phi=img>MD;
phi0=phi;

Lv=xor(phi(:,1:(width-1)),phi(:,2:width));
Lh=xor(phi(1:(height-1),:),phi(2:height,:));

L=Lv(1:(height-1),:)+Lv(2:height,:);
L=L+Lh(:,1:(width-1))+Lh(:,2:width);

K0=sum(sum(L));
lambda=40;
lambda2=-4*lambda;
Num_Changes=1;
E0=Hs(1).TD+Hs(2).TD + lambda*K0
```

```
while Num_Changes>0,
  Num_Changes=0;
  for i=2:height-1,
    ip1=i+1;
    im1=i-1;
    for j=2:width-1,
      jp1=j+1;
      jm1=j-1;
      pij=phi(i,j);
      pijn=~pij;
      Uij=img(i,j);
      % Key elements of the greedy algorithm:
      % Predict change in fidelity if pixel is moved.
      DF=RdeltaS(Hs(1+pij),Uij) + AdeltaS(Hs(1+pijn),Uij);
      % Predict change in length if pixel is moved.
      DL=(pij~=phi(i,jp1)) + (pij~=phi(i,jm1)) + ...
         (pij~=phi(ip1,j)) + (pij~=phi(im1,j)) - 2.0;
      % Move the pixel if the energy will decrease
      if (DF+lambda2*DL)<0
        phi(i,j)=pijn;
        HT=RemoveV(Hs(1+pij),Uij);
        Hs(1+pij)=HT;
        HT=AddV(Hs(1+pijn),Uij);
        Hs(1+pijn)=HT;
        Num_Changes=Num_Changes+1;
      end
    end
  end
  Lv=xor(phi(:,1:(width-1)),phi(:,2:width));
  Lh=xor(phi(1:(height-1),:),phi(2:height,:));

  L1=Lv(1:(height-1),:)+Lv(2:height,:);
  L1=L1+Lh(:,1:(width-1))+Lh(:,2:width);

  K1=sum(sum(L1));
  E1=Hs(1).TD+Hs(2).TD + lambda*K1
end % while Num_Changes>0
% set up piecwise constant image:
imgs=img;
for i=1:2,
  C(i)=0.5*(Hs(i).MXM+Hs(i).MXP);
end
for i=1:height,
  for j=1:width,
    imgs(i,j)=C(1+phi(i,j));
  end
end
imshow(img),figure,imshow(imgs),figure,imshow(uint8(63*L1));
```

The function *InitH* sets up the data structures described in section 4.2.

```
function H=InitH(X);
% H=InitH(X) creates the data structure described below when
% X is a vector of values of type UINT8
% arranged so that size(X)=[N,1].
% The data structure produced is as follows:
%   H.hist - histogram of the values in X
%   H.chist - cumulative values of H.hist
%   H.m
%   H.p is either 0 or 1 so that N=2*H.m + H.p
%   H.MXM = X(m+p) when X is sorted in ascending order
%   H.MXP = X(m+1) with 1 as the first index
%   H.TD = the total deviation from the median of X
% Note that the value associated with both the histogram and the
% cummulative histogram is the index-1.
N=size(X,1);
m=int32(floor(N/2));
p=int32(N-2*m);
T.hist=histc(X, 0:255);
T.chist=T.hist;
i=int32(1);
if (p==0),
  % N is even
  while(T.chist(i)<m),
    j=i+1;
    T.chist(j)=T.chist(i)+T.hist(j);
    i=j;
  end
  T.MXM=i-1;
  while(T.chist(i)<m+1),
    j=i+1;
    T.chist(j)=T.chist(i)+T.hist(j);
    i=j;
  end
  T.MXP=i-1;
else
  % N is odd
  while(T.chist(i)<m+1),
    j=i+1;
    CP=T.chist(i);
    T.chist(j)=CP+T.hist(j);
    i=j;
  end
  T.MXM=i-1;
  T.MXP=T.MXM;
end
while(i<256),
  j=i+1;
  T.chist(j)=T.chist(i)+T.hist(j);
  i=j;
end
T.TD=abs((0:255)-double(T.MXM))*double(T.hist);
T.m=m;
T.p=p;
H=T;
%END InitH
```

The function *TotDev2* searches for the split in the intensity histogram that will result in the minimum L_1 fidelity term. Note that it only computes the total deviation at discrete points in the range of the image. This is in contrast to the calculation of the more general S curves described in chapter 3.

```
function SA=TotDev2(H)
% SA=TotDev2(H) returns these values in the Array SA:  SA(1) is the
% number of occurences of the minimum and should be 1.  SA(2) is
% the point where the best split will occur.  SA(3) is the
% value of the minimum value of 'S.'  The minimum point is found
% looking only at the points in the histogram.

% find all the values of S in the points of the histogram
S=zeros([256 1]);
for i=0:255,
  S(i+1)=SifSplit(H,i);
end
[Smin,imin]=min(S);
Nmins=sum(S==Smin);
% check for adjacent minimums
if (Nmins>1)
  i=1;
  while (S(imin+i)==Smin)
    i=i+1;
  end
  if i==Nmins
    % All the minimums are adjacent. Use the average index.
    imin=mean(imin:(imin+Nmins-1));
    Nmins=1;
  end
```

```
end
SA=[Nmins imin-1 Smin];
```

The function *SplitH* divides the histogram of the image into two parts. It is used to separate the intensity values for an initial condition with the minimum value of the fidelity term. There is a nested function named *SetUp* that makes separate data structures for each portion. These are then used as part of the minimization algorithm to record the histograms of the to phases of the segmentation.

```
function G=SplitH(H,Xs)
% G=SplitH(H,Xs) divides H into two histograms.
% G(1) contains all the values <= Xs.
% G(2) contains all the values > Xs.
% Xs should be uint8 and H should be the data structure
% described in function InitH.
% G(1), G(2) will have the same form as H.
if ~strcmp(class(Xs),'uint8')
  error('split level must be uint8');
end
if Xs==255
  error('unable to split with maximum uint8');
end
HL=H;
HU=H;
is=int32(Xs+1);
HL.hist(is+1:end)=0;
HU.hist(1:is)=0;
G=[SetUp(HL), SetUp(HU)];
% end SplitH

function H=SetUp(H)
% from the histogram H.hist, adds the rest of the parameters.
N=double(sum(H.hist));
m=int32(floor(N/2));
p=int32(N-2*m);
T=H;
T.chist(1)=T.hist(1);
i=int32(1);
if (p==0),
  % N is even
  while(T.chist(i)<m),
    j=i+1;
    T.chist(j)=T.chist(i)+T.hist(j);
    i=j;
  end
  T.MXM=i-1;
  while(T.chist(i)<m+1),
    j=i+1;
    T.chist(j)=T.chist(i)+T.hist(j);
    i=j;
  end
  T.MXP=i-1;
else
  % N is odd
  while(T.chist(i)<m+1),
    j=i+1;
    CP=T.chist(i);
    T.chist(j)=CP+T.hist(j);
    i=j;
  end
  T.MXM=i-1;
  T.MXP=T.MXM;
end
while(i<256),
  j=i+1;
  T.chist(j)=T.chist(i)+T.hist(j);
  i=j;
end
T.TD=abs((0:255)-double(T.MXM))*double(T.hist);
T.m=m;
T.p=p;
H=T;
%END SetUp
```

The functions *AdeltaS* and *RdeltaS* predict how the total deviation of a segmentation component will change if the given element is added or removed. They exploit the results of lemma 2 on page 40. The two functions are used by the greedy algorithm to decide whether or not to move a pixel from one segment to another.

```
function D=AdeltaS(H,v)
%D=AdeltaS(H,v)
%  D is the change in Total Deviation if the value v
%  were to be added to H.  H is not changed.
v1=int32(v);
if v1<=H.MXM
    DV=H.MXM-v1;
elseif v1>=H.MXP
    DV=v1-H.MXP;
else
    DV=0;
end
D=double(DV);
```

```
%END AdeltaS

function D=RdeltaS(H,v)
%D=RdeltaS(H,v)
%  D is the change in Total Deviation if the value v
%  were to be removed from H.  H is not changed.
v1=int32(v); % v is typically uint8
if v1<=H.MXM
    DV=v1-H.MXP;
elseif v1>=H.MXP
    DV=H.MXM-v1;
else
    DV=0;
end
D=double(DV);
%END RdeltaS
```

The functions *RemoveV* and *AddV* actually change the histograms of each segment when the greedy algorithm decides that such a change is beneficial. These are time consuming operations in MATLAB because parameters are passed by value. All that is really needed is to change a few elements of the data structure.

```
function H1=RemoveV(H,v);
% H1=RemoveV(H,V)
%   Returns an new histogram with the value V removed, the left and
%   right medians, and total deviation adjusted.
v1=int32(v);
if H.hist(v1+1)==0
  error('Attempted to remove a non-occuring value');
end

DTD=RdeltaS(H,v1);
if H.p==0
  if v1<=H.MXM
    MXM=H.MXP;
  else
    MXM=H.MXM;
  end
  MXP=MXM;
  H.p=1;
  H.m=H.m-1;
else
  % find X(m)
  i=int32(1);
  while H.chist(i)<H.m
    i=i+1;
  end
  XM=i-1;
  % find X(m+2)
  i=H.MXM+1;
  while H.chist(i)<H.m+2
    i=i+1;
  end
  XM2=i-1;
  if v1<H.MXM
    MXM=H.MXM;
    MXP=XM2;
  elseif v1>H.MXM
    MXM=XM;
    MXP=H.MXM;
  else
    MXM=XM;
    MXP=XM2;
  end
  H.p=0;
end
H.MXM=MXM;
H.MXP=MXP;
H.TD=H.TD+DTD;
%
v1=int32(v)+1; % need because MatLab uses 1 based array indexing
H.hist(v1)=H.hist(v1)-1;
H.chist(v1:256)=H.chist(v1:256)-1;
H1=H;
%END RemoveV

function H1=AddV(H,v);
% H1=AddV(H,v)
%   Returns an new histogram with the value v added, the left and
%   right medians, and total deviation adjusted.
v1=int32(v);
DTD=AdeltaS(H,v1);
if H.p==0
  if v1<=H.MXM
    MXM=H.MXM;
  elseif v1>=H.MXP
    MXM=H.MXP;
  else
    MXM=v1;
  end
  MXP=MXM;
  H.p=1;
else
  % find X(m) (this algorithm is correct, but inefficient)
  i=int32(1);
  while H.chist(i)<H.m
    i=i+1;
  end
  XM=i-1;
  %find X(m+1)
  XM1=H.MXM; % H.MXP would be the same since N=2m+p is odd.
  % find X(m+2)
  i=H.MXM+1;
  while H.chist(i)<H.m+2
    i=i+1;
  end
  XM2=i-1;
  % find the new medians
  if v1<=XM
    MXM=XM;
    MXP=XM1;
  elseif v1>=XM2
    MXM=XM1;
    MXP=XM2;
  elseif v1>=XM1
    MXM=XM1;
    MXP=v1;
  else
    MXM=v1;
    MXP=XM1;
  end
  H.m=H.m+1;
  H.p=0;
end
H.MXM=MXM;
H.MXP=MXP;
H.TD=H.TD+DTD;
%
v1=int32(v)+1; % need because MatLab uses 1 based array indexing
H.hist(v1)=H.hist(v1)+1;
H.chist(v1:256)=H.chist(v1:256)+1;
H1=H;
%END AddV
```

Appendix C

Notes and acknowledgments regarding the images

All the images used as examples are from public sources or from the author's private collection. Information and ackowledgments for each image are listed in this appendix. References to the "USC SIPI database" mean the Signal and Image Processing Institute image database

`http://sipi.usc.edu/database/`

at the University of Southern California.

1. Figure 2.5 on page 34 is a standard test image from the USC SIPI database.

2. Figure 4.1 on page 69 is from the author's private collection. The child is the author's granddaughter. The image is included with her parent's permission for use in this thesis and this thesis only.

3. Figure 4.2 on page 69 is a standard test image from the USC SIPI database. The original image is 512 × 512 pixels. It was down-sampled to 256 × 256 to produce the curve.

4. Figure 5.1 on page 79 is from the authors private collection. It is a photograph of the family dog 'Charlie' captured by Elizabeth Mason on the date shown.

5. Figure 5.2 on page 80 is another standard test image from the USC SIPI database. It is the familiar *Lena* image that appears so frequently in the image analysis literature.

6. Figure 5.8 on page 84 is a synthetic image. It is called a gaussian spot because the pixel values represent the value of a two dimensional gaussian. The 3σ point of the spot is near the edge of the image, so it contains most of the dynamic range. The image was constructed with the following Java code segment.

```
for (int x=0; x<width; x++)
  for (int y=0; y<height; y++) {
    r=(double)((x-centerX)*(x-centerX)+(y-centerY)*(y-centerY));
    GB[y][x]=scale*(1.0d - Math.exp(r/sigma2));
  }
```

Notice that the image is inverted in the sense that the darker areas represent larger values of the function.

7. Figure 6.1 on page 88 is a synthetic image constructed with the following code:

```
X=[]; img=[];
for j=1:256
    X(j)=200+50*(j-1)/256; end
for j=1:256
    img=[img ; X]; end
hs=32; cntr=128; rng=(cntr-hs):(cntr+hs); img(rng,rng)=25;
```

   The image illustrates characteristics that can cause local minima. The border is for visual separation and is not part of the image.

8. Figure 6.2 on page 90 shows the Andromeda Galaxy also known as M31. Image credit: NOAO/AURA/NSF. Obtained from

   `http://www.noao.edu/image_gallery/html/im0424.html`

Bibliography

[1] Milton Abramowitz and Irene A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. NBS applied mathematics series number 55. National Bureau of Standards, Washington, D.C., June 1964. Ninth Dover printing; tenth Government Printing Office printing.

[2] Luis Alvarez, Pierre-Louis Lions, and Jean-Michel Morel. Image selective smoothing and edge detection by nonlinear diffusion. II. *SIAM Journal on Numerical Analysis*, 21(3):845–866, June 1992.

[3] P. Arabie, L. J. Hubert, and G. De Soete, editors. *Clustering and Classification*. World Scientific, 1997.

[4] Peter Bloomfield and William L. Steiger. *Least Absolute Deviations*. Birkhäuser, 1983.

[5] Tien D. Bui, Song Gao, and Q. H. Zhang. New image segmentation models. In M. W. Marcellin, editor, *Proceedings of The Seventh IASTED International Conference on Signal and Image Processing, Honolulu, Hawaii, USA August 15-17, 2005*, pages 400–403. ACTA Press, 2005.

[6] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[7] T. Chan, S. Esedoglu, F. Park, and A. Yip. Recent development in total variation image restoration. Technical Report 1, University of California Los Angeles, January 2005.

[8] Tony Chan and Luminita Vese. Image segmentation using level sets and the piecewise-constant Mumford-Shah model. Technical Report 14, University of California Los Angeles, July 2000.

[9] Tony F. Chan and Selim Esedoglu. Aspects of total variation regularized L_1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

[10] Tony F. Chan, Selim Esedoglu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. Technical Report 54, University of California Los Angeles, September 2004.

[11] Tony F. Chan and Jianhong (Jackie) Shen. *Image Processing and Analysis.* Society for Industrial and Applied Mathematics, Philadelphia, 2005.

[12] Tony F. Chan, Jianhong (Jackie) Shen, and Luminita Vese. Variational PDE models in image processing. *Notices of the AMS*, 50(1):14–26, January 2003.

[13] Tony F. Chan and Luminata A. Vese. A level set algorithm for minimizing the Mumford-Shah functional in image processing. Technical Report 13, University of California Los Angeles, April 2000.

[14] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.

[15] Charles K. Chui. *Wavelets: A Mathematical Tool for Signal Analysis*, chapter 2, pages 30–32. SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[16] Ronald H. Y. Chung, Nelson H. C. Yung, and Paul Y. S. Cheung. An efficient parameterless quadrilateral-based image segmentation method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1446–1457, September 2005.

[17] Douglas Comer. A note on median split trees. *ACM Transactions on Programming Languages and Systems*, 2(1):129–133, January 1980.

[18] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on theory of computing, Shaker Heights, Ohio, United States*, pages 151–158, New York, 1971. ACM Press.

[19] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to algorithms*, chapter 17, pages 329–355. MIT electrical engineering and computer science series. McGraw-Hill, New York, 1990.

[20] Jérôme Darbon and Marc Sigelle. A fast and exact algorithm for total variation minimization. Technical Report 2, École Nationale Superieure des Télécommunications, January 2005. Départment Traitement du Signal et des Images Group Traitements Statistiques et Applications aux Communications.

[21] Rodney G. Downey et al. Cutting up is hard to do: the parameterized complexity of K-cut and related problems. *Electronic Notes in Computer Science*, 78:1–14, April 2003. Copy in authors files obtained from http://www.sciencedirect.com.

[22] Selim Esedoglu and Yen-Hsi Richard Tsai. Threshold dynamics for the piecewise-constant Mumford-Shah functional. Technical Report 63, University of California Los Angeles, October 2004.

[23] Lawrence C. Evans and Ronald F. Gariepy. *Measure theory and fine properties of functions.* CRC Press, Boca Raton, 1992.

[24] Mário Figueiredo. Bayesian image segmentation using wavelet-based priors. In *Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2005. IEEE Computer Society.

[25] Walter D. Fisher. On grouping for maximum homogeneity. *Journal of the American Statistical Association*, 53(284):789–798, December 1958.

[26] Song Gao and Tien D. Bui. Image segmentation and selective smoothing by using Mumford-Shah model. *IEEE Transactions on Image Processing*, 14(10):1537–1549, October 2005.

[27] Roman Garnett, Timothy Huegerich, Charles Chui, and Wenjie He. A universal noise removal algorithm with an impulse detector. *IEEE Transactions on Image Processing*, 14(11):1747–1754, November 2005.

[28] Teofilo F. Gonzalez. On the computational complexity of clustering and related problems. In R. F. Drenick and F. Kozin, editors, *System Modeling and Optimization: Proceedings of the 10th IFIP Conference, New York City, USA, August 31-September 4, 1981*, volume 38 of *Lecture Notes in Control and Information Sciences*, pages 174–182, Berlin/Heidelberg, 1982. Springer. Library of Congress call number QA402 S958 1982; ISBN: 0387116915.

[29] Leo Grady and Eric L. Schwartz. Isoperimetric graph partitioning for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):469–475, March 2006.

[30] Matthias Heiler and Christoph Schnörr. Natural image statistics for natural image segmentation. *International Journal on Computer Vision*, 63(1):5–19, 2005.

[31] F. K. Hwang. Optimal partitions. *Journal of Optimization Theory and Applications*, 34(1):1–10, May 1981. Library of Congress call number QA402.5.A13.

[32] John Immerker. Fast noise variance estimation. *Computer Vision and Image Understanding*, 64(2):300–302, 1996.

[33] Zoltan Kato and Ting-Chuen Pong. A markov random field image segmentation model using combined color and texture features. In Wladyslaw Skarbek, editor, *Proceedings of the Ninth International Conference on Computer Analysis of Images and Patterns, Warsaw, Poland, September 5-7, 2001*, volume 2124 of *Lecture Notes in Computer Science*, pages 547–554. Springer, 2001.

[34] Leonard Kaufman and Peter J. Rousseeuw. *Finding groups in data : an introduction to cluster analysis*. Wiley, 1990.

[35] Qifa Ke and Takeo Kanade. Robust subspace computation using L_1 norm. Technical Report 172, Carnegie Mellon University School of Computer Science, 2003.

[36] Qifa Ke and Takeo Kanade. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2005. IEEE Computer Society.

[37] Mirko Křivánek and Jaroslav Morávek. NP-hard problems in hierarchical tree clustering. *Acta Informatica*, 23(3):311–323, June 1986.

[38] R. Manmatha and Jamie L. Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1212–1225, August 2005.

[39] Boriana L. Milenova and Marcos M. Campos. O-cluster: Scalable clustering of large high dimensional data sets. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, page 290. IEEE Computer Society, 2002.

[40] Jean-Michel Morel and Sergio Solimini. *Variational Methods in Image Segmentation.* Birkhäuser, Boston, 1995.

[41] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989. Library of Congress call number QA1 C718.

[42] Mila Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers. *SIAM Journal on Numerical Analysis*, 40(3):965–994, 2002.

[43] Stanley Osher and Ronald P. Fedkiw. Level set methods: An overview and some recent results. Technical Report 8, University of California Los Angeles, September 2000.

[44] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, November 1988.

[45] Stanley Osher, Andrés Solé, and Luminita Vese. Image decomposition and restoration using total variation minimization and the H^{-1} norm. Technical Report 57, University of California Los Angeles, October 2002.

[46] Thierry De Pauw and Didier Smets. On explicit solutions for the problem of Mumford and Shah. *Communications in Contemporary Mathematics*, 1(2):201–212, May 1999. Obtained from http://www.worldscinet.com.

[47] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[48] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, pages 259–268, 1992.

[49] B. A. Sheil. Median split trees: A fast lookup technique for frequently occurring keys. *Communications of the ACM*, 21(11):947–958, November 1978.

[50] Bing Song. *Topics in Variational PDE Image Segmentation, Inpainting and Denoising*. PhD thesis, University of California Los Angeles, 2003.

[51] Bing Song and Tony Chan. A fast algorith for level set based optimization. Technical Report 68, University of California Los Angeles, December 2002.

[52] Sinisa Todorovic and Michael C. Nechyba. Dynamic trees for unsupervised segmentation and matching of image regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1762–1777, November 2005.

[53] Luminita A. Vese and Stanley J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19(3):553–572, December 2003.

[54] Markus von Oehsen. *Multiscale Methods for Variational Image Denoising*. PhD thesis, Universität zu Lübeck, 2002.

[55] Yang Wang, Kia-Fock Loe, and Jian-Kang Wu. A dynamic conditional random field model for foreground and shadow segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):279–289, February 2006.

[56] Robert Weinstock. *Calculus of Variation*. McGraw-Hill, 1952.

[57] Wotao Yin, Donald Goldfarb, and Stanley Osher. Image cartoon-texture decomposition and feature selection using the total variation regularized L_1 functional. Technical Report 1, Columbia Univiersity, 2005. Computational Optimization Research Center (CORC) technical report.

[58] Jian Yu. General C-means clustering model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1197–1211, August 2005.

[59] Lu Yu, Qiao Wang, Lenan Wu, and Jun Xie. Mumford-Shah model with fast algorithm on lattice. Preprint in author's files received from Charles Chui, January 2006.

[60] G. Udny Yule. *An Introduction to the Theory of Statistics*, pages 144–145. Charles Griffin and Company, London, 1911.