RICE UNIVERSITY

A Coding Theoretic Approach To Image Segmentation

by

Unoma Ndili

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Science

Approved, Thesis Committee:

Dr. Robert Nowak, Chair
Assistant Professor
Electrical and Computer Engineering

Dr. Richard Baraniuk
Professor
Electrical and Computer Engineering

Dr. Hyeokho Choi
Faculty Fellow
Electrical and Computer Engineering

Houston, Texas

April, 2001

A Coding Theoretic Approach To Image Segmentation

Unoma Ndili

Abstract

Using a coding theoretic approach, we achieve unsupervised image segmentation by implementing Rissanen's concept of *Minimum Description Length* (MDL) for estimating piecewise homogeneous regions in images. MDL offers a mathematical foundation for balancing brevity of descriptions against their fidelity to the data. Our image model is a Gaussian random field whose mean and variance functions are piecewise constant. Our model is aimed at identifying regions of constant intensity (mean) and texture (variance).

Based on a multi-scale encoding approach, we develop two different segmentation schemes. One algorithm is based on an adaptive (greedy) rectangular partitioning, while the second algorithm is an optimally-pruned wedgelet-decorated dyadic partitioning scheme. We compare the two algorithms with the more common signal plus constant noise schemes, which accounts for variations in mean only.

We explore applications of our algorithms on *Synthetic Aperture Radar* (SAR) imagery. Based on our segmentation scheme, we implement a robust *Constant False alarm Rate* (CFAR) detector towards Automatic Target Recognition (ATR) on *Laser Radar* (LADAR) and *Infra-Red* (IR) images.

Acknowledgments

I would like to express my deepest gratitude and appreciation to my thesis advisors and mentors: Drs. Richard Baraniuk and Robert Nowak, for their excellent guidance, purposeful leadership and words of encouragement through out the project. I also deeply appreciate the unwavering support, advice and insightful contributions of Dr. Hyeokho Choi.

I thankfully acknowledge Clayton Scott and Yi Wan who painstakingly proof read my thesis and offered very useful suggestions. Thanks to all my friends here at Rice, especially my office mates and members of the Digital Signal Processing group for their friendly assistance and encouragement.

I am extremely grateful to my family for all their wonderful support and patience. Special thanks to my brothers, Amaechi and Awele, for their faith in me. I would like to acknowledge especially, my sister Uju, who listened to me all the time and gave me words of inspiration, strength and hope.

My special thanks to my loving parents, Professor and Mrs. Frank Ndili, whose vision, support and prayers over the years have remained foundational to my motivation and success.

Finally, my deepest gratitude is to God who is there with me, now and always.

For whom the son shall set free, he shall be free indeed!

john 8[36]

Contents

Illustrations

Chapter 1

Introduction

Image processing can be broadly classified into two main areas: Image enhancement techniques which are geared toward improving the overall perceptual quality of an image (e.g. denoising), and image analysis. This project is mainly concerned with image segmentation, a problem of image analysis in which we need to further process an image in a way that renders it suitable for subsequent analysis.

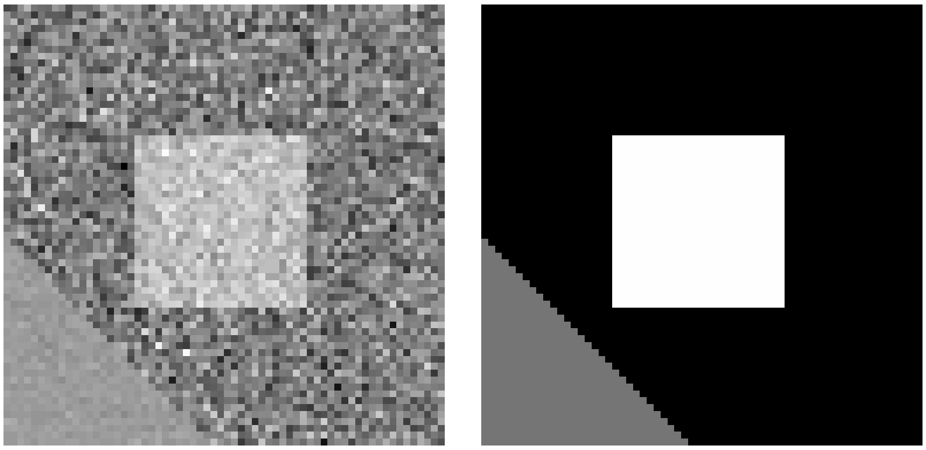


Figure 1.1 Illustration of segmentation (a) Original pixel-based image. (b) The segmented image defined by the location of boundaries and representative values for each region.

The overall aim of image segmentation is to divide up an image into meaningful connected regions or areas that are each homogeneous with respect to some statistical characteristics, such that adjacent regions differ with respect to the same criteria. A casual look at Fig 1.1 reveals the simple idea underlying image segmentation. There are obviously three regions in this image, each of which has a distinct brightness or gray level. Moreover, the three regions are spatially distinct: they lie within a defined boundary. Image segmentation is the process by which a computation translates the original image in Fig 1.1(a) into a description of the three regions, with representative properties as shown in Fig 1.1(b), where each distinct segment is represented with a common parameter, in this case its average intensity.

Image segmentation requires that we identify and isolate components of interest within an image. We use the following examples to illustrate the usefulness of segmentation.

1. A meteorologist presented with a satellite image showing various weather formations may require further information about the particular size and distribution of specific cloud formations in relation to the area under consideration. The first step would be to isolate regions of the satellite image representing possible cloud formations.

2. A surgeon may require that Magnetic Resonance (MR) images of his patient's brain be processed to yield regions of possible 'activity', to help in detecting formations of a tumor. To enable this, the MR image needs to be segmented into component regions representing tumor, or else tumor-free areas.

3. In military applications, large datasets from long-range radar images of enemy territory need to be processed to accurately identify possible 'target' locations and enable precise combat or attack strategies. Image segmentation is used in identifying different regions in these radar images, so that areas which are statistically similar to target regions are kept for further processing, while all other clutter regions are 'thrown away'.

In all three cases, it is required that homogeneous regions of an image under certain pre-defined criteria or characterization, be isolated and identified. This is the process generally referred to as image segmentation. Segmentation is often claimed as the most important image processing application [WS88]. This is because image segmentation is often used as an essential front end step in several subsequent image processing applications. Such applications include pattern recognition, automatic object detection, motion analysis and tracking, image compression, stereo vision, scene interpretation and image database retrieval algorithms; the performance of which often rely on a high quality of image segmentation [Teu93]. For this reason, image segmentation has received a lot of attention and a number of different approaches have been proposed in the literature. These approaches basically fall into four broad classes, namely: Classification-based methods [CD00, CMP00, CK98], Edge-based methods [FLJ95], Shape-based methods [Li92] and Region-based methods [WS88]. A general review of image segmentation techniques may be obtained from [WS88, Teu93].

We note that the principal gain of segmentation is a tremendous reduction of dimensionality. A pixel based representation of an image typically has a dimension several times more than that required to identify common regions by the same statistic, because segmentation aims at removing redundancy in the data.

While it may be easy for the human eye to determine areas of similar characteristics just by observing an image, it is not so trivial to devise computational algorithms to segment an image. It is necessary to first pre-define the criteria based on which segmentation is reasonable. This criteria may vary from image to image. It is also important to allow a flexible scheme that enables an algorithm to divide out a region if it considers it sufficiently different from its neigbours. Excessive flexibility however, may result in over-segmentation which does not gain much over a pixel-representation of the image. In addition segmentation algorithms are usually computationally intensive, often requiring searches over a large feature space. This is restrictive in applications that require real-time image segmentation.

In addition, certain segmentation schemes require prior *training* or supervision [CMP00, VCB00]. Such algorithms follow a two-step procedure: training to obtain an accurate statistical model for the data, followed by segmentation. Training requires several reliable sets of training data for every possible segmentation class, from which a set of parameters called the characteristic signature of the underlying class, is extracted. Given the image data, we are then able to determine the class with which the data has the closest match. Key issues with supervised schemes are: the definition of what constitutes the signature for a class and the fact that often insufficient training data results in unreliable class signatures.

In this thesis we consider a coding theoretic approach to image segmentation which achieves **unsupervised** segmentation. This enables us to segment an image without training, into spatially homogeneous regions, the nature and number of which we have very little prior knowledge. We apply Rissanen's concept of Minimum Description Length (MDL) [Ris89, Sai94, KS99] to estimate piecewise homogeneous regions of an image. The MDL principle provides a mathematical foundation for data dependent model selection based on an optimal trade-off between brevity of model description and its fidelity to the data. It does this by penalizing overly complex segmentation representations. Our approach is a region-based scheme which attempts to cluster unknown image data into regions based on spatial similarity among pixels. Our predefined criteria for homogeneity are the mean and variance of regions. That is, if either the mean or the variance (or both) of a region differs from that of any of its adjacent regions, then we want to identify and isolate that region, and represent it with its underlying constant mean and variance.

The image model we assume is a Gaussian random field whose mean and variance functions are piecewise constant across the image. Our segmentation scheme is aimed at automatically identifying regions of both varying mean (intensity) and/or variance (unstructured texture). This model differs from the so called*signal plus constant noise* models used in most wavelet-based multi-scale image analysis procedures [HY00, CK98, CD00]. Constant noise models identify and segment regions based only on differences in their mean values, ignoring first order textural properties of the image.

Our approach is motivated by the fact that textural properties in images can provide important information about objects in the image. Several texture-only segmentation schemes have been developed to identify textural characteristics in images [CD00, FMLCC98]. Other unsupervised schemes have achieved segmentation based only on intensity values, without consideration for textural variations (so called constant variance models) [VNBmF00, SBB+00]. To the best of our knowledge, this scheme is the first in the literature to propose a generic unsupervised segmentation algorithm based on both mean and variance. This finds applications in a wider class of natural and radar images including SAR imagery.

We consider images as generically consisting of arbitrarily shaped regions, with each region having its own mean and variance (to capture first order statistics of any region). This view implies that we do not necessarily need an accurate statistical model of the data-generating distribution for an image, so that a model (such as our assumed Gaussian distribution) which captures this variation in mean and/or variance suffices.

Suppose the m-by-n image y under consideration is composed of connected regions of pixels, with each pixel assumed to be independently distributed according to a

Gaussian density. Our model is:

$$y(i_1, i_2) = \mu(i_1, i_2) + \sigma(i_1, i_2)\ z \quad 0 \leq i_1 \leq m, \ \ 0 \leq i_2 \leq n, \tag{1.1}$$

where $z \sim$ i.i.d. $\mathcal{N}(0, 1)$ so that y is distributed according to the normal density $\mathcal{N}(\mu(i_1, i_2), \sigma^2(i_1, i_2))$. The probabilistic model in Eqn.(1.1) is useful in implementing the asymptotically *optimal* encoding scheme according to Shannon [Sha48] (see Theorem 1). The MDL criterion requires that the parameters of the assumed model as well as the data under this model, be transmitted to the receiver. We adopt a multi-scale progressive coding scheme in which we initially transmit to the receiver the parameters (or equivalently sufficient statistics for the parameters) of the assumed model, and then the data using the applicable probability model. Under this assumption, we must obtain the conditional distribution of the data, given that these sufficient statistics are assumed already known to the receiver.

We propose two different approaches to this optimal progressive coding approach to image segmentation. The two approaches involve a tradeoff in terms of accuracy versus speed. The **Adaptive Recursive Partitioning** (ARP) is a greedy scheme which achieves rectangular (not necessarily dyadic) tessellations while the **Wedgelet Decorated Dyadic Partitioning** (WDDP) is an optimal multi-scale analysis restricted to dyadic partitions but in addition allows for wedge-splits. The WDDP is based on the dyadic CART algorithm originally developed by Donoho in [Don99]. The difference is that Donoho's method is a constant variance algorithm whereas our model also enables the detection of changes in variance in addition to changes in intensity, as will be illustrated in our results.

The remaining part of this thesis is organized as follows: In Chapter 2 we review basic coding theory and the MDL principle. We also derive the conditional density required by our Gaussian model in the MDL criteria. Chapter 3 develops our two schemes: the ARP and WDDP algorithms. We analyze and compare the two schemes with the constant variance model [Don99]. In Chapter 4, we present applications of our schemes for segmentation of synthetic aperture radar (SAR) data. We also discuss a robust constant false alarm rate (CFAR) detector for automatic target detection in laser radar (LADAR) and uncooled Infra-Red (IR) imagery. We show examples to illustrate the performance of these algorithms. Finally we present concluding remarks in Chapter 5.

Chapter 2

Basic Coding Concepts and the MDL Principle

In this chapter we summarize relevant facts about source coding theory and the MDL principle. Taking a coding theoretic approach to the image segmentation problem, we assume that the image to be segmented is an information source $(\mathsf{X}^n, p(\mathsf{X}^n))$ defined by the alphabet X^n and a probability function p, with domain X^n. This information source is to be efficiently modeled, optimally encoded and transmitted to a receiver with the fewest possible number of bits.

2.1 Optimal Codes

One of the main objectives of coding in information theory is to shorten the description of a data string. This uses the basic idea that short codewords are assigned to frequently occurring data points, while the rare ones are assigned longer codewords.

Given a finite set $X^n = (x_1, x_2, \cdots x_n)$ of data sequences of length n, a code is a one-to-one function from X^n to B^*, the set of all finite binary strings. An important practical restriction on the codes we are interested in is that no codeword is allowed to be the prefix of another code word. It is well known [CA92] that such *prefix* codes have desirable properties such as: 'instantaneous' decodability and satisfying the important Kraft's inequality given as:

$$\sum_{x \in X^n} 2^{-L(x)} \leq 1$$

$L(x)$ is the length of the codeword x and $\sum_{x \in X^n} L(x)p(x)$ is the mean length of the code words.

An *optimal prefix code* is defined as one with the shortest mean code length per symbol. Given *any* distribution p on X^n, the optimal prefix code has a length function equal to $\mathsf{L}_p(X^n) = -\log_2 p(X^n)$.

In this thesis we assume logarithm to base 2 so that the code lengths are in bits. If we now suppose that the elements of X^n are generated according to a known *true* probability distribution q defined on X^n, then the mean length is bounded from below by the entropy $H(q)$ according to Shannon's Theorem [Sha48].

> **Theorem 1** *(Shannon's Source Coding Theorem)* Suppose elements of X^n are generated according to a true probability distribution q. For any prefix code on X^n, the expected code length $L_p(X^n)$ is bounded below by $H(q)$, the entropy of $q(X^n)$. That is,
>
> $$L_p(X^n) \geq H(q) \equiv -\sum_{x \in X^n} q(x) \log_2 q(x)$$

We adopt the convention that $\log 0$ is negative infinity and $0 \log 0 = 0$. Shannon's theorem basically states that no prefix code for the set X^n exists with mean length below the entropy. In this sense then, we may take $-\log q(X^n)$ as the optimal code length we should attempt to achieve.

We must emphasize however that any distribution p defined on X^n, not necessarily the data-generating or true distribution, can be used to encode the data X^n. In fact, in most statistical applications, the true distribution q is rarely known, and to a large extent, most applications are concerned with codes built from approximations of the true distribution.

For our purposes we take the *optimal* code length as defined for a given probability distribution $p(X^n|\underline{\Theta})$ i.e., $\mathsf{L}_p(X^n)$, where $\underline{\Theta}$ is the set of parameters that define the distribution. Here, $p(X^n|\underline{\Theta})$ may or may not be equivalent to the data generating distribution. Broadly, codes based on $p(X^n|\underline{\Theta})$ remove *redundancy* from the data without loss of information.

2.2 The MDL Principle

The purpose of a probabilistic model is to explain observed statistical properties of the data. The need for model selection arises when we have a set of K models $\{p_k(X^n|\underline{\Theta_k})\}_{k=1}^{K}$, competing to explain the available data. A good choice is considered a concise model that provides a good fit to the data. We must therefore imbue our algorithms with an urge to favor simple representations over complex ones. That is, if there are several descriptions compatible with the observed image, we select the most parsimonious.

The *Minimum Description Length* MDL criterion decides that among the K possibilities, the 'best' model is the one that minimizes the description length required to code the data, obtained by assuming a *two-part* code given as:

$$L(X^n) = L(\underline{\hat{\Theta}_k}) - \log p(X^n|\underline{\hat{\Theta}_k}), \quad k = 1, \ldots K, \tag{2.1}$$

where $L(\underline{\hat{\Theta}_k})$ is the code length required to describe the *Maximum Likelihood Estimate* (MLE) $\underline{\hat{\Theta}_k}$ of the parameter set $\underline{\Theta_k}$ such that the decoder unambiguously knows the model under which the code for the data was obtained. This is a generally known as a penalized likelihood with $\log p(X^n|\underline{\hat{\Theta}_k})$ as the likelihood term and $L(\underline{\hat{\Theta}_k})$ as the penalty term. The penalty represents the cost required to encode the estimated parameter values $\underline{\hat{\Theta}_k}$. This *two-part* code was suggested by Leclerc [Lec89] for applying the MDL principle to image segmentation. A block diagram illustrating a general MDL-based coding system for an image is shown in Fig.2.1.

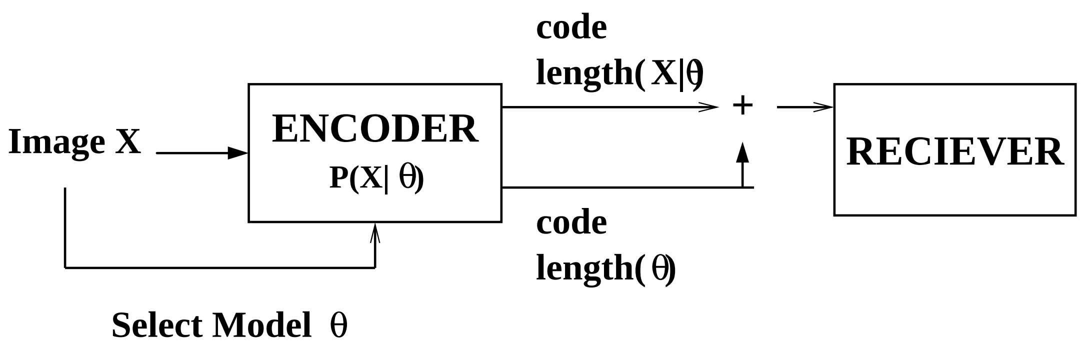


Figure 2.1 Block diagram illustrating the MDL-based coding scheme

MDL is used as a tool to decide what model (or class of model) 'best' suits the image data by penalizing more complex representations. In effect MDL basically provides a mathematical foundation for the optimal trade-off between the complexity (or brevity) of a model and the inaccuracy or deviations of the model from the data. This idea is illustrated in Fig. 2.2, where to transmit the original pixel-based image shown in (a) would require a lot of bits to encode each individual pixel's value. This representation would definitely be most accurate but also very complex as per code length. If instead, we use a simpler representation such as shown in Fig. 2.2(b), we would be able to encode the data with far fewer bits while loosing some accuracy in the individual values of the pixel and location of the boundary.

For example, suppose we want to transmit the boundary between the two regions in this image. If we pick a curve so complex that all the deviations from the true boundary are zero, nothing will be gained since encoding such a curve requires that we code the positions of all the observations that fall on it. On the other hand, if we take the curve too simple, we need a lot of bits to encode the remaining large deviations from 'truth'. Somewhere between these extremes, MDL attempts to find the optimally smooth curve we desire.

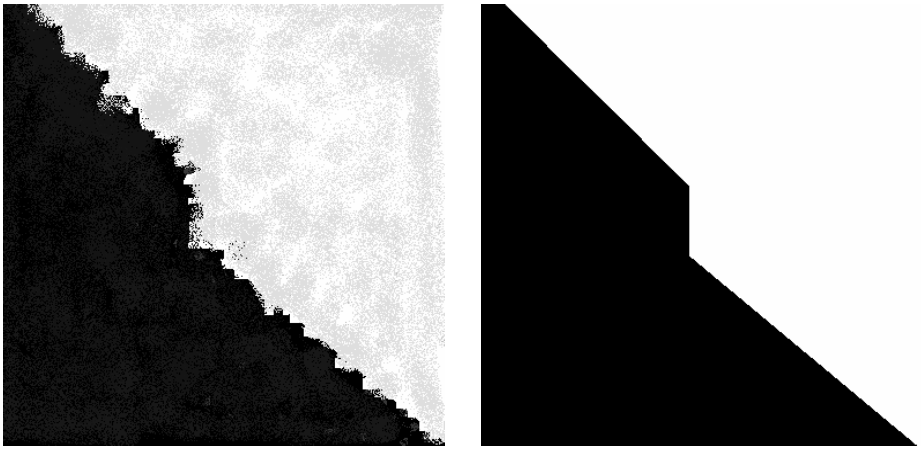


Figure 2.2 Illustration of trade-off between complexity and accuracy of model. (a) The original image. (b) A less accurate but also less complex representation of (a).

When applying MDL, our focus is on casting statistical models as a means of generating codes, with the resulting code lengths providing a metric by which we can compare competing models. We note that we can compare code lengths without actually generating the codes, thereby making the implementation details of encoding somewhat unimportant. MDL has been successfully used in several image analysis/segmentation problems [VNBmF00, NF99, Lan99, Li92].

2.2.1 Two-stage Description Length

For the two-stage coding scheme, we first choose a member of the class of distributions to model our data with, and then use this distribution to encode X^n.

The penalty term: In the first stage, we communicate an estimate $\underline{\hat{\Theta}_k}$ (obtained by, say, MLE or some Bayes procedure). It is necessary to address how to encode this parameter estimate. This is typically done [HY01] by discretizing a compact parameter space with precision $\frac{1}{\sqrt{n}}$ for each member of $\underline{\hat{\Theta}}$, and then transmitting $\underline{\hat{\Theta}_k}$ with a uniform encoder. Rissanen ([Ris89, Ris96]) shows that this choice of precision is optimal in regular parametric families.

The intuitive argument is that $\frac{1}{\sqrt{n}}$ represents the magnitude of the estimation error in $\underline{\hat{\Theta}_k}$ and hence there is no need to encode the estimator with greater precision. Therefore, assuming the standard parametric rate, we will require a total of $-\log_2 \frac{1}{\sqrt{n}} = \frac{1}{2}\log_2 n$ bits to transmit each parameter.

The likelihood term: In the second stage, we transmit the data given a parameter estimate. This term seeks to maximize the likelihood of the data fitting our model over all possible parameter values $\underline{\Theta_k} \in \underline{\Theta}$, i.e., it finds the parameter estimate that maximizes $p(X^n)$. The most popular estimation technique is derived from the Maximum Likelihood (ML) Principle pioneered by Fisher [Edw72]. If $L(\underline{\Theta})$ is constant over all model classes, then the MDL principle seeks a model that minimizes $-\log p(X^n|\underline{\Theta})$. This is equivalent to maximizing the log likelihood $\log p(X^n|\underline{\Theta}))$ among all densities in the family, so that MDL coincides with MLE in parametric estimation problems in this case. Obviously, the MLE without the penalty term will yield the data exactly, resulting in the most complex, most accurate representation with no savings in description length. The penalty term balances this complexity of representation with accuracy of description of the image.

2.3 MDL For Gaussian Data

Under a continuous Gaussian distribution assumption, the elements of $\underline{\hat{\Theta}_k}$ are real-valued and have to be truncated to finite precision in order to yield finite code-length $L(\underline{\hat{\Theta}_k})$. We use the standard $\frac{1}{2}\log n$ code-length (explained above) for each components of $\underline{\Theta_k}$ based on asymptotic approximations [Ris89, HY01]. To encode the image given our progressive transmission scheme, we derive below the conditional density for the data, given the parameter estimates (or equivalently, their corresponding sufficient statistics).

2.3.1 Conditional Density for Gaussian with Unknown Mean and Variance

Let us assume that $X^n = (x_1, \ldots, x_n)$ is a set of i.i.d. $\mathcal{N}(\mu, \sigma^2)$ samples corresponding to a homogeneous region to be transmitted. Our coding scheme based on the MDL criteria first describes the parameters (in this case sample mean μ and sample variance σ^2), or equivalently, the corresponding sufficient statistics, of the prevailing model to the receiver. The sufficient statistics that have to be collect to obtain estimates of μ and σ^2 are the sum $\sum_{i=1}^n x_i$ and sum of squares $\sum_{i=1}^n x_i^2$. These constitute the information that is initially sent to the decoder.

Next, to describe the data under this assumed parametric model, given that the receiver already knows the parameters, we need to determine the conditional density $-\log p(X^n | \sum_i^n x_i, \sum_i^n x_i^2)$ based on which we build the code for the data.

Our Gaussian model parameterized by the mean μ and variance σ^2 has a density function given as:

$$p(X^n|\mu, \sigma^2) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_{i=1}^n x_i^2 - 2\mu\sum_{i=1}^n x_i + \mu^2\right)\right\} \tag{2.2}$$

Eqn. 2.2 may be expressed in exponential family form with two natural parameters $\zeta_1 = \frac{\mu}{\sigma^2}$ and $\zeta_2 = -\frac{1}{2\sigma^2}$ for which $\sum x_i \equiv t_1$ and $\sum x_i^2 \equiv t_2$ are the corresponding sufficient statistics. Using Neyman's factorization [BS94], we can write $p(X^n|\zeta_1, \zeta_2) =$

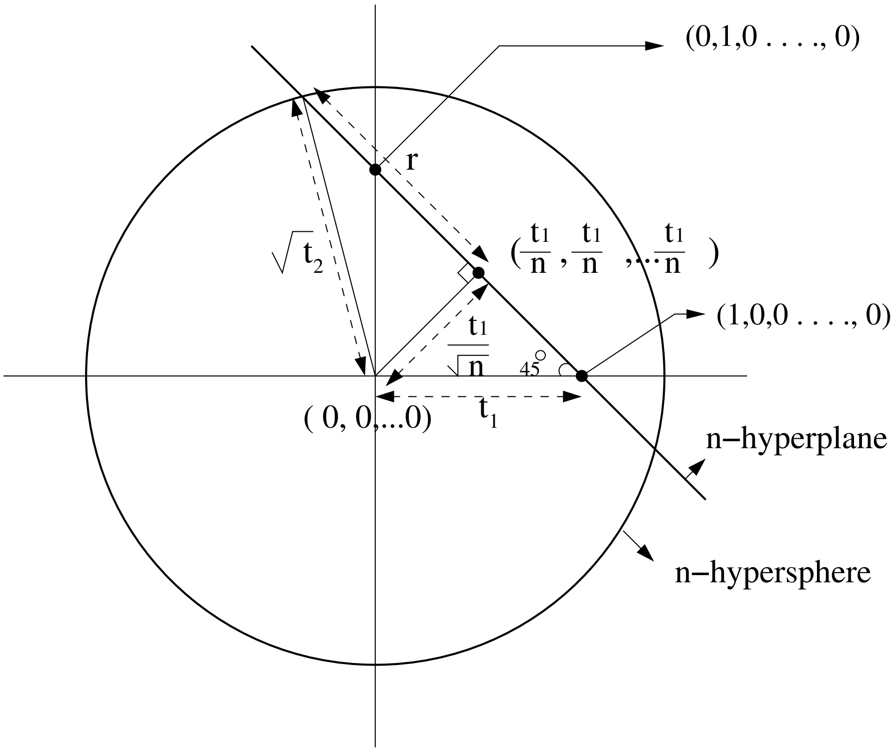


Figure 2.3 Deriving the radius of an n-1 sphere

$p(X^n|t_1,t_2)p(t_1,t_2|\zeta_1,\zeta_2)$, so that we have the following probabilistic interpretation:

$$\begin{aligned} p(X^n|\zeta_1,\zeta_2) &= \left(\frac{-\zeta_2}{\pi}\right)^{\frac{n}{2}} \exp\left(\frac{n\zeta_1^2}{4\zeta_2}\right) \exp\left(\zeta_1 \sum x_i + \zeta_2 \sum {x_i}^2\right) \\ &= \underbrace{\psi(\zeta_1,\zeta_2)\exp(\zeta_1 t_1 + \zeta_2 t_2)}_{\propto p(t_1,t_2|\zeta_1,\zeta_2)} \underbrace{1}_{\propto p(X^n|t_1,t_2)} \end{aligned} \tag{2.3}$$

where ψ is a function.

Surprisingly the conditional density $p(X^n|t_1,t_2)$ is proportional to 1 showing that it is uniform on the constraint set $\mathcal{C}(t_1,t_2) = (X^n : \sum x_i = t_1 \text{ and } \sum {x_i}^2 = t_2)$. The normalizing constant is then simply the surface area of the constraint set $\mathcal{C}$ which is an $(n-1)$-sphere resulting from the intersection of an $n-$sphere of radius $\sqrt{t_2}$ centered at the origin, with an $(n-1)$ dimensional hyper-plane. We proceed to determine the radius of this $(n-1)$-sphere to be able to find its surface area.

We note that the hyper-plane intersects the axis of the hyper-sphere at $45°$ in R^n. The shortest distance from the hyper-plane to the origin is given by the vector from the point $(\frac{t_1}{n}, \cdots, \frac{t_1}{n})$ on the hyper-plane to the origin. This vector has a length $\frac{t_1}{\sqrt{n}}$, and is perpendicular to the hyper-plane. We may then apply the Pythagorem theorem [JKB94] to obtain the radius r of this $(n-1)$-sphere as $r = \sqrt{t_2 - \frac{{t_1}^2}{n}}$. This derivation is illustrated in Figure 2.3. Using the standard formula for the surface area of an n-sphere given as: $r^{n-1}\frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}$. We obtain the uniform density as the inverse of this surface area so that we have the joint p.d.f. of X^n conditioned on t_1 and t_2 as:

$$p(X^n|t_1, t_2) = \begin{cases} \frac{\Gamma(\frac{n-1}{2})r^{2-n}}{2\pi^{\frac{n-1}{2}}} & \Leftarrow \quad X^n \in \mathcal{C}(t_1, t_2) \\ 0 & \Leftarrow \quad X^n \notin \mathcal{C}(t_1, t_2). \end{cases} \tag{2.4}$$

with $\Gamma(\cdot)$ denoting Euler's gamma function.

Chapter 3

Two Segmentation Algorithms

In this chapter we consider two different approaches to image segmentation based on our image model in Eqn. 1.1, where the underlying intensity and variance functions of our assumed independent Gaussian distributed image data are both varying. MDL has been discussed in the previous section as a tool for model selection, and will be used as the basic building block in both algorithms to decide between models classes. That is, given all reasonable possible parametric model representations of an image (or a region of the image), we determine appropriate code lengths for each model as per MDL and choose as the best representation, the one that gives us the minimum code length. Recall that according to MDL, we first code and transmit sufficient statistics for the model parameters, and then the data using the conditional distribution obtained in Eqn.2.4, given that model.

3.1 Multi-scale Tree Based Approach

Suppose we want to encode and transmit an m-by-n image $y(i_1, i_2)$, for $0 \leq i_1 \leq m$, $0 \leq i_2 \leq n$, to a receiver. We use a multi-scale tree based approach as illustrated in 3.1. .20.0

We start by defining a tree structure on our segmentation process, similar to the idea of regression trees used in statistical classification [Edw72]. The entire image is represented by the root node of this tree. At each level, we decide the best coarse or average representation of the image data in a region, and recursively refine this

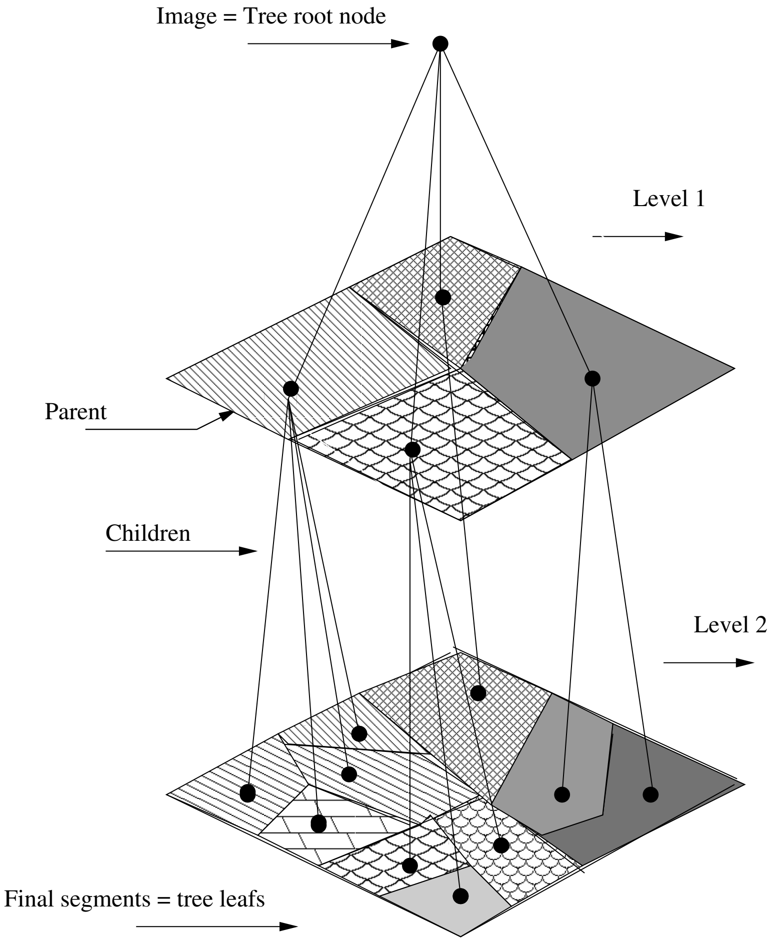


Figure 3.1 Multi-scale representation of the segmentation of an image into homogeneous regions.

representation as we proceed down the tree, until we achieve the best segmentation of the image. The final segments are represented as the leafs of this tree. The number of segments we may split into at each level, and how we split a region is determined by the possibilities allowed by our algorithms.

Our two approaches basically differ in the way that this tree is grown, either from top down, in a greedy way as in the Adaptive Recursive Partitioning, or using a bottom up pruning scheme which is optimal as in the Wedgelet Decorated Dyadic Partitioning. Both approaches have their merits and demerits as we will discuss in the next section.

We now proceed to describe our two algorithms in details. First we define some notation common to both algorithms. Given any region R of the entire image y of size m_R-by-n_R, let y^R describe the data in region R i.e. $y \in R$ with mean μ_R and variance σ_R^2. Let us define $t_1^R \equiv \sum_{y \in R} y$ and $t_2^R \equiv \sum_{y \in R} y^2$. We may carve R into H non-overlapping 'sibling' regions each represented as R_h for $h = 1, \cdots H$. We are now ready to describe our algorithms.

3.2 Adaptive Recursive Partitioning

The **Adaptive Recursive Partitioning** (ARP) algorithm recursively examines a subblock R (starting with the entire image), to determine whether it is best represented as homogeneous with a common mean and variance under model class 1, or else split at an arbitrary pixel location into either two homogeneous rectangle (vertically or horizontally) under model class 2, or four homogeneous rectangles (with one common vertex) under model class 3. If a decision to split is made, we must also determine what the best split point $\mathbf{k}$ is. Under model class 2 or class 3, let us denote the resulting rectangular regions as R_h ($h = 1, 2$ and $h = 1, 2, 3, 4$) respectively, each region having a different mean and/or variance from its siblings. Each of the resulting subblocks is then recursively examined in this way to decide how it is best described, and processing on a region only stops when we decide on model class 1 or we reach trivial pixel partitions [NF99, CK98].

The decision to either split a region or retain it as homogeneous is made by using the MDL criteria explained in chapter 2. In addition, the best split point $\mathbf{k}$ is also decided based on the MDL principle. The description length of any given model consists of the code length required to indicate a given model class, plus the number of bits required to code the model parameter and then the data given that model.

First we transmit sufficient statistics for the model parameters (sample mean and sample variance) of a region, and then transmit the data in that region based on the parametric model assumed. We initialize this algorithm by sending the sum and sum of squares of the entire image y. The progressive scheme then enables us take advantage of previously transmitted information, since if a region is split into H regions, we only need to transmit sufficient statistics for $H-1$ regions given that one set of statistics may be obtained from previously transmitted mean and variance of the 'parent' block. The ARP algorithm admits three possible model classes:

Model Class 1: Under model class 1, assuming sufficient statistics for the model parameters have already been transmitted, we only need to code and transmit the data as a homogeneous block. Given the uniform distribution obtained in section (2.3.1), the description length for this model class is simply:

$$L_1(R) = -\log p(y^R | t_1^R, t_2^R) \tag{3.1}$$

Processing of a subblock is stopped once it is determined to best be described as homogeneous.

Model Class 2: Under this model class we may split R into 2 sub-rectangles R_h, $(h = 1, 2)$ horizontally or vertically. We examine all possible pixel-split points and decide on the one that gives the minimum code length as per MDL. For this model we must describe the position of the split point to the receiver. Since there are $(m_R - 1)$ ways to horizontally split R, and $(n_R - 1)$ ways to vertically split R, we will need $\log(m_R - 1)$ and $\log(n_R - 1)$ bits to encode each horizontal and vertical split, respectively.

In addition, the sufficient statistics for the estimated parameters of one of the resulting regions, i.e., either $t_1^{R_1}, t_2^{R_1}$ or $t_1^{R_2}, t_2^{R_2}$ will be encoded to enable us describe the data in R_h under different models. Using a worst case analysis, noting that $t_1^R \geq t_1^{R_h}$ and $t_2^R \geq t_2^{R_h}$ (because of the positivity of image data), we require at most $\frac{1}{2}\log(n_R m_R)$ for each statistic, as described in section 2.2.1. Finally, we encode the data in the two regions under the different models according to Shannon's optimal code length [Sha48], using the conditional density obtained in Eqn. 2.4. As an example, if we consider splitting R horizontally, the resulting MDL description length would be given as:

$$\begin{aligned} L_2\{R(j)\} &= -\log p(y^{R_1}|t_1^{R_1(j)}, t_2^{R_1(j)}) - \log p(y^{R_2}|t_1^{R_2(j)}, t_2^{R_2(j)}) \\ &\quad + \log(m_R - 1) + \log(n_R m_R). \end{aligned} \tag{3.2}$$

for $j = 1, 2, \cdots, (m_R - 1)$.

Model Class 3: For this model class we may split R into 4 sub-rectangles having one common vertex. The description length is very similar to that of model class 2 except that in this case, there are $J = (m_R - 1)(n_R - 1)$ possible split positions and three sets of sufficient statistics to be transmitted. The description length under this model class is given as:

$$\begin{aligned} L_2\{R(j)\} &= -\log p(y^{R_1}|t_1^{R_1(j)}, t_2^{R_1(j)}) - \log p(y^{R_2}|t_1^{R_2(j)}, t_2^{R_2(j)}) + 3\log n_R m_R \\ &\quad - \log p(y^{R_3}|t_1^{R_3(j)}, t_2^{R_3(j)}) - \log p(y^{R_4}|t_1^{R_4(j)}, t_2^{R_4(j)}) + \log J. \end{aligned} \tag{3.3}$$

for $j = 1, 2, \cdots J$.

In summary, the ARP proceeds as follows: We start by assuming that the decoder knows the size of the image and that the total sum t_1^y and sum of squares t_2^y of pixel values in the entire image have been already transmitted. We then compute and compare code lengths for the three model classes and choose which ever model yields the minimum.

That is, if $L_1(R) < [\min\{L_2\{R(j)\}\}, \min\{L_3\{R(j)\}\}]$ we represent the image as one single piece with one mean and one variance estimated from t_1^y and t_2^y. Otherwise if $\min\{L_2\{R(j)\}\} < [L_1(R), \min\{L_3\{R(j)\}\}]$ there is one best partition of the image at $\mathbf{k}$, the point that attains the code length $\min\{L_2\{R(j)\}\}$, into 2 subblocks. Else, we split into 4 rectangles again at the point which achieves the codelength obtained from $\min\{L_3\{R(j)\}\} < [L_1(R), \min\{L_2\{R(j)\}\}]$.

If either $\min\{L_2\{R(j)\}\}$ or $\min\{L_3\{R(j)\}\}$ was selected, then each resulting rectangular region is examined recursively and split under model class 2 or 3 until either model class 1 (i.e. L_1) is selected or we get to trivial pixel partitions. Each iteration corresponds to a lower level in our multi-scale tree and each node represents the decision made among the three possible classes. All subblocks may be represented with the MLE of one or both of its parameters i.e., mean and variance of the data inside the subblock. A brief pseudo-code of the ARP algorithm is given in Fig. 3.2.

The ARP algorithm was originally developed for Poisson data by Nowak and Figueredo [NF99] and was used for the segmentation of SAR data [VNBmF00] based on intensity function only. It has been adapted here for the Gaussian model of unknown mean and unknown variance, to capture variations in both variance and intensity in regions of the data.

The ARP is a greedy algorithm because at each level we ignore the fact that each subblock may be further subdivided thus achieving an even shorter code length. Therefore at each stage an optimal decision is made however, the overall best possible segmentation may not be achieved. Figure 3.3 shows a simple tree-map illustration of partitioning produced using the ARP on a test image with regions differing in means and variances.

Adaptive Recursive Partitioning algorithm

1. Input: y of size $mn = m \times n$;
$t_1 = \sum_{i_1,i_2}^{mn} y(i_1,i_2)$, $t_2 = \sum_{i_1,i_2}^{mn} y(i_1,i_2)^2, r = \sqrt{t_2 - \frac{t_1^2}{mn}}$;

Homogeneous region
2. Calculate $L_1 = -\log(\phi(mn, r))$;

Horizontal splits
for $j = 1, \cdots, m$ {
$t_1(j) = \sum_{i_1,i_2}^{jn} y(i_1,i_2)$ and $t_2(j) = \sum_{i_1,i_2}^{jn} y(i_1,i_2)^2$;
obtain $r_{top}(j)$ and $r_{bottom}(j)$
3. Calculate $L_2(j) = -\log\phi(jn, mn, r_{top}(j), r_{bottom}(j))$
end }

Vertical splits
for $l = 1, \cdots, n$ {
obtain $t_1(l), t_2(l), r_{left}(l)$ and $r_{right}(l)$
4. Calculate $L_2(l) = -\log\phi(ml, mn, r_{left}(l), r_{right}(l))$
end }

Four-way splits
for $p = 1, \cdots, m$ {
for $q = 1, \cdots, n$ {
5. Calculate $L_3(p,q) = -\log(\phi(pq, mn, r_{top-left}(p,q), r_{top-right}(p,q),$
$r_{bottom-left}(p,q), r_{bottom-right}(p,q)))$
end }
end }

if $L_1 < [\min\{L_2(j)\}, \min\{L_3(j)\}]$; {
Output: $y = $mean$(y(i_1,i_2))$;
stop;

else if $[\min L_2(j), L_2(l)] < [L_1, \min\{L_3(p)\}]$;
k = find j or l for $\min L_2(j), L_2(l)$;
Input: $y_{top}(k), y_{bottom}(k)$ or $y_{left}(k), y_{right}(k)$;
goto line 1.

else k = find p, q for $\min L_3(p,q) < [L_1(R), \min\{L_2(j)\}, \min\{L_2(l)\}]$;
Input: $y_{top-left}(k), y_{top-right}(k), y_{bottom-left}(k), y_{bottom-right}(k)$;
goto line 1.
end }

Figure 3.2 Psuedo-code for the Adaptive Recursive Partitioning algorithm

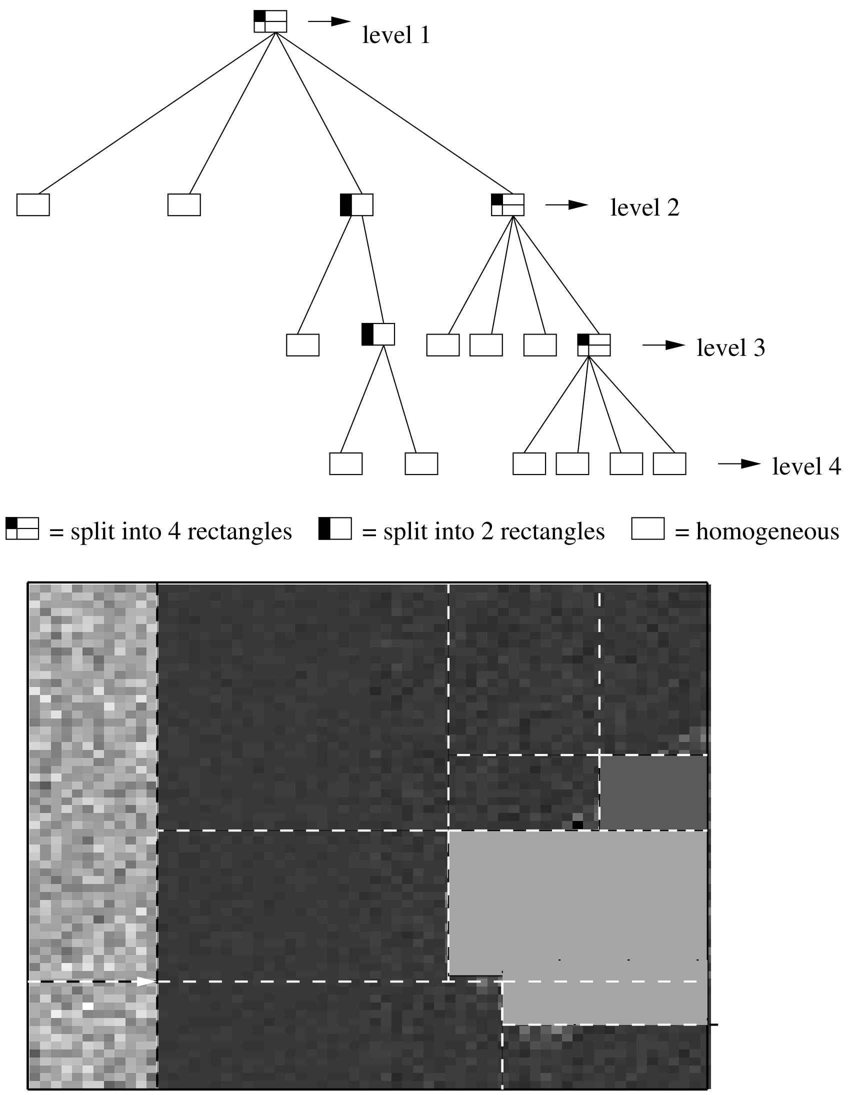


Figure 3.3 Segmentation tree map illustrating ARP algorithm used to segment the synthetic image (below).

3.3 Wedgelet Decorated Dyadic Partitioning

The goal of the **Wedgelet Decorated Dyadic Partitioning** (WDDP) algorithm is to implement an optimal parsing of the observed image. This is considered here since our proposed ARP algorithm is sub-optimal as stated in the previous section.

The WDDP developed here, is an extension of the Classification and Regression Tree (CART)-based wedgelet algorithm first proposed by Donoho [Don99]. The CART theorem [BFOS84, CRM99] asserts that given a function $\xi(T)$ which assigns numerical values to a quad tree T and its subtrees, and obeys a certain additivity property, the optimal subtree is obtained by a breadth-first bottom up pruning. This idea is used here: Our multi-scale analysis places on the observation, a dyadic tree grown from the bottom (i.e., starting from pixel values until we reach a representation for the entire image). Each node of this tree is decorated with a wedgelet. The notion of wedgelets is explained in the next section. Due to the prohibitive complexity required to search through all possible sub-trees, we restrict our analysis to dyadic observations and partitions.

3.3.1 Wedgelets

A 'wedge' is defined here on a dyadic $n_R \times n_R$ region R of an $n \times n$ image $y(i_1, i_2)$ for $0 \leq i_1, i_2 \leq n_R$, as a line segment from a pixel-vertex on one edge of R, to another pixel-vertex on another edge, splitting R into two polygons. A complete wedge dictionary for image y consists of wedges at all possible orientations, scales and locations of the image. To reduce the complexity of our algorithm, we do not search through this complete dictionary of all possible wedge-splits for each R, since this collection has a cardinality of $O(n^4)$ and would be computationally intensive to implement at all nodes. Instead, in forming our wedge dictionary, we only use vertices marked off at $\tau = 4$ equi-distant points on all edges of dyadic blocks of edge size greater than 4 pixel units. This ensures that at each scale, the cardinality of our wedge dictionary for any such block is an absolute constant $B = 80$ (i.e., $6\tau^2 - 4\tau$). For blocks of size less than 4, we consider all possible pixel vertex wedges so that for example, a 2×2 block has only $\tau = 2$ and $B = 16$ possible wedges. An illustration of our restricted wedge-dictionary on dyadic blocks at different scales is shown in the left diagram in Fig. 3.4.

Wedgelets are basis functions that enable us to represent wedge-splits at different locations and orientations. Let the indicator function of say, the left polygon resulting from a wedge-split on R, be denoted as $w_{b,R}^{l}$. Then the collection of wedgelet functions given as:

$$w_b(R) = \{1_R\} \cup \left\{w_{b,R}^{l}\right\} \qquad \text{for} \quad b = 1, \cdots B,$$

expresses all ways of wedge-splitting R into two polygons.

A *wedgelet decomposition* $W(R)$ of region R is defined as the vector of all inner products of the wedgelet function $w_b(R)$ with the dyadic block R, i.e., $\langle w_b(R), R\rangle$ for $b = 1, \cdots, B$. The cardinality of the wedgelet decomposition is also restricted to B and can therefore be rapidly computed in $O(n^2 \log n)$ time.

For the multi-scale WDDP algorithm, we perform the wedgelet decomposition at each node at all scales of the dyadic tree, so that each node may be decorated with a wedgelet. The diagram on the right of Fig. 3.4 is used to illustrate this concept.

Wedgelet decomposition adds so much more flexibility to our proposed partitioning scheme, enabling non-dyadic tessellations of our observation space. This also results in a non-blocky segmentation and is very useful since any arbitrary boundary may be approximated with a chain of wedgelets at all scales. This idea is illustrated in Figure 3.5, where the solid line indicates an arbitrary boundary, and the dotted lines are wedgelet approximations at different scales, locations and orientations. More on the wedgelet idea and a complete proof of the optimality of a wedgelet-decomposition representation for image data may be found in Donoho's original paper [Don99].

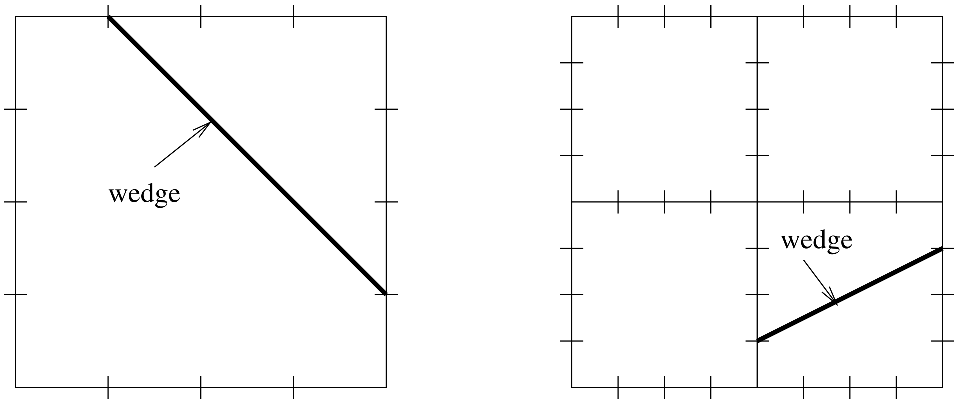


Figure 3.4 Illustrating the restricted dictionary of wedge-splits for dyadic blocks at different scales. Only four equi-distant vertices are allowed on each edge.

3.3.2 The WDDP Algorithm

The WDDP is an optimal recursive, **bottom-up** partitioning algorithm based on a *hereditary* tree structure. The terminal nodes of the resulting optimal subtree are non-overlapping dyadic partitions, or else wedgelet splits of a dyadic block into two polygons. The partitioning is determined by the MDL criterion based on our assumed model of piecewise constant mean and variance functions.

In this case unlike the ARP algorithm, since we start from the bottom of the tree with pixels of the image, we do not require any pre-transmitted parameter estimates (of total sum and sum of squares). The description length given by the MDL principle at each level satisfies the required additive property stated by the CART theorem. This implies that the description lengths can be optimized sequentially to achieve an optimal tree representation of our image data.

Given that $y(i_1, i_2)$ for $i_1, i_2 = 1, \cdots n$, is an n-by-n dyadic image, let $J = \log_2 n$ denote the tree depth and $j = 1, \cdots J$ indicate the scale such that the tree is rooted at

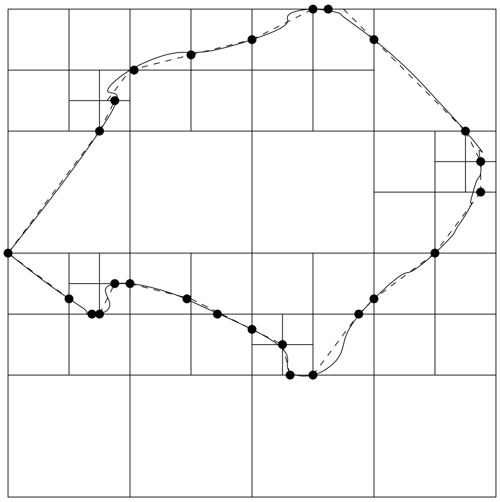


Figure 3.5 Illustrating a wedgelet-chain approximation (dotted chain-line) of an arbitrary boundary (solid line) at all scales and orientations.

the entire image y which is the coarsest scale $j = 1$. A standard quad-split on a dyadic region R produces only four dyadic 'children' we will denote as $R_c = \{R_{c1}, \cdots R_{c4}\}$, while a standard quad-merge on four children results in their parent. The additivity property requires that the description length $L(R)$ for R be a sum of the description length of its four children as:

$$L(R) = \sum_{c=1}^{4} L(R_c),$$

This enables parents inherit description lengths from their children, thereby reducing the computation required at each node.

The optimal tree is built according to the following recursive hereditary scheme: Starting from the bottom with standard quad-merges on individual pixels, we obtain 2×2 dyadic-block representation of $y(i_1, i_2)$. We perform quad-merges recursively, at each stage proceeding upwards by always inheriting the best possible cumulative description length for a block at that level. For a dyadic region R, the associated node on the tree is decorated with a description cost which is obtained as the best of the following three models.

Model Class 1: Under this model class we represent R as a homogeneous region with one mean and variance, for which we need to encode the sufficient statistics for their estimates. We then encode the data under the assumed parametric model, given that these statistics have been transmitted using Eqn. (2.4). The penalty term is the number of bits required to transmit t_1 and t_2 therefore the description length is:

$$L_1(R) = -\log p(y^R | t_1^R, t_2^R) + 2 \log n_R, \tag{3.4}$$

Model Class 2: Perform a wedge-split on R at B different orientations to obtain two polygons denoted as $R_1(b)$ and $R_2(b)$ for $b = 1, \cdots, B$. For this model class, we need to code the individual (2 sets of) sufficient statistics for the sample mean and sample variance of $R_1(b)$ and $R_2(b)$, to enable us describe the data in each of these regions under different models. Again, since $t_1^R \geq (t_1^{R_1(b)}, t_1^{R_2(b)})$ and $t_2^R \geq (t_2^{R_1(b)}, t_2^{R_2(b)})$, we require at most $\log n_R$ for each of these statistics.

In addition to encoding the sufficient statistics for the parameters, and the data in each of the resulting polygons under different models, we also need to describe the location of the splitting wedge. Since there are exactly B possible wedge orientations, we require $\log B$ bits to encode this information. Our description length is given as:

$$\begin{aligned} L_2\{R(b)\} &= -\log p(y^{R_1(b)} | t_1^{R_1(b)}, t_2^{R_1(b)}) + \log B + 4 \log n_R \\ &\quad -\log p(y^{R_2(b)} | t_1^{R_2(b)}, t_2^{R_2(b)}) \end{aligned} \tag{3.5}$$

for $b = 1, \cdots, B$. The 'best' wedgelet representation of R is the one that corresponds to the minimum code length in this class, given as: $L_2(R) = \min_b L_2\{R(b)\}$.

Model Class 3: For this class we perform a quad-split on R into its 4 children R_{ch}, where each child may be represented under a different model. The code length of each individual child is inherited from its 'descendants' at previous (lower) levels, as the cumulative best codelength representing the optimal segmentation of that region. That is, the code length that corresponds to the decision that was considered as the 'best' representation of each child (as per MDL) is inherited by the parent. The description length for a dyadic block R under this model is then simply the sum of the best description length for its four children, and is simply given as:

$$L_3(R) = \sum_{h=1}^{4} \min\{L_1(R_{ch}), L_2(R_{ch}), L_3(R_{ch})\} \tag{3.6}$$

At each node on the tree, the 'best' description length L is the minimum of the three possible code lengths from the model classes i.e. $L = \min(L_1, L_2, L_3)$. This is the code length inherited at the next level up the tree under model class 3.

Basically, for a dyadic block at each stage, the possible choices are either to represent the block as homogeneous, to split the block with a wedge, or else represent the block with the 'best' possible subtree inherited from its children. After the optimal tree is grown in this way, we prune back the tree by halting further processing on a subblock whenever the best representation turns out to either be model class 1 (code length L_1) or model class 2 (code length L_2). The leafs of this tree are thus either homogeneous dyadic blocks or polygons (from wedgelet splits). The image y is then progressively transmitted by sweeping down this optimal tree. We shown a psuedo-code for the WDDP algorithm in Fig. 3.6.

This algorithm was developed by Donoho for a constant variance image model using a complexity penalized sum of squares as a cost function, and performs well for segmentation based only on mean (intensity) property of an image. Our WDDP scheme extends Donoho's constant variance dyadic CART algorithm by enabling the detection of changes in variance in addition to intensity based segmentation. It is more flexible and has applications to a wider class of images as shown by our results.

Wedgelet Decorated Dyadic Partitioning algorithm

Initialize: image y of size $n^2 = n \times n$, $J = \log_2(n)$;

1. Input: pixels $y(i_1, i_2)$
   for $j = 1, \cdots, J-1$ {
   for $i_1, i_2 = 1 : 2^j : n$ {
   $t_1 = \sum y(i_1, i_2)$, $t_2 = \sum y(i_1, i_2)^2, r = \sqrt{t_2 - \frac{t_1^2}{2^j}}$;

Homogeneous region

2. Calculate $L_1(i_1, i_2, j) = -\log(\phi(j, r))$;
   end }

Wedgelet splits

for $b = 1, \cdots, B$ {
obtain $r_{left}(b)$ and $r_{right}(b)$

3. Calculate $L_2(b) = -\log \phi(j, r_{left}(b), r_{right}(b))$
   end }

$L_{inherit}(i_1, i_2) = \min[L_1, \min L_2(b)]$

Dyadic splits

4. Calculate $L_3(j+1) = \sum[L_{inherit}(i_1, i_2), L_{inherit}(i_1+1, i_2),$
   $L_{inherit}(i_1, i_2+1), L_{inherit}(i_1+1, i_2+1)]$;

end }

Pruning

for $j = J, \cdots, 1$ {
if $L_1 < [\min\{L_2(b)\}, \min\{L_3\}]$; {
Output: $y =$mean$(y(i_1, i_2))$;
stop;
else if $\min L_2(b) < [L_1, \min\{L_3\}]$;
k = find b for $\min L_2(b)$;
Output: $y =$ mean $(y_{left}), (y_{right})$;
stop;
else (dyadic merge)
Input: parent of $[y_{child1}, y_{child2}, y_{child3}, y_{child4}]$;
goto line 1.
end }
end }

Figure 3.6 Psuedo-code for the Wedgelet Decorated Dyadic Partitioning algorithm

A tree map illustrating the WDDP algorithm on a synthetic image is shown in Fig. 3.7.

3.4 Speed versus Accuracy

In both the ARP and WDDP schemes, we use a multi-scale approach to place a tree structure on the process of partitioning the image. The ARP however, works from the top of the image down, without consideration for the possible partitions that may yield better representations of the image. Because the ARP is not optimal and does not search through all possible subtree representations of the image, it is extremely fast, inspite of the fact that we require that at any stage, it searches through all possible pixel partitions of a given region.

On the other hand, the WDDP algorithm is aimed at achieving the optimal representation of the image. The added flexibility of searching through all possible wedge-splits at all nodes on this tree is a computational burden which we minimize by restricting our partitions to a fixed dyadic tree.

The optimality and accuracy achieved by the WDDP is traded off with the speed achieved by the ARP. In general though, both algorithms as fast enough to be implemented in real time, and our choice would depend on the application for which a given scheme is needed.

3.5 Comparison Results of Segmentation Algorithms

Figure 3.8(a) is a synthetic test image used to illustrate the flexibility and usefulness of our schemes. The region on the left labeled as A has a different intensity but the same variance from the adjacent region labeled as B. Region C on the right has

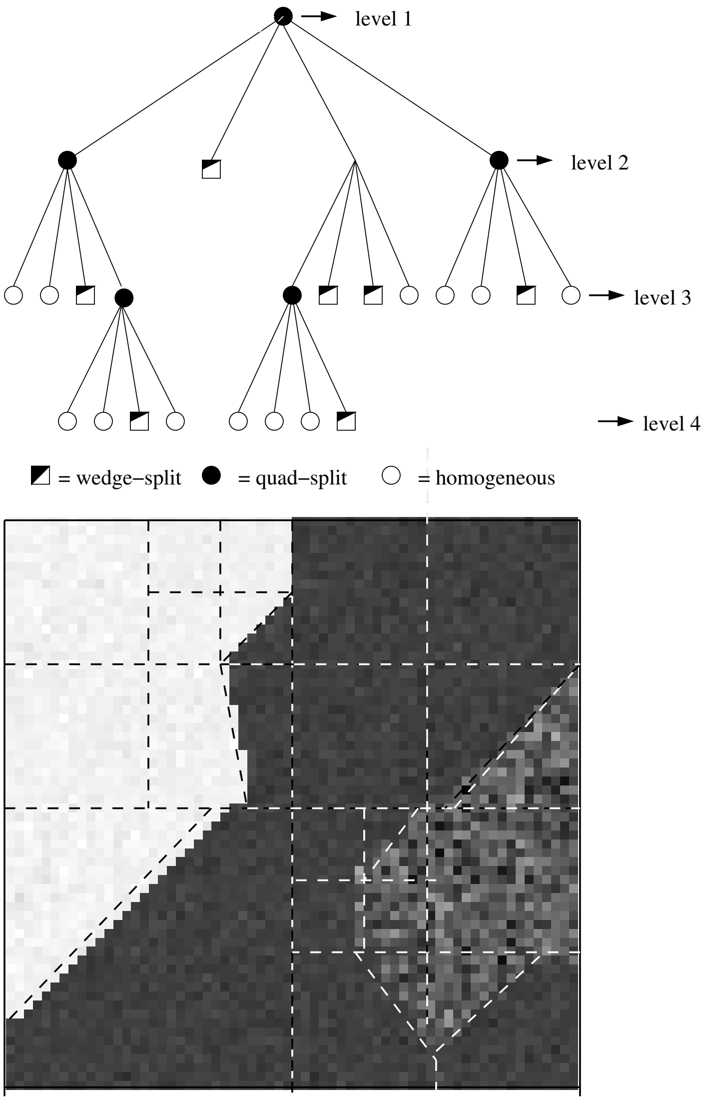


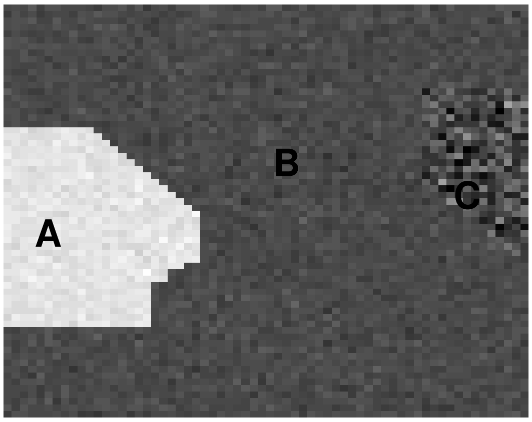
Figure 3.7 Tree map illustrating WDDP algorithm on a synthetic image with linear boundaries. The tree leafs are either dyadic blocks or polygons.

the same intensity but a different variance from B. The regions were made to have arbitrary (non-rectangular) shapes to illustrate the wedgelet decomposition idea.

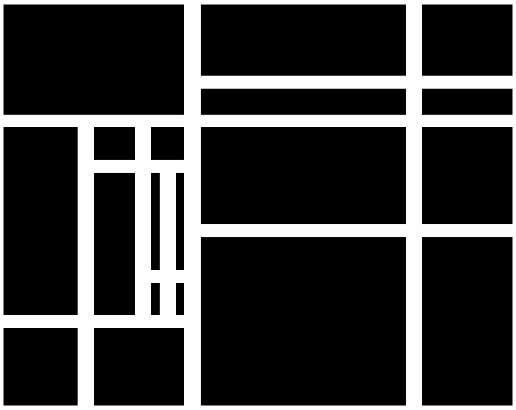
We have segmented this test image using our ARP and WDDP algorithms, both of which result in a piecewise constant mean and variance function for the image. Figures

3.8(b) and (c) show the resulting segmentation maps for the ARP and WDDP algorithms respectively, applied to this test image. We have chosen to use segmentation maps to illustrate where splitting occurs for each algorithm.

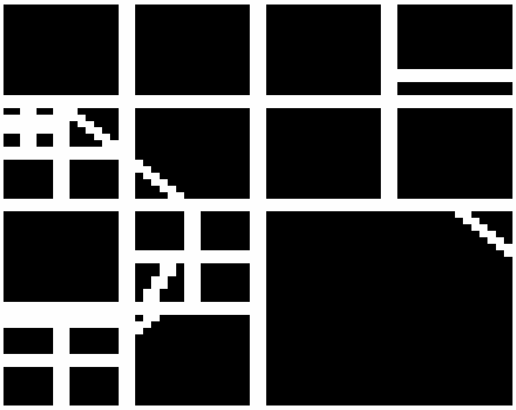
Both algorithms detect the change in texture on the right patch C. The ARP is not restricted to dyadic partitions and so we see rectangular tessellations of the observation, however, the WDDP does a better job approximating arbitrary shapes with wedges and is not blocky. Figure 3.8(d) is the segmentation map for the constant-variance wedgelet-decorated dyadic CART algorith developed in [Don99]. This scheme segments as well as the WDDP in region with varying intensity, but fails to detect vital changes in variance based on region C. This difference in the criterion for segmentation captures the main contribution of our segmentation algorithms. Our model enables one extra degree of freedom in segmentation and is therefore more flexible and better suited for the segmentation of a wider class of images requiring both intensity and textural characterization.



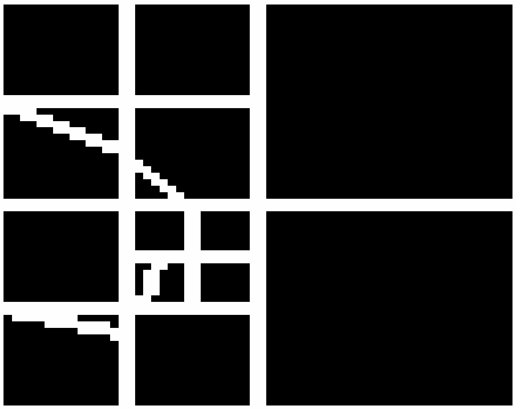
(a)



(b)



(c)



(d)

Figure 3.8 Results on synthetic test image. (a) The original image. Segmentation maps obtained using the following schemes: (b) The ARP: results in a blocky segmentation while segmenting all three regions (A,B,C). (c) The WDDP: better approximates the linear (arbitrary) boundaries while detecting all three regions. (d) The constant-variance dyadic CART performs as well as (b) in intensity based segmentation of region (B) but misses the texture-varying region (C).

Chapter 4

Applications and Results

In this chapter we discuss the performance of both our algorithms to applications for segmentation of SAR, LADAR and Infra-Red images of landscape and military scenes. The aim in these applications is to correctly identify and isolate object represented by statistically homogeneous regions. This is useful in compression of the typically huge data sets by identifying important 'targets' and throwing away what we consider as 'clutter'. Segmentation is also useful as a first step to automatic target recognition, which has several important applications in artificial intelligence, military operations, etc.

Our interest and investigation of segmentation in these different imagery is motivated by the fact that the definition of *clutter* and *target* is entirely dependent on the particular scenario and imaging technique [KHO93]. Clutter, man-made or natural can be just as interesting, just as complex and structured as the targets of interest. Automatic target detection problems are challenging because one cannot possibly hope to provide models for every type of target and clutterer the system may encounter. For each application, we investigate the important statistical features of homogeneous regions of the image, and find that our algorithms are suitable for segmenting the imagery presented in this chapter.

Our unsupervised segmentation approach is flexible and ensures that we isolate statistically varying regions, to accommodate whatever target the system may encounter. Supervised schemes require reliable training data which is often unavailable.

Also, the fact that our algorithms admit partitioning based on variance (as well as intensity) variation adds one degree of freedom over constant-variance model, to the segmentation of these images.

Since segmentation is often a first step to other image processing algorithms, we discuss a next step for target detection based on our segmentation schemes. Specifically, we introduce a robust CFAR detection scheme for LADAR and uncooled Infra-Red (IR) imagery and present results to illustrate performance.

4.1 Segmentation of Synthetic Aperture Radar

Synthetic Aperture Radar (SAR) is a coherent imaging technology which records both the amplitude and phase of measured radar back-scattered radiation. SAR data processing entails forming an image from these radar backscatter returns, followed by processing to analyse the formed image. SAR images provide broad area, long-range, all-weather,day-or-night remote sensing useful in applications such as mapping, search-and-rescue, mine detection, navigation and control, target recognition, etc.

SAR images suffers from a noise-like phenomenon known as *speckle*. Speckle results because each resolution cell contains many scatterers which return signals with randomly distributed phases, resulting in interference. This gives the images a grainy appearance and makes edge detection and segmentation of SAR images challenging. The problem of speckle may be overcome to some degree by averaging several independent observations of the same scene to produce multi-look images [DSL98]. SAR images can also be de-speckled using advanced expensive filtering techniques.

In SAR images, a large pixel value (bright pixel) represents a strong received signal. The strength of the received signal depends on several things, including the radar wavelength and relative size of the scatterers. Smooth, flat surfaces reflect very little radiation so that roads and still water will appear dark and smooth while fields or woodland often give a textured appearance. For this reason, both intensity and textural properties in SAR imagery provide important perceptual cues to identify candidate targets.

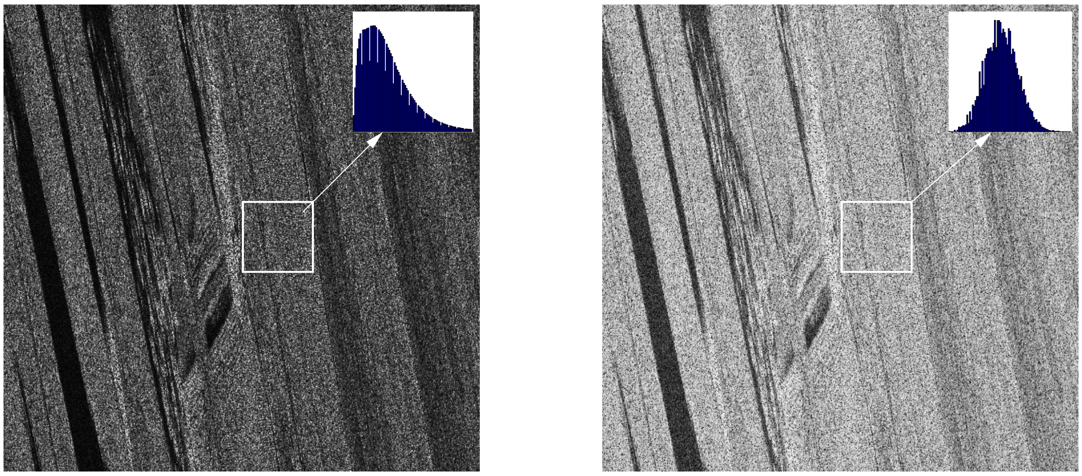
Segmentation plays a key role in SAR image analysis. Given a high-quality segmentation, important tasks such as adaptive noise filtering, classification and target detection can be greatly simplified. Also, since SAR images are typically huge datasets, we may achieve compression by 'throwing away' unwanted clutter regions.

Several supervised texture-only segmentation schemes have been developed for SAR data, requiring training prior to segmentation [FMLCC98, VCB00]. Other unsupervised schemes have achieved segmentation based only on intensity (mean) values without consideration for textural variations (i.e., constant variance models) [VNBmF00]. Our scheme is the first to propose an unsupervised segmentation of SAR imagery based on both mean (intensity) and variance (texture).

4.1.1 Statistical Model for SAR Imagery

In this section we examine the statistics of SAR data for the sake of completeness. We restate that since the MDL principle is not concerned with obtaining a code length based on the data generating distribution, any distribution defined on the data, not necessarily the true distribution can be used to encode the data [HY01, CA92]. This means that our MDL criterion based on a Gaussian assumption will do a good job on a generic image with regions of constant mean and/or variance irrespective of the true distribution of the data.

Suppose we have a randomly *speckled* SAR data set Z^* which is typically complex. Typically SAR data has been observed to have a skewed asymmetric distribution. The pixels y of SAR amplitude image $Y = |Z|$ are well modeled with a two parameter



(a) (b)

Figure 4.1 (a) Original SAR amplitude image with inbox showing the skewed (Rayleigh) distribution of the data in a homogeneous region. (b) SAR amplitude image after a simple square-root transform with inbox showing the approximately Gaussian distribution of the data in the corresponding homogeneous region.

Rayleigh distributed having a p.d.f.:

$$p_Y(y) = \frac{y - \mu_y}{\sigma_y^2} e^{-(y-\mu_y)^2/(2\sigma_y^2)}, \qquad y \geq \mu_y, \;\; \sigma_y^2 > 0, \tag{4.1}$$

where μ_y and σ_y^2 are the mean and variance of the distribution. We employ the estimating equations $E[Y] = \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ (sample mean) and $Var(Y) = s^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y})^2$ (sample variance) [Ser80] to obtain simple moment estimators for σ_y^2 and μ_y as $\hat{\sigma_y^2} = 2s^2/(4-\pi)$ and $\hat{\mu_y} = \overline{Y} - \hat{\sigma_y}\sqrt{\pi/2}$.

In [JKB94], it is shown that for a variable Y with a distribution whose mean and variance are functions of the same parameter s such as we have for the Rayleigh, a simple variance-stabilizing transform such as the logarithm or square-root of the original variable results in a variable which is approximately Gaussian. For example,

$X = \sqrt{Y}$ has a p.d.f approximately Gaussian and given as:

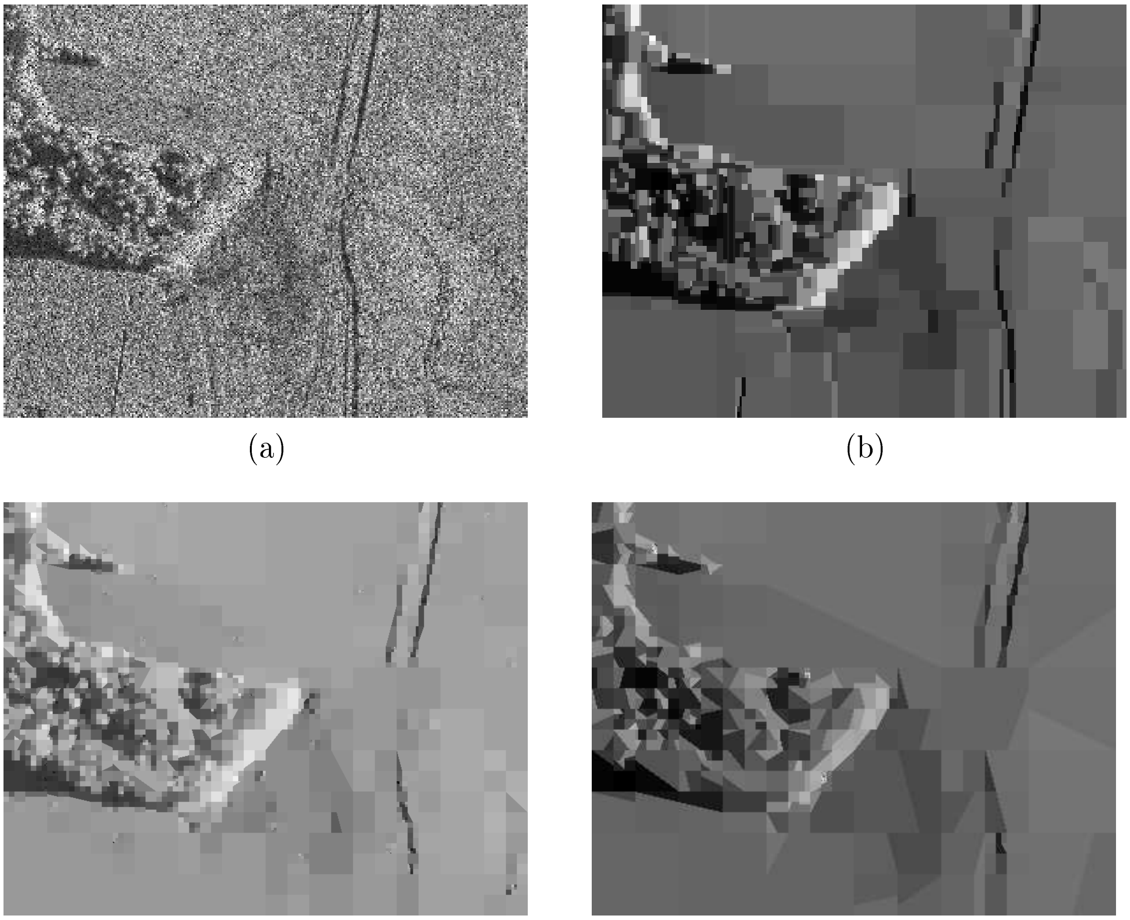
$$p_X(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-(x-\mu_x)^2/(2\sigma_x^2)}, \tag{4.2}$$

The effect of this simple transformation is illustrated in Fig. 4.1 where (a) shows an example of an original SAR amplitude image from the MSTAR database [(pu95]. Inset in the top right corner of the image is the histogram illustrating the skewed asymmetric distribution of a homogeneous region (outlined in the image). Fig. 4.1(b) shows the corresponding square-root transformed SAR image and the histogram of the homogeneous region. The Rayleigh distribution is approximately Gaussian after the this simple transformation. For our purposes we apply this simple transformation to our data and assume X is generated by a Gaussian process. We use this as the input to our algorithms to ensure that our image model is suited to the SAR data. .20.0

4.1.2 Illustrative Segmentation Results in SAR Imagery

We proceed to apply our two segmentation algorithms on real SAR data sets. Figure 4.2(a) shows an original 512×512 one-look MSTAR SAR amplitude data HB06170. Figure 4.2(b,c,d) show the ARP, WDDP and constant-variance CART algorithms applied to the SAR image. Visual inspection of these images indicate that the WDDP outperforms the other algorithms. The ARP gives a blocky effect while the constant variance CART algorithm produces an under segmented image.

Another illustration is shown in Fig. 4.3. The original (a) is a 512×512 multi-look SAR image of agricultural landscape in Flevoland, northern Netherlands [DSL98]. The image was acquired with 12.5cm of pixel spacing and has four-looks.Figures 4.3(b) and (c) are the resulting mean plots of the segmented image using the ARP and WDDP respectively. We see that both algorithms correctly smooth the data in



(a) (b) (c) (d)

Figure 4.2 MSTAR SAR amplitude data HB06170 (a) original image (b) ARP-segmented image (c) WDDP-segmented image (d) constant-variance WDDP segmentation

areas with rectangular regions. WDDP however does a better job in regions (in the right of the image) that have non-vertical and non-horizontal boundaries. We have tested our algorithm on several SAR images with similar results.

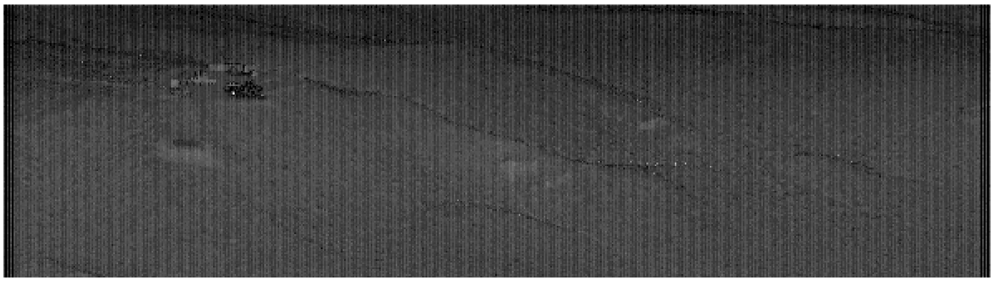
(a) (b) (c)

Figure 4.3 Multi-look SAR image of agricultural fields in Flevoland, Netherlands, (a) original image (b)mean plot of ARP segmented image (c)mean plot of WDDP segmented image

4.2 Target Detection in LADAR Imagery

Laser Radar (LADAR) is a promising imaging technology because it is able to collect high resolution 2-D and 3-D images by raster scanning a field of view, and is used as an effective and versatile remote sensing technique. The laser radar transmits a series of laser pulses, one for each pixel. Each pulse passes through the atmosphere and reflects off the first opaque objects it contacts. The reflected light returns to the sensor and then undergoes optical heterodyne detection followed by intermediate frequency (IF) filtering and peak detection.

A LADAR sensor typically collects two channels of data: intensity and range data. An illustrative example is shown in Figure 4.4. The intensity image is a measure of the reflected energy which takes lighting, color variations and textural properties etc. into account. The range image is formed by measuring the time delay between the peaks of the transmitted and the detected waveforms. The range data is a measure of physical distance which unlike intensity, is unaffected by illumination from other sources such as the weather, internal heat, texture etc [SRC95]. The basic difference



(a)

(b)

Figure 4.4 Original LADAR Images (a) intensity and (b) range

in the formation of the intensity and range images inspires the difference in the way we process them.

LADAR target signatures are much more robust as their range information is not dependent on ambient illumination artifacts, camouflage color schemes, or temperature variations. LADAR can provide enhanced weather penetration, and under the waters surface imaging. Shape, distance, velocity, and vibration can also be measured reliably through optically dense media such as fog, rain, sea water, etc, at short or long ranges.

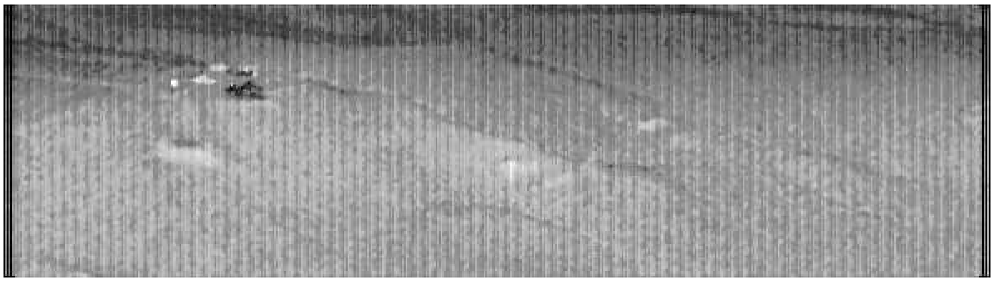
As a result, laser radars have numerous diagnostic research applications and military applications in flight control, obstacle avoidance, target detection, and fire control. This robust measurement capability also supports a variety of industrial and commercial applications.

LADAR images are degraded by the combined effects of laser speckle and local oscillator shot noise. The former is due to the rough-surfaced nature of encountered objects when compared to the laser wavelength, which causes constructive and destructive interference in the reflected signal. The latter is the fundamental noise involved in optical heterodyne detection. Speckle noise degrades LADAR images through anomalies which occur when a deep speckle fade combines with a strong noise peak resulting in measurements which are substantially different from the true value [KSW98].

An imaged scene may contain objects which appear 'smooth' or textured depending on certain factors including reflectance. The goal in segmenting an image of such a scene would then be to identify these objects with different statistical characteristics before target recognition can be performed. Several algorithms proposed in the literature perform segmentation based only on one of these features (intensity or texture) [YA00, JNG92] or use a multi-stage decoupled scheme [GL00, CNA90]. The ARP and WDDP algorithms developed in this work perform segmentation based on intensity and texture and are therefore well suited for segmentation of LADAR imagery.

4.2.1 Preprocessing the Input LADAR Images

Figures 4.4(a) and (b) are typical real LADAR intensity and range images collected from experimental ground-imaging sensors aboard an aircraft. The output has been scaled to 256 gray levels. In the intensity image, strong reflections are displayed as bright whites while negligible reflectance is in dark gray. For the range image, the lighter shades indicate greater distance from the sensor to the scene point. The images are 164×594 pixels containing some targets situated on a hill in a sloping



(a)

(b)

Figure 4.5 Median filtered LADAR images (a) intensity (b) log-range

terrain. The pixel spacing is assumed to be large enough so that the measurements are statistically independent.

Previous work [KCG+98] has shown that LADAR images have very large-error pixels which are uniformly distributed, with the remaining pixels having a Gaussian distributed error. The single-pixel statistical model of the observed $m \times n$ LADAR data (stacked into a vector), denoted as L_i for $i = 1, \cdots, mn$ given the true values $\overline{L_i}$, has the probability density function given as [KSW98]:

$$Pr_{l|\bar{l}}(L|\overline{L}) = \prod_{i}^{mn} \left[[1 - Pr(A)] \frac{\exp\left(-\frac{(L_i - \overline{L}_i)^2}{2(\delta L)^2}\right)}{\sqrt{2\pi(\delta L)^2}} + Pr(A)\frac{1}{\Delta L} \right] \text{ for } L_{min} \leq L_i \leq L_{max}$$

where $Pr(A)$ is the probability of anomaly, i.e., the probability that speckle and shot noise effects combine to give a measurement more than one resolution cell from the true value; ΔL is the width of the laser radar's uncertainty interval and δL is the local measurement accuracy (or precision).

The first term equals the probability that the measurement is not anomalous times a Gaussian p.d.f. with mean equal to the true range value. The second term is the probability that the pixel is anomalous times a uniform distribution over the entire uncertainly interval. Our initial preprocessing is aimed at suppressing this second term so that our Gaussian model may be used. We achieve this by a using a simple median filter. Even though more sophisticated methods exist to reduce these sensor dependent effects (e.g., maximum likelihood wavelet based method [KSW98]), this *ad hoc* median filter is sufficient for our application. Figure 4.5(a) shows the median filtered intensity image.

The log of the median filtered range image is shown in Fig. 4.5(b). A mesh plot of the log of the range image would reveals that range data is linearly varying (planar). Gradient filtering the logarithm of the range image yields an approximately piecewise constant image shown in Fig. 4.6.

4.2.2 Segmentation of LADAR Intensity Image

The ARP algorithm is used to segment the intensity image, resulting in a mean plot shown in figure 4.7(a). The algorithm does a good job of segmenting out homogeneous regions while staying robust to outliers. Figure 4.7(b) is a simple edge detector used on the gradient log-range image. This is appropriate since the gradient filter basically extracts boundaries where sharp slope changes occur. The segmented intensity plot is very useful on its own as a front end step fed into other processing steps for target detection and recognition. One such application is the robust CFAR detection scheme described below.

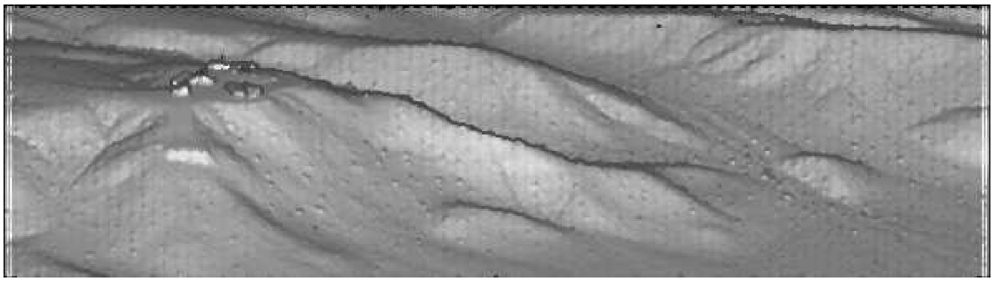


Figure 4.6 Gradient image of log range data

4.2.3 Robust CFAR Detection

The CFAR detector is often used as a prescreener to identify potential targets in LADAR images on the basis of pixel brightness [TROW99]. The conventional cell averaging (CA) CFAR detector examines a reference window W of samples surrounding the test pixel and returns bright pixels as suspect targets. Every pixel is sequentially examined as a test pixel. If we denote X_t as the amplitude of the test pixel, $\overline{X_s}$ and $\hat{\sigma}_s$ as the estimated mean and the estimated standard deviation of the surrounding samples in W, the following rule is used:

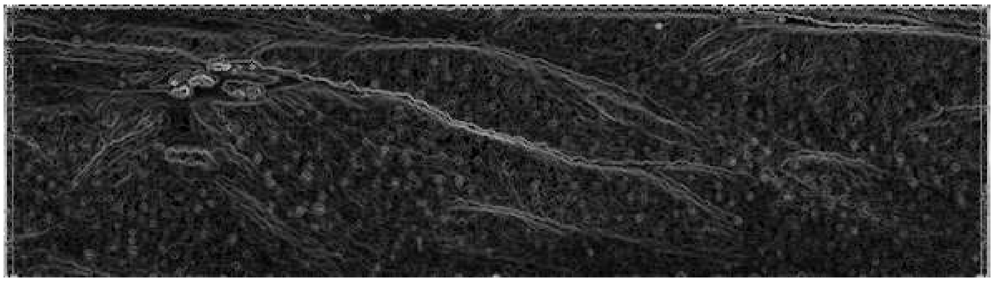
$$\frac{X_t - \overline{X_s}}{\hat{\sigma}_s} \geq \tau \Longrightarrow \text{declare target}$$

$$\frac{X_t - \overline{X_s}}{\hat{\sigma}_s} < \tau \Longrightarrow \text{declare clutter}$$

The threshold τ is a constant that determines the false alarm rate. This statistic is computed for each pixel and the resulting image is called a CFAR image. To detect the presence of a target in the test pixel, the binary test of hypothesis is carried out between the target absent hypothesis and the alternative target present hypothesis.

Some of the problems with this simple scheme is that it is not adaptive to the varying-size of regions since W is fixed. It also performs poorly if the background

(a)

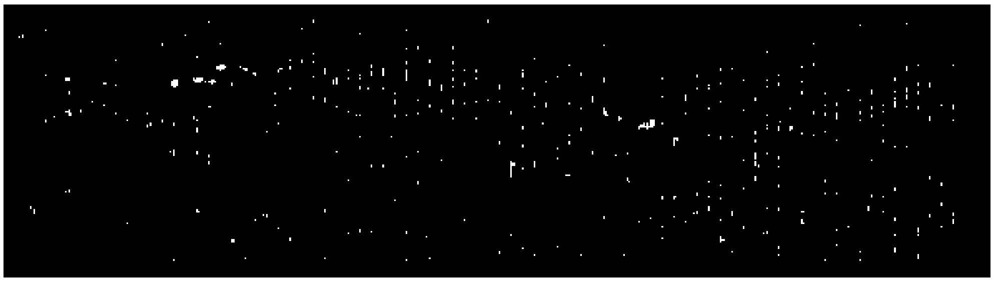


(b)

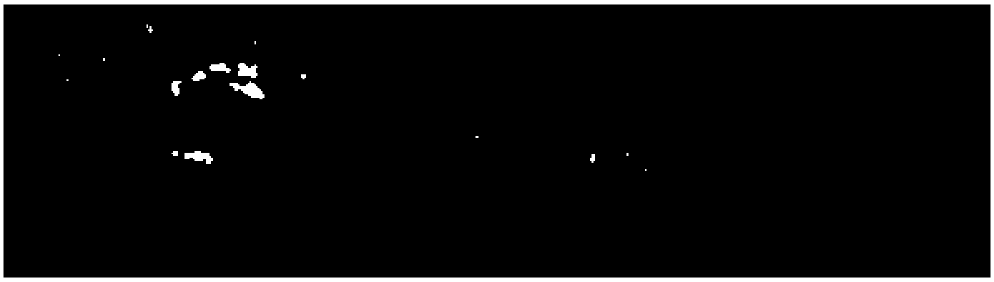
Figure 4.7 (a) Mean plot of the segmented LADAR intensity image. (b) Edge detector on gradient log-range image

window W is a non-homogeneous background, as in multiple targets and edge situations [JK98]. Also there are boundary problems associated with this scheme since pixels at the edge of the image do not have sufficient surrounding pixels in W to give reasonable CFAR statistics.

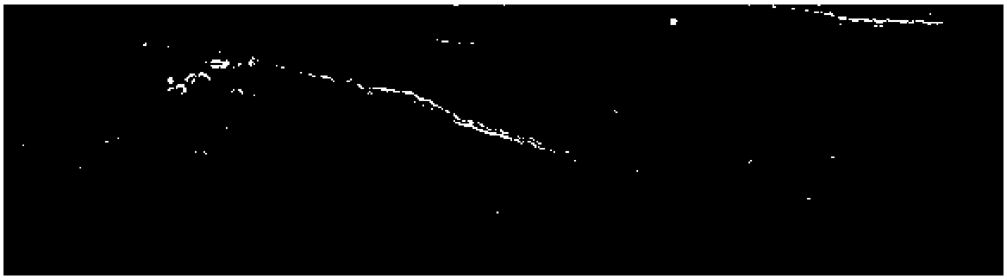
To resolve these issues, we implement a robust CFAR detector which uses estimates of the statistics obtained from our segmentation scheme on the intensity image. That is, for each test pixel, $\overline{X_s}$ and $\hat{\sigma}_s$ are the underlying mean and variance values obtained for the subblock into which X_t falls, in the segmented image. The resolution of our segmentation is adjusted such that targets of small sizes are 'immersed' in larger segments. Larger segments imply more reliable estimates. The



(a)



(b)



(c)

Figure 4.8 (a)Using the conventional CA-CFAR detector with a 9 x 9 window on the intensity image (b) Robust CFAR detector on the intensity image. (c) Thresholded edge-range image.

statistics obtained from these regions are also more robust because they adapt to the size of homogeneous regions and therefore give more accurate estimates of the mean and standard deviation. They do not suffer from clutter boundary edge problems within the image. This robust scheme is less sensitive to outliers that distract the conventional cell-averaging CFAR scheme. We also avoid the afore-mentioned image boundary problems since every pixel is part of a region.

For target detection on the range data, edge information obtained from the gradient log-range image is thresholded to detect strong edges where gradients change. This edge image gives reliable information and is compared with the results from the robust CFAR detection on the intensity image.

Discussions on our Robust CFAR detector

Our robust CFAR detector is appropraite for small sized targets and detects such targets because of the flexibiity obtained by weighting the penalty term to achieve more coarse segmentation. In the event that our achieved segments are smaller than target sizes in the image, our CFAR detection scheme would not be appropraite and will yield unacceptable results.

Another simple and robust CFAR detection scheme possible (but which we did not implement) using our segmented image involves using the segments as the test cells and surrounding regions or segments as the window from which the CFAR statistics are obtained. This is similar the the simple cell averaging CFAR scheme, except that in this case we do not work with pixel values, but instead with regions resulting from our segmentation.

4.2.4 Illustrative Results on LADAR Imagery

Figure 4.8(a) shows the output from the conventional (CA) CFAR algorithm using a sliding window of 9×9 pixels around the test pixel. We have used a threshold that reflects the 85th percentile of bright pixels in the CFAR image. Several false alarms can be seen and the target regions are not distinct. To improve detection using this algorithm would require a larger window size but with a greater penalty at the image boundary. Figure 4.8 (b) shows the result of using our robust CFAR algorithm on the intensity image. We are able to get rid of several of the false positives with more distinct 'activity' regions. Figure 4.8 (c) is the thresholded edge image of the gradient log-range image indicating boundaries where sharp gradient change occurs.

4.3 Target Detection in Uncooled InfraRed Images

This section briefly presents preliminary results on the segmentation and target detection of Infra-Red images. Infra-red imagers are detector and lens combinations used to take thermal representations of objects that emit infra-red energy (or heat). An illustrative example of an IR image of a single T72 tank is shown in Fig.(4.9). Thermal infra-red imagers translate thermal energy transmitted in the infrared wavelength (1 micron to 100 microns), into data that can be processed into a visible light spectrum display.

All objects above 0^o Kelvin emit thermal infrared energy so thermal imagers can passively see all objects regardless of ambient light. In other words thermal imagers let you "see" heat. This implies that, unlike optical images, visible light from a light source is not required for infra-red imaging. For this reason, IR images are capable of providing very detailed images of situations invisible to the naked eye, and performs well in a wider range of environmental conditions. Infra-Red imaging has found application in several areas including military (land and airborne), night or day surveillance, medical imaging, remote security monitoring, non-destructive testing, process control and predictive/preventive maintenance. Uncooled IR images are low cost equipment in which the charge couple device (CCD) chip has no provision for cooling and so cannot be used to take single long exposures due to the problems of thermal noise.

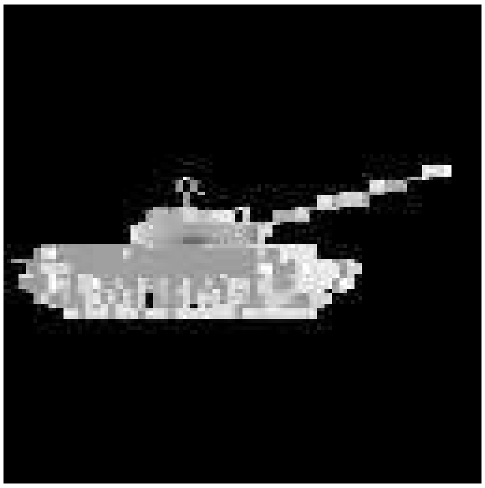
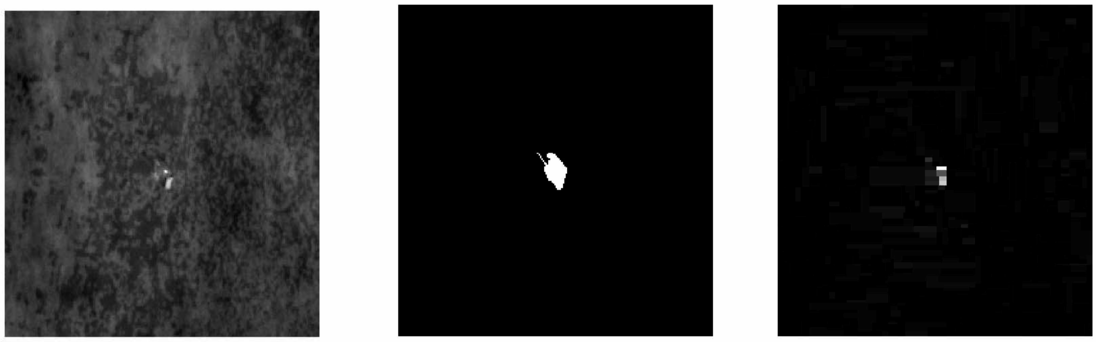


Figure 4.9 Illustrative example of an uncooled IR Image of a T72 tank. The tank is ‘hotter’ and therefore brighter than the background.

4.3.1 Illustrative Results on Uncooled IR Imagery

The Uncooled IR data sets used in this section are 220×220 real data sets containing only one target, which has been embedded in the pre-collected clutter background. The range of the IR sensor is 1000m, and the contrast is at 1.0 delta temperature. Truth images are provided to provide performance measures.

Fig 4.10(a) shows the original IR image and (b) is the truth image. Fig 4.10(c) is a variance representation from our adapted texture-based segmentation, derived in the appendix(A). This scheme correctly identifying the heavily occluded target. This result is very promising when compared to several segmentation algorithms in the literature, for low contrast uncooled IR images. This segmented image may then be simply thresholded to achieve a binary image, yielding regions of interest.



(a) (b) (c)

Figure 4.10 Illustrating target detection on uncooled IR data (a) The original image (b)The truth image. (c) Variance plot from our multi-scale variance-based segmentation

Acknowledgement

We would like to thank Alan Van Nevel, Carey Schwartz and Jack Riddle* of the Naval Air Warfare Center, China Lake, California and Raytheon Company* for providing several real LADAR and uncooled infra-red data sets used in this section of our work.

Chapter 5

Conclusions and Future Work

We have investigated a novel approach to segmenting signals using a coding theoretic approach of Minimum Description Length (MDL). Approximating piecewise homogenous regions of same intensity and/or textural properties, we assume our image is modeled as an i.i.d. Gaussian random field whose mean and variance functions may vary. Our image model is intended to capture variations in both intensity and textural properties in the image. We also present conditional densities which will enable easy adaptations of our schemes to intensity-only and texture only segmentations.

We develop two multi-scale schemes which differ in their approach, trading off speed versus accuracy. The Adaptive Recursive Partitioning is a progressive, greedy scheme that is not restricted to dyadic regions. The Wedgelet Decorated Dyadic Partitioning is an optimal fixed tree scheme restricted to dyadic partitions but allows for flexible wedge-splits at all tree nodes. Both schemes are computationally efficient and unsupervised. We discuss comparisons with an existing constant-variance scheme. We investigate applications of our segmentation algorithms to Synthetic Aperture Radar (SAR) imagery and discuss a robust Constant False Alarm Rate (CFAR) detector based on our segmentation results, for Automatic Target Detection in Laser Radar (LADAR) and Infra-Red imagery.

Appendix A

Conditional Density for Other Variants Under a Gaussian Model

A.0.2 Known Variance, Unknown Mean

For the case of known variance and unknown mean, we assume that $X^n = (x_1, \cdots, x_n)$ is a set of i.i.d. $\mathcal{N}(\mu, \sigma^2)$ samples modeled as:

$$y(i) = \mu(i) + \sigma.z(i) \qquad 0 \leq i \leq n,$$

with $z(i) \sim$ i.i.d. $\mathcal{N}(0,1)$. This is the constant noise model that enables intensity based segmentation. The variance σ^2 is constant across the image and known to the reciever. The sufficient statistic $\sum_i x_i \equiv t_1$ is transmitted progressively to the receiver. At each stage, we need to encode the data using the conditional distribution $p(X^n|t_1)$ according to the MDL criterion. To determine this distribution, we start by factoring $P(X^n|\mu)$ according to the standard exponential family form as:

$$p(X^n|\mu) \quad = \quad \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\frac{-\mu^2}{2\sigma^2}\right)^n \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n} x_i^2\right\} \exp\left(\frac{\mu}{\sigma^2}\sum_1 x_i\right)$$

where $\frac{\mu}{\sigma^2}$ is the natural parameter of the distribution. Using the theory of sufficient statistics and Neyman's factorization [BS94], we can write $p(X^n|\mu) = p(X^n|t_1)p(t_1|\mu)$ so that we have:

$$p(t_1|\mu) \propto \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\frac{-\mu^2}{2\sigma^2}\right)^n \exp\left(\frac{\mu}{\sigma^2}t_1\right)$$

$$\text{and} \quad p(X^n|t_1) \propto \exp\left\{-\frac{1}{2\sigma^2}t_1\right\} \equiv \phi(X^n)$$

Note that conditioned on t_1 the observations are independent of parameter μ. Normalising to obtain a valid density:

$$p(X^n|t_1) = \begin{cases} \frac{\phi(X^n)}{\int_{\mathcal{C}(\sqcup)} \phi(X^n) d(X^n)} & \Leftarrow \quad X^n \in \mathcal{C}(t_1) \\ 0 & \Leftarrow \quad X^n \notin \mathcal{C}(t_1). \end{cases} \tag{A.1}$$

where $\mathcal{C}(t_1) = X^n : (\sum_i^n x_i = t_1)$ is the constraint set which is a hyperplane in R^n. The integration required for normalization is

$$\int_{\mathcal{C}(t_1)} \phi(X^n) d(X^n) = \frac{2^{\frac{n}{2}-1}}{\sqrt{\frac{n}{2}}} (\pi\sigma^2)^{\frac{n-1}{2}} \exp\{-\frac{t_1^2}{2n\sigma^2}\} \equiv Z(t_1, n)$$

Therefore

$$p(X^n|t_1) = \begin{cases} \frac{1}{Z(t_1,n)} \left\{\exp\left(-\frac{1}{2\sigma^2} \sum x_i^2\right)\right\} & \Leftarrow \quad X^n \in \mathcal{C}(t_1) \\ 0 & \Leftarrow \quad X^n \notin \mathcal{C}(t_1). \end{cases} \tag{A.2}$$

This is the form of distribution we would use if we considered only intensity-based segmentation for our algorithms.

A.0.3 Known Mean, Unknown Variance

In this case of known variance σ^2 but unknown common mean μ, our signal is modeled as:

$$y(i) = \mu + \sigma(i).z(i) \qquad 0 \leq i \leq n$$

The sufficient statistic for σ^2 is $\sum_i^n (x_i - \mu)^2 \equiv t_2$ and is assumed already transmitted to the decoder. The natural parameter for the exponential family form is now $\xi = \frac{-1}{2\sigma^2}$ so that:

$$p(X^n|\xi) \quad = \quad \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^n x_i^2\right\} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\frac{-\mu^2}{2\sigma^2}\right)^n \exp\left(\frac{\mu}{\sigma^2} \sum_1 x_i\right) \tag{A.3}$$

where $\frac{\mu}{\sigma^2}$ is the natural parameter of the distribution, ψ and ϕ are some functions. Again, using the theory of sufficient statistics and Neyman's factorization [BS94], we may write:

$$p(X^n|\mu) = \left(\frac{-\xi}{\pi}\right)^{\frac{n}{2}} \phi(\xi) \exp\{\xi \sum_{i=1}^{n} (x_i - \mu)^2 \}$$

Note that the function $\phi(X^n)$ is equal to one, which means that $p(X^n|t_2)$ is uniform over the constraint set $\mathcal{C}(t_2) = X^n : \sum_i^n (x_i - \mu)^2 = t_2$ which is an $n-$dimensional hypersphere of radius $\sqrt{t_2}$ centered at $(\mu, \cdots \mu)$. The normalizing constant is then the surface area of this $n-$sphere which is given as $t_2^{\frac{2\pi}{\Gamma(\frac{n}{2})}}$. Therefore the joint p.d.f. of X^n conditioned on t_2 is given as:

$$p(X^n|t_1, t_2) = \begin{cases} \frac{\Gamma(\frac{n}{2}) t_2^{-\frac{n-1}{2}} \pi^{-\frac{n}{2}}}{2} & \Leftarrow \quad X^n \in \mathcal{C}(t_2) \\ 0 & \Leftarrow \quad X^n \notin \mathcal{C}(t_2). \end{cases}$$

This is the form of distribution we would use if we considered only variance-based segmentation for our algorithms.

Bibliography

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wordsworth & Brook, Pacific Grove, CA, 1984.

[BS94] J. Bernardo and A. Smith. *Bayesian Theory*. J. Wiley and Sons, Chichester, UK,, 1994.

[CA92] T. Cover and J. A.Thomas. *Elements of Information Theory*. J. Wiley and Sons, New York, 1992.

[CD00] Mary L. Comer and Edward J. Delp. The EM/MPM algorithm for segmentation of textured images: Analysis and further experimental results. *IEEE Trans. on Image Processing*, 9 No.10:1731–1744, Oct. 2000.

[CK98] Vittorio Castelli and Ioannis Kontoyiannis. An efficient recursive algorithm for classification using wavelets. *IBM Journal of Research and Development,*, 42(2):253–268, March. 1998.

[CMP00] Hung-Chih Chiang, Randolph L. Moses, and Lee C. Potter. Model-based classification of radar images. *IEEE Trans. on Info. Theory*, 46 No.5:1842–1854, Aug. 2000.

[CNA90] C. Chu, N. Nandhakumar, and J. K. Aggarwal. Image segmentation using laser radar data. *Pattern Recognition*, 23:569–581, 1990.

[CRM99] Isreal Cohen, Shalom Raz, and David Malah. Translation-invariant denoising using the Minimum Description Length criterion. *Signal Processing*, 75, No 3:201–223, June. 1999.

[Don99] David L. Donoho. Wedgelets: Nearly-minimax estimation of edges. *Annals of Statistics*, 27:859–897, 1999.

[DSL98] J. Dias, T. Silva, and J. Leitao. Adaptive restoration of speckled SAR images using a compound random markov field. *Proceedings of ICIP*, TA3.02:859–897, 1998.

[Edw72] A. W. F. Edwards. *Likelihood.* Cambridge University Press, New York, 1972.

[FLJ95] Mario A. T. Figueiredo, Jose M. N. Leitao, and Anil K. Jain. Unsupervised contour representation and estimation using B-splines and a Minimum Description Length criterion. *IEEE Trans. on Image Processing*, 9 No.6:1075–1087, 1995.

[FMLCC98] R. Fjrtoft, P. Marthon, A. Lops, and E. Cubero-Castan. An optimal multiedge detector for SAR image segmentation. *IEEE Trans. Geoscience and Remote Sensing (TGARS)*, 36(3):793–802, May 1998.

[GL00] Jill R. Goldschneider and Alan Q. Li. Variational segmentation by piecewise facet models with application to range imagery. *Proc. ICIP*, 2000.

[HY00] Mark H. Hansen and Bin Yu. Wavelet thresholding via MDL for natural images. *IEEE Trans. on Infomation Theory*, 46 No.5, Aug. 2000.

[HY01] Mark H. Hansen and Bin Yu. Model selection and the principle of Minimum DescriptionLength. *Journal of the American Statistical Association,*, to appear, 2001.

[JK98] Kyung-Tae Jung and Hyung-Myung Kim. Performance analysis of generalized order statistic cell averaging CFAR detector with noncoherent integration. *Institute of Electronics, Information and Communication Engineers Trans. Fundamentals*, E81-A:1201–1209, June. 1998.

[JKB94] Norman L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univarite Distributions Vol. 1.* J. Wiley and Sons, Chichester, UK, 1994.

[JNG92] Anil Jain, Tim Newman, and Micheal Goulish. Range-intensity histogram for segmenting LADAR images. *Pattern Recognition Letters*, 13:41–56, Jan. 1992.

[KCG+98] J. Kostakis, M. Cooper, T. Green, M. Miller, J. Sullivan, J. Shapiro, and D. Snyder. Multispectral sensor fusion for ground-based target orientation estimation: FLIR, LADAR, HRR. *Proc of SPIE, Automatic Target Recognition VIII*, 3371:500–507, April. 1998.

[KHO93] Daniel Kreithen, Shawn D. Halversen, and Gregory J. Owirka. Discriminating targets from clutter. *The Lincoln Laboratory Journal*, 6 No.1:25–50, 1993.

[KS99] H. Krim and I.C. Schick. Minmax description length for signal denoising and optimal representation. *IEEE Trans. on Information Theory*, 45(3), April. 1999.

[KSW98] A. Koksal, J. H. Shapiro, and W. M. Wells. Model-based object recognition using Laser Radar range imagery. *Proc SPIE, Automatic Target Recognition VIII*, 3718:256–266, 1998.

[Lan99] Aaron D. Lanterman. Minimum Description Length understanding of infrared scenes. *Proc. SPIE, Automatic Target Recognition VIII*, 3371, 1999.

[Lec89] Y. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.

[Li92] Mengniang Li. Minimum Description Length based 2D shape description. *Report from Computer Vision and Active Perception Lab.*, 3717, Oct. 1992.

[NF99] R. D. Nowak and M. A. T. Figueiredo. Unsupervised progressive parsing of poisson fields using Minimum Description Length criteria. *Proc. ICIP*, pages 26–29, 1999.

[(pu95] MSTAR (public). Clutter DBA7b02AA 2-CD set, Sept. 1995.

[Ris89] J. Rissanen. *Stochastic Complexity in Statistical inquiry.* World Scientific Publishers, Singapore, 1989.

[Ris96] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transaction on Information Theory*, 42:40–47, 1996.

[Sai94] N. Saito. Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion. *Wavelets in Geophysics*, Foufoula-Georgiou and Kumar (eds.), Academic Press, 1994.

[SBB$^+$00] D. Stewart, D. Blacknell, A. Blake, R. Cook, and C. Oliver. An optimal approach to SAR image segmentation and classification. *IEEE. Trans. on Radar, Sonar and Navigation*, 147, June. 2000.

[Ser80] Robert J. Serfling. *Approximation Theorems of Mathematical Statistics.* J. Wiley and Sons, Chichester, UK,, 1980.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 47:143–157, July. 1948.

[SRC95] Magnus Snorrason, Harald Ruda, and Alper Caglayan. Automatic Target Recognition in Laser Radar imagery. *Proc. ICASSP*, 1995.

[Teu93] Jan Teuber. *Digital Image Processing.* Prentice Hall, New York, 1993.

[TROW99] J. Tian and Jr R. O. Wells. A CFAR-enhanced image codec for SONAR ATR. *Proc. of SPIE tech. conf. on applications in signal and image Processing*, VII, July. 1999.

[VCB00] V. Venkatachalam, H. Choi, and R. Baraniuk. SAR image segmentation using wavelet-domain hidden markov tree models. *Proc. SPIE International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Algorithms for SAR Imagery,*, VII, April. 2000.

[VNBmF00] V. Venkatachalam, R. Nowak, R. Baraniuk, and m. Figueiredo. Unsupervised SAR image segmentation using recursive partitioning. *Proc. SPIE International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Algorithms for SAR Imagery VII,*, 2, April. 2000.

[WS88] Roland Wilson and Michael Spann. *Image Segmentation and Uncertainty.* J. Wiley and Sons, New York, 1988.

[YA00] Yongjian Yu and Scott T. Acton. Polarimetric SAR image segmentation using texture partitioning and statistical analysis. *Proc. ICIP*, Sept. 2000.