GEOMETRIC MODEL-BASED ESTIMATION FROM PROJECTIONS

by

Jerry Ladd Prince

B.S., University of Connecticut (1979)
S.M., Massachusetts Institute of Technology (1982)
E.E., Massachusetts Institute of Technology (1986)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1988

Signature of Author _____
Department of Electrical Engineering and Computer Science
December 27, 1987

Certified by _____
Alan S. Willsky
Thesis Supervisor

Accepted by _____
Arthur Smith
Chairman, Department Committee on Graduate Students

GEOMETRIC MODEL-BASED ESTIMATION FROM PROJECTIONS

by

Jerry Ladd Prince

Submitted to the Department of Electrical Engineering and Computer Science on January 2, 1988 in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

This thesis addresses the problem of image reconstruction from noisy and limited- or sparse-angle projections. An algorithm is presented for the estimation of the maximum *a posteriori* (MAP) estimate of the full sinogram (which is an image of the 2-D Radon transform of the object) from the available data. It is implemented using a primal-dual constrained optimization procedure, which solves a partial differential equation in the primal phase using an efficient local relaxation algorithm, followed by a simple Lagrange multiplier update in the dual phase. The sinogram prior probability is given by a Markov random field (MRF) which includes information about the mass, center of mass, and convex hull of the object, and about the smoothness, fundamental constraints, and periodicity of the 2-D Radon transform. The geometric information reflected in the MRF formulation is estimated hierarchically, in part by new set reconstruction algorithms developed herein. These algorithms in turn are based on probabilistic estimation formulations which incorporate prior information about the size, position, and shape of the object. In particular, knowledge of the eccentricity, orientation, and boundary curvature may be used. This thesis contributes to the state of knowledge in the field of computed tomography, particularly in those disciplines in which noise and limited-data is a problem, and in the field of computational geometry, where our probabilistic formulations provide new and interesting insights on the problem of convex set reconstruction from support line measurements.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering

Acknowledgements

There are many people who have contributed to the success of my thesis. In particular, I would like to thank Alan Willsky, my thesis advisor, who saw some potential in me at the outset, arranged for my financial support, and continued to encourage my research through steady and insightful dialogue. Thanks to Alan, I have accomplished the goals that I set for myself at the beginning of my Ph.D. program. I would also like to thank Profs. Dimitri Bertsekas, Bernard Levy, George Verghese, and Berthold Horn for their comments and assistance at various stages in the research. I give special thanks to Prof. Sanjoy Mitter for helping to provide the facilities in which my research was conducted — the Laboratory for Information and Decision Systems — and for his enthusiasm for and support of my research activities throughout.

I would also like to thank my associates in the laboratory. First, there are the many graduate students with whom I have shared an office, and without whom my education would not have been as pleasant: Peter Doerschuk (the laugh), Carey Bunks (DUBL), Richard Lamb (hardware), Robin Rohlicek (PC), Ahmed Tewfik (the answer), Ramine Nikoukhah (the question), Cuneyt Ozveren (coffee), and Ken Chou (software bugs). The staff at LIDS has always been very helpful and kind to me. In particular, I would like to thank Kathleen O'Sullivan, Betty Lou McClanahan, Bob Bruen, Peaco Todd, Bill Hanway, and Arthur Giordani for all their assistance over the past few years. And thanks to Prof. Pierre Humblet, who took charge of the computing facility to make it really useful for me and the rest of LIDS.

I wish to acknowledge the various funding agencies that have supported me and my research. This research has been supported by the National Science Foundation under grants ECS-83-12921 and ECS-87-00903, and the U.S. Army Research Office under grants DAAG29-84-K-0005 and DAAL03-86-K-0171. Also, I have been fortunate to have been the recipient of two fellowships, one due to the Schlumberger-Doll Corporation and the other due to the U.S. Army Research Office.

Finally, I would like to give my deepest thanks to my wife Carol, and to her family and mine. It was through their encouragement that I was able to embark on this adventure, and through ther love and support that I was able to complete it.

Thank you all.

to Carol and my parents.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

This thesis addresses the problem of *reconstruction from projections*, the theoretical and practical aspects of which have received much attention over the past two decades. Among the applications that use the currently available reconstruction techniques are medicine, optics, material science, astronomy, geophysics, and NMR. The most widely known application of this theory is the problem of medical transmission X-ray tomography. In this discipline, "pencil beam" X-rays are fired from many angles through a single cross section of the body, effectively measuring *line integrals* of the 2-D X-ray density function corresponding to the various tissues in the cross section. A collection of line integrals obtained over lines with the same angle but different lateral positions forms a 1-D function called a *projection.* Given a set of projections taken from many different angles, an image of the density function may be reconstructed and used in diagnosis. For many medical conditions this tomographic approach to imaging of the body leads to greatly improved imagery over conventional (chest-type) X-ray images and has proven to be of great benefit in medical diagnosis.

In this thesis we are primarily concerned with the *algorithms* used to reconstruct the images, and in particular, we address two issues related to inadequacies in the data acquisition: 1) noisy data and 2) limited- and sparse-angle data. A measurement of a line integral may be noisy in the medical problem described above, for

example, if only low energy X-rays are used. The data acquisition would be restricted to a limited angular range if there were an obstruction, for example, and might be only sparsely sampled if there was a requirement to obtain the data extremely rapidly. It is in these situations that the usual algorithms fail and alternate methods must be used. The methods we use incorporate prior knowledge about the shape and structure of objects in the cross section, and utilize a hierarchical reconstruction method that allows a controlled utilization of the available prior information and of the results from previous stages. The following prior knowledge is used in the reconstruction algorithm:

1. The values of line integrals taken over lines close in either lateral displacement (with the same angle) or angle (with the same lateral displacement) tend to be similar in value.

2. The Radon transform (see Section 1.2) of any cross section obeys certain fundamental mathematical constraints, the so-called *consistency conditions.*

3. The convex support of the cross-section density function uniquely specifies the support of the Radon transform.

The approach we have developed is a *projection-space method* since the prior knowledge and primary processing takes place in Radon space rather than in object space. The difficulty with this approach is that knowledge about the object is not conveniently incorporated in Radon space, but the advantages are that optimum probabilistic-based algorithms are more easily stated (and solved) in Radon space since the noise is white in this domain. We have chosen to model the Radon transform — the picture of which is called a *sinogram* — as a Markov random field (MRF) which incorporates the three properties listed above. Then we solve a large-scale optimization problem to give the MAP estimate of the sinogram, from which convolution back-projection (CBP) is used to reconstruct an image of the object. The hierarchical aspect of the algorithm involves primarily the estimation of quantities related to the constraints mentioned in item 2 above, and the estimation of the convex support of the object, mentioned in item 3.

It is important to point out that in conventional medical X-ray tomography a patient will receive a relatively high dose of radiation over a full range of angles in order to produce clear and distinct pictures for visual diagnosis. Therefore, the techniques we develop in this thesis are not (necessarily) intended to benefit this discipline. However, there are a few applications in medicine and in other fields where the data is, in fact, noisy or is available over limited or sparse angles. Some of these applications are:

- Low-Dose X-ray Tomography Screening.
- Emission CT.
- Real-time CT.
- Ultrasound Tomography.
- Cold Core Imaging.

This chapter is organized as follows. In the following three sections we present background related to the Radon transform and to reconstruction methods, both for high signal-to-noise ratio (SNR), full-view cases and for low-SNR, limited-view cases. In Section 1.5 we discuss the specific contributions of this thesis and Section 1.6 gives a summary of the organization of the thesis document.

1.2 The Radon Transform

The modern mathematical literature related to reconstruction from projections began with an historic paper by Johann Radon in 1917 [71], in which it was shown that a function defined on the plane is completely specified by knowledge of the integral of the function over each line in the plane. Radon also showed that a function defined in space (3-D) is completely determined by knowledge of the integral of the function over each plane in space. Fritz John [41] generalized these results to higher dimensions in 1934, and today we call the function determined by integrating

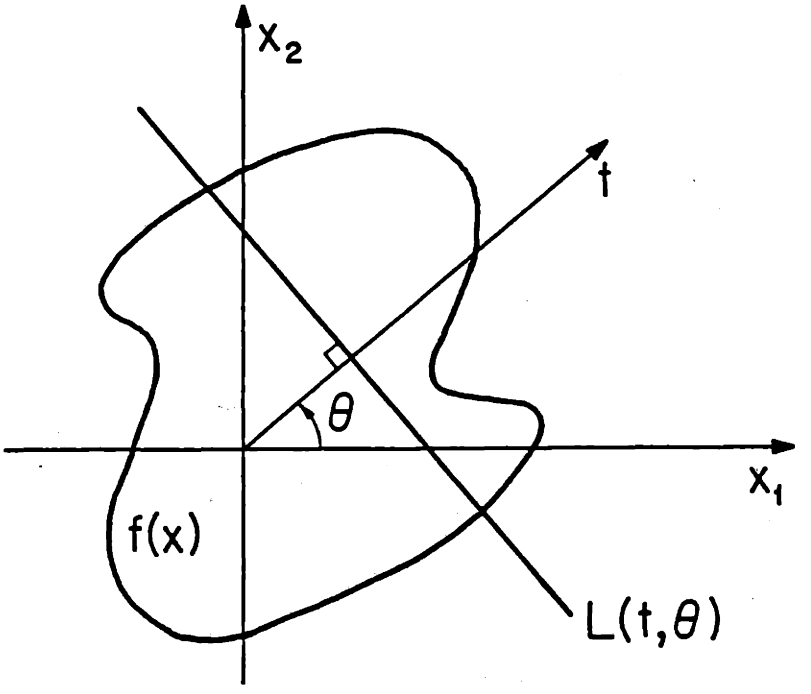


Figure 1.1: The geometry of the 2-D Radon transform.

$f : \mathbb{R}^n \rightarrow \mathbb{R}^1$ over all hyperplanes of dimension $n-1$, the Radon transform of f.[1] This thesis is exclusively concerned with the functions defined on the plane, and hence, integrals over lines; we now give precise mathematical definitions for the 2-D case.

Consider a single valued function $f(x)$ defined on the plane as depicted in Fig. 1.1. We denote the integral of f along the line $L(t,\theta)$ by $g(t,\theta)$. The function g for all values of t and θ is called the *2-D Radon transform* of f. For a single value

[1] In this thesis, the symbol $\mathbb{R}^n$ denotes the real n-dimensional Euclidean vector space. Also, the symbol S^n denotes the set of all $(n+1)$-dimensional vectors with unit length — i.e., S^n is a generalized unit sphere, the unit n-sphere.

of θ, g is a function of t only, and is called the *projection* of f at angle θ. The Radon transform of $f(x)$ may be written as

$$g(t,\theta) = \int_{x\in\mathbb{R}^2} f(x)\delta(t - x\cdot\omega)\,dx \tag{1.1}$$

where $\omega = [\cos\theta \;\; \sin\theta]^{\mathrm{T}}$, and $\delta(\cdot)$ is the Dirac delta function. It turns out that not just any function $g(t,\theta)$ is a valid Radon transform of some function $f(x)$. There are inherent mathematical conditions that constrain $g(t,\theta)$ to lie in a particular functional subspace defined on the cylinder $\mathbb{R}^1 \times S^1$. These conditions, referred to as the *consistency conditions*, are important for some reconstruction algorithms and will be discussed in greater detail in later chapters.

The fact that (1.1) is invertible (for a wide class of functions) is well known. Deans [20] describes many of the known exact inversion formulas. Except under certain (usually impractical) circumstances, however, it is not possible to determine f exactly given the value of g for only a *finite* number of lines $L(t,\theta)$. It has been the primary concern of engineers and physicists in this field, over the last 20 years or so, to study approximate inversion algorithms given such a finite measurement set. The performance of any particular algorithm depends on the nature of the measurements – their number and arrangement, and their noise properties – and often on the nature of the object itself.[2] In this thesis, we are primarily concerned with the case of low signal to noise ratio (SNR) and limited- or sparse-angle measurement configurations. Also, we consider only *parallel-ray* projections, not fan-beam projections, for example. We now briefly discuss some of the predominant reconstruction methods.

1.3 Reconstruction Methods

1.3.1 Full-View, High SNR Methods

By far, the most generally effective approximate inversion algorithms are the so-called *transform methods* (see for example [33,49]), among which are the well-known

[2] Throughout this thesis, we shall often refer to the support of the function $f(x)$ or the function itself as the *object*.

Fourier and convolution backprojection (CBP) methods. These methods are simply discrete approximations of the exact inversion formulas. Implicit to these methods, however, are the requirements of high SNR and full angular coverage. When either or both of these conditions fail to be true, the reconstructions are poor.

Most other full-view reconstruction algorithms can be described as finite-series expansion methods. In these methods, the object is approximated by a finite sum of weighted basis functions $\psi_i(x)$ as

$$f(x) \cong \sum_{j=1}^{N} f_j \psi_j(x) \ . \tag{1.2}$$

One then seeks an estimate of the vector $f = [f_1 \ldots f_N]^T$ — a finite number of parameters. Furthermore, the measurements, rather than being simply sampled versions of (1.1), may be more generally described by the integration of $f(x)$ over measurement basis functions $\phi_i(x)$, as

$$g_i = \int_{x \in \mathbb{R}^2} f(x) \phi_i(x) \, dx \ .$$

Forming the measurement vector $g = [g_1 \ldots g_M]^T$, we see that the measurement process is described by a simple linear transformation of f as

$$g = Hf + e \ .$$

with an error term e, because of the finite sum describing $f(x)$. The matrix H, called the projection matrix, has elements given by

$$H_{ij} = \int_{x \in \mathbb{R}^2} \phi_i(x) \psi_j(x) \, dx \ .$$

If M measurements are taken then H is an $M \times N$ matrix — H may be rectangular or square.

Once the choice of basis functions has been made, the reconstruction problem is nothing more than solving the linear equation $g = Hf + e$. But the solution may be very difficult because of the large dimensionality of the problem and the lack of sparsity in H. Furthermore, since H may be rectangular, the equation may be inconsistent, in which case there may be no exact solution, or more commonly,

it may be underdetermined, in which case there are infinitely many solutions from which one must be chosen. Many of the proposed algorithms and their criteria for optimality are described in [33] and more recently in [15]. It should be noted that the most popular algorithm among these is the so-called algebraic reconstruction technique (ART), also known as Kacmarz' algorithm.

Of primary interest to this thesis is the fact that the above description of the object and the measurements represent modeling choices. In the literature, several choices of the basis functions ϕ_i and ψ_j have been made. The choice of measurement basis functions are limited, naturally, because they must reflect the geometry of the imaging apparatus. On the other hand, the object basis functions, for which one has much more freedom of choice, are most often chosen to be shifted 2-D gate functions representing the pixels of the picture to be reconstructed [33]. Other choices have been considered, however, including polar pixels [14], natural pixels [13,12], and more recently families of locally overlapping functions [31].

1.3.2 Noisy Measurements

When $f(x)$ is assumed to exactly representable by the finite series expression in (1.2), then the noisy observation vector may be written as

$$y = Hf + n$$

where n is a noise vector with a known probability distribution. Once in this form, the optimum solution may be sought using maximum likelihood if no prior information about the object is assumed [73], or from a Bayesian approach [34,31, 27], or from a linear minimum variance approach [95,97].

Again, the importance of the above methods to this thesis is that probabilistic modeling has been included in the overall model. In particular, these researchers have made choices concerning the prior probability of the object. Furthermore, this prior is expressed on the vector f, which is related to the object through a choice of basis functions.

We now make several comments regarding modeling. First, Wood [95] pointed out that the distribution of linear attenuation (this is $f(x)$, also called the density)

in a typical medical scan is not well-modeled by a Gaussian density. Therefore, the object and the noise cannot be jointly Gaussian — and the linear minimum variance estimate it not the true Bayesian estimate. Second, Buonocore [12] pointed out that the choice of object basis functions can have a marked effect on the accuracy to which the true object covariance function can be approximated by a covariance matrix on f.

Finally, we note that there has been very little research on the general continuous Bayesian estimation problem in which one seeks to estimate $f(x)$ given a prior on f and noisy observations $y(t,\theta)$ over the entire domain of the Radon transform. Instead, most researchers begin by expanding $f(x)$ in a finite series as we have described, develop a prior on the vector of coefficients, and use a finite observation equation. Apparently, a major problem in obtaining results in this area has been that the stochastic characterization of the observations $y(t,\theta)$ is not trivial. Tasto [85,84] sought to characterize $y(t,\theta)$ when $f(x)$ is modeled as a stationary Gaussian random field windowed by the unit disk — but this did not lead to analytically tractable results. More recently, Jain and Ansari [40] have obtained some results on the second order characterization of $y(t,\theta)$ assuming the object is a stationary random field of infinite extent. Under this assumption they obtain a Wiener filter that obtains the minimum variance estimate of $f(x)$ from $y(t,\theta)$. One interesting auxiliary result that they prove is that $y(t,\theta)$ is white in θ for stationary random fields (and their filter is based on this result). But this result can be misleading. As we shall see below, data known over a limited angular view can be used quite effectively to estimate the missing angles. What this tells us is that choosing a stationary random field to model a function that has finite support is not a good choice.

1.3.3 Limited-Angle Tomography

There are many applications where the geometry does not allow complete data to be acquired. Most often the data is lacking in angular samples; the projections are

measured at either sparsely sampled angles or over a small angular range.[3] Either case leads to poor performance of the usual transform method inversion algorithms. We now briefly discuss algorithms which have been especially developed for this situation.

Several investigators have developed what might be called "modified transform methods" to account for the missing angles. Louis [53] uses the Radon transform consistency relations to estimate the missing projections; ordinary CBP is then used to reconstruct the object estimate. We shall have more to say about this method in later chapters. Davison and Grunbaum [19] designed optimum angle-dependent filters to be used in place of the usual so-called rho filter normally used in CBP. Inoye [39] used analytic continuation in the Fourier plane, followed by an inverse Fourier transform — this method proved to be very sensitive to noise. More recently, Reeds and Shepp [72] have developed a method called "squashing", which accomplishes limited-angle reconstruction using a scale compression, followed by a modified CBP, followed by a scale expansion. The primary advantage of squashing is computational efficiency. These methods are effective in relieving the major artifacts caused by the lack of angular measurements. They do not, however, provide a general framework in which one may introduce more prior knowledge concerning the probabilistic or geometric nature of the objects.

Methods based on the finite-series expansion of $f(x)$ using the pixel basis functions account for most other limited-angle reconstruction methods. It is clear that under this model, the missing angles amount to making fewer measurements, thus reducing the size of the projections matrix H. In early efforts, no attempts were made to estimate the missing projections. Instead, the problem was solved in almost identical fashion to the full-angle problem, using ART or other iterative techniques [63]. The major problem with this approach was the production of streaking artifacts in the images (see [15]). Wood, Macovski, and Morf [96] showed that with the addition of an object prior covariance, the linear minimum variance estimate gave superior reconstructions. More recently, Buonocore [12] chose an unorthodox

[3]However, there are cases where data is missing in t. Lewitt and Bates [50,51,52] have studied cases of hollow and truncated projections — where data is missing in central and outer parts of t, respectively.

natural pixel image basis and derived a fast linear minimum variance estimator of the missing projections assuming a white object covariance function (and a circular object support region).

The absence of observed data over part of the projection space (or equivalently, part of Fourier space) has led to a class of algorithms that may be called *iteration between spaces* [83,19]. This method has also been generalized to apply to many types of known prior constraints by the method called *projection onto convex sets* (POCS) [98,79]. In general terms, these methods work as follows. The algorithm starts with the measurements over the known angles and places zeros elsewhere. It then iterates between the measurement space (or Fourier space) and the object space, imposing known constraints (known data, positivity, support, etc.) by projecting the current function onto convex constraint spaces. Continuing in this way, the algorithm simultaneously estimates the missing projections (or their Fourier transforms) and the object, constrained by whatever constraints that have been imposed along the way.

From the modeling point of view, one problem with these iteration between spaces methods is that they don't allow incorporation of any probabilistic description of the object or measurement noise (except through moment matching techniques [87]). Also, iteration between projection space and object space inevitably incurs a high degree of finite-pixel error over time [12]. To get around the latter problem, Kim, Kwak, and Park [64,43] have proposed a method which estimates the missing projections through iteration entirely in projection space.

1.4 Geometric Model-Based Reconstruction

The underlying object models discussed so far have been proposed primarily to attempt to represent a broad class of possible cross-sections. Methods such as POCS attempt, during the reconstruction, to force the object parameters into a constrained subset of their possible values. A more direct approach — geometric modeling — is to parameterize the object directly in the class of interest, thus (possibly) reducing the number of parameters and hence the number of objects these parameters may

represent. Relatively few efforts have been made along these lines, however, perhaps because of the the overwhelming success of the usual reconstruction algorithms in the full-view and high SNR cases. It is expected that it is in the low SNR and limited-view cases that geometric modeling will reap great rewards.

The work by Rossi and Willsky [74,75] has greatly influenced the course of this research to date. In this work, the object was represented by a known profile, with a very small number of embedded unknown (and non-random) parameters: center position, density, radius, eccentricity, and orientation. These parameters were then estimated using a maximum likelihood formulation. As an example, the simplest case they considered was a disk object whose radius and density were known, but whose center position was unknown. Thus, only two parameters were to be estimated: the vector components of the object's position.

The advantage of posing the problem in this fashion is that one can expect better performance estimating only two parameters as opposed to the 65,000 (pixels) one may typically attempt to estimate. Another advantage is that this estimator can be studied analytically in great detail to determine its performance (resolution and robustness). There are disadvantages, however. First, the class of objects is very small in this case; therefore, the applicability is narrow (however, its performance in estimating parameters such as location was found to be extremely robust in the presence of modeling errors). Second, the parameters are embedded non-linearly in the model, leading to difficulty in finding jointly optimal solutions — instead, a suboptimal hierarchical approach was taken.

The geometric modeling approach of Rossi and Willsky was expanded upon by Bresler and Macovski [7,8,9]. These researchers considered the 3-D X-ray transform (see [54]); thus, they observed the 2-D projection of a function of three dimensions. The objects being imaged were blood vessels (enhanced by contrast agent and digital subtraction angiography). To model this class of objects, the authors chose sequences of equal length 3-D cylinders, each with unknown radius, position, and orientation. The reconstruction proceeded hierarchically via a three step procedure which we will not repeat here. It is important to point out, however, that their algorithms contained aspects which did not appear in Rossi and Willsky's work. For

one, it required the detection of many objects in the field rather than just one, and the subsequent (suboptimal) estimation of those object's parameters. Their model also included a probabilistic description of the relationship between cylinders (but this relationship was not used in the detection phase).

Other investigations into geometric model-based reconstruction have been made. Chang and Shelton [16,17] and more recently, Horn [37] and [23] have investigated the reconstruction of binary objects. The latter authors concluded that two (noise-free) projections taken exactly 90 degrees apart are sufficient to exactly reconstruct a binary object. Hanson [30] has investigated the reconstruction of axially symmetric objects (from a single 2-D radiograph).

1.5 Contributions

This thesis makes contributions in several areas. The first area is in the approach we use to incorporate prior knowledge in the problem of reconstruction from projections. In particular, we have shown that prior knowledge of several types, including knowledge of a prior probability on sinograms, the consistency conditions, and the relationship between convex object support and the support on the Radon transform, can be used effectively and efficiently to improve reconstructions. The fact that this information is incorporated in Radon space allows us to implement an algorithm which finds the optimal MAP estimate of the *sinogram* without transforming the data into object space — a step used in other methods which invariably causes a loss of information due to computational errors and adds considerably to the overall processing time. The object image is produced by using convolution backprojection (CBP) on the final sinogram MAP estimate.

A second contribution is in the area of algorithms for reconstruction from projections. Here, because of the constraints that are imposed as a result of the consistency conditions and because of the form of the prior probability on sinograms we have developed a novel approach to reconstruction which solves a partial differential equation on the sinogram domain, subject to the consistency constraints. Because the sinogram prior is a local Markov random field (MRF), the solution of

the partial differential equation is very efficient and in fact may be implemented in parallel with great speed. Because of the constraints, however, several iterations may have to be made with a correction for the constraints incorporated between iterations.

A third contribution of this thesis is in the area of computational geometry. Here, we have developed a theory of shape estimation, and in particular, we have developed methods to reconstruct convex sets from noisy support line measurements. The purpose of these methods in this thesis is to estimate the convex support of objects, which may then be used hierarchically in our sinogram reconstruction algorithm. However, the methods may have much broader application in fields which involve the reconstruction of convex sets, e.g., robotics [37], chemical component analysis [29,38], and silhouette imaging [89,90]. The novel aspects of this theory of set reconstruction is its incorporation of noise statistics, the ability to incorporate prior shape information, and in the incorporation of fundamental constraints on sets of support lines.

A fourth contribution is embodied in the hierarchical reconstruction algorithm itself. In this algorithm, we initially estimate various geometric properties of the object and sinogram, the results of which are passed into a sinogram estimation algorithm of the type mentioned above. At each stage, information about the *reliability* of the estimates is also passed into the succeeding stage, thus allowing the final stages to emphasize or de-emphasize the available geometric information.

1.6 Organization

The thesis is organized as follows. Chapter 2 reviews several subjects which are considered to be essential background to the subsequent chapters. Chapter 3 begins this thesis' contributions with the development of sinogram MAP estimation techniques using a Markov random field sinogram prior probability. This chapter is primarily concerned with an approach to full reconstruction using certain prior information. Chapter 4 develops an approach to consistent and optimal estimation of the convex support of an object from noisy support line estimates. Chapter 5

continues this development by discussing several potential prior probabilities on sets of support lines. In Chapter 6 we discuss methods to make initial estimates of the positions of support lines from individual projections. Chapter 7 merges the ideas of Chapters 3, 4, 5, and 6 into one hierarchical algorithm for producing full object reconstruction. Chapter 8 takes a more fundamental look at the consistency of the Radon transform, and proposes an algorithm which incorporates more of the consistency conditions than those of Chapter 3. Finally, we summarize our results and propose further research in Chapter 9.

Chapter 2

PRELIMINARIES

2.1 Overview

In this chapter we present the theoretical background which is common to many of the subsequent chapters. Section 2.2 describes the general theory of Markov random fields (MRF's) and Gibbs densities, which are used in many places in the thesis as prior probabilities on sinograms. In Section 2.3, we describe the concept of maximum *a posterior* (MAP) estimation and then present in Section 2.4 a particular algorithm for computing MAP estimates — called simulated annealing — which can be of value for large-scale problems in which the prior probability is given by an MRF. In Section 2.5 we present ideas related to the convex support of objects and the support of the Radon transform, and in Section 2.6 we we discuss the consistency of the Radon transform, a topic which was briefly mentioned in Chapter 1. Finally, Section 2.7 gives a general description of constrained optimization algorithms and a more detailed description of a quadratic programming (QP) code called ZQPCVX, which is used in various places in the thesis.

2.2 Markov Random Fields and Gibbs Densities

The sinogram prior which will be developed in Section 3.2 is a Markov random field (MRF). MRF's have recently emerged as a powerful tool for probabilistic modeling of phenomena defined on lattice systems [26], [5]. We now give general background

on MRF's and on their joint probability densities, Gibbs densities.

Markov random fields extend the concept of Markov chains to multiple dimensions. The goal is to be able to specify local probabilistic interactions of random variables through their conditional probabilities. Arbitrary specification of such conditional probabilities in the multidimensional case does not always lead to a consistent joint density, however. In order to show the consistency relationships, which become very important in the specification of MRF's, we need some definitions.

Consider a set of random variables $(x_1, \ldots, x_n)$ defined on the set of lattice sites $S = \{1, \ldots, n\}$. We associate to S a *graph structure*, such that each site is connected (via non-directed arcs) to a number of other sites in S. The set of sites connected to site $s \in S$ is denoted N_s, and is called the *neighborhood* of s (note that $s \notin N_s$).

An MRF is defined through its conditional probabilities in the following way. The vector of random variables $x = (x_1, \ldots, x_n)$ is an MRF if, given the prescribed graph structure, the joint probability and conditional probabilities obey

$$(a) \qquad p(x) > 0 \quad \forall x \in \Omega_x$$

$$(b) \quad p(x_s \mid \{x_r : r \in S - \{s\}\}) = p(x_s \mid \{x_r : r \in N_s\}).$$

It is important to point out that given a set of arbitrary conditional probabilities, it is not always possible to construct a consistent joint density $p(x)$. The nature of the consistency constraints was discovered by Besag [5], and are most easily stated by discussing the required form of the joint density.

Because of property (a), the joint density may be written as

$$p(x) = \tfrac{1}{Z} e^{-U(x)/T} \quad x \in \Omega_x \tag{2.1}$$

Probability density functions (pdf's) with the above form are called *Gibbs densities*. T is a real constant called the *temperature*, and $U(x)$ is called the *energy function*. Z is a real constant chosen to make the density integrate to one over Ω_x; Z is referred to in the literature as the *partition function*, the functional dependency being on T, in general. We shall call any possible sample realization of x a *configuration*.

As it is, the Gibbs density is not very restrictive — any joint density may be written in the form of (2.1) provided that Ω_x is chosen to exclude all zero probability

states. What restricts the form of $p(x)$ is the prescribed graph structure. Besag showed that, given a graph structure, the energy function in the Gibbs density must have the form

$$U(x) = \sum_{i \in C_1} V_i(x_i) + \sum_{(i,j) \in C_2} V_{ij}(x_i, x_j) + \sum_{(i,j,k) \in C_3} V_{ijk}(x_i, x_j, x_k) + \cdots + V_{1,\ldots,n}(x_1, \ldots, x_n) \tag{2.2}$$

Here, the sums are taken over groups of sites called *cliques*. A clique is defined to be either a single site or a collection of sites each pair of which are neighbors of each other. In the notation above, the set C_i contains all the cliques with i sites. Once the graph structure is chosen, the freedom in specifying an MRF remains in the functions $\{V\}$, called *potential functions*, each of which depends on the values of i random variables and their indices (site locations).[1]

This interesting dependence allows one to specify an MRF by specifying the energy function $U(x)$, through its potential functions, $\{V\}$. This has two advantages over specifying the conditional probabilities. First, in defining the $\{V\}$'s, one automatically defines a set of conditional probabilities which is consistent with its underlying graph structure. Second, the potential functions provide a very natural way of describing affinities and differences among site values. For example, if we expect two neighboring site values to have similar values then the potential function should have a large value when the difference between the two values is large, and a small value when the difference is small. This specification then relates directly to the joint probability; it says that the probability of a configuration with similar neighboring site values is larger than with differing neighboring site values.

Example:

Suppose the lattice is square with $n = mxm$ sites (see Fig. 2.1a). Define the graph to be nearest-neighbor connections and nearest-diagonal connections, such that each site has 8 neighbors. Let the connections wrap around at the edges (such a lattice is said to be toroidal), as shown in Figs. 2.1a and 2.1b. Fig. 2.1c shows the possible sets of cliques. From the theory given above, the energy function may be

[1] We will on occasion abbreviate our notation for the potential functions by omitting the arguments, e.g. V_i will stand for $V_i(x_i)$, and V_{ij} will stand for $V_{ij}(x_i, x_j)$, etc..

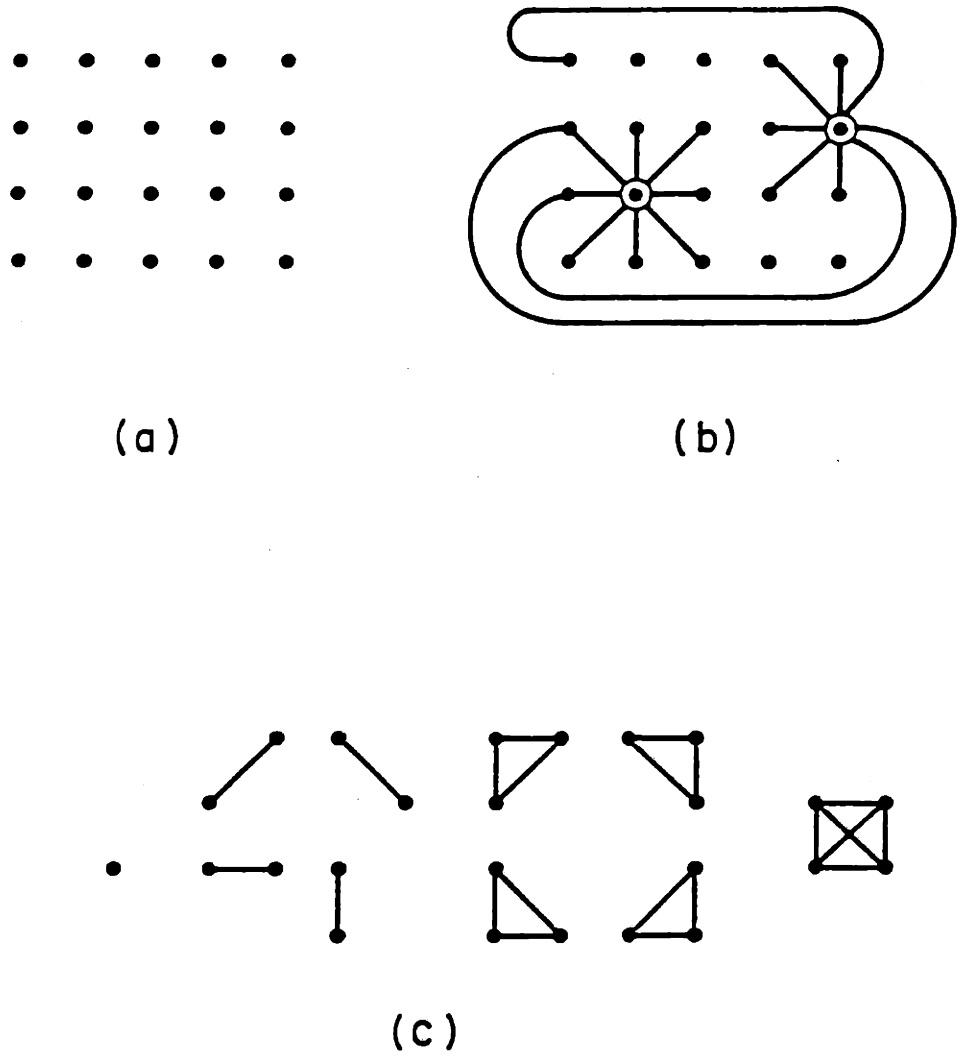


Figure 2.1: (a) A lattice, (b) two sites and their neighbors, and (c) the structure of the cliques.

written in its fullest generality as

$$U(x) = \sum_{i \in C_1} V_i(x_i) + \sum_{(i,j) \in C_2} V_{ij}(x_i, x_j) + \sum_{(i,j,k) \in C_3} V_{ijk}(x_i, x_j, x_k) + \sum_{(i,j,k,l) \in C_4} V_{ijkl}(x_i, x_j, x_k, x_l) \quad (2.3)$$

because the largest clique has only four sites. Now, one is free to define the potential functions. The V_i's, or *self-potentials*, correspond to knowledge concerning the site values themselves. For example, the values may be influenced by the observed data as when $p(x)$ represents a posterior density [26]. In the model to be proposed in Section 3.2, we use the self-potentials to reflect information that is known about the support of the object. The V_{ij}'s, or *pair-potentials*, relate to the interaction of two neighboring sites. They may be used to reflect how alike neighboring site values are expected to be.

2.3 MAP Estimation

Maximum *a posteriori* (MAP) estimation is often used for parameter estimation when the parameters have a known prior probability distribution [91]. MAP estimation is the focus of Chapter 3, and is used throughout this thesis in various contexts. The method presumes that the joint density between the unknown parameter (vector) x and the observation (vector) y is known, and is given by $p(x, y)$. Then given an actual observation $y = Y$, we define the MAP estimate as that value of x which maximizes the conditional density of x given $y = Y$ (the posterior or *a posteriori density*), or

$$\hat{x}_{map} = \underset{x \in \Omega_x}{\operatorname{argmax}}\, p(x|y = Y) \quad (2.4)$$

One may derive alternate expressions for $\hat{x}_{map}$ using Bayes' rule:

$$\hat{x}_{map} = \underset{x \in \Omega_x}{\operatorname{argmax}}\, p(x, y = Y) \quad (2.5)$$

$$= \underset{x \in \Omega_x}{\operatorname{argmax}}\, \ln p(y = Y|x) + \ln p(x) \quad (2.6)$$

In all these expressions the set Ω_x represents the set of feasible parameters, thus this is the set over which the prior probability $p(x)$ integrates to one.

2.4 Optimization by Simulated Annealing

Simulated Annealing (SA) was recently introduced by Kirkpatrick [45] as a general approach to solving large-scale combinatorial optimization problems where traditional algorithms are often inadequate because of the lack of derivative information and the presence of many local minima. The method is tied closely to statistical mechanics concepts because of the use of the Gibbs density in the formulation and implementation of the algorithm. The method seeks to minimize a cost function — the Gibbs energy — by finding the configuration which is most probable, i.e. maximizes the Gibbs density (see equation (2.1)). If the Gibbs density is in the form of a posterior density, then the optimal configuration is the MAP estimate.

2.4.1 General Description

Here is a general description of the SA algorithm. Suppose the problem is to find x^* that solves

$$\text{(S)} \qquad \begin{array}{ll} \text{minimize} & E(x) \\ \text{subject to} & x \in \Omega_x \end{array}$$

where $x \in \mathbb{R}^n$ is a vector of parameters and $E : \mathbb{R}^n \rightarrow \mathbb{R}^1$ is the objective function. We form the Gibbs density as

$$p(x) = \frac{1}{Z} e^{-E(x)/T}$$

where Z is the partition function and T is a real parameter called the temperature.[2] As mentioned above, (S) may be restated as

$$\text{(S)} \qquad \begin{array}{ll} \text{maximize} & p(x) = e^{-E(x)/T} \\ \text{subject to} & x \in \Omega_x \end{array}$$

Here are two key observations:

- The solution to (S) is independent of T, and

[2]The usual Gibbs density has kT, where k is the Boltzmann constant, but it is not necessary here.

- As T is lowered (from some large positive number) toward zero, the probability becomes concentrated (i.e., markedly peaked) at x^*, the solution to (S). [3]

The success of SA depends upon these two facts and on the the existence of algorithms which *simulate* the system. Simulating the system, in this case, means obtaining independent sample configurations $x_i \in \Omega_x$ chosen according to the probability rule $p(x)$ for any specific value of T. We shall discuss one such simulation algorithm, the Metropolis algorithm, below.

There exists two distinct components to the SA algorithm: 1) Simulation, and 2) Annealing. The approach is to simulate the system at a high temperature and slowly lower the temperature — the prescription for lowering the temperature is called the *annealing schedule*. At the lowest temperatures, the sample configurations will be close to or have the exact value of one of the global solutions x^*. The analogy to physical annealing (of metals) is quite strong in some cases, and we may learn about the behavior of SA by analogy. For example, in the physical annealing of crystalline materials it is essential that the temperature is lowered very slowly when passing through "critical" temperature regions. If this is not done, then irregularities in the crystal will result (spin glasses, etc.) which implies that the crystal is not at its lowest energy state. These alternate configurations correspond to local minima in the energy function, which we wish to avoid. Therefore, to successfully implement SA, it is essential to make both the simulation and annealing stages fast, but not so fast that the final estimates correspond to local minima.

A potentially very important aspect for decreasing the computation time of an SA implementation occurs when the Gibbs density describes a *local* MRF. Here, local does not necessarily mean that sites have nearby neighbors, but instead, that the sites have a *small number* of neighbors. In this case, the simulation stage may often be accomplished with a high degree of parallel computation. And even if parallel computation is not available, the computation on a scalar machine is often significantly reduced.

[3] If there is more than one solution to (S) then the probability becomes concentrated on the set of solutions.

Our primary purpose for introducing SA in this thesis is because of its close ties to MRF modeling. The implementation in Chapter 3 for a particular energy function turns out to be a rather inefficient algorithm for the sinogram MAP estimation problem. However, SA is very general, and would still be applicable even if the MRF that we specified had a far more complex form. We now describe the simulation stage of SA — the Metropolis algorithm.

2.4.2 The Metropolis Algorithm

The Metropolis Algorithm (MA) is a Monte Carlo technique developed by Metropolis et. al. in 1954 [60] for numerical computation of the statistics of an N-particle system in thermodynamic equilibrium. Essentially, this algorithm allows one to compute a sequence of independent samples of a Gibbs density. It is now a commonly used technique in the study of MRF's, and as a stage in the SA algorithm. We describe the general method here; in Chapter 3 we describe specific changes which we made to account for the constraints in the sinogram MAP problem.

Let $x = [x_1, \ldots, x_N]^T$ be a vector of random variables, where x_i may take on a finite number of values. The vector x may be thought of as the state of a system, where we impose the restriction that there be only a finite number of states in the system. The objective of MA is to obtain a sample configuration X (i.e., a sample realization of x) that is obtained with probability given by the Gibbs measure

$$p_x(X) = \frac{1}{Z} e^{-U(X)/T} .$$

The procedure begins with a feasible configuration X^0, then makes random perturbations, forming a random sequence X^0, X^1, ... of feasible configurations. The Metropolis algorithm specifies the rules for the random perturbations so the derived sequence may be thought of as coming from a Markov chain whose limiting state probabilities are identical to $p_x(X)$. In practice, one halts the process at some iteration k, and uses X^k as a sample configuration chosen approximately from $p_x(X)$.

The Metropolis Algorithm:

1. *Initialization.* Choose any arbitrary initial feasible state X^0. Set $k = 0$.

2. *Site Visitation.* Choose a site j out of all sites $1, 2, \ldots, N$ with uniform probability.

3. *Site Replacement.* This stage consists of three parts. First one makes a random perturbation of the state, producing a new state called the proposal. Then one calculates the change in energy caused if the proposal were accepted, and finally one either accepts or rejects the proposal.

   (a) *Proposal.* Choose a new value $\tilde{X}_j^k$ from the set of allowable states using a random number distributed uniformly over the set.

   (b) *Energy Change.* Compute the change in energy $\Delta U^k = U(\tilde{X}^k) - U(X^k)$ that would result if the jth element of the vector X^k were changed, where U is the energy in the Gibbs density.

   (c) *New State.* If $\Delta U^k \leq 0$ set $X^{k+1} = \tilde{X}^k$. Otherwise generate a random number r, uniformly distributed between 0 and 1. If $r \leq e^{-\Delta U^k/T}$ set $X^{k+1} = \tilde{X}^k$, otherwise set $X^{k+1} = X^k$.

4. *Repeat.* Let $k = k + 1$. Go to Step 2.

Many proofs of the convergence of the Metropolis algorithm exist (cf. [60], [44], [59], [24], [88]). What we mean by convergence is that the Markov chain formed by the algorithm has an invariant probability which is equivalent to the Gibbs density. A key idea necessary for convergence which we will require in Chapter 3 concerns the selection of the next site in Step 2 and selection of the proposal in Step 3. Taken together, these two steps, which select a new state which may or may not be accepted in Step 3(c), will be called the *selection* step. Let us think of the Markov chain that would be generated if we accepted each selection. Denoting two states in the chain by r and s, and the probability of selecting s given that the current state is r by q_{rs}, the proof of convergence requires that $q_{sr} = q_{rs}$, a condition which is known as *strong reversibility.* Furthermore, it is required in this new chain that each state be able to reach any other state in a finite number of steps. If we let Q denote the one-step transition probability of this chain, then the two above conditions are equivalent to the conditions that Q be symmetric and irreducible [24].

The conditions given above are satisfied in the statement of the algorithm given above. However, in Chapter 3 we will encounter constraints which will not permit changing only one site value at a time while simultaneously satisfying the constraints. We will therefore propose a modification of the selection process which can be made to satisfy the symmetry and irreducibility conditions.

One final note on the proof of convergence. It is common to find in the literature implementations of the Metropolis algorithm which scan through the sites in a deterministic fashion, thus modifying the random site visitation rule of Step 2.[4] However, it is easy to see that this method does not meet the strong reversibility requirement of the selection process which we described above, since after making a change in the value of site i, for example, it is impossible to make a transition back to the original state by changing the value at site $i + 1$. Marroquin [59] has shown that for any deterministic scan procedure (most notably, the raster scan and any parallel updating scheme that updates first one set of sites, then another) the above algorithm still has an invariant probability, but it is not equivalent to the desired Gibbs density. He has further shown that using such a method in an SA algorithm can lead to solutions which are very different from the known optimum. It should be noted that a computationally more intensive method known as the Gibbs sampler, which was developed by Geman and Geman [26], also forms a Markov chain whose invariant probability is the Gibbs density, and this method can be implemented using deterministic or parallel updating so long as each site is visited infinitely often.

2.4.3 Annealing Schedule

The annealing schedule prescribes the method by which the temperature is lowered in the SA algorithm [45]. In this section we describe the particular adaptive schedule that we have employed in our experiments. In addition, we describe the stopping criteria used for both the Metropolis algorithm and SA as a whole.

[4] In fact, Metropolis et. al. actually used such a procedure.

Stopping Criterion for the Metropolis Algorithm

To obtain a sample configuration from a Gibbs density, ideally, one would like to let the Metropolis algorithm run forever. In practice we must choose a stopping time that has allowed the chain to get close to equilibrium. But also, in the context of SA, it is reasonable to terminate the MA stage for other reasons as well, e.g. the temperature may be much too high or the algorithm appears to be stuck at a local minimum. We implement a heuristic method due to Bunks [10,11].

At each site replacement, there are three possible results:

1. $\Delta U \leq 0$ and the proposal is accepted.
2. $\Delta U > 0$ and the proposal is accepted.
3. $\Delta U > 0$ and the proposal is rejected.

Let us denote the number of occurrences of each outcome by $C1$, $C2$, and $C3$, and the number of occurrences over a trailing window of length W by $W1$, $W2$, and $W3$.[5] The total number of site replacements is $C = C1 + C2 + C3$. If this exceeds 100 sweeps (i.e., $C > 100$ times the total number of sites) then equilibrium is deemed not to have been achieved, but the Metropolis stage is terminated anyway — this outcome has an affect on the temperature scheduler described in the following section. The only other way for the Metropolis algorithm to terminate is if $C1+C2$, the total number of accepted changes (whether up or down) has exceeded 10 sweeps.

The Temperature Schedule

After the Metropolis stage, the temperature must be lowered. We use an adaptive method which attempts to determine the condition of the system in order to follow a fast annealing schedule, but not so fast that the estimate gets stuck in a local minimum. Following Bunks [10,11], we implement a method which follows one of two different temperature scales depending on the outcome of the MA stage. The idea is to use the statistics $W1$, $W2$, and $W3$ to determine whether the system is at

[5] All counters are reset to zero when the temperature is changed by the temperature scheduler.

or near a local minimum. If so, the temperature is not lowered very much, or the system will never have an opportunity to escape from that local minimum.

At or near a local minimum, there will be a tendency for proposals to have higher energy than the current configuration, and if the current temperature seems to be too low for an escape to be made then there will be a high number of rejections. Therefore, we consider the ratio $R = W2/(W2 + W3)$, which ranges between zero and one. When R is low it is clear that very few upward energy proposals were accepted, and therefore it is likely that the current configuration is at or near a local minimum. In this case we follow the slow scale, which lowers the temperature only a small amount at each stage. When the system first begins the slow scale the current temperature is saved in the variable T_c and the counter k is set to 2. Then, as long as the slow scale is being followed, the next temperature is determined according to the rule $T_{k+1} \leftarrow T_c \ln 2/\ln k$, where k is incremented after every temperature update.[6] We say that R is low if it is less than 0.25, this value having been chosen because it yields good experimental results. When R is high (i.e. $R \geq 0.25$), the fast temperature scale is initiated. In this case the temperature is always halved, that is $T_{k+1} \leftarrow T_k/2$, where k is the counter set to 1 when the fast scale is initiated. A typical annealing process would begin at a high temperature, follow the fast scale for some time, then enter the slow scale during a critical region, often return to the fast scale, and finally finish on the slow scale. Examples will be shown in a subsequent section.

SA Stopping Criterion

As the temperature is made lower and lower, the configuration eventually "freezes" into a state that rarely changes. Our version of SA stops when three consecutive MA stages have allowed the maximum number of site replacements to be attempted. The final optimum value is then given by the configuration that has achieved the lowest energy during the entire annealing algorithm.

[6]The slow scale was chosen to resemble the theoretical result by Geman and Geman [26] which has a rate given by $C/\ln k$, C being a scalar quantity related to the number of sites and the energy function. In their result, however k is a *site visitation* counter, whereas in ours k is an *equilibrium* counter.

2.5 Object Support and Radon Transform Support

One of the important geometric aspects of objects which we explore in this thesis is convex support. What we are most interested in is the smallest convex set which supports the function $f(x)$, and how this set represents itself in Radon space. In this section, we begin a discussion related to the convex support of objects; this subject is continued in more detail in Chapters 4 and 5. The basic idea is that if the function f is zero outside of the set $\mathcal{F}$, which is itself a subset of the disk with radius T centered at the origin, then the Radon transform of f must be zero outside a set $\mathcal{G} \subset \mathcal{Y}_T$ where

$$\mathcal{Y}_T = \{(t,\theta) \mid -T \leq t \leq T, 0 \leq \theta \leq \pi\}. \tag{2.7}$$

We now explore this relationship in more detail.

Consider the function $f(x) : \mathbb{R}^2 \rightarrow \mathbb{R}^1$ which is zero outside D_T, where D_T is the disk of radius T centered at the origin. Let $\mathcal{F}$ be the set of points for which $f(x) \neq 0$. Clearly, we must have that $\mathcal{F} \subset D_T$. Now consider the Radon transform $g(t,\theta)$ of f, and the unit vector $\omega = [\cos\theta \ \sin\theta]^{\mathrm{T}}$. With reference to Fig. 2.2 and to the definition of the Radon transform in (1.1), we see that for any given ω, the value of the Radon transform must be zero for $t \geq t_+$ and $t \leq t_-$. Here, t_+ is the lateral position of the line perpendicular to ω which is positioned as far as possible in the $+\omega$ direction so it just grazes the set $\mathcal{F}$; t_- is the lateral position of the line perpendicular to ω which is positioned as far a possible in the $-\omega$ direction so it just grazes the set $\mathcal{F}$. In Chapter 4 we identify t_+ and t_- as *support values* and the corresponding lines as *support lines* of the set $\mathcal{F}$. Knowledge of both t_+ and t_- for all θ in $[0,\pi)$ determines the convex hull of $\mathcal{F}$, denoted hul($\mathcal{F}$), which is, by definition, the smallest convex set which contains $\mathcal{F}$.

We now define the set $\mathcal{G}$ to be

$$\mathcal{G} = \{(t,\theta) \in \mathcal{Y}_T \mid t_-(\theta) \leq t \leq t_+(\theta)\} \tag{2.8}$$

where, for clarity, we have explicitly indicated the functional dependence of t_+ and t_- on θ. We show one example of such a set in Fig. 2.3. Note that because of the

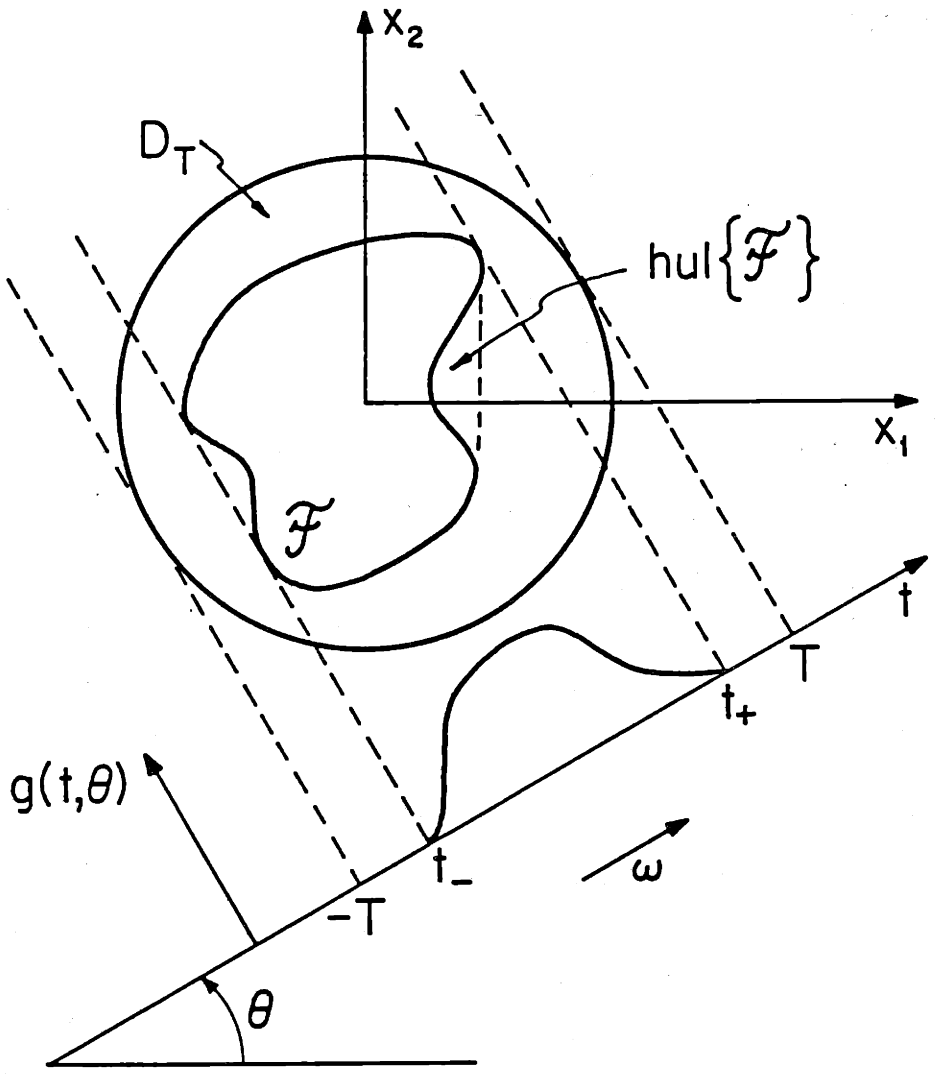


Figure 2.2: The convex support of an object and the support of a projection.

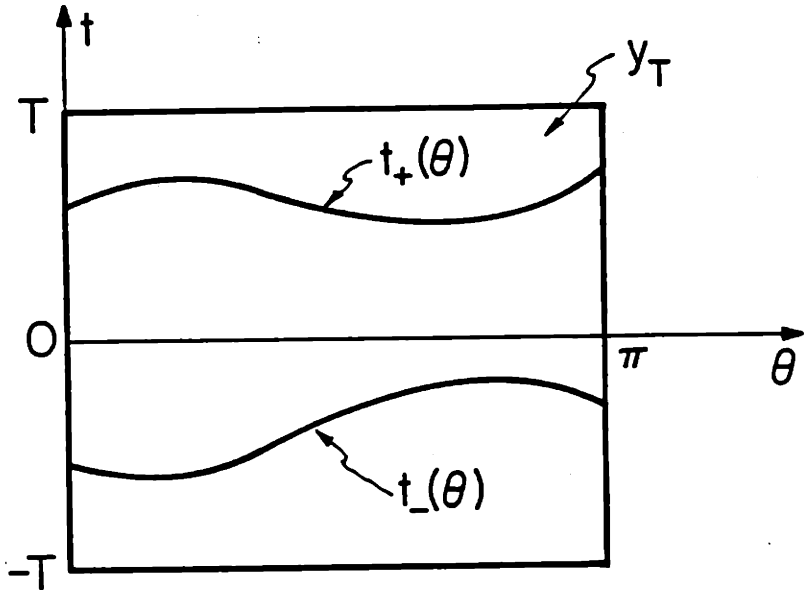


Figure 2.3: The support of a Radon transform.

definition of $\mathcal{G}$, we must have $g(t,\theta) = 0$ when $(t,\theta) \notin \mathcal{G}$. For a given object support set $\mathcal{F}$, we think of $\mathcal{G}$ as the matching region of support in Radon space. However, although $\mathcal{F}$ uniquely determines $\mathcal{G}$, it is clear that $\mathcal{G}$ uniquely determines hul($\mathcal{F}$), not $\mathcal{F}$ itself. This is why we are primarily concerned with the convex support of f, since this is what may be determined directly from knowledge of $\mathcal{G}$.

It is the objective of Chapters 4, 5, and 6 to attempt to estimate $\mathcal{G}$ from noisy and incomplete data. Chapter 4 is concerned primarily with the constraints that sets of support lines must obey, while Chapter 5 develops methods to include prior shape and size information in the estimation of $\mathcal{G}$. Chapter 6 is concerned with the estimation of the values t_+ and t_- from a single noisy projection. In Chapter 3 we assume that an estimate of $\mathcal{G}$ is available, and we define a prior probability on sinograms which gives a low probability to sinograms which have non-zero values outside of $\mathcal{G}$.

2.6 Consistency of the Radon Transform

An important fact that recurs throughout this thesis is that not all functions $g : \mathbb{R}^1 \times S^1 \rightarrow \mathbb{R}^1$ are valid 2-D Radon transforms. In the first part of this section we present several well-known mathematical properties of Radon transforms that are known as the consistency relations. We will see that a valid Radon transform, that is, a function which is the Radon transform of some function $f : \mathbb{R}^2 \rightarrow \mathbb{R}^1$, is constrained to lie in a particular functional subspace of the space of all real functions. This subspace is characterized by the property that g is even in t and ω and by the property that certain generalized Fourier coefficients of g must be zero. These properties are exploited in the constraint-based reconstruction methods of Chapter 8. In the second part of this section we show that the mass and center of mass of the object f appear in two of the lower order constraints on g. These constraints are incorporated in the Markov random field model developed in Chapter 3 and are used again in Chapter 7.

The Consistency Theorem [55,32]

The Consistency Theorem given below is stated in terms of the n-dimensional Radon transform given by

$$g(t,\omega) = \int_{x \in \mathbb{R}^n} f(x)\delta(t - \omega \cdot x)\, dx \tag{2.9}$$

where $\delta(\cdot)$ is the Dirac delta function, and ω is a unit vector in $\mathbb{R}^n$, i.e. $\omega \in S^{n-1}$. We motivate the theorem by considering two simple facts. First, consider the geometry of the 2-D Radon transform shown in Fig. 1.1. We see that the line $L(t,\theta)$ is also given by $L(-t,\theta+\pi)$. Then if we denote the integral of f along L by $g(t,\omega)$ where $\omega = [\cos\theta \ \sin\theta]^T$ is the unit vector pointing in the θ direction, we see that it must be true that $g(t,\omega) = g(-t,-\omega)$. This symmetry condition is also true for the n-dimensional Radon transform since the coordinates (t,ω) and $(-t,-\omega)$ each describe the same hyperplane $P = \{x \in \mathbb{R}^n \mid x \cdot \omega = t\}$ (over which integration of $f(x)$ takes place).

The second fact may be seen by multiplying g by the powers of t, integrating in t, and manipulating using the definition of the n-dimensional Radon transform

given in (2.9) as follows

$$\begin{aligned}\int_{-\infty}^{\infty} g(t,\omega)t^k\,dt &= \int_{-\infty}^{\infty} t^k \int_{x\in\mathbb{R}^n} f(x)\delta(t-\omega\cdot x)\,dx\,dt \\ &= \int_{x\in\mathbb{R}^n} f(x)(\omega\cdot x)^k\,dx \qquad k=0,1,2,\ldots \end{aligned} \tag{2.10}$$

We see that the expression in (2.10) must be a homogeneous polynomial of degree k in $\omega_1,\omega_2,\ldots,\omega_n$. We now give the theorem.

Let S denote the space of rapidly decreasing C^∞ function on $\mathbb{R}^n$, where $\mathbb{R}^n$ is the real n-dimensional Euclidean space. We now characterize the consistency of the Radon transform for functions belonging to S with the following theorem.

Theorem 2.1 (Consistency Theorem) *In order for $g(t,\omega)$ to be the Radon transform of a function $f\in S$, it is necessary and sufficient that*

(a) $g\in S(\mathbb{R}^1\times S^{n-1})$,

(b) $g(t,\omega)=g(-t,-\omega)$, *and*

(c) the integral

$$\int_{-\infty}^{\infty} g(t,\omega)t^k\,dt \tag{2.11}$$

be a homogeneous polynomial of degree k in $\omega_1,\omega_2,\ldots,\omega_n$ for all $k\geq 0$.

Condition (c) may be replaced by the following condition (d): If $S_l(\omega)$ is a spherical harmonic of degree l, and if $k<l$ then

$$\int_{|\omega|=1}\int_{-\infty}^{\infty} g(t,\omega)t^k S_l(\omega)\,dt\,d\omega = 0\ . \tag{2.12}$$

Proof *See [32],[55], and comments in Appendix 2.A.* □

To see how generalized Fourier coefficients arise from this theorem (see Louis [53]), we observe that if the support of f is the disk of radius T centered at the origin, then the limits of the two integrals over t in (2.11) and (2.12) may be replaced by $\pm T$. In the same two integrals, let us now replace the term t^k by $P_k(t)$, a polynomial in t of degree k, which is a member of a complete orthonormal (CON)

set of functions defined on the interval $[-T,T]$. For example, in Chapter 8 $P_k(t)$ are the Legendre polynomials. Then the product $P_k(t)S_{lm}(\omega)$ represents a CON set of functions over $\mathbb{R}^1 \times S^{n-1}$, and therefore, condition (d) requires that a certain infinite set of generalized Fourier coefficients, given by the double integral on the left hand side of (2.12), must be zero. We shall exploit this situation in Chapter 8.

Since this thesis is concerned only with the 2-D Radon transform we now restate the Consistency Theorem for the 2-D case as follows.

Theorem 2.2 (2-D Consistency Theorem) *In order for $g(t,\theta)$ to be the 2-D Radon transform of a function $f \in \mathcal{S}$, it is necessary and sufficient that*

(a) $g \in \mathcal{S}(\mathbb{R}^1 \times S^1)$,

(b) $g(t,\theta+\pi) = g(-t,\theta)$, *and*

(c) the integral

$$\int_{-\infty}^{\infty} g(t,\theta)t^k\,dt \tag{2.13}$$

be a homogeneous polynomial of degree k in $\cos\theta$ and $\sin\theta$ for all $k \geq 0$.

Condition (c) may be replaced by the following condition (d): If $k < l$ then

$$\int_0^{2\pi}\int_{-\infty}^{\infty} g(t,\theta)t^k \frac{1}{\sqrt{\pi}}\cos l\theta\,dt\,d\theta = 0 \quad \textit{and} \tag{2.14}$$

$$\int_0^{2\pi}\int_{-\infty}^{\infty} g(t,\theta)t^k \frac{1}{\sqrt{\pi}}\sin l\theta\,dt\,d\theta = 0 \quad . \tag{2.15}$$

Proof *Follows directly from Theorem 2.1.* □

Here, we have recognized that there are always exactly two spherical harmonics of degree l and that they are given by $\frac{1}{\sqrt{\pi}}\cos l\theta$ and $\frac{1}{\sqrt{\pi}}\sin l\theta$ (see Appendix 2.A). In Chapter 8 we will combine conditions (b) and (d) and the symmetry of the sine and cosine functions to obtain an even more concise representation of valid 2-D Radon transforms.

Mass and Center of Mass Constraints

The mass and center of mass constraints constrain the two lowest order moments of $g(t,\theta)$ as follows. The mass constraint tells us that the integral from $-T$ to T of any projection, which may be thought of as the mass of the projection, must have the same value, equal to the integral of $f(x)$ over D_T. If, for example, a noisy measurement of a true Radon transform has any two projections which do not integrate to the same value *then the measurement is not a valid Radon transform*, and it follows that an inverse transform is theoretically undefined. The center of mass constraint tells us that the (1-D) center of mass of a given projection is equal to the projection of the (2-D) center of mass of the object onto the ω-axis. From this one can see that the collection of centers of mass of the projections for different θ must be a cosinusoidal function with period 2π. If that is not true for a given measurement then, again, the measurement is not a valid Radon transform. These two facts are easily shown using the Consistency Theorem, as follows.

Consider condition (c) in Theorem 2.2. For $k = 0$, condition (c) requires that $\int_{-\infty}^{\infty} g(t,\theta)\,dt$ is a constant — the same constant for all θ. From the manipulations in (2.10) we see that

$$m = \int_{-\infty}^{\infty} g(t,\theta)\,dt = \int_{x\in\mathbb{R}^2} f(x)\,dx \quad \forall\theta \tag{2.16}$$

where the constant m will be referred to as the *mass* of $f(x)$. Clearly, if f is known to be zero outside D_T, the disk of radius T centered at the origin, then we have

$$m = \int_{-T}^{T} g(t,\theta)\,dt = \int_{x\in D_T} f(x)\,dx \quad \forall\theta \tag{2.17}$$

which we will refer to as the *mass constraint* for the 2-D Radon transform.

For $k = 1$, condition (c) of Theorem 2.2 maintains that $\int_{-\infty}^{\infty} g(t,\theta)t\,dt$ is a polynomial in $\cos\theta$ and $\sin\theta$ of degree one. Therefore, the center of mass $c(\theta)$ of the projection $g(t,\theta)$ must satisfy (assuming bounded support as above)

$$c(\theta) = \frac{1}{m}\int_{-T}^{T} g(t,\theta)t\,dt = a\cos\theta + b\sin\theta \tag{2.18}$$

for some real constants a and b. Equation (2.18) shall be referred to as the *center of mass constraint*. It is also easy to see using the manipulations in (2.10) that the

center of mass of the projection $g(t,\theta)$ is the projection of the 2-D center of mass of $f(x)$. Indeed, if (R,ϕ) denote the polar coordinates of the center of mass of the object, it can be shown that [74]

$$c(\theta) = R\cos(\theta - \phi) \ .$$

When we impose the center of mass constraint in Chapters 3 and 7, we assume that the object is centered at the origin, and therefore that $c(\theta) = 0 \ \forall\theta$. This corresponds to assuming that the object is located at the origin, or that we had previously adjusted the measurements for a known object location so that it now appears that the object is located at the origin. This pre-shifting may always be accomplished provided one has a field of view (i.e. D_T) large enough to encompass both the original object and the object shifted to the origin. We assume this to be the case.

2.7 Constrained Optimization Algorithms

This thesis is primarily concerned with modeling and estimation. Most of the models proposed herein have natural constraints imposed on the model parameters, and hence, the estimation algorithms that are developed are constrained optimization algorithms. The general form of a constrained optimization algorithm is given by

$$\begin{array}{ll} \underset{x\in\Omega_x}{\text{minimize}} & f(x) \\ \text{subject to} & g(x) \geq 0 \text{ and } h(x) = 0 \end{array}$$

The function $f(x)$ is called the objective function, the set Ω_x is some large set usually taken to be $\mathbb{R}^n$, and the functions $g(x)$ and $h(x)$ define the inequality and equality constraints respectively.

In this thesis we are mostly interested in constraints that are linear, and in objective functions which are either linear or quadratic. The problem which has linear constraints and a linear objective function is called a linear programming (LP) problem and may be written as

$$\underset{x\in\Omega_x}{\text{minimize}} \quad c^{\mathrm{T}}x$$

$$\text{subject to} \quad Ax \geq b \ .$$

Note that the vector inequality is taken component by component and that any equality constraint may be written as two inequality constraints. It can be shown [56] that one need consider only a finite number of possible solutions in order to solve an LP, hence, it is inherently a combinatorial optimization problem. Many codes exist which exploit the structure of the LP to reduce the time from that required for an exhaustive search (cf. [48,67]). Thus although there are pathological examples which may cause certain algorithms to search all possible answers, LP codes are generally considered to be efficient algorithms. Note that the LP formulation does not guarantee a unique solution and, of course, there may be no solution at all. Good codes are written to detect these situations.

When the constraints are linear but the objective function is quadratic, the optimization problem is called a quadratic program (QP). It may be written in the most general form as

$$\begin{aligned} \underset{x \in \Omega_x}{\text{minimize}} \quad & \frac{1}{2} x^T G x + p^T x \\ \text{subject to} \quad & Ax \geq b \ . \end{aligned}$$

Clearly, the same comments apply regarding the equality constraints as in the LP. If G is positive definite then there is a unique solution to the QP, otherwise a unique solution cannot be guaranteed. As in the LP, it is also possible that there is no solution. Finding the solution to a QP is also combinatorial in the following sense. Suppose one knew what inequality constraints the optimal solution would satisfy with equality — these are the so-called *active constraints* at the solution. Then, since all these active constraints are linear equality constraints, the solution could be found exactly by Lagrange multiplier techniques. By definition, there is only a finite number of inequality constraints, therefore, there is only a finite number of possible combinations which could form the optimal set of active constraints at the solution. Then, in principle, one could search through all possible sets of active constraints, solving the exact equations at each stage. The optimum solution is the one out of this finite set which minimizes the objective function.

As in LP, there are more efficient ways to seek the solution to a QP. One of the best methods is incorporated in the Fortran code called ZQPCVX, which is documented in [66] (with corrections in [65]). This code uses duality to estimate the final Lagrange multipliers at each stage in the search. The estimated Lagrange multipliers serve as guides toward the optimal solution, thereby allowing the code to bypass many obviously non-optimal active sets. We use ZQPCVX extensively throughout this thesis to solve the many QP problems as they arise. Furthermore, in one problem in Chapter 5, we use successive QP stages, solved using ZQPCVX, interspersed with an unconstrained line search to solve a non-quadratic optimization problem.

2.A Comments on the Consistency Theorem

In Section 2.6, we have stated Theorem 2.1, the Consistency Theorem, in a slightly different fashion than the equivalent theorems in either Helgason [32] or Ludwig [55]. Helgason proves the theorem without part (d), while part (c) of Ludwig's theorem says that the polynomial is of degree $\leq k$, rather than making the somewhat stronger statement about homogeneity that we have made. Ludwig included condition (d) and stated that it is *equivalent* to his condition (c). In our statement, (d) is *not* equivalent to (c), but it may replace it and the theorem still holds. The distinction is that condition (d) requires (b) also in order to conclude (c), while Ludwig did not need (b) since his (c) is weaker. To clarify the matter completely, we now prove that condition (d) may replace (c) in Theorem 2.1.

First, we require some facts about the real spherical harmonic functions $S_l(\omega)$ (cf. [22],[20]). We call $S_l(\omega)$ a spherical harmonic of degree l if $S_l(\omega)$ is a harmonic polynomial (therefore it satisfies Laplace's equation) in $\omega_1, \ldots, \omega_n$ which is homogeneous of degree l (therefore for $\alpha > 0$, $S_l(\alpha\omega) = \alpha^l S_l(\omega)$). Harmonic polynomials of different degrees are orthogonal on S^{n-1} and for each degree l there are $N(n,l)$ orthonormal spherical harmonics. Furthermore, the set of functions S_{lm}, $m = 1, \ldots, N(n,l); l = 0, 1, \ldots$, where m indexes the different orthonormal spherical harmonics of degree l, forms a complete orthonormal (CON) set of functions on the sphere S^{n-1}. The general formula for $N(n,l)$ is available in [22]; all that we need to know in this thesis is that $N(2,l) = 2$, i.e. in two dimensions there are only two orthonormal spherical harmonics for any degree l. In fact, in 2-D the two orthonormal spherical harmonics of degree l are given by:

$$S_{l1}(\omega) = \frac{1}{\sqrt{\pi}} \cos l\theta \tag{2.19}$$

$$S_{l2}(\omega) = \frac{1}{\sqrt{\pi}} \sin l\theta \tag{2.20}$$

where $\omega = (\cos\theta, \sin\theta)$. Another important property of the $S_{lm}(\omega)$ is that they satisfy the symmetry condition [20]

$$S_{lm}(-\omega) = (-1)^l S_{lm}(\omega). \tag{2.21}$$

Now we proceed to prove that in Theorem 2.1 (and hence in Theorem 2.2) condition (d) may replace condition (c).

Proof For notational convenience we denote

$$I_k(\omega) = \int_{-\infty}^{\infty} g(t,\omega)t^k\,dt \ .$$

If (c) is true then $I_k(\omega)$ is a (homogeneous) polynomial of degree k, hence $I_k(\omega)$ is orthogonal over S^{n-1} to any spherical harmonic of degree $l > k$. Therefore, (c) implies (d). Now let us assume (a), (b), and (d) to be true; we will show that these conditions imply (c). From condition (d) we may conclude that $I_k(\omega)$ is a polynomial in ω of degree $\leq k$, and, hence, that it may be expanded in orthonormal spherical harmonics as follows

$$I_k(\omega) = \sum_{l=0}^{k}\sum_{m=1}^{N(n,l)} a_{lm}S_{lm}(\omega) \ . \tag{2.22}$$

Now, using the symmetry of $g(t,\omega)$ in (b) we may make the following manipulations:

$$\begin{aligned} I_k(\omega) &= \int_{-\infty}^{\infty} g(t,\omega)t^k\,dt \\ &= \int_{-\infty}^{\infty} g(-t,-\omega)t^k\,dt \\ &= (-1)^k\int_{-\infty}^{\infty} g(t,-\omega)t^k\,dt \\ &= (-1)^k I_k(-\omega) \ . \end{aligned} \tag{2.23}$$

Combining (2.22) and (2.23), and using the symmetry property of the spherical harmonics given in (2.21), yields

$$I_k(\omega) = \sum_{l=0}^{k}\sum_{m=1}^{N(n,l)} a_{lm}(-1)^{l+k}S_{lm}(\omega) \ . \tag{2.24}$$

Now we see that in order for $I_k(\omega)$ to satisfy both (2.22) and (2.24) it must be true that $a_{lm} = 0$ for $l+k$ odd. Therefore, since $S_{lm}(\omega)$ is a *homogeneous* polynomial of degree l we see that $I_k(\omega)$ is the sum of homogeneous polynomials in ω of degrees $k, k-2, k-4, \ldots$. Now, since $\|\omega\|^2 = \omega_1^2 + \omega_2^2 + \ldots + \omega_n^2 = 1$, we may multiply any term in the expansion by $\|\omega\|^2$ without changing the value of the expansion,

but this increases the degree of the term by 2. Hence, we may multiply any term enough times so that its degree is exactly k, which proves that $I_k(\omega)$ is *homogeneous* with degree k. □

Chapter 3

MAP ESTIMATION OF SINOGRAMS

3.1 Introduction

This chapter develops several methods for the use of certain types of prior knowledge to reconstruct images from noisy and incomplete projections. These methods assume prior information about the object's support, mass, and center of mass. In addition, they assume that the objects under consideration (e.g. x-ray density cross-sections) have non-negative values only (positivity constraint). We incorporate this information by first specifying a Markov random field (MRF) prior probability on sinograms. The remainder of the chapter concentrates on finding efficient algorithms to determine the MAP estimate of sinograms from noisy and incomplete observations.

The object's positivity, mass, and center of mass are incorporated as constraints on the set of feasible sinograms. In other words, any sinogram that has a negative value, or does not have the pre-specified mass and center of mass, has zero probability. The support information, on the other hand, is incorporated as a "penalty" term; sinograms with non-zero values outside the specified region of support are less likely than those with the same values inside the region but zero outside. In subsequent chapters, we develop methods to determine support, mass, and center of mass from the data. These methods serve as "preprocessors" to the methods in this chapter. We will see that the estimates of mass and center of mass are much more accurate than the estimate of support, with justifies the treatment of these

estimates as constraints and penalties, respectively.

The full reconstruction is accomplished in two stages. First, we estimate a full sinogram from partial, noisy measurements using maximum a posteriori (MAP) estimation techniques. The object estimate is then obtained by ordinary convolution backprojection (CBP) [33]. As mentioned in Chapter 2, other researchers have used estimation of the smoothed sinogram as a preprocessing stage to CBP, but few have attempted to incorporate knowledge about the object directly in sinogram space as we do here. The usual methods of applying constraints (POCS, and others) involve iteration between object and sinogram domains. A disadvantage of this latter method is that the transformation between these stages is not exact when dealing with finite data, and inevitably, as the iteration proceeds, a form of cumulative computational error, called "finite pixel error" [12] reduces the accuracy of the result.

The main advantage of our sinogram MAP approach is that the additive noise is white in this space, leading to a relatively straightforward statement of the MAP estimate. Unfortunately, the incorporation of object constraints and other prior knowledge is difficult because what we may naturally understand and describe in object space is not always easily translated to sinogram space. What we develop herein serves as a framework for further development of sinogram MAP methods. As we proceed, we shall make comments as to possible extensions, and remark on the additional complexity such additions may create.

This chapter begins with a development of a Markov Random field (MRF) on the sinogram lattice that includes the positivity, mass, and center of mass constraints. This model also introduces smoothing terms which allows us to specify sinograms which tend to be locally smooth. Furthermore, we incorporate the convex support of the object by including a term which tends to cause the estimated sinograms to be zero outside the region of support. At this point, quite a bit of information is assumed to be known about the object. In subsequent chapters we will indicate how some of this information may be estimated and then used in a hierarchical reconstruction scheme. Chapters 4, 5, and 6, for example, develop methods for the estimation of the convex support of an object given noisy projections. Chap-

ters 7 and 8 discuss the estimation of mass and center of mass, and present the full hierarchical reconstruction method. These chapters also expand on the role that constraints play in general to this approach to reconstruction from projections.

3.2 Sinogram MRF and MAP Estimation

Much of this thesis is concerned with estimating smoothed sinograms. In this section we explore one possible sinogram prior probability which takes the form of a Markov random field (MRF) with constraints. We then write the formal expression for the sinogram MAP estimate given noisy and incomplete observations. The remaining sections in this chapter are concerned with algorithms for actually computing these estimates.

3.2.1 A Sinogram MRF

This section specifies the sinogram prior density $p(g)$ as a MRF on sinograms constrained to be in Ω_g, a set to be defined precisely below which contains only sinograms which meet the positivity, mass, and center of mass constraints. Using $p(g)$ and the observation equation (3.14), the form of the MAP estimate $\hat{g}_{map}$ is then derived. The ingredients needed to define a MRF (as outlined in Chapter 2) are: 1) the lattice and graph structure, 2) the potentials functions (or the conditional probabilities), 3) the feasible configurations (the state space), and 4) the boundary conditions. We begin by discussing the lattice.

The Sinogram Lattice

As discussed in Chapter 1, a sinogram is an image of the Radon transform (or measured Radon transform) of an object over the truncated domain $\mathcal{Y}_T$ (see equation (2.7)), with brighter intensities corresponding to larger values of $g(t, \theta)$.

In order to define the MRF, however, we require a finite lattice system, rather than the continuum of points given in $\mathcal{Y}_T$. Therefore, for the purposes of defining the MRF, we define the *sinogram lattice* which is a rectangularly sampled version

of $\mathcal{Y}_T$ given by

$$\mathcal{Y}_S = \{(t,\theta) \mid t = \frac{2T}{n_d} i, i = -\frac{n_d-1}{2}, \ldots, \frac{n_d-1}{2},\ \theta = \frac{\pi}{n_v} j, j = 0, \ldots, n_v\} \qquad (3.1)$$

where n_d is the number of sample points in t, and n_v is the number of sample points in θ. For simplicity, we will always take n_d to be odd so that there is always a sample at $t = 0$, and n_v will always be even with a sample at $\theta = 0$.

For convenience we adopt the following notation for the remainder of this section (refer also to Section 2.2 for terminology). A *site* in the sinogram lattice will be denoted $s = (i,j)$ and the set of all sites by S. A *site value* will be denoted in several ways: $g_s = g_{ij} = g(t_i, \theta_j)$. The collection of all site values will be called the *discrete sinogram* or just the *sinogram* when the meaning is clear from the context. Note, that a site in the sinogram lattice corresponds to a line in the plane passing through the disk D_T. In particular, the site (i,j) corresponds to the line $L(t_i, \theta_j) = \{(x,y) \in \mathbb{R}^2 \mid x\cos\theta_j + y\sin\theta_j = t_i\}$.

The Potential Functions

Pair-Potentials The physics of this problem does not specify for us a prior probability on sinograms. We rely on experimentation and intuition to surmise what a reasonable form for a prior might be, given what we know about the types of objects under consideration and the transformation of those objects via the Radon transform.

After much thought and experimentation, one key idea has driven us to implement what turns out to be the simplest kind of MRF. This idea is simply that sinograms tend to be locally smooth. A prior that produces such sample functions is an MRF with potential functions that prescribe an affinity between nearest neighbors — this is the so-called nearest-neighbor "blob" model [18].[1] Let $s, r \in S$ denote sites that are either vertical or horizontal nearest neighbors. To prescribe affinities between the sinogram values defined on these sites we define the vertical

[1] At one point in the research we considered using an 8-neighbor scheme [69] but abandoned this idea after finding no significant intuitive or experimental advantages.

pair-potential function as

$$V_{(sr)} = b_v(g_s - g_r)^2 \tag{3.2}$$

and the horizontal pair-potential function as

$$V_{\langle sr \rangle} = b_h(g_s - g_r)^2 \tag{3.3}$$

where (sr) and $\langle sr \rangle$ represent pairs of adjacent sites in the vertical and horizontal directions, respectively.[2] The positive constants b_v and b_h allow one to make the vertical and horizontal affinities of different strength, thus making this a non-isotropic random field [44]. In this thesis, we choose the constants b_v and b_h *a priori*; however, it is possible to use the actual data to estimate these coefficients as part of a hierarchical estimation algorithm [27].

The potential functions defined in (3.2) and (3.3) are not the only choices we could have made. This choice of quadratic penalty more strongly penalizes site values which are very different. Geman and McClure [27] suggest that there are benefits to choosing pair-potentials of the form

$$V_{sr} = \frac{-1}{1 + |\Delta g|^2/\tau}$$

where $\Delta g = g_s - g_r$ and τ is a positive constant. This form has the effect that very large outliers (Δg very large) and only moderately large outliers are penalized almost equally. This tends to have the effect of allowing relatively sharp boundaries between regions of strongly different intensity. While this clearly has advantages in standard image processing, we did not feel that this intuitive reasoning had strong enough justification in the sinogram setting to give up the quadratic property that we so strongly exploit in subsequent sections. However, the simulated annealing technique described and implemented in Section 3.3 is sufficiently general to allow the use of any kind of potential function, as we shall see.

Self-Potentials Here, we attempt to take advantage of knowledge of an object's support within the measurement disk. As developed in Chapter 2, the object's

[2] In a sinogram, t is the vertical coordinate, and θ is the horizontal coordinate.

support $\mathcal{F}$ implies a matching region of support $\mathcal{G}$ within the Radon transform domain $\mathcal{Y}_T$ (see Section 2.5) If either set is known exactly then we should insist that sinogram values in the region

$$\bar{\mathcal{G}} = \mathcal{Y}_T - \mathcal{G} \tag{3.4}$$

be exactly zero. However, we shall only assume that an estimate of the sinogram support is available, and that we have a measure of that estimate's reliability. Then the prior we specify seeks to make sinogram values outside the region of support zero, but not as strongly if the support estimate is not reliable. The support self-potential is defined as

$$V_s = \begin{cases} \kappa g_s^2, & (t_i, \theta_j) \in \bar{\mathcal{G}} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

where g_s stands for the value of the sinogram at site s=(i,j), and κ is a positive constant which is used to reflect the support measurement's reliability. In this chapter, κ will be set very high, since only known support functions will be used in the experiments. In Chapter 7, however, we will indicate how κ may be chosen as part of a hierarchical estimation scheme, the first part of which is support estimation.

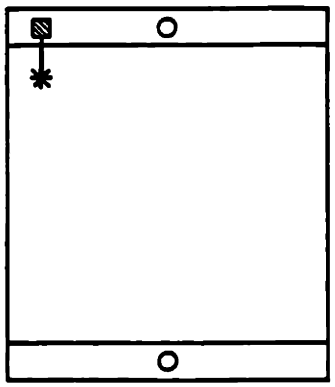
The Graph Structure

Neighborhoods The form of the potential functions described above dictate the required neighborhood structure. In fact, only nearest neighbors are required to accommodate both the pair-potentials and self-potentials described above. For the nearest neighbor construct, the most general form of the MRF energy function is

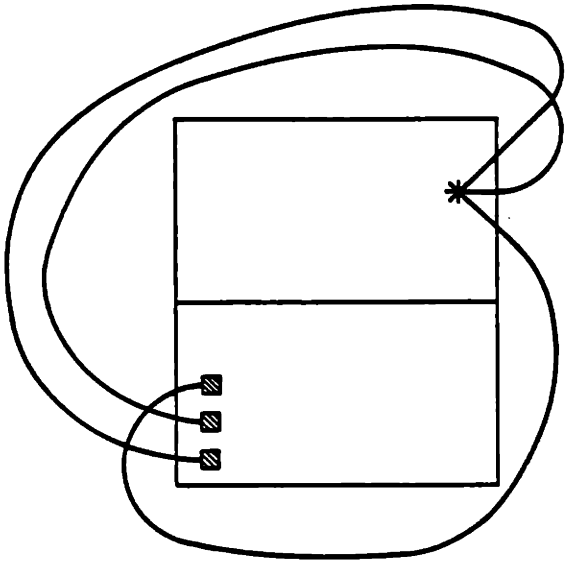
$$U(g) = \sum_{s \in C_1} V_s(g_s) + \sum_{(s,r) \in C_2} V_{sr}(g_s, g_r) \tag{3.6}$$

since the largest clique has only two sites. Having defined both the pair and self potentials, then, except for the boundary conditions and the state space, the form of the MRF is completely specified.

Boundaries Having chosen the neighborhood structure, we must consider what happens to sites near the boundaries (see Fig. 3.1). Since all objects are known to be



(a)



(b)

Figure 3.1: (a) The vertical and (b) horizontal boundaries of the sinogram MRF.

zero outside the disk of radius T, the boundary value above and below the sinogram must be zero. However, the boundaries to the left and right of the sinogram must be treated differently. Here we use the fundamental property of the Radon transform stated in Chapter 1

$$g(t,\theta) = g(-t,\theta+\pi)$$

to conclude that the neighbors wrap around in a toroid that is twisted or flipped about the t-axis (see Fig. 3.1b) — the sinogram is actually defined on a Mobius strip.

In order to compute the potential functions for sites defined near the boundary, we will sometimes require the site value g_{kl} where k and l are not necessarily in the ranges $1 \leq k \leq n_d$ and $1 \leq l \leq n_v$. While, technically, we have not defined any sites in the lattice outside this range, and thus no site values, the boundary conditions allow us to make the following identification, which summarizes the site value computation:

$$g_{kl} = \begin{cases} g_{kl}\,, & 1 \leq k \leq n_d \text{ and } 1 \leq l \leq n_v \\ 0\,, & k < 1 \text{ or } k > n_d\,, \text{any } l \\ g_{k,\, l \bmod n_v}\,, & 1 \leq k \leq n_d \text{ and } \lfloor (l-1)/n_v \rfloor \text{ even} \\ g_{n_d+1-k,\, l \bmod n_v}\,, & 1 \leq k \leq n_d \text{ and } \lfloor (n-1)/n_v \rfloor \text{ odd} \end{cases} \tag{3.7}$$

where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x, and $l \bmod n_v$ indicates l evaluated modulo n_v so that the result is in the range $1, \ldots, n_v$. This equation is used directly in the simulated annealing algorithm proposed in Section 3.4 and in the local relaxation algorithm of Section 3.5, and is used indirectly in deriving the form of the Hessian matrix for the quadratic programming method of Section 3.3.

Sinogram Constraints

As discussed previously, we intend to impose mass, center of mass, and positivity as constraints on the space of feasible sinograms, Ω_g. Hence, we assume that the object is non-negative, and that we know its mass and the position of its center of mass. Furthermore, we make the assumption that in a preprocessing stage (see

Chapters 7 and 8), the sinogram was adjusted so that the object's center of mass is at the origin.

The positivity constraint is obvious: an object with non-negative values produces a sinogram with non-negative values, or

$$g_{ij} \geq 0 \quad 1 \leq i \leq n_d, \; 1 \leq j \leq n_v \tag{3.8}$$

The mass and center of mass constraints are discrete approximations to the integral expressions of (2.17) and (2.18) from Chapter 2. Thus letting m denote the object's mass we have

$$\frac{2T}{n_d} \sum_{i=1}^{n_d} g_{ij} = m \quad \forall j, \; 1 \leq j \leq n_v \tag{3.9}$$

and

$$\frac{1}{m} \frac{2T}{n_d} \sum_{i=1}^{n_d} t_i g_{ij} = 0 \quad \forall j, \; 1 \leq j \leq n_v \tag{3.10}$$

where

$$t_i = \frac{2T}{n_d}(i - \frac{n_d + 1}{2}) \tag{3.11}$$

is the lateral position of the ith line.

The set of feasible discrete sinograms, Ω_g, sometimes called the state space (for reasons that become apparent in Section 3.4), consists of all real matrices of dimension n_d by n_v which satisfy (3.8), (3.9), and (3.10). Of course, when one views the discrete sinogram as a vector, the constraints may also be written in explicit vector form (see Section 3.3). We see that each constraint is linear, the positivity constraint being a linear *inequality* constraint, the others linear *equality* constraints. The presence of constraints does not make it any harder to formally define the MRF since the effect of the constraints is felt primarily in the partition function Z, which we normally do not need to compute. But the constraints make the computation of the MAP estimate more difficult since the algorithms must be constrained optimization methods, which are generally more complicated and time consuming than unconstrained methods [56]. Sections 3.3 and 3.4 describe different algorithms which incorporate all three constraints, while Section 3.5 only uses the two equality constraints.

3.2.2 The MAP Formulation

The Gibbs Prior

Having now defined all the elements of the MRF it is a simple matter to write down the joint probability density for the discrete sinogram prior. Clearly, the prior is given by

$$p(g) = \frac{1}{Z} e^{-U(g)} \quad g \in \Omega_g \tag{3.12}$$

where g stands for the vector of sinogram site values and Z is given by

$$Z = \int_{g \in \Omega_g} e^{U(g)} \, dg$$

so that $p(g)$ integrates to one. The function $U(g)$ is the energy function defined in (3.6).

The Sinogram MAP Estimate

We assume that noisy observations of the true site values are available over a (possibly) limited- or sparse-angle subset $\mathcal{Y}_O$ of $\mathcal{Y}_T$ and that the observations are given by

$$y_{ij} = g_{ij} + n_{ij} \quad (t_i, \theta_j) \in \mathcal{Y}_O \tag{3.13}$$

where the n_{ij} are independent zero-mean Gaussian random variables with variance σ^2. Letting g stand for the vector of true sinogram site values and letting y and n stand for the vector of observations and noise samples, respectively, we may write the observation equation in vector form as

$$y = Sg + n \tag{3.14}$$

where S is a matrix that *selects* the observations as follows. In the measurement configuration we consider, a column of the matrix given by $[g_{ij}]$ is either observed completely (in additive noise) or not observed at all. Suppose, for the purposes of this discussion only, we form g by stacking the columns of $[g_{ij}]$, stacking all the *observed* columns first from the top proceeding downwards, and the remaining

columns following in any order. Then, denoting the number of observed columns by n_o, we see that S is given by

$$S = [\, I \,|\, 0 \,]$$

where I is the $n_o n_d \times n_o n_d$ identity matrix. Clearly, the length of each of the vectors y and n is given by $n_o n_d$.

Now, having an observation equation given by (3.14) and a prior probability on g given by (3.12), we may now derive the form of the MAP estimate $\hat{g}_{map}$. Denoting the noise covariance matrix by K_n $(= \sigma^2 I)$ we may use (3.14) to write the conditional measurement density (zero mean, jointly Gaussian) as

$$p(y|g) = |2\pi K_n|^{-1/2} \exp\left(-\frac{1}{2}(y - Sg)^T K_n^{-1}(y - Sg)\right). \tag{3.15}$$

Then given the sinogram prior of (3.12) and the definition of conditional probability we may write the joint probability of the sinogram and the observations as

$$\begin{aligned} p(g,y) &= p(y|g)p(g) \\ &= \frac{|2\pi K_n|^{-1/2}}{Z} \exp\left(-\left(\frac{1}{2}(y - Sg)^T K_n^{-1}(y - Sg) + U(g)\right)\right) \\ &= \frac{1}{Z'} \exp\left(-U'(g,y)\right) \quad . \end{aligned} \tag{3.16}$$

Clearly, the joint density is also a Gibbs density as suggested by the use of a new constant of integration Z' and joint energy function $U'(g,y)$. It is also easy to see that the posterior density is a Gibbs density given by

$$\begin{aligned} p(g|y) &= p(g,y)/p(y) \\ &= \frac{|2\pi K_n|^{-1/2}}{Z\, p(y)} \exp(-(\frac{1}{2}(y - Sg)^T K_n^{-1}(y - Sg) + U(g))) \\ &= \frac{1}{Z_p} \exp(-U'(g,y)) \end{aligned} \tag{3.17}$$

the only change being the new partition function Z_p. Because of the fact that our observation equation is not convolutional, the graph structure corresponding the posterior MRF is identical to the prior [26]. The only significant change between

the two MRF's is the form of the energy function, which now contains a self potential term which couples the observations y to the sinogram g.

The MAP estimate, which maximizes either (3.16) or (3.17) with the observations substituted into the expressions (see Section 2.3), is clearly that g which minimizes U'. Hence, the MAP estimate is given by

$$\hat{g}_{map} = \underset{g \in \Omega_g}{\operatorname{argmin}} U'(g, y = Y) \tag{3.18}$$

$$= \underset{g \in \Omega_g}{\operatorname{argmin}} \frac{1}{2\sigma^2}(Y - Sg)^{\mathrm{T}}(Y - Sg) + U(g) \tag{3.19}$$

where we have used the fact that $K_n = \sigma^2 I$.

In the following section we solve the MAP problem of (3.19) exactly using large scale quadratic programming (QP) techniques. After demonstrating the extreme computational burdens of the QP approach we proceed to describe a very general iterative relaxation method, namely simulated annealing (SA), which ties in very nicely with the Gibbs formulation and is a highly parallelizable algorithm. Finally, in Section 3.5 we rewrite the original problem without the positivity constraint using a continuous variational formulation with constraints. The solution of the resulting partial differential equation using Kuo's local relaxation technique [47] yields an extremely efficient solution to the estimation problem. Following the development of these three algorithms, we have a large section of experimental results. The experiments compare the algorithms against each other and against CBP under a wide variety of situations including varying signal to noise ratio, different incomplete observation scenarios, different sinogram resolutions, and different objects. Furthermore we will explore the effects of changing b_h and b_v, the two smoothing parameters introduced in this section, the weighting κ, and the effect of incorrect support.

3.3 Quadratic Programming Algorithm

The sinogram MAP estimate given in (3.19) may be found exactly by solving the quadratic programming (QP) problem

$$\text{(Q)} \qquad \begin{aligned} &\text{minimize} && \frac{1}{2}g^{\mathrm{T}}Gg + p^{\mathrm{T}}g \\ &\text{subject to} && Ag = b \text{ and } Bg \geq c \end{aligned}$$

In the following subsection we reveal the precise form of the matrices G, A, and B, and the vectors b, c, and p. Then, the solution of (Q) using ZQPCVX (see Section 2.7) is discussed and experimental results shown. The intent of this section is to show that a formal solution leading to an exact answer is possible but impractical. The formalism developed herein, however, is used in subsequent sections.

3.3.1 The QP Formulation

Let g be the column vector made by stacking the columns of the sinogram, with the rightmost column on the bottom. Let us first examine the constraints on g. The mass constraint of (3.9) may be be written in matrix form as

$$A_1 g = b_1 \tag{3.20}$$

where

$$A_1 = \begin{bmatrix} e^{\mathrm{T}} & & 0 \\ & e^{\mathrm{T}} & \\ 0 & & \ddots \\ & & & e^{\mathrm{T}} \end{bmatrix} \tag{3.21}$$

$$b_1 = m\frac{n_d}{2T}e \tag{3.22}$$

where e is the vector of dimension n_d containing all 1's, and m is the mass. The center of mass constraint given in (3.10) may be written in matrix form as

$$A_2 g = 0 \tag{3.23}$$

where

$$A_2 = \begin{bmatrix} t^{\mathrm{T}} & & 0 \\ & t^{\mathrm{T}} & \\ 0 & & \ddots \\ & & & t^{\mathrm{T}} \end{bmatrix} \tag{3.24}$$

Self Potentials	1. Support	$V_s = \begin{cases} 0 & (t_i, \theta_j) \in \mathcal{G} \\ \kappa g_s^2 & (t_i, \theta_j) \notin \mathcal{G} \end{cases}$
	2. Observations	$V_s = \begin{cases} \frac{1}{2\sigma^2}(y_s - g_s)^2 & (t_i, \theta_j) \in \mathcal{Y}_O \\ 0 & \text{otherwise} \end{cases}$
Pair Potentials	3. Vertical	$V_{\langle sr \rangle} = b_v(g_s - g_r)^2$
	4. Horizontal	$V_{\langle sr \rangle} = b_h(g_s - g_r)^2$

Table 3.1: Potential Functions in the Posterior Density.

where t is the vector of detector positions $t = [t_1 t_2 \ldots t_{n_d}]^T$ where t_i is given in (3.11).

The two equality constraints may be combined to form the QP linear equality constraint

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} g = \begin{bmatrix} b_1 \\ 0 \end{bmatrix} \tag{3.25}$$

The positivity constraint, which forms the only QP linear inequality constraint, is easily written as

$$Ig \geq 0 \tag{3.26}$$

which completes the specification of the vectors and matrices involved in the constraints.

In order to determine the form of the matrices in the objective function, it is useful to recall the terms that make up the posterior energy function (which is the objective function). Table 3.1 identifies the four different terms. To simplify the notation in the sequel we define the following indicator function notations

$$\bar{\chi}_G(t, \theta) = \begin{cases} 1 & (t, \theta) \in \bar{\mathcal{G}} \\ 0 & \text{otherwise} \end{cases} \tag{3.27}$$

and

$$\chi_Y(t,\theta) = \begin{cases} 1 & (t,\theta) \in y_O \\ 0 & \text{otherwise} \end{cases} \tag{3.28}$$

Then, by expanding the terms in the Table 3.1 and incorporating the MRF boundary conditions it is not difficult to see that the QP objective function may be written as

$$f(g) = \frac{1}{2} g^{\mathrm{T}} G g + p^{\mathrm{T}} g + k \tag{3.29}$$

where the square matrix G is given by

$$G = \left[\begin{array}{c|c|c|c} \begin{matrix} q_{1,1} & v & & \\ v & q_{2,1} & \ddots & \\ & \ddots & \ddots & v \\ & & v & q_{n_d,1} \end{matrix} & \begin{matrix} u & & & \\ & u & & \\ & & \ddots & \\ 0 & & & u \end{matrix} & & \begin{matrix} & & & u \\ & & u & \\ & \iddots & & \\ u & & & \end{matrix} \\ \hline \begin{matrix} u & & 0 & \\ & u & & \\ & & \ddots & \\ & & & u \end{matrix} & \begin{matrix} q_{1,2} & v & & \\ v & q_{2,2} & \ddots & \\ & \ddots & \ddots & v \\ & & v & q_{n_d,2} \end{matrix} & \begin{matrix} \ddots \end{matrix} & \\ \hline & \begin{matrix} \ddots \end{matrix} & \begin{matrix} \ddots \end{matrix} & \begin{matrix} u & & \\ & u & \\ & & \ddots \\ 0 & & u \end{matrix} \\ \hline \begin{matrix} & & u \\ & u & \\ \iddots & & \\ u & & \end{matrix} & & \begin{matrix} u & 0 \\ & u \\ & & \ddots \\ & & & u \end{matrix} & \begin{matrix} q_{1,n_v} & v & & \\ v & q_{2,n_v} & \ddots & \\ & \ddots & \ddots & v \\ & & v & q_{n_d,n_v} \end{matrix} \end{array}\right] \tag{3.30}$$

and

$$\begin{aligned} q_{i,j} &= 2\left(\kappa \bar{X}_G(t_i,\theta_j) + \frac{1}{2\sigma}\chi_Y(t_i,\theta_j) + 2b_v + 2b_h\right) \\ v &= -2b_v \text{ and} \\ u &= -2b_h \ . \end{aligned}$$

The remainder of $f(g)$ is determined by specifying

$$p = -\frac{1}{\sigma^2} y$$

and

$$k = \frac{1}{2\sigma^2} y^T y \ .$$

The constant k doesn't affect the outcome of the minimization, however, so it is dropped from the expression in (Q).

It is important to see how the boundary conditions are satisfied by the terms given above. The boundaries above and below are zero. This is incorporated by the presence of an extra b_v term at both corners of each diagonal block with the absence of any corresponding cross term; therefore, terms such as $b_v(g_s - g_r)^2$ where $s \in S$ and r is above or below the lattice has $g_r = 0$ implied. The twisted boundaries to the left and right of the lattice are incorporated by the doubly symmetric diagonal blocks on the top right and bottom left.

3.3.2 Implementation

The minimization problem (Q) is a quadratic program and, in principle, may be solved in finite time by standard quadratic programming methods [56],[28]. However, these methods cannot, in general, take advantage of the sparsity or special structure of the Hessian matrix G because of the requirement of forming an exact factorization of G, which is usually not sparse. For this reason, the memory requirements of a finite time QP code effectively determine the largest size problem that can be solved in this fashion. In Appendix 3.C we analyze the memory and time requirements of ZQPCVX (see Chapter 2) for five different sinogram sizes. It is obvious that the requirements for all but the smallest sinogram are prohibitive. Therefore, we have experimental results using finite time QP codes only for sinograms with $n_d = 41$ and $n_v = 30$. The results of several experiments are given in Section 3.6.

3.4 Simulated Annealing Algorithm

In this section we show how $\hat{g}_{map}$ of (3.19) may be found using simulated annealing (SA). The essential features of SA have been presented in Chapter 2, so that we present here only those aspects which are significantly different. As we shall see, the primary difference is in the handling of constraints, which results in a different Metropolis update procedure. It is important to point out once again that SA is a very general optimization procedure which is well-suited to problems with very complicated objective functions which are often non-differentiable and filled with local minima. In contrast, the sinogram MAP objective function in (3.19) (or (Q) of section 3.3) has only a single local minimum — the global minimum. And furthermore, the objective function in this case is, in fact, differentiable. SA, unlike the QP algorithm of the previous section and the local relaxation algorithm presented in the next section, does not exploit these facts, and therefore, would not be the first choice to solve the given problem. This development of SA is valuable in three respects, however. First, and most importantly, SA will still work with almost no changes for any case where the objective function is made non-differentiable and/or full of local minima. We could produce this situation by making relatively minor changes to the pair potentials presented in Section 3.2. Second, the fact that constraints are an inherent part of our problem requires the implementation of a modified Metropolis update which we develop below. While this modification is realized simply as a more complicated state-space move, it is done in such a way as to make the increased computation burden as small as possible. Third, SA in general is a highly parallelizable algorithm [26], and requires only a small amount of memory. We point out, however, that the addition of constraints, which changes the nature of the state-space update, also makes the structure of a parallel algorithm more difficult to deduce. We do not discuss this issue, however.

3.4.1 Constrained Metropolis Algorithm

Because of the presence of constraints in our problem, the ordinary Metropolis algorithm is not adequate. We present the modified version in this section. The

problem with the ordinary Metropolis update may be demonstrated by considering updating a single site $s = (i, j)$. Let the current site value be g_{ij} and the proposed new site value be $\tilde{g}_{ij} \neq g_{ij}$. Since the current sinogram is feasible, the proposed change will result in non-feasible sinogram since the mass constraint (3.9) cannot be maintained. In fact, it is clear that at least two site values per column must be changed simultaneously in order to satisfy the mass constraint; one site may add some mass but another must subtract an identical amount. But we go one step further — the center of mass constraint requires that a *third* site be included as well. Thus at each site replacement, a triplet of sites from a single column must be simultaneously updated. Let us look at this procedure in more detail.

Suppose sites q, r, and s belong to a single column in the sinogram; let the current site values be denoted by g_q, g_r, and g_s, respectively. The goal is to specify a random perturbation rule that satisfies the Metropolis reversibility requirement and that produces a feasible sinogram. Let the three proposed site values be denoted by $\tilde{g}_q$, $\tilde{g}_r$, and $\tilde{g}_s$. Then in order to maintain both the mass and center of mass constraints we must have that

$$\tilde{g}_q + \tilde{g}_r + \tilde{g}_s = m = g_q + g_r + g_s \tag{3.31}$$

and

$$t_q\tilde{g}_q + t_r\tilde{g}_r + t_s\tilde{g}_s = c = t_q g_q + t_r g_r + t_s g_s \tag{3.32}$$

where the t's are the t coordinates of the corresponding sites. Note that m and c are the mass and center of mass of just these three sites, not the entire projection, and are in fact calculated directly from the current site values. These two constraints may be written in vector form as

$$g^{\mathrm{T}}e = m \tag{3.33}$$

$$g^{\mathrm{T}}t = c \tag{3.34}$$

where the vector g denotes either vector $g = [g_q, g_r, g_s]^{\mathrm{T}}$ or $\tilde{g} = [\tilde{g}_q, \tilde{g}_r, \tilde{g}_s]^{\mathrm{T}}$, e is the vector of ones, and $t = [t_q, t_r, t_s]^{\mathrm{T}}$. Forming the difference vector $\Delta g = \tilde{g} - g$, it follows that

$$\Delta g^{\mathrm{T}}e = 0 \tag{3.35}$$

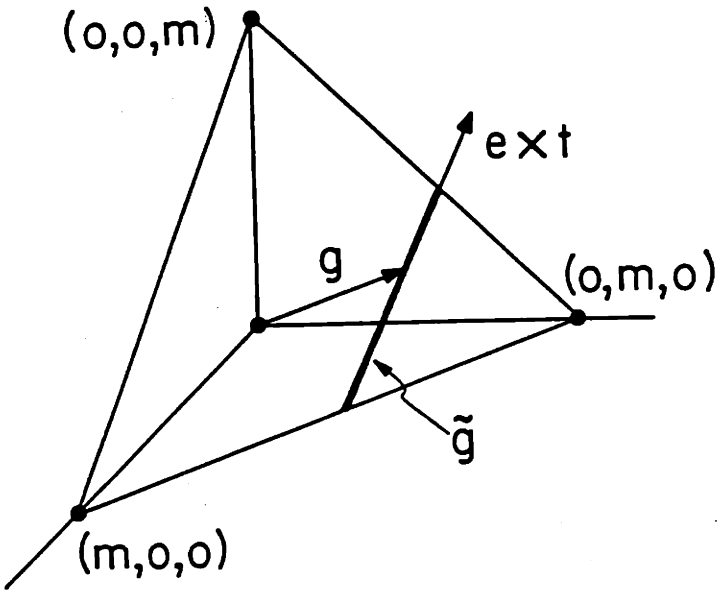


Figure 3.2: The geometry of the 3-site update.

$$\Delta g^T t = 0 . \tag{3.36}$$

Since e and t are linearly independent, then Δg is determined up to a multiplicative constant α by the cross product

$$\Delta g = \alpha(e \times t). \tag{3.37}$$

But one more constraint is necessarily imposed: non-negativity. This constraint restricts the range of multipliers α that will produce non-negative proposals.

Thus, in order to produce a feasible proposal, we must first calculate m and c, the cross product $e \times t$, and the range of feasible α's. Then, we randomly choose a multiplier from the range of feasible multipliers, which completely determines the three proposed site values.

Fig. 3.2 shows the geometry of the three-site update. The proposal (that is, the random perturbation of the three site values) we seek, vector $\tilde{g}$, must lie in the triangle shown because the three-site mass m cannot change, and because $\tilde{g}$ must have positive entries. Furthermore, it must lie on the line $g + \alpha(e \times t)$ because the

three-site center of mass c must remain constant. Thus $\tilde{g}$ must lie on the darkened line segment.

What remains is to specify the random rule used to select α. There are two basic approaches to choosing the random perturbation. The first approach is simply to choose $\tilde{g}$ randomly from along the feasible line segment, ignoring the current value of g. If this approach is used, a uniform probability rule over the line segment must be used in order to satisfy the Metropolis reversibility requirement. As Metropolis et. al. pointed out [60], this type of rule has the potential to waste a tremendous amount of time in the computations since there may be a good chance that the proposal will differ strongly from the current value. This is of even greater concern in SA, when as one lowers the temperature, one expects to make fewer and fewer large state-space moves.

This problem may be avoided by taking a second approach to the random perturbation: choose $\tilde{g}$ to be a local perturbation about the current value g. In this case we center a line segment of some length (presumably smaller that the length of the feasible line segment) about the current point g, and choose a $\tilde{g}$ with uniform probability from the intersection of the two line segments. (Again the uniform probability rule is required in order to satisfy reversibility.) The edge effects may be satisfied by either wrap-around or wrap-back. This method has the advantage that the perturbations can be made smaller (by selecting progressively shorter overlaying line segments) as the temperature decreases, thus improving the efficiency of the computations.

The fact that three sites must be updated simultaneously raises another question: how does one choose the sites for updating? We elected to use a part raster, part random rule. We sequence through the sinogram in raster fashion. At each site we randomly (uniform probability) choose two "co-sites" from the current column, making sure that three distinct sites are selected. Theoretically, even this rule violates the Metropolis reversibility requirement, but two effects make this violation practically negligible. First, there is a random element to the selection, and second, there is a large number of sites, which decreases the effect of non-reversibility (see [59]).

3.4.2 Initialization

We must also specify the initial configuration prior to the first iteration. Since each update proceeds from feasible sinogram to feasible sinogram, we must clearly begin with a feasible sinogram. Other than that requirement, it is irrelevant what particular sinogram is used, provided that the initial temperature is high enough. Accordingly, we begin with a sinogram which is constant everywhere, the value at each site being given by $g_{ij} = n_d m/2T$. Clearly, from (3.9) and (3.10) we see that the initial sinogram has the correct mass and center of mass, and furthermore, it is clearly non-negative.

3.5 Local Relaxation Algorithm

This section develops the theory and implementation of a fast iterative non-stochastic algorithm for computing $\hat{g}_{map}$. Here is an overview of the stages to be detailed below. The key step in developing this method is to write the vector minimization problem of (3.19) as a minimization problem involving an unknown function $g(t,\theta)$ over the *continuous* domain $\mathcal{Y}_T$. This new minimization problem is then solved analytically using variational calculus techniques in which Lagrange multipliers are used to account for the equality constraints. For this section only, we drop the positivity constraint, thus possibly allowing solutions to have negative numbers. The necessary conditions for a solution to the minimization problem are in the form of a partial differential equation (PDE), with constraints, having as unknowns both the Radon transform function $g(t,\theta)$ and the Lagrange multiplier functions. But the Lagrange multiplier functions may be solved for analytically, leaving a more complicated integro-differential equation (IDE) which depends only on the function $g(t,\theta)$. At this stage, an approximation which is valid near the solution produces an approximate PDE which has a form which may be solved numerically after suitable discretization. We implement Kuo's local relaxation algorithm [47], a very efficient method which may be implemented in parallel, to solve this PDE. Finally, we discuss cases where the aforementioned approximations are not valid, and describe Lagrange multipler update methods to compensate for this inadequacy.

3.5.1 Variational MAP Formulation and Solution

The Minimization Problem

Consider the problem, which we refer to as (V), of minimizing

$$I = \iint_{y_o} \frac{1}{2\sigma^2}(y-g)^2\,dt\,d\theta + \iint_{\mathcal{G}} \kappa g^2\,dt\,d\theta + \iint_{y_T} \left[\beta\left(\frac{\partial g}{\partial t}\right)^2 + \gamma\left(\frac{\partial g}{\partial \theta}\right)^2\right] dt\,d\theta \tag{3.38}$$

subject to the equality constraints

$$\begin{aligned} J_1 &= m = \int_{-\mathrm{T}}^{\mathrm{T}} g(t,\theta)\,dt \\ J_2 &= 0 = \frac{1}{m}\int_{-\mathrm{T}}^{\mathrm{T}} t g(t,\theta)\,dt \end{aligned} \tag{3.39}$$

and boundary conditions

$$\begin{aligned} g(\mathrm{T},\theta) &= g(-\mathrm{T},\theta) = 0 \\ g(t,0) &= g(-t,\pi) \end{aligned} \tag{3.40}$$

where κ, β, and γ are positive constants. This problem is a continuous formulation of the sinogram MAP problem of (3.19). Clearly, the first term in I is analogous to the first term in (3.19) — both represent a penalty which seeks to keep the estimate close to the observations. The second two terms are analogous to the two terms of $U(g)$, given in (3.6). There is a term comprising the support information — this matches the self-potential clique summation of $U(g)$. And finally, there is the integral which contains two terms involving the square of the two partial derivatives of g — this term corresponds to the summation of the pair-potential terms in $U(g)$. The two integral constraints in (3.39) are exactly the mass and center of mass contraints. Finally, the boundary conditions, which include the twisted boundary, are stated in (3.40).

Using the indicator functions for the observations, $\chi_Y(t,\theta)$, and support, $\bar{\chi}_G(t,\theta)$, as defined in Equations (3.28) and (3.27) respectively, we may write I using a single double integral sign as

$$I = \iint_{y_T} \kappa\bar{\chi}_G g^2 + \beta\left(\frac{\partial g}{\partial t}\right)^2 + \gamma\left(\frac{\partial g}{\partial \theta}\right)^2 + \frac{1}{2\sigma^2}\chi_Y(y-g)^2\,dt\,d\theta\ . \tag{3.41}$$

The problem is now in the form of a classical variational problem which may be solved using standard variation calculus techniques (see [36], for example).

Partial Differential Equation

A necessary condition for $g(t,\theta)$ to be a solution to (V) is that it satisfy the following second order partial differential equation (PDE) (see complete development in Appendix 3.A)

$$\left(2\kappa\bar{X}_G + \frac{1}{\sigma^2}\chi_Y\right) g - 2\beta\frac{\partial^2 g}{\partial t^2} - 2\gamma\frac{\partial^2 g}{\partial \theta^2} = \frac{1}{\sigma^2}\chi_Y y - \lambda_1(\theta) - \lambda_2(\theta)t \tag{3.42}$$

and the additional boundary condition

$$\frac{\partial g(t,0)}{\partial t} = \frac{\partial g(-t,\pi)}{\partial t} \ . \tag{3.43}$$

In addition, $g(t,\theta)$ must satisfy the original constraints and boundary conditions. It is important to note that (3.42) contains three unknown functions: $g(t,\theta)$, and two Lagrange multiplier functions $\lambda_1(\theta)$ and $\lambda_2(\theta)$ (one for each constraint). To simplify the expressions in the remainder of this section we use the notation g_t and g_{tt} to stand for the first and second partial derivatives of $g(t,\theta)$ with respect to t, respectively, and g_θ and $g_{\theta\theta}$ to stand for the first and second partial derivatives of $g(t,\theta)$ with respect to θ, respectively.

Now we have that the solution must satisfy (3.42) in addition to the constraints (3.39) and boundary conditions (3.40) and (3.43). But as it stands, (3.42) cannot be solved without first determining λ_1 and λ_2. It turns out that λ_1 may be determined analytically by integrating both sides of (3.42) and simplifying; λ_2 may be determined by first multiplying both sides by t, then integrating and simplifying. The results of this work (detailed in Appendix 3.B) are

$$\lambda_1(\theta) = \frac{-1}{2T}\left[\int_{-T}^{T} 2\kappa\bar{X}_G g\,dt - 2\beta g_t\Big|_{-T}^{T} + \frac{m}{\sigma^2}\chi_Y - \frac{1}{\sigma^2}\int_{-T}^{T}\chi_Y y\,dt\right] \tag{3.44}$$

$$\lambda_2(\theta) = \frac{-3}{2T^3}\left[\int_{-T}^{T} 2t\kappa\bar{X}_G g\,dt - 2\beta t g_t\Big|_{-T}^{T} - \frac{1}{\sigma^2}\int_{-T}^{T} t\chi_Y y\,dt\right] \tag{3.45}$$

which when substituted into (3.42) leaves an integro-differential equation in a single unknown (function), $g(t,\theta)$, given by

$$\left(2\kappa\bar{X}_G + \frac{1}{\sigma^2}\chi_Y\right) g - 2\beta g_{tt} - 2\gamma g_{\theta\theta} = \tag{3.46}$$
$$\frac{1}{\sigma^2}\chi_Y y - \frac{1}{2T}\left(\int_{-T}^{T}\frac{1}{\sigma^2}\chi_Y y\,dt - \int_{-T}^{T} 2\kappa\bar{X}_G g\,dt - \frac{m}{\sigma^2}\chi_Y + 2\beta g_t(T,\theta) - 2\beta g_t(-T,\theta)\right)$$
$$- \frac{3t}{2T^3}\left(\int_{-T}^{T}\frac{1}{\sigma^2}t\chi_Y y\,dt - \int_{-T}^{T} 2t\kappa\bar{X}_G g\,dt + 2\beta t g_t(T,\theta) - 2\beta t g_t(-T,\theta)\right).$$

Approximations

Fortunately, the integral terms involving g on the right-hand side of (3.46) may often be suitably approximated at or near the solution. Near the solution, we expect that $g(t,\theta) \approx 0$ for $(t,\theta) \in \bar{\mathcal{G}}$, especially when κ is large. Hence, we may make the approximations

$$\int_{-T}^{T} 2\kappa\bar{X}_G g\,dt \approx 0 \quad \text{and} \quad \int_{-T}^{T} 2t\kappa\bar{X}_G g dt \approx 0.$$

Furthermore, the terms in (3.46) involving the partial derivative g_t evaluated at $\pm T$ may often be taken to be zero. This assumption is not valid for large β, nor is it valid for angles θ for which there are no observations. With the above substitutions, the integro-differential equation of (3.46) becomes a PDE given by

$$\left(2\kappa\bar{X}_G + \frac{1}{\sigma^2}\chi_Y\right) g - 2\beta g_{tt} - 2\gamma g_{\theta\theta} =$$
$$\frac{1}{\sigma^2}\chi_Y y - \frac{1}{2T\sigma^2}\int_{-T}^{T}\chi_Y y\,dt + \frac{m}{2T\sigma^2}\chi_Y - \frac{3t}{2T^3\sigma^2}\int_{-T}^{T} t\chi_Y y\,dt \tag{3.47}$$

which is an approximate version of the original PDE of (3.42). This PDE, however, unlike the original, has a single unknown, $g(t,\theta)$, and may be solved numerically by any of several techniques for solving elliptic partial differential equations. In the cases where the above approximations are not valid, the solution to (3.47) will not meet the required constraints. This is because the approximations we made above essentially attempted to estimate the final value of the Lagrange multipliers, and when this estimate is incorrect the constraints will not be met. Therefore, after a discussion of the numerical methods required to solve (3.47), we will discuss the

implementation of a Lagrange multiplier update method which is incorporated if the solution to (3.47) does not meet the constraints.

3.5.2 Numerical Methods

The first stage in any numerical method to solve an elliptic PDE is to discretize the equation. As in previous sections $g(t,\theta)$ and $y(t,\theta)$ will be discretized evenly in t over the range $[-T,T]$ using n_d samples (n_d is odd with one sample at $t=0$), and evenly in θ over the range $[0,\pi)$ using n_v samples (n_v is even with a sample at $\theta=0$). This describes a rectilinear grid with different vertical and horizontal sample spacing given by $\Delta_t = 2T/n_d$ and $\Delta_\theta = \pi/n_v$, respectively. The usual approximations to the second partial derivatives of g are then given as

$$\begin{aligned} g_{tt} &\approx \frac{g_{i+1,j} - 2g_{i,j} + g_{i-1,j}}{\Delta_t^2} \\ g_{\theta\theta} &\approx \frac{g_{i,j+1} - 2g_{i,j} + g_{i,j-1}}{\Delta_\theta^2} \quad . \end{aligned}$$

Then we have

$$2\beta g_{tt} + 2\gamma g_{\theta\theta} \approx \frac{2\beta}{\Delta_t^2}\left(g_{i+1,j} - 2g_{i,j} + g_{i-1,j}\right) + \frac{2\gamma}{\Delta_\theta^2}\left(g_{i,j+1} - 2g_{i,j} + g_{i,j-1}\right) \quad .$$

Since there are no other terms on either side of (3.47) which require explicit use of the grid spacings Δ_t and Δ_θ (other than to approximate the values of the integrals involving y), it is convenient to define new constants, $\hat{\beta}$ and $\hat{\gamma}$, as

$$\hat{\beta} = \frac{\beta}{\Delta_t^2} \quad \text{and} \quad \hat{\gamma} = \frac{\gamma}{\Delta_\theta^2} \quad .$$

Then the PDE of (3.47) may be approximated at an interior point by the finite difference equation[3] [47]

$$d_{i,j}g_{i,j} - r_{i,j}g_{i+1,j} - l_{i,j}g_{i-1,j} - t_{i,j}g_{i,j+1} - b_{i,j}g_{i,j-1} = s_{i,j} \tag{3.48}$$

where

$$l_{i,j} = 2\hat{\beta} \tag{3.49}$$

[3] This approximation is called a 5-point stencil [47].

$$
\begin{aligned}
r_{i,j} &= 2\hat{\beta} \\
b_{i,j} &= 2\hat{\gamma} \\
t_{i,j} &= 2\hat{\gamma} \\
d_{i,j} &= 4\hat{\beta} + 4\hat{\gamma} + \left(2\kappa\bar{\chi}_G + \frac{1}{\sigma^2}\chi_Y\right)\Big|_{(t_i,\theta_j)} \\
s_{i,j} &= \left(\frac{1}{\sigma^2}\chi_Y y - \frac{1}{2T\sigma^2}\int_{-T}^{T}\chi_Y y\,dt + \frac{m}{2T\sigma^2}\chi_Y - \frac{3t}{2T^3\sigma^2}\int_{-T}^{T} t\chi_Y y\,dt\right)\Big|_{(t_i,\theta_j)}
\end{aligned}
$$

Equation (3.48) is also valid for boundary points when the computations for the boundary points use the equation (3.7) given in Section 3.2.1.

It is clear that one may interpret the set of equations given by (3.48) for all j, $j = 1,\ldots,n_d$ and i, $i = 1,\ldots,n_v$ as a vector equation

$$Ag = s \ . \tag{3.50}$$

In fact,if κ, β, and γ are chosen properly, the matrix A is exactly the matrix G of the QP solution. We shall have more to say on the relationship between the two problems in the discussion at the end of this section.

Local Relaxation

Several traditional methods (c.f. [93]) including Jacobi, simultaneous over-relaxation (SOR), and Chebyshev semi-iterative relaxation methods may be employed to solve (3.50). However, we have chosen to implement a relatively new method due to Kuo, Levy, and Musicus [47] which has been shown to have very favorable convergence properties, and is relatively easy to implement. This method, in addition, has been shown to be ideally suited for parallel implementation.

Our implementation of Kuo's local relaxation algorithm (KLR) follows [47] exactly; we have included the basic equations in Appendix 3.D. One point related to convergence is worthy of comment. In the initialization phase, KLR calculates an array of *local relaxation parameters*, $\omega_{i,j}$, one per site, which are theoretically optimum for a particular boundary condition which our problem does not satisfy (because of the Mobius strip or twisted boundary property). Therefore, although KLR still converges to the correct solution, it may do so slower than the predicted

convergence rates. As we shall see in the following section, however, no slow-down is evident in the experiments; the rate is still of order $\sqrt{N}$, where N is the total number of points in the grid.

3.5.3 Lagrange Multiplier Updates

Using the methods above, we obtain a numerical solution to (3.47), which may solve the original problem (V) if the approximations related to the two Lagrange multiplier functions are accurate. If the approximations were not accurate then the solution will not satisfy the constraints, and we must seek an alternate method to find the solution to (V). In this section we describe a method known as the generic primal-dual algorithm [4] which involves updating the Lagrange multipliers between stages and solving (3.42) directly at each stage using the present estimates of the Lagrange multipliers.

First, we recognize that the approximate PDE given in (3.47) is precisely the PDE of (3.42) with the following substitutions

$$\lambda_1(\theta) = \frac{-1}{2T}\left[\frac{m}{\sigma^2}\chi_Y - \frac{1}{\sigma^2}\int_{-T}^{T}\chi_Y y\,dt\right] \tag{3.51}$$

$$\lambda_2(\theta) = \frac{-3}{2T^3}\left[-\frac{1}{\sigma^2}\int_{-T}^{T}t\chi_Y y\,dt\right] \tag{3.52}$$

If the constraints are not adequately met by the solution of (3.47) then, following Bertsekas [4], we may update these functions using the following formulas

$$\lambda_1^{k+1}(\theta) = \lambda_1^k(\theta) + \alpha\left(m - \int_{-T}^{T} g(t,\theta)\,dt\right) \tag{3.53}$$

$$\lambda_2^{k+1}(\theta) = \lambda_2^k(\theta) + \alpha\left(0 - \frac{1}{m}\int_{-T}^{T} tg(t,\theta)\,dt\right) \tag{3.54}$$

where α is a positive constant, and k is an iteration counter. The constant α is chosen large enough so that convergence to the correct Lagrange multipliers (and, hence, the correct solution to (V)) are found quickly, yet not so large that the sequence will not converge. Bertsekas [4] describes more precisely the trade-offs in the selection of α, and relates this generic primal-dual method to the method of

multipliers, about which a great deal of theory is known. In our experiments, α is chosen empirically to yield a good rate of convergence for a given problem.

3.6 Experimental Results

3.6.1 Overview

In this section, we present the results of several simulation studies, each designed to demonstrate a different aspect of the sinogram MAP algorithms described in the earlier sections of this chapter. The object that is used for all of the simulations in this section is an ellipse with the letters M I T in its interior, as shown in Fig. 3.3. The ellipse is centered at the origin and rotated 45 degrees in the clockwise direction, and has two values: 0 outside of the ellipse and 1 within the body of the ellipse, except within the letters, where the value is 0. The object is described *analytically* — i.e., equations are used to describe the ellipse and rectangles comprising the interior letters — so that the noise-free sinogram shown in Fig. 3.4a may be calculated to arbitrary accuracy.[4] The sinogram has 81 rows and 60 columns corresponding to $n_d = 81$ and $n_v = 60$.[5] Fig. 3.4b shows a reconstruction — using convolution backprojection (CBP) — of the ellipse object from the noise-free data in Fig. 3.4a. The reconstructed image has dimensions 81 by 81.

Fig. 3.4c shows a noisy sinogram, created by adding independent samples of zero-mean Gaussian noise with variance σ^2 to each element of the true sinogram of Fig. 3.4a. The signal to noise ratio (SNR) of this sinogram is 3.0dB, where we use the following definition of SNR:

$$\mathrm{SNR} = 10 \log \frac{\frac{\pi}{n_v}\frac{2T}{n_d}\sum_{j=1}^{n_v}\sum_{i=1}^{n_d} g^2(t_i,\theta_j)}{\sigma^2}, \tag{3.55}$$

[4] In the figures of this chapter (and of Chapters 7 and 8) which contain four images of either sinograms or objects or both, the separate images are designated by panels (a)–(d), which correspond to the images at the (a) top left, (b) top right, (c) bottom left, and (d) bottom right.

[5] Instead of calculating approximate *line* integrals, we calculate approximate *strip* integrals of width $2T/n_d$ (see [33] for comments on the physical implications of this difference). Also, the sinograms we show correspond to $g(t,\theta)$ over the angular range $[\pi/2, 3\pi/2)]$.

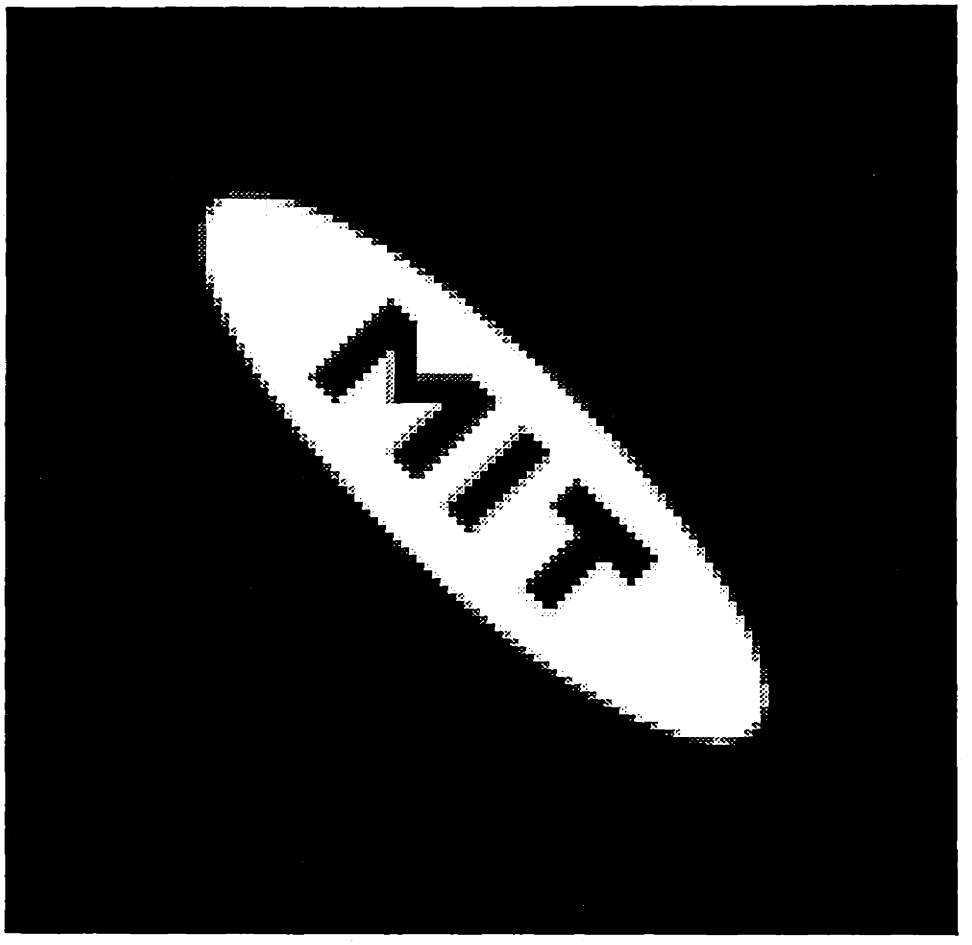


Figure 3.3: The MIT ellipse.

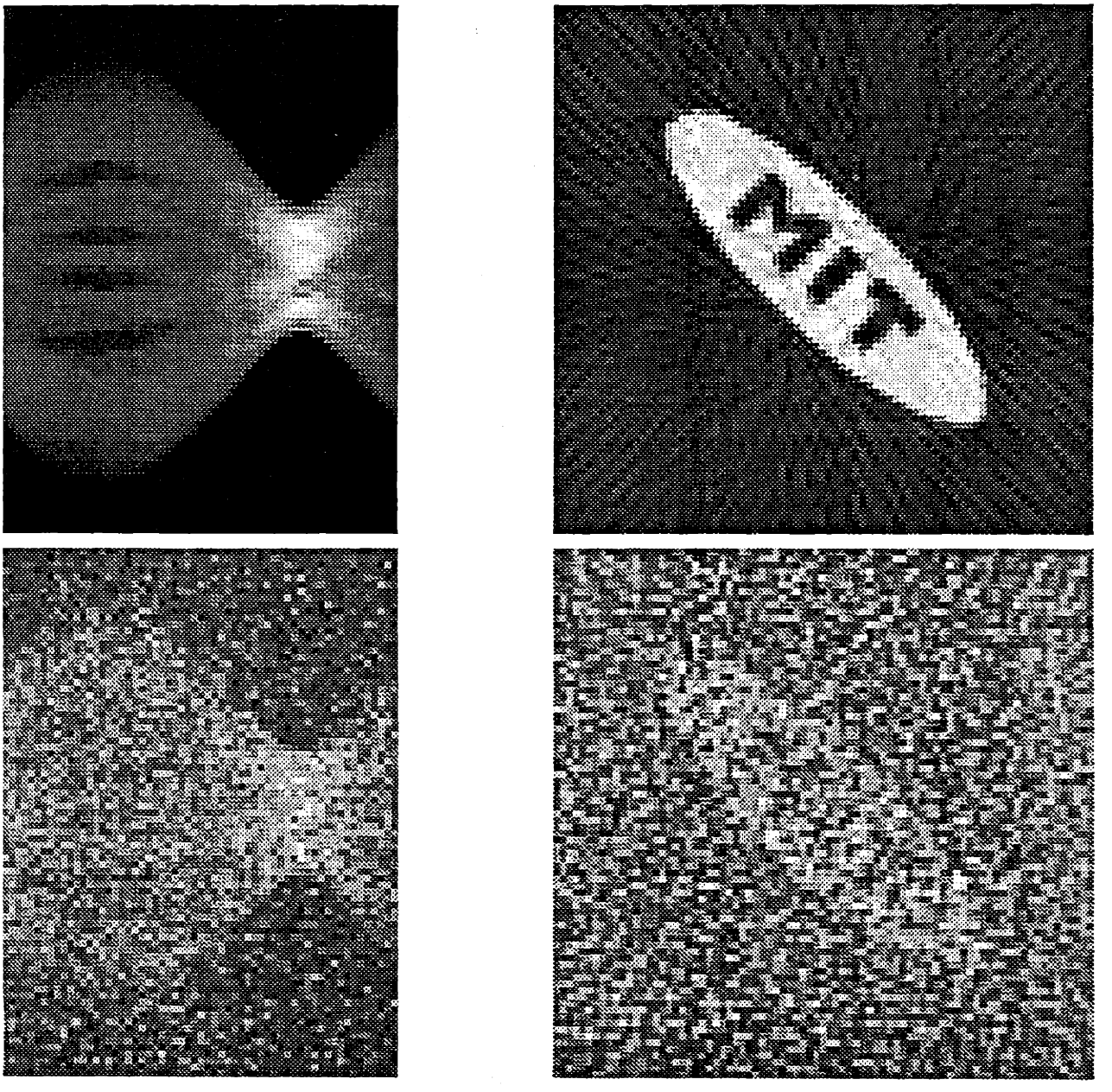


Figure 3.4: (a) A noise-free sinogram, (b) and its reconstruction. (c) A noisy sinogram (SNR=3.0dB), (c) and its reconstruction.

where $g(t_i, \theta_j)$ is the true sinogram. Fig. 3.4d shows a reconstruction of Fig. 3.4c using CBP.

We also consider several cases where the data is incomplete. Figs. 3.5a and 3.5b show reconstructions from, respectively, 15 and 10 evenly spaced projections (starting with the left-most projection) of a 10.0dB noisy observation of the ellipse of Fig. 3.4a. Since the true sinogram has 60 views, these two reconstructions correspond to having observed projections spaced 12.0 degrees and 18.0 degrees apart, respectively. Figs. 3.5c and 3.5d show reconstructions from the first (left) 40 projections and the last (right) 40 projections, respectively, of the 10.0dB sinogram mentioned above. Each of these four reconstructions was made using CBP, where the missing projections were assumed to be zero.

All the experiments in this section use the local relaxation (LR) algorithm of Section 3.5. They are designed to show the effect of the choice of γ and β, and the effect of κ, on the full-view, sparse-view, and limited-view cases. The center of mass is known to be zero in each example, however, the mass is estimated. In order to compare the performance of the algorithm for objects (which will be introduced in Chapter 7) other than the MIT ellipse we *normalize* each sinogram to unit mass before processing with the local relaxation algorithm. If this were not done, then the coefficients β, γ, and κ would not have the same qualitative effect on low mass sinograms as on large mass sinograms. Also, we estimate the noise variance using the top and bottom row of the (normalized) sinogram, and use this estimate as the true variance in the computations.

3.6.2 Effect of Smoothing Coefficients

The coefficient γ has the effect of smoothing or blurring the sinogram in the horizontal direction; the coefficient β has a similar smoothing effect in the vertical direction. Fig. 3.6 shows sinogram MAP estimates resulting from the full-view observations of Fig. 3.4c, using no support information. Fig. 3.6a corresponds to $\gamma = 0.05$ and $\beta = 0.01$, Fig. 3.6b to $\gamma = 0.5$ and $\beta = 0.01$, Fig. 3.6c to $\gamma = 0.05$ and $\beta = 0.1$, and Fig. 3.6d to $\gamma = 0.005$ and $\beta = 0.001$. Reconstructions of these sinograms using CBP are shown in the respective panels of Fig. 3.7. The reconstruction in Fig. 3.7a

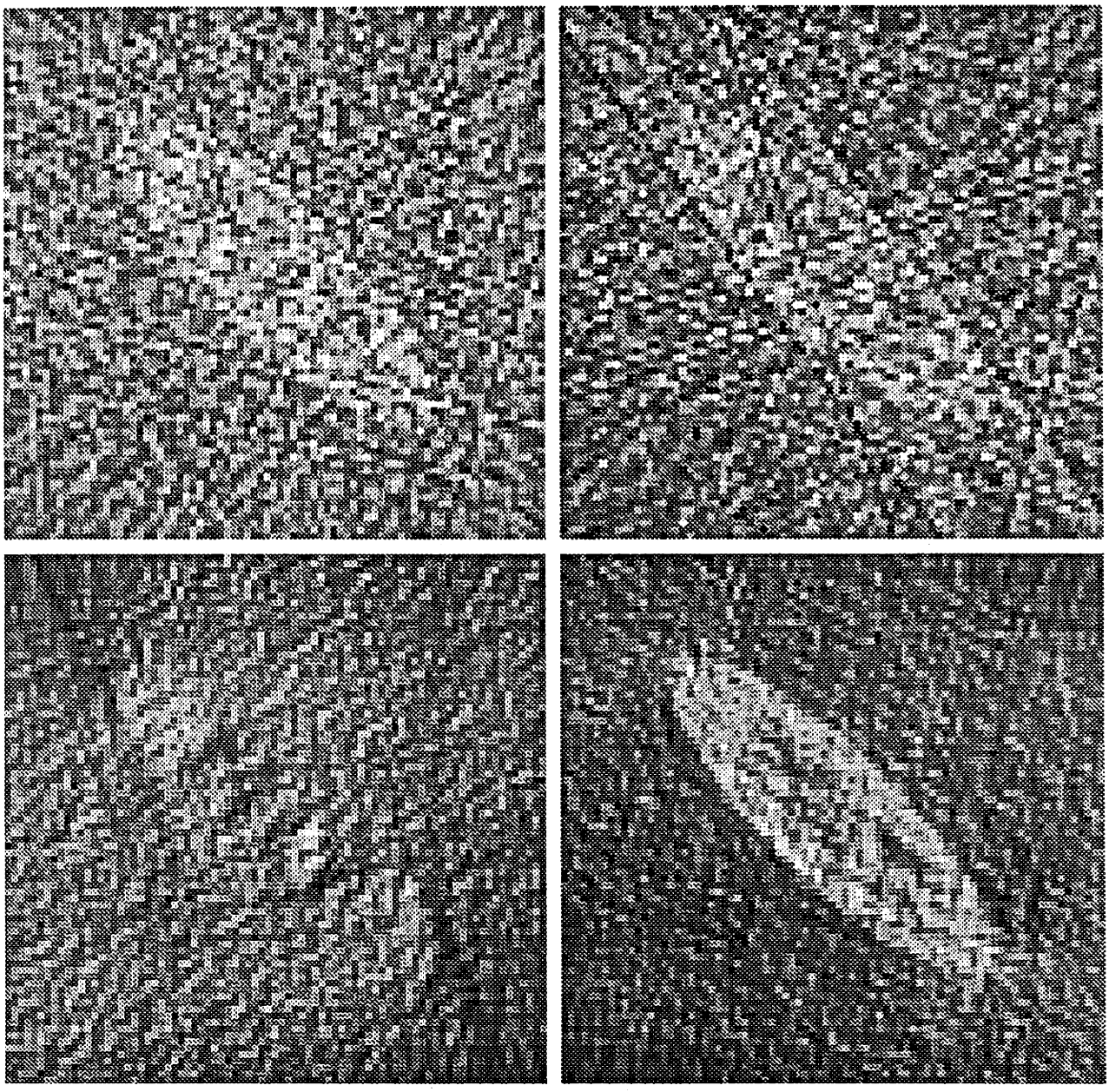


Figure 3.5: Reconstructions from a noisy sinogram (SNR=10.0dB) from (a) 15 sparse views, (b) 10 sparse views, (c) left-most 40 views, and (d) right-most 40 views.

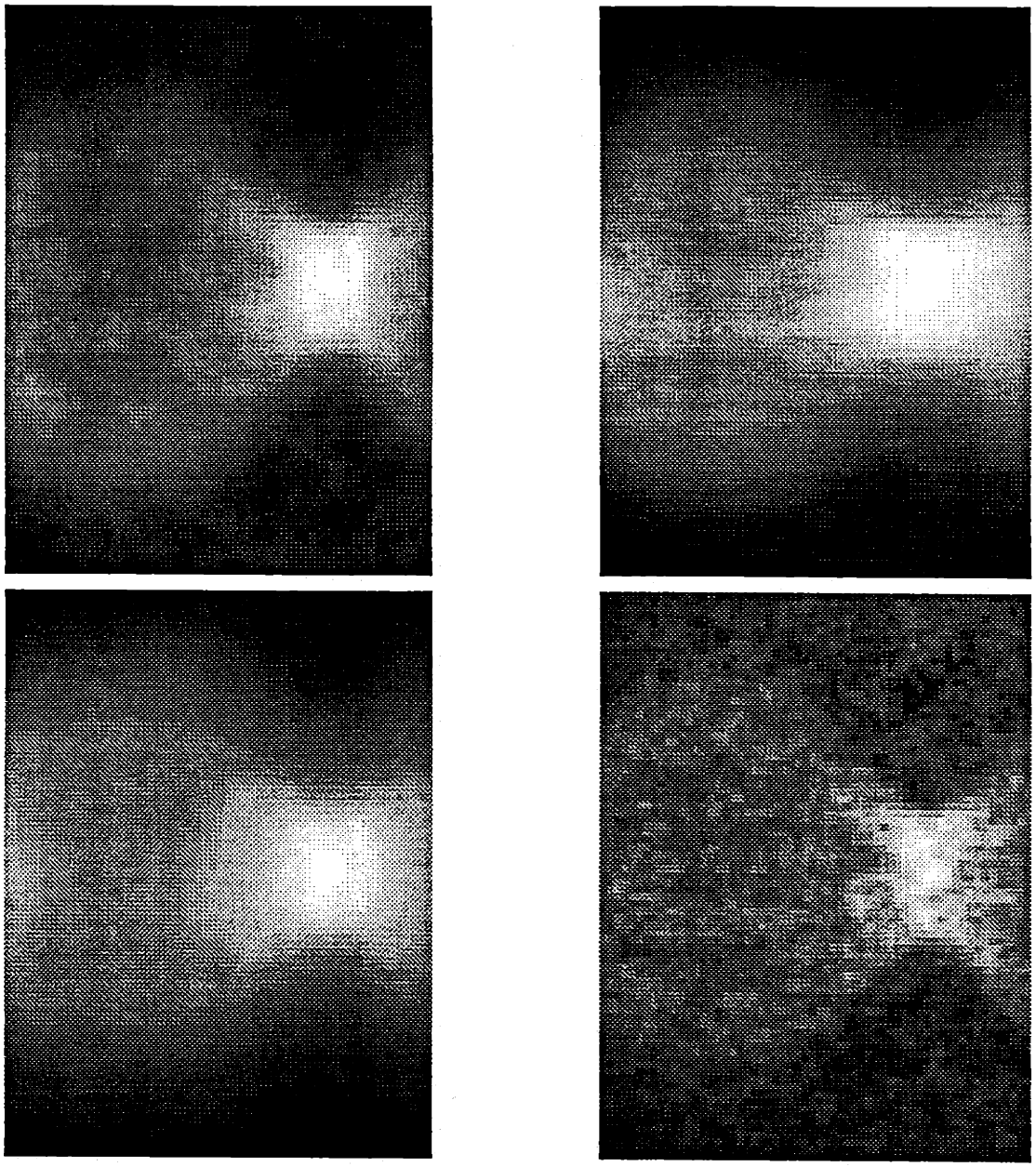


Figure 3.6: Estimates produced by the LR algorithm with (a) $\gamma = 0.05$ and $\beta = 0.01$, (b) $\gamma = 0.5$ and $\beta = 0.01$, (c) $\gamma = 0.05$ and $\beta = 0.1$, and (d) $\gamma = 0.005$ and $\beta = 0.001$.

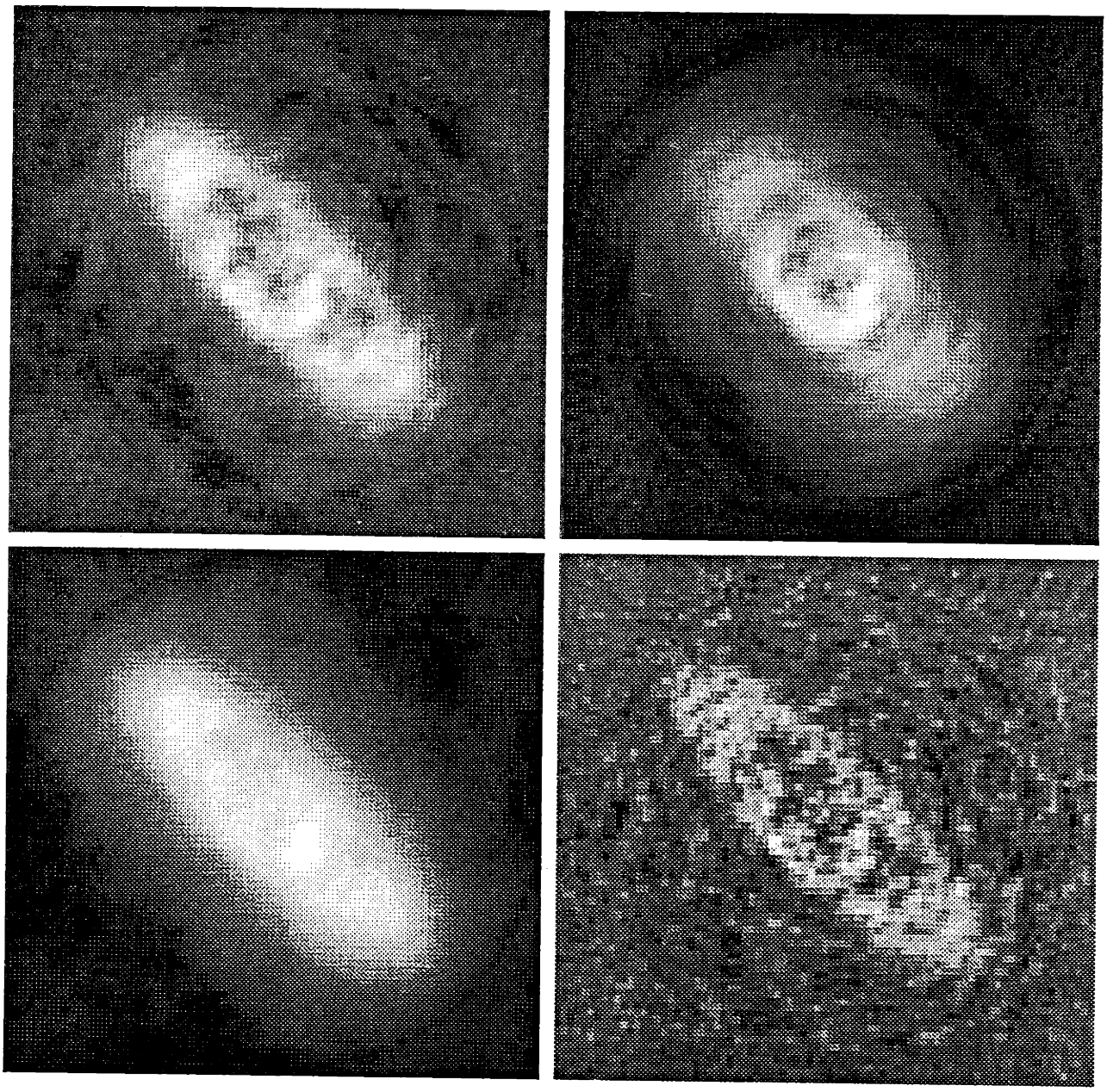


Figure 3.7: CBP reconstructions from Fig. 3.6.

— which used what has empirically shown to be good smoothing coefficients — should be compared to the unprocessed CBP reconstruction of Fig. 3.4d.

It should be noted from Fig. 3.7b that excessive smoothing of the sinogram in the horizontal direction results in *circular* blurring of the reconstructed image. Similarly, the haziness of the image in Fig. 3.7c results from excess smoothing of the sinogram in the vertical direction, which effectively produces a low-pass filtering effect on each projection. There is noticeable improvement in both reconstructions shown in Figs. 3.7a and 3.7b over that in Fig. 3.4d; however, there are important differences. For one, the contrast between the ellipse body and the background is better for the larger smoothing coefficients of Fig. 3.7a. However, that enhancement also accompanies a decreased definition of the ellipse boundary. The legibility of the internal letters, however, appears to be best in the highest contrast image shown in Fig. 3.7a.

3.6.3 Effect of Known Support

Fig. 3.8 shows the effect of varying κ for known (correct) support. The different values of κ are given by (a) $\kappa = 0.0$, (b) $\kappa = 5.0$, (c) $\kappa = 10.0$, and (d) $\kappa = 10,000$. In each case, the local relaxation algorithm used the full-view observations of Fig. 3.4c and $\gamma = 0.05$ and $\beta = 0.01$. The object reconstructions were made using full-view CBP, and should be compared to those of Figs. 3.4d and 3.7a,b,c,d.

We see from the set of experiments shown in Fig. 3.8 that known support sharpens the boundary of the ellipse considerably. However, in the image with the sharpest boundary (Fig. 3.8d), the letters in the ellipse are not as legible as the images in the other panels — the contrast of the letters does not appear to be as large. This is likely to be due to the mass constraint, which for κ large, has to produce an estimate which has all its mass (for a given projection) between the two support values. But, in addition there is a smoothness requirement which is attempting to reduce abrupt variations within the projections. This may have the overall effect of increasing the magnitude of (normally small) values of line integrals which pass through the internal letters.

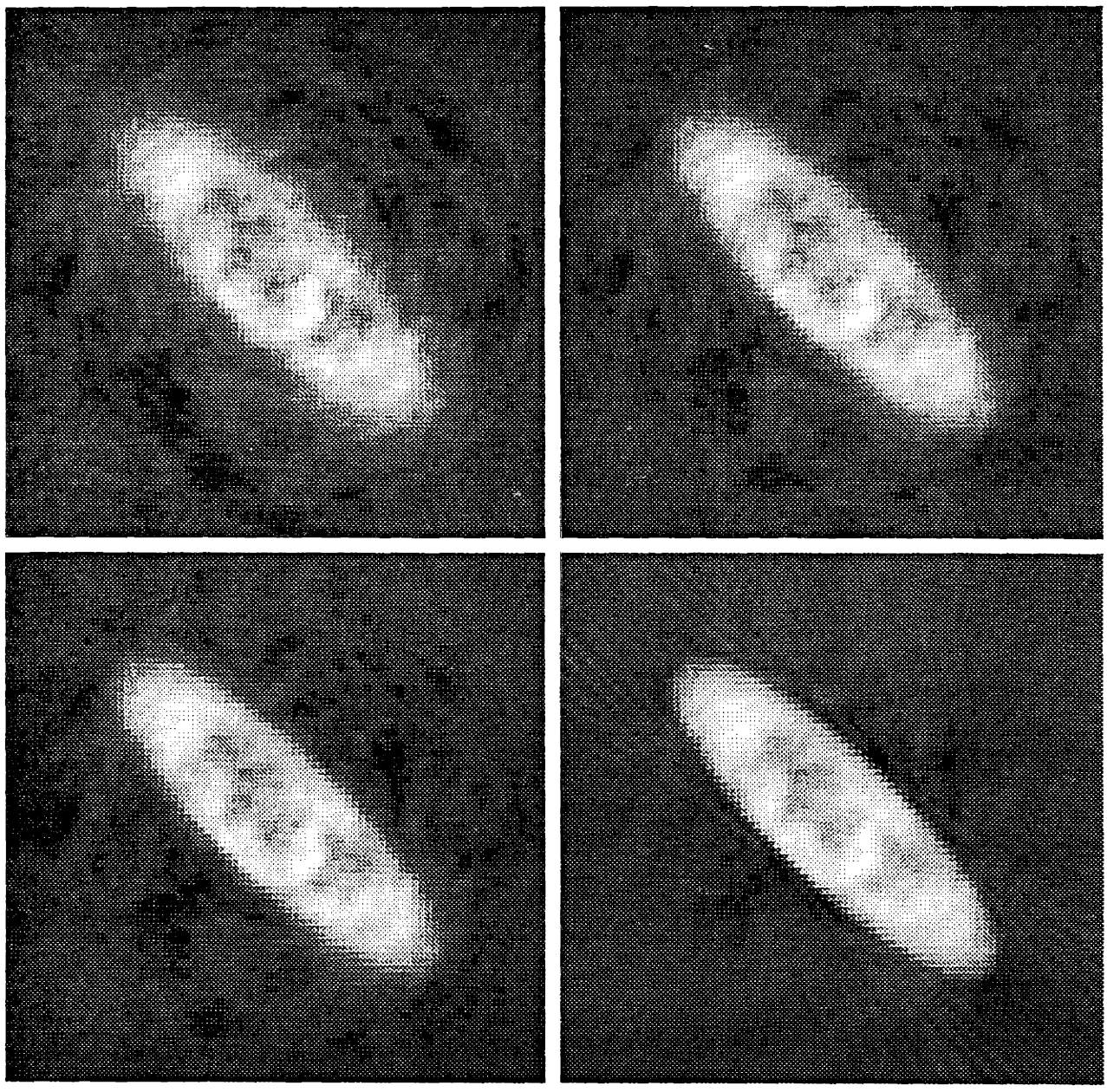


Figure 3.8: Effect of known support for (a) $\kappa = 0.0$, (b) $\kappa = 5.0$, (c) $\kappa = 10.0$, and (d) $\kappa = 10,000.0$.

3.6.4 Effect of Incorrect Support

In this set of experiments we examine the effect of using support information which is *incorrect*. Fig. 3.9 shows results where the support corresponds to an ellipse which has been rotated 90 degrees from the correct orientation. The observed sinogram is that of Fig. 3.4c, and the LR algorithm used the smoothing coefficients $\gamma = 0.05$ and $\beta = 0.01$. The different reconstructions — made using CBP — correspond to setting κ to (a) $\kappa = 0.0$, (b) $\kappa = 5.0$, (c) $\kappa = 10.0$, and (d) $\kappa = 10,000$.

This set of experiments shows that as κ grows larger, the image values outside the assumed region of support grow smaller. Eventually, this effect overwhelms the evidence of the observations and virtually obliterates the parts of the true ellipse which lie outside of the incorrect support. But the effect of the mass constraint and the smoothing coefficients also affect the appearance of the final image. Since each projection has mass m, when the support width is incorrectly narrowed, and κ is too large, then the sinogram values must be very large within the region of support *just to accommodate the required mass*, and the values will typically be very much larger than the observations. As mentioned previously, this will have the effect of reducing the contrast of the inner details of these projections, and the effect on the image is to eliminate contrast within even the intersection of the correct support and the incorrect support. On those projections which have support values that are much too wide, it is the smoothing terms which dominate. In order to lower the overall energy of the sinogram (that is, the energy term in the Markov random field), the vertical pair-potentials or equivalently, the vertical derivatives should be small. Therefore, these projections tend to become as smooth as possible over the prescribed support and contribute to the image a "shadow" ellipse which corresponds to the incorrect support.

Fig. 3.10 shows a sequence of reconstructions (using LR followed by CBP) which have kept κ to the constant 5.0, but vary the orientation of the assumed object support. In these reconstructions, we have used the support of an ellipse that has the same size and eccentricity at the true object support, but has been rotated in the counter-clockwise direction by (a) 0.0 degrees, (b) 15.0 degrees, (c) 30.0 degrees, and (d) 45.0 degrees.

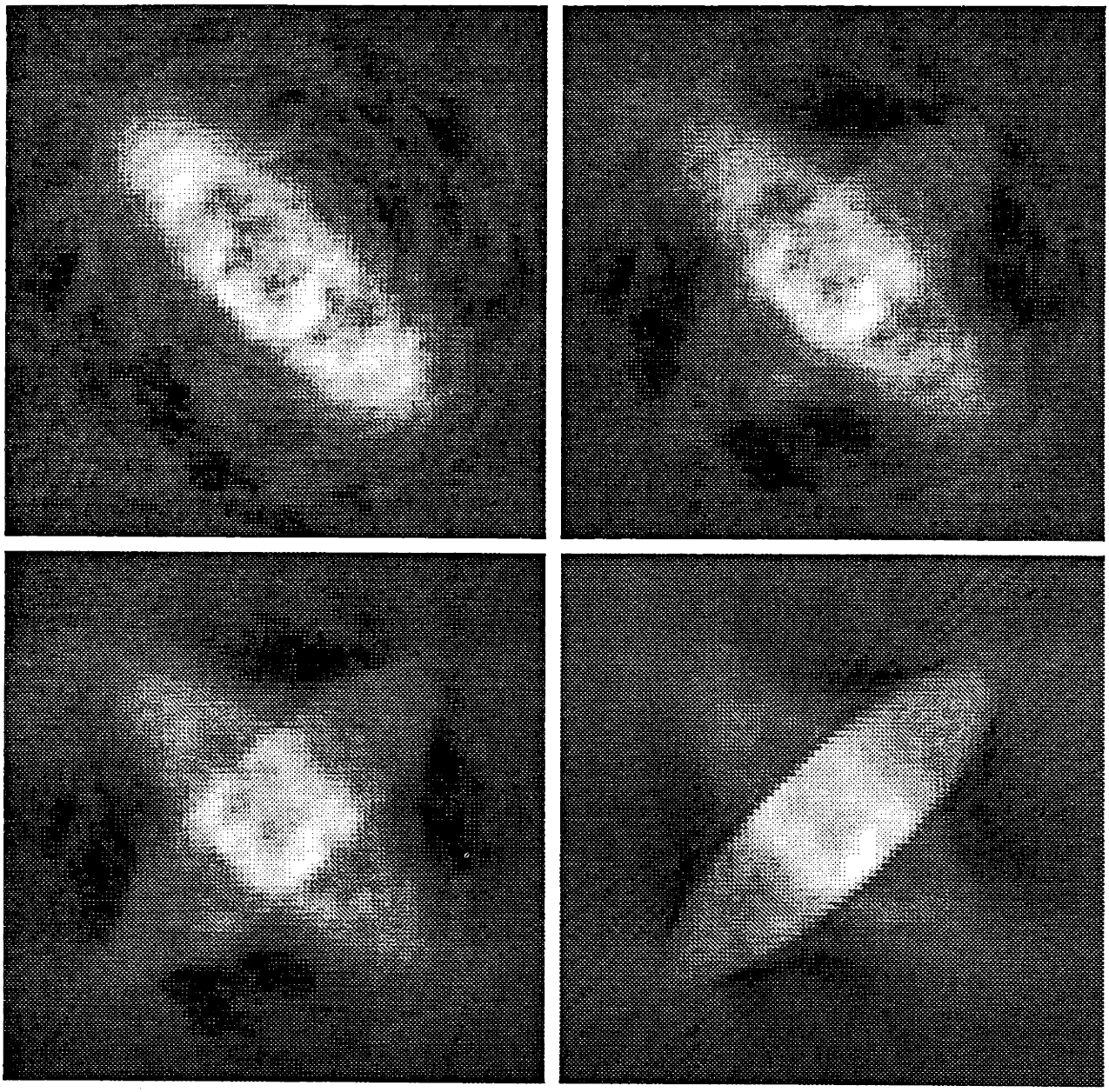


Figure 3.9: Effect of incorrect support for (a) $\kappa = 0.0$, (b) $\kappa = 5.0$, (c) $\kappa = 10.0$, and (d) $\kappa = 10,000.0$.

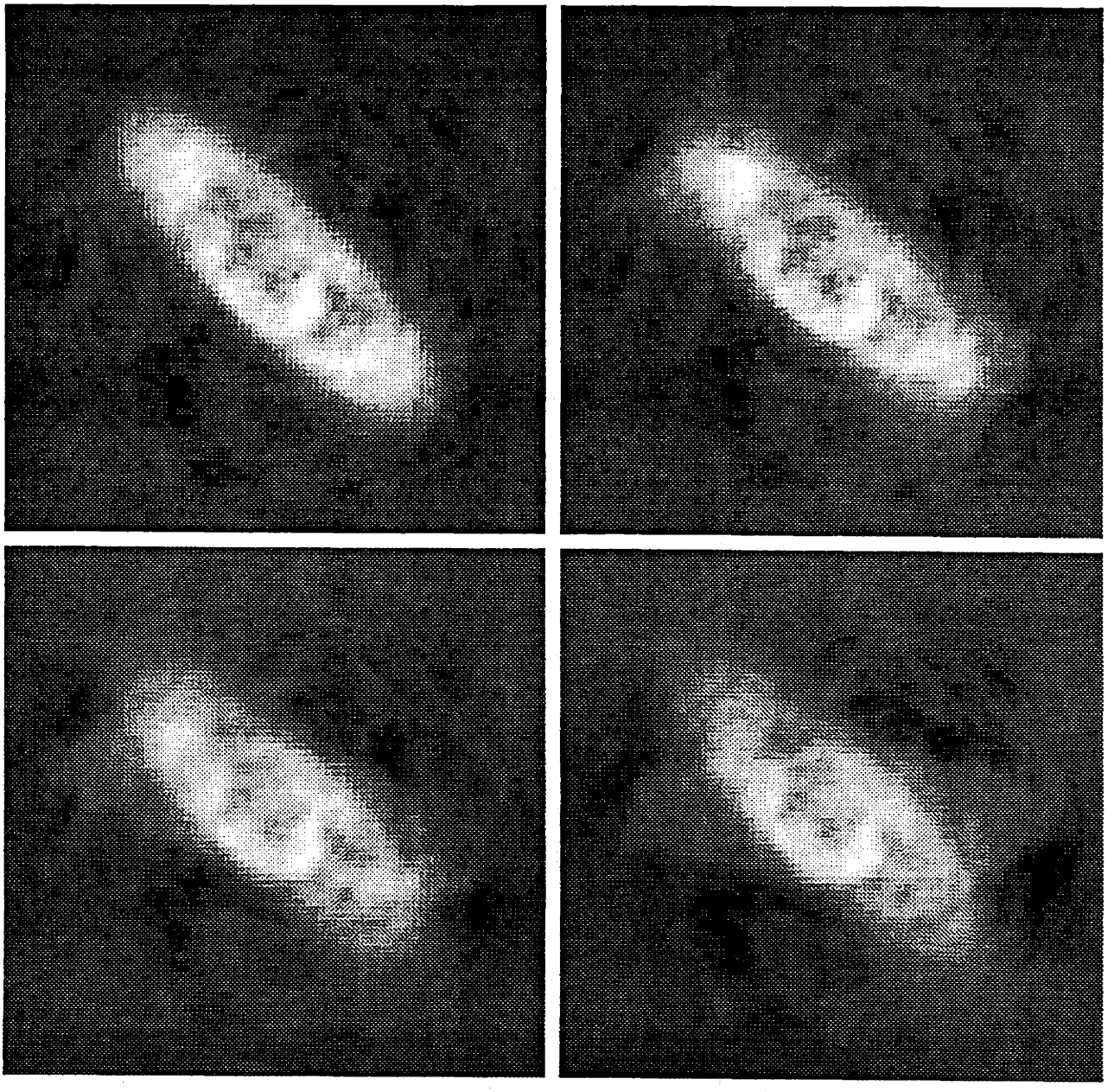


Figure 3.10: Effect of using an incorrect support which is rotated counter-clockwise by (a) 0.0 degrees, (b) 15.0 degrees, (c) 30.0 degrees, and (d) 45.0 degrees.

This set of experiments shows that a modest choice of κ, together with a less severe support error will produce an image which retains many of the details of the true image with only a small "shadow" due to the incorrect support. However, it is clear that an incorrect support estimate can produce results much worse than having not introduced any support information whatsoever (compare these results to that of Fig. 3.9a).

In Fig. 3.11 we show a sequence of reconstructions (using LR followed by CBP) which have used $\kappa = 5.0$, but with support which is the incorrect size. Figs. 3.11a and 3.11b show two cases where the support is too small, and Figs. 3.11c and 3.11d show two cases where the support is too large. Overall, the size of the support increases from Fig. 3.11a to Fig. 3.11d. The reconstruction using the correct support and $\kappa = 5.0$ may be seen in Fig. 3.10a.

We may conclude from this set of experiments that is is preferable to err on the side of using a support estimate that is too large than too small, in general. Although the boundaries are not as sharp when the support is too large, the loss of contrast in the interior and the effect of double-boundaries for small support is much more undesirable.

3.6.5 Sparse-Angle Studies

Fig. 3.12 shows the results of the sparse-angle experiments for this chapter. The (a) and (b) images in this figure correspond to the 15-view and 10-view 10dB sparse-angle cases (see Section 3.6.1), respectively, where $\gamma = 0.05$ and $\beta = 0.01$ and the support is known and $\kappa = 10,000$. The (c) and (d) images correspond to the 15-view and 10-view cases, respectively, with the same smoothing coefficients, but with $\kappa = 0.0$ — i.e., no known support information is used.

This experiment demonstrates nicely the potential of the LR algorithm. In either sparse-angle case, the contrast of the image is improved over those in Fig. 3.5 dramatically. And while the boundary is quite sharp as expected in the case of $\kappa = 10,000$, it is quite clear what the shape of the object is in case of $\kappa = 0.0$. The loss of contrast in the interior of the ellipse when $\kappa = 10,000$ remains evident in these experiments, however.

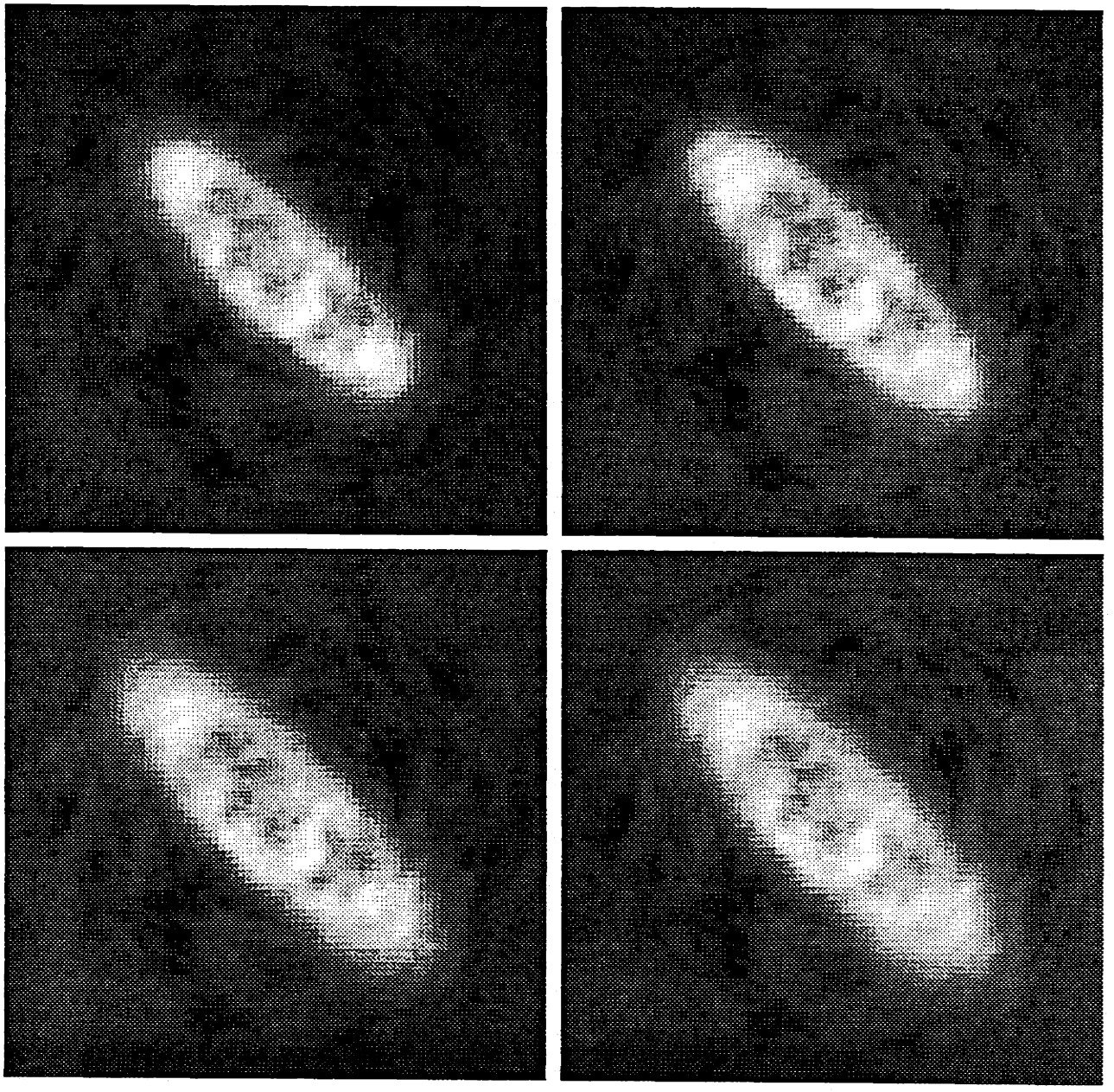


Figure 3.11: Effect of using an incorrect support which is too small in (a) and (b) and too large in (c) and (d).

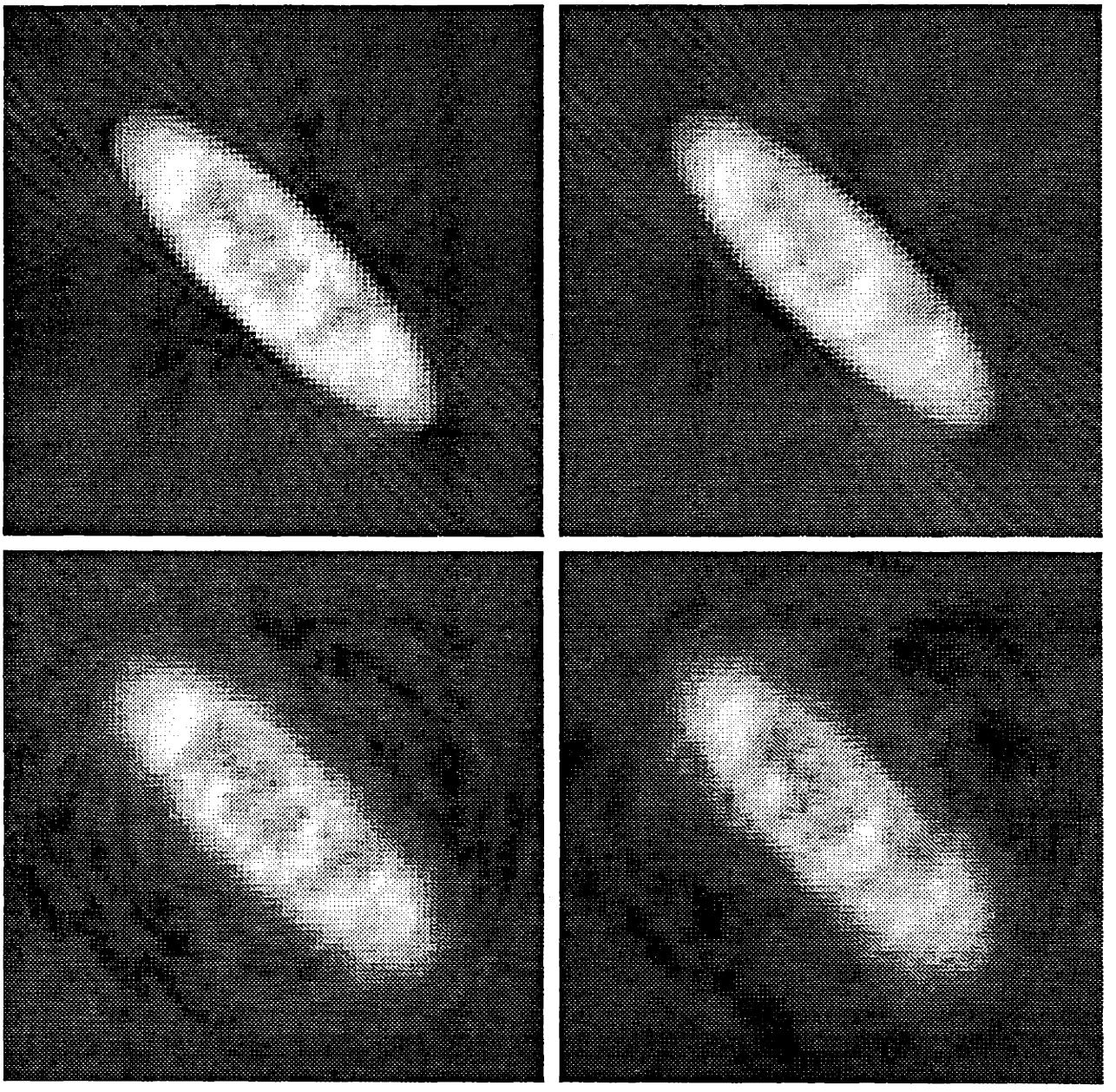


Figure 3.12: Sparse-angle studies (LR with $\gamma = 0.05$ and $\beta = 0.01$). (a) 15 observed projections and known support. (b) 10 observed projections and known support. (c) 15 observed projections and no support. (d) 10 observed projections and no support.

3.6.6 Limited-Angle Studies

Fig. 3.13 shows the results of several limited-angle studies. The (a) and (b) images are reconstructions obtained with known support (with $\kappa = 10,000$) from the two limited-angle cases described in Section 3.6.1. The experiment resulting in panel (a) uses the first 40 (left-most) projections, whereas panel (b) uses the last 40 (right-most) projections. Panels (c) and (d) correspond to the same observations as in (a) and (b), respectively, but in these cases no support information was used. As in the sparse-angle studies, the smoothing coefficients for all four studies were $\gamma = 0.05$ and $\beta = 0.01$.

These limited-angle studies show behavior which is similar to the sparse-angle studies. The boundary of the ellipse is quite sharp, as expected, in the case of $\kappa = 10,000$, and there is an accompanying loss of contrast in the interior. The images generated using $\kappa = 0.0$ have different problems, however. In particular, the image in Fig. 3.13c shows good contrast in the letters in the interior but is unable to provide any boundary definition on the long sides of the ellipse. This is because the leftmost 40 projections which are observed view the ellipse from the *broadside*, and as such do not contain information about the narrow ellipse dimension. The image in Fig. 3.13d suffers from the opposite problem. There is a loss of definition of the letters in the interior because many of the projections that would normally be obtained from the broadside of the ellipse are missing. It is in the first case that support knowledge can aid tremendously as we shall see in Chapter 7. Unfortunately, when projections from the broadside of the ellipse are missing, there is little that our method can do to provide any additional clarity of the interior detail.

3.7 Discussion

In this chapter we have developed a reconstruction method based on MAP estimation principles. The method uses a prior probability on the *sinograms* — which is a Markov random field (MRF) — and uses known noise statistics, of the noisy and (possibly) incomplete observations, to specify the form of the MAP estimate.

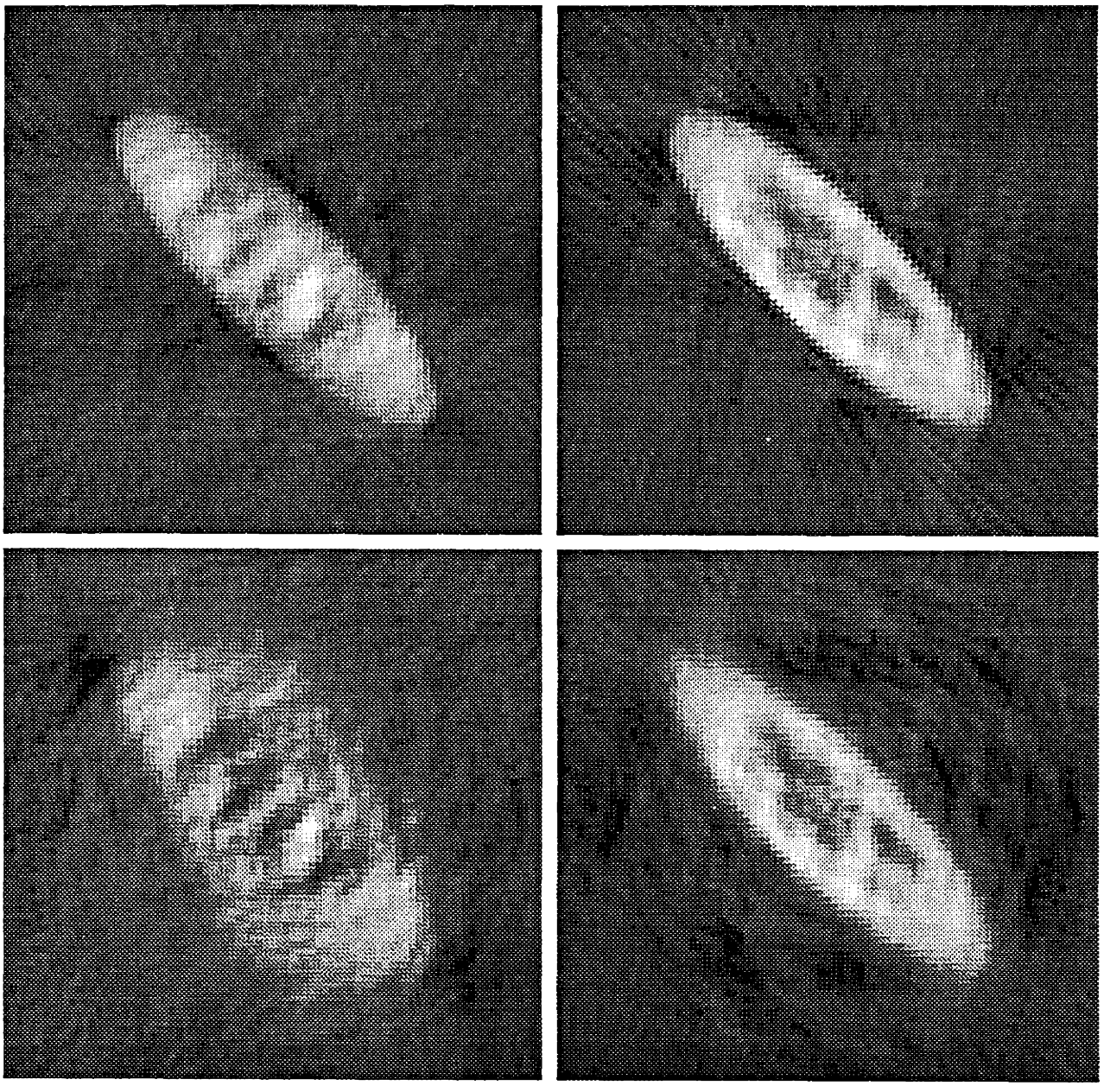


Figure 3.13: Limited-angle studies (LR $\gamma = 0.05$ and $\beta = 0.01$). (a) Left 40 projections and known support. (b) Right 40 projections and known support. (c) Left 40 projections and no support. (d) Right 40 projections and no support.

The MRF presented in Section 3.2 incorporates the prior information that line integrals obtained over lines slightly different in lateral displacement (but with the same angle) tend to have similar values, as do line integrals taken from lines having slightly different angles (but with the same lateral displacement). Also, we impose *constraints* on the space of feasible sinograms so that only those which have projections (columns of the sinogram) with a constant mass, and with zero center of mass, and for which all sinogram values are positive, are feasible. The MRF also specifies boundary conditions which are consistent with the fact that the object is entirely contained within the disk of radius T, and that the sinogram is defined on a Mobius strip.

Because the constraints are linear and we assume the noise to be Gaussian, and because of our choice of potential functions for the MRF, the form of the MAP estimation problem is a quadratic program (QP). In Sections 3.3, 3.4, and 3.5 we present three possible approaches to solving this QP. The solution of Section 3.3 is a straightforward implementation of the QP using standard QP code, and is shown in Appendix 3.C to be entirely impractical to solve on today's computers, for any reasonable size problems. This method does have the advantage, however, that it solves the problem *exactly* (to machine precision, of course) in finite time. The simulated annealing method of Section 3.4 is a very general method which has close ties to our use of a MRF as a prior probability. However, because of its generality, it is a very inefficient way to solve a QP because it does not take advantage of the structure of the problem. One interesting result that was developed in this section, however, is the development of a particular form of *constrained* Metropolis updating, a topic which is starting to receive some attention among researchers in large-scale optimization [25]. The final algorithm we developed in Section 3.5 is based upon a variational interpretation of the MAP estimation problem, and leads to essentially the same optimization problem as proposed in Section 3.2 but without the positivity constraint. Posing the problem in this way, however, suggested a method of solution based on the solution of an elliptic partial differential equation (PDE) on the sinogram domain. Even with the necessity of keeping and updating Lagrange multipliers, this method is fast and memory efficient, and is used in all

our studies in the simulations of Section 3.6 and in future chapters.

The simulations presented in Section 3.6 show the range of results that may be obtained using this MAP approach. Of particular interest, are the results presented for the known (correct) support with κ large, and with the smoothing coefficients given by $\gamma = 0.05$ and $\beta = 0.01$, because it is this case which we have observed yields the best empirical results. The improvement over the unprocessed CBP reconstructions shown in Figs. 3.4 and 3.5 is quite dramatic. In particular, the boundary of the ellipse is made much sharper, and the letters within the ellipse can be made out in all of the processed cases, and in none of the unprocessed cases. However, we have not compared this method with any of the other methods proposed in the literature, therefore, it is not possible to claim that this method represents an improvement over *all* existing methods. The *framework* of this solution methodology, however, may set the stage for much further creativity, both in the specification of prior knowledge and in the implementation of reconstruction methods. Our exploration of support estimation in the next several chapters, the hierarchical reconstruction algorithm of Chapter 7, and the constraint-based methods of Chapter 8 represent some of this potential development. We also discuss some ideas for further research in Chapter 9.

3.A Derivation of Variational PDE

The statement of the variational problem (V) appears in equations (3.38), (3.39), and (3.40). In this appendix, we show that the necessary condition for $g(t,\theta)$ to solve (V) is that it satisfy the PDE given in (3.42) together with the additional boundary condition given in (3.43). In this section the notations g_t and g_{tt} stand for the first and second partial derivatives of $g(t,\theta)$ with respect to t, respectively, and g_θ and $g_{\theta\theta}$ stand for the first and second partial derivatives of $g(t,\theta)$ with respect to θ, respectively.

3.A.1 Formal Statement of the Necessary Conditions

Referring to equation (3.41) we see that the problem is to find g that minimizes

$$I = \iint_{y_T} F(t,\theta,g,g_t,g_\theta)\,dt\,d\theta \tag{3.56}$$

where

$$F(t,\theta,g,g_t,g_\theta) = \kappa\bar{\chi}_G g^2 + \beta g_t^2 + \gamma g_\theta^2 + \frac{1}{2\sigma^2}\chi_Y (y-g)^2 \ ,$$

and that satisfies the constraints and boundary conditions mentioned above. Lagrange multiplier techniques are used in order to transform the constrained problem into an unconstrained problem. Accordingly, we define (see [36])

$$\begin{aligned}\tilde{J}_1 &= \iint_{y_T} \tilde{G}_1\,dt\,d\theta = \int_\theta \lambda_1(\theta)\left(\int_t (g(t,\theta) - m\delta(t))\,dt\right)d\theta = 0\\ \tilde{J}_2 &= \iint_{y_T} \tilde{G}_2\,dt\,d\theta = \int_\theta \lambda_2(\theta)\int_t tg(t,\theta)\,dt\,d\theta = 0\end{aligned}$$

and proceed to minimize (unconstrained)

$$\tilde{I} = I + \tilde{J}_1 + \tilde{J}_2 = \iint_{y_T} \tilde{F}(t,\theta,g,g_t,g_\theta)\,dt\,d\theta \ .$$

Following Hildebrand [36], three *admissible test functions* are introduced, $\nu_1(t,\theta)$, $\nu_2(t,\theta)$, and $\nu_3(t,\theta)$ in order to form the three-parameter family of *comparison functions*

$$p(t,\theta) = g(t,\theta) + \epsilon_1\nu_1(t,\theta) + \epsilon_2\nu_2(t,\theta) + \epsilon_3\nu_3(t,\theta) \ .$$

The test functions $\nu_1(t,\theta)$ are admissible in the sense that if g satisfies the boundary conditions then $g+\epsilon_i\nu_i$ for any ϵ_i also satisfies the boundary conditions. Hence, any comparison function $p(t,\theta;\epsilon_1,\epsilon_2,\epsilon_3)$ automatically satisfies the boundary conditions when g does. However, for any arbitrary choice of ν_1, ν_2, and ν_3, the ϵ_i's can not (necessarily) be chosen independently, since the equality constraints $\tilde{J}_1$ and $\tilde{J}_2$ also must be satisfied by the comparison functions p. This is why three test functions are required.

The variational technique now tells us to minimize

$$\tilde{I}(\epsilon_1,\epsilon_2,\epsilon_3) = \iint_{y_T} \tilde{F}(t,\theta,p,p_t,p_\theta)\,dt\,d\theta$$

with respect to the three ϵ's. The necessary conditions are

$$\left.\frac{\partial\tilde{I}}{\partial\epsilon_1}\right|_{\epsilon_1=0} = \left.\frac{\partial\tilde{I}}{\partial\epsilon_2}\right|_{\epsilon_2=0} = \left.\frac{\partial\tilde{I}}{\partial\epsilon_3}\right|_{\epsilon_3=0} = 0\ . \tag{3.57}$$

This equation leads directly to the necessary conditions we desire. The rest of this appendix is simplification and substitution.

3.A.2 Derivation of the Euler-Lagrange Equation

The derivatives in (3.57) are given by

$$\frac{\partial\tilde{I}}{\partial\epsilon_1} = \iint_{y_T} \frac{\partial\tilde{F}}{\partial p}\frac{\partial p}{\partial\epsilon_i} + \frac{\partial\tilde{F}}{\partial p_t}\frac{\partial p_t}{\partial\epsilon_i} + \frac{\partial\tilde{F}}{\partial p_\theta}\frac{\partial p_\theta}{\partial\epsilon_i}\,dt\,d\theta \quad i=1,2,3$$

which, when $\epsilon_i = 0$, becomes

$$\left.\frac{\partial\tilde{I}}{\partial\epsilon_i}\right|_{\epsilon_i=0} = \iint_{y_T} \frac{\partial\tilde{F}}{\partial g}\nu_i + \left(\frac{\partial\tilde{F}}{\partial g_t}\nu_{it} + \frac{\partial\tilde{F}}{\partial g_\theta}\nu_{i\theta}\right)\,dt\,d\theta \quad i=1,2,3\ . \tag{3.58}$$

The two bracketed terms in the integrand may be expanded using the product rule for differentiation as

$$\frac{\partial\tilde{F}}{\partial g_t}\frac{\partial\nu_i}{\partial t} + \frac{\partial\tilde{F}}{\partial g_\theta}\frac{\partial\nu_i}{\partial\theta} = \frac{\partial}{\partial t}\left(\frac{\partial\tilde{F}}{\partial g_t}\nu_i\right) + \frac{\partial}{\partial\theta}\left(\frac{\partial\tilde{F}}{\partial g_\theta}\nu_i\right) - \left[\frac{\partial}{\partial t}\left(\frac{\partial F}{\partial g_t}\right)\nu_i + \frac{\partial}{\partial\theta}\left(\frac{\partial\tilde{F}}{\partial g_\theta}\right)\nu_i\right]\ .$$

Then, the first two terms on the right hand side of the above equation may be simplified using the divergence theorem ($\iint_V \mathrm{div}F\,dv = \int_{\partial V} F\cdot n\,ds$), and with this

substitution made, (3.57) and (3.58) yield

$$\left.\frac{\partial \tilde{I}}{\partial \epsilon_i}\right|_{\epsilon=0} = \iint_{y_T} \frac{\partial \tilde{F}}{\partial g}\nu_i - \frac{\partial}{\partial t}\left(\frac{\partial \tilde{F}}{\partial g_t}\right)\nu_i - \frac{\partial}{\partial \theta}\left(\frac{\partial \tilde{F}}{\partial g_\theta}\right)\nu_i \, dt\, d\theta + \int_{\partial y_T}\left(\frac{\partial \tilde{F}}{\partial g_t}n_t + \frac{\partial \tilde{F}}{\partial g_\theta}n_\theta\right)\nu_i \, ds = 0 \quad i = 1,2,3 \tag{3.59}$$

where n_t and n_θ are the coordinates of the outward pointing unit normal on the boundary ∂y_T. This is the formal statement of the necessary conditions. We now simplify the above expressions.

Since (3.59) holds for all admissible test functions, it must hold for those test functions which are *zero* on the boundary. In this case, then, the contour integral in (3.59) is trivially zero, hence we must have that

$$\iint_{y_T} \frac{\partial \tilde{F}}{\partial g}\nu_i - \frac{\partial}{\partial t}\left(\frac{\partial \tilde{F}}{\partial g_t}\right)\nu_i - \frac{\partial}{\partial \theta}\left(\frac{\partial \tilde{F}}{\partial g_\theta}\right)\nu_i \, dt\, d\theta = 0 \quad i = 1,2,3 \tag{3.60}$$

which is sometimes called the *weak form* of the necessary conditions (see [81]).

Now (3.60) must hold for all admissible test functions, hence we must have also that

$$\int_{\partial y_T}\left(\frac{\partial \tilde{F}}{\partial g_t}n_t + \frac{\partial \tilde{F}}{\partial g_\theta}n_\theta\right)\nu_i \, ds = 0 \quad i = 1,2,3 \tag{3.61}$$

which may be called the *integrated boundary conditions*. We shall return to analyze this requirement later. However, first we wish to show that for this problem, the weak form of equation (3.60) implies the *Euler-Lagrange equation*

$$\frac{\partial \tilde{F}}{\partial g} - \frac{\partial}{\partial t}\left(\frac{\partial \tilde{F}}{\partial g_t}\right) - \frac{\partial}{\partial \theta}\left(\frac{\partial \tilde{F}}{\partial g_\theta}\right) = 0 \tag{3.62}$$

which, when $\tilde{F}$ is substituted in, becomes the desired PDE.

To see why (3.62) is true let us rewrite (3.60) in symbolic form as

$$\iint L\{\tilde{F}\}\nu_i \, dt\, d\theta = 0 \tag{3.63}$$

where $L = \frac{\partial}{\partial g} - \frac{\partial}{\partial t}\frac{\partial}{\partial g_t} - \frac{\partial}{\partial \theta}\frac{\partial}{\partial g_\theta}$ is a differential operator. Equation (3.63) may be expanded as

$$\iint L\{F\}\nu_i \, dt\, d\theta + \iint L\{\tilde{G}_1 + \tilde{G}_2\}\nu_i \, dt\, d\theta = 0 \ . \tag{3.64}$$

The second term in the right hand side may be simplified by denoting $\tilde{G}_1 = \lambda_1 G_1$ and $\tilde{G}_2 = \lambda_2 G_2$. Then we have

$$\begin{aligned} L\{\lambda_i G_i\} &= \frac{\partial}{\partial g}(\lambda_i G_i) - \frac{\partial}{\partial t}\left(\frac{\partial}{\partial g_t}(\lambda_i G_i)\right) - \frac{\partial}{\partial \theta}\left(\frac{\partial}{\partial g_\theta}(\lambda_i G_i)\right) \\ &= \lambda_i \frac{\partial G_i}{\partial g} - \frac{\partial}{\partial t}\left(\lambda_i \frac{\partial G_i}{\partial g_t}\right) - \frac{\partial}{\partial \theta}\left(\lambda_i \frac{\partial G_i}{\partial g_\theta}\right) . \end{aligned}$$

But, $\partial G_i / \partial g_\theta = 0$, so

$$\begin{aligned} L\{\lambda_i G_i\} &= \lambda_i \frac{\partial G_i}{\partial g} - \lambda_i \frac{\partial}{\partial t}\left(\frac{\partial G_i}{\partial g_t}\right) \\ &= \lambda_i L_2\{G_i\} \end{aligned}$$

where $L_2 = \frac{\partial}{\partial g} - \frac{\partial}{\partial t}\frac{\partial}{\partial g_t}$ is a second differential operator.

Now we may write (3.64) as

$$\int_\theta \int_t L\{F\}\nu_i \, dt + \int_t \lambda_1 L_2\{G_1\}\nu_i + \lambda_2 L_2\{G_2\}\nu_i \, dt \, d\theta = 0 \ . \tag{3.65}$$

Consider (3.65) for $i = 1, 2$ only. We argue below that ν_1 and ν_2 may be chosen so that $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$ satisfy (3.65). Then, since ν_3 is arbitrary, *its coefficient must be zero.* This yields exactly the Euler-Lagrange equation of (3.62).

To justify the above conclusion we choose ν_1 so that

$$\begin{aligned} \int_t L_2\{G_2\}\nu_1 \, dt &= 0 \text{ and} \\ \int_t L_2\{G_1\}\nu_1 \, dt &\neq 0 \ . \end{aligned}$$

Then, $\lambda_1(\theta)$ may be chosen to satisfy (3.65) without regard to $\lambda_2(\theta)$ since its coefficient is zero. Then repeat the procedure by choosing first ν_2 and solving for $\lambda_2(\theta)$. The above may always be done provided that $L_2\{G_1\} \neq 0$ and $L_2\{G_2\} \neq 0$ which is certainly true.

3.A.3 The Explicit Equilibrium Equations

The major results of this appendix are contained formally in equations (3.61) and (3.62). We now make these equilibrium equations explicit for our problem by substituting our expression for $\tilde{F}$ and simplifying. We begin by computing the derivatives of $\tilde{F}$.

The Derivatives

The required derivatives of $\tilde{F}$ are easily computed as

$$\begin{aligned}
\frac{\partial \tilde{F}}{\partial g} &= 2\bar{\chi}_G g - \frac{1}{\sigma^2}\chi_Y(y-g) + \lambda_1 + \lambda_2 t \\
\frac{\partial \tilde{F}}{\partial g_t} &= 2\beta g_t \\
\frac{\partial}{\partial t}\left(\frac{\partial \tilde{F}}{\partial g_t}\right) &= 2\beta g_{tt} \\
\frac{\partial \tilde{F}}{\partial g_\theta} &= 2\gamma g_\theta \\
\frac{\partial}{\partial \theta}\left(\frac{\partial \tilde{F}}{\partial g_t}\right) &= 2\gamma g_{\theta\theta}
\end{aligned} \tag{3.66}$$

The Additional Boundary Condition

For this section only, we denote the boundary of $\mathcal{Y}_T$ by ∂D, and the four sides of $\mathcal{Y}_T$, starting on the right and proceeding counterclockwise, by ∂D_1, ∂D_2, ∂D_3, and ∂D_4 (see Fig. 3.14). Since our original boundary conditions in (3.40) *specify* the value of g on ∂D_2 and ∂D_4, then our test functions ν_i must be zero on ∂D_2 and ∂D_4. However, g is only partially specified on ∂D_1 and ∂D_3 by the condition $g(t,0) = g(-t,\pi)$. Equation (3.61) forces an addition boundary condition on g, however, as is usually the case with a free boundary in a variational problem [36].

To see how this additional boundary condition comes about, we write (3.61) as

$$\int_{\partial D} (2\beta g_t n_t + 2\gamma g_\theta n_\theta)\, \nu_i \, ds = 0$$

which becomes

$$\int_{\partial D_1} 2\beta g_t \nu_i \, ds + \int_{\partial D_3} 2\beta g_t \nu_i \, ds = 0$$

since ν_i is zero on ∂D_2 and ∂D_4. Substituting the limits for t and the appropriate constant values for θ, and dividing through by 2β, this becomes

$$\int_{-T}^{T} g_t(t,\pi)\nu_i(t,\pi)\, dt - \int_{-T}^{T} g_t(-t,0)\nu_i(-t,0)\, dt = 0 \ .$$

Now, since ν_i must satisfy the boundary conditions of (3.40), we have that $\nu_i(t,\pi) = \nu_i(-t,0)$, which implies the desired additional boundary condition of (3.43).

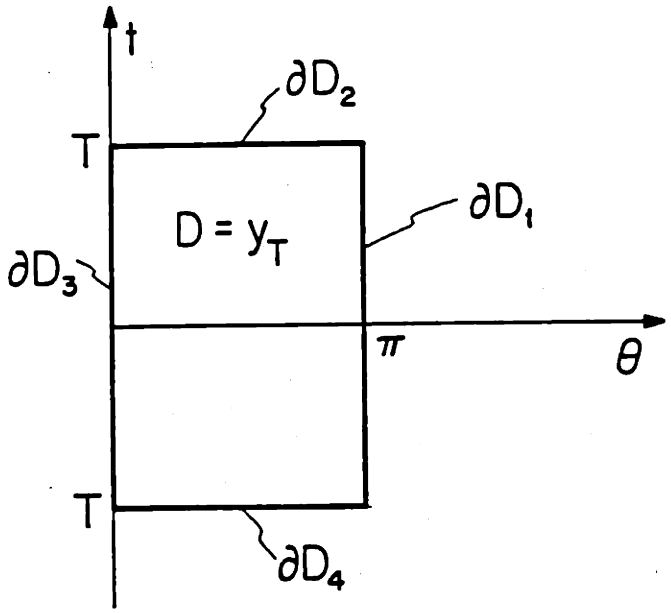


Figure 3.14: The sinogram domain and its boundaries.

The Equilibrium PDE

It is a simple matter to substitute the derivatives in (3.66) into (3.62) to produce the required PDE of (3.42). This equation, together with the original constraints and boundary conditions, and the above boundary condition, form the solution to the minimization problem (V) stated in Section 3.5. Of course, the solution to this equilibrium equation is merely a *stationary function* of (V) — we have not shown sufficient conditions that would guarantee that a stationary function of this problem is, in fact, a global minimum, or even a local minimum. However, because of the convex structure of the variational problem, together with the linear constraints, we can in fact conclude that a stationary function found in this manner is in fact a global minimum of the original variational problem (V) [57].

3.B Lagrange Multiplier Derivations

In this section we derive the expressions for $\lambda_1(\theta)$ and $\lambda_2(\theta)$ given in (3.44) and (3.45) in Section 3.5.1. We begin by integrating both sides of (3.42) with respect to t over the range $[-T, T]$ as

$$\int_{-T}^{T} 2\kappa\bar{X}_G g + \frac{1}{\sigma^2}\chi_Y g - 2\beta g_{tt} - 2\gamma g_{\theta\theta}\, dt = \int_{-T}^{T} \frac{1}{\sigma^2}\chi_Y y - \lambda(\theta) - \lambda_2(\theta) t\, dt \qquad (3.67)$$

The first term on the left-hand side of (3.67) cannot be simplified, however we may simplify the second term. Since χ_Y, by assumption, is either 1 or 0 for any given θ and since the integral of g over $[-T, T]$ is just the mass of the projection, m, then the second term becomes $\frac{m}{\sigma^2}\chi_Y$. The third term becomes just $-2\beta g_t|_{-T}^{T}$, and the fourth term is zero since $\int g_{\theta\theta}\, dt = \frac{\partial^2}{\partial\theta^2}\int g\, dt$ and $\int g\, dt$ is constant in θ. On the right-hand side of (3.67) we cannot simplify the first term, but the second term becomes $-2T\lambda_1(\theta)$ and the third term integrates to zero. The above simplifications lead directly to the desired result in (3.44).

To determine λ_2 we multiply both sides of (3.42) by t and integrate in t over $[-T, T]$ yielding

$$\int_{-T}^{T} 2t\kappa\bar{X}_G g + \frac{t}{\sigma^2}\chi_Y g - 2t\beta g_{tt} - 2t\gamma g_{\theta\theta}\, dt = \int_{-T}^{T} \frac{t}{\sigma^2}\chi_Y y - t\lambda(\theta) - t^2\lambda_2(\theta)\, dt \quad (3.68)$$

The first term on the left-hand side of (3.68) does not simplify; however, the second term is always zero since the center of mass of each observed projection is zero (and we assume that χ_Y has complete projections). The third term may be integrated by parts as

$$\begin{aligned}\int_{-T}^{T} 2\beta t g_{tt}\, dt &= 2\beta t g_t|_{-T}^{T} - 2\beta\int_{-T}^{T} g_t\, dt \\ &= 2\beta t g_t|_{-T}^{T} - 2\beta g|_{-T}^{T}\end{aligned}$$

and the second term of the above expression is zero because of the boundary condition $g(T) = g(-T) = 0$, which leaves just the first term. The fourth term on the left-hand side of (3.68) integrates to zero for the same reasons given for the fourth term of (3.67). On the right-hand side of (3.68), the second term is zero, and the third term yields $2T^3\lambda_2(\theta)/3$. After rearrangement, we are left with (3.45), as desired.

3.C Time and Memory Requirements for ZQPCVX

In this appendix we discuss the time and memory requirements for ZQPCVX to solve the problem denoted by (Q) in Section 3.3.

Memory is of major concern here since several large matrices and their factorizations are stored explicitly. As discussed in Chapter 2, we consider sinograms of several different sizes in this thesis. We wish to compare the storage requirements for the different size models. Let us denote by N the total length of a sinogram (viewed as a vector), i.e., $N = n_v n_d$. And let M denote the total number of constraints, combining equality and inequality; it is easy to see that $M = n_d n_v + 2n_v$. Table 3.2 shows the storage requirements of the matrices used in ZQPCVX as a function of N and M in (a), and in (b) the actual data storage requirements for the different sinogram models. Clearly, the storage requirements for even the modest sized model 3 is enormous at 1900 megabytes. By today's standards, only model 1 is even worth considering implementing. With only a small modification to ZQPCVX (which was not written with such large problems in mind) we implemented model 1 on several test cases; the results are shown in Section 3.5.

Although we have experimental timing results in Section 3.5, it is useful to consider where the burden of computation lies in the ZQPCVX program. In fact, like many iterative algorithms, it is difficult to make a precise timing analysis *a priori* since we do not know how many iterations will be necessary. We may, however, consider what is required for initialization and then what is required for each iteration. The dominant calculation in the initialization phase is the computation of the Cholesky factorization of G. This step alone involves approximately $N^3/6$ multiplications [4]. Powell [66] estimates that the total number of multiplications per iteration is

$$NM + \frac{16}{3}N^2 - 3N|S_l| + \frac{5}{3}|S_l|^2$$

where $|S_l|$ is the number of active constraints in this iteration. Since we do not know *a priori* how many constraints will be active during the computation, let us

Variable	No. Elements
A	NM
G	N^2
g	N
p	N
Workspace	$1.5N^2 + 5.5N + M$
Total	$2.5N^2 + NM + 7.5N + M$

(a)

Model	n_d	n_v	Mbytes (R*4)
1.	41	32	24
2.	81	64	379
3.	121	96	1900
4.	161	128	5973
5.	201	160	14533

(b)

Table 3.2: Storage Requirements for ZQPCVX.

Model	Multiplies $x10^6$ (lowerbound)	Time, hrs 10 μs per multiply
1.	$376 + 2k$	$1 + .006k$
2.	$23,200 + 32k$	$64 + .1k$
3.	$262,000 + 163k$	$725 + .5k$
4.	$1,458,000 + 510k$	$4,052 + 1.5k$
5.	$5,543,000 + 1,241k$	$15,400 + 3.5k$

Table 3.3: Timing Estimates for ZQPCVX.

concentrate on a lower bound. Clearly, each iteration will require at least $NM + 3N^2/16$ multiplications (if Powell's estimate is accurate). Then, for each sinogram model we may construct a lower bound on the number of multiplies in the complete algorithm. We show this in Table 3.3. It is apparent from the table that simply doing the Cholesky decomposition on such large matrices is prohibitive. Once again, model 1 is the only reasonable size to implement in this fashion.

3.D The Local Relaxation Method [47]

In this appendix, we present the essential equations required to implement Kuo's local relaxation algorithm to solve (3.47). We assume the PDE to be of the form[6]

$$-p\frac{\partial^2 u}{\partial x_1^2} - q\frac{\partial^2 u}{\partial x_2^2} + \varsigma(x_1, x_2)u = f(x_1, x_2),$$

$(x_1, x_2) \in [0,1] \times [0,1]$, and to satisfy the conditions given in [47]. Then the PDE is approximated by the 5-point stencil

$$d_{i,j}u_{i,j} - ru_{i+1,j} - lu_{i-1,j} - tu_{i,j+1} - bu_{i,j-1} = s_{i,j},$$

with

$$l = p, \quad r = p, \quad b = q, \quad t = q,$$

$$d_{i,j} = 2p + 2q + \varsigma_{i,j}h^2, \quad s_{i,j} = f_{i,j}h^2$$

[6] A more general form for the PDE and resultant local relaxation solution is given in [47].

where h is the grid spacing and $\varsigma_{i,j}$ is defined as $\varsigma(ih, jh)$. Each grid point is assigned a color, either red or black, according to an alternating pattern as on a checkerboard. Then the local relaxation procedure can be written as:

red points ($i + j$ is even):

$$u_{i,j}^{(n+1)} = (1-\omega_{i,j})u_{i,j}^{(n)} + \omega_{i,j} d_{i,j}^{-1} \left(l u_{i-1,j}^{(n)} + r u_{i+1,j}^{(n)} + b u_{i,j-1}^{(n)} + t u_{i,j+1}^{(n)} + s_{i,j} \right),$$

black points ($i + j$ is odd):

$$u_{i,j}^{(n+1)} = (1-\omega_{i,j})u_{i,j}^{(n)} + \omega_{i,j} d_{i,j}^{-1} \left(l u_{i-1,j}^{(n+1)} + r u_{i+1,j}^{(n+1)} + b u_{i,j-1}^{(n+1)} + t u_{i,j+1}^{(n+1)} + s_{i,j} \right),$$

where $\omega_{i,j}$ is called the *local relaxation parameter* and is given by

$$\omega_{i,j} = \frac{2}{1 + \sqrt{1 - \rho_{i,j}^2}},$$

where

$$\rho_{i,j} = \frac{2}{d_{i,j}} \left(p \cos \frac{\pi}{M_1 + 1} + q \cos \frac{\pi}{M_2 + 1} \right).$$

Chapter 4

SUPPORT LINE CONSISTENCY

4.1 Introduction

This chapter begins a three chapter sequence which deals exclusively with the problem of support estimation. In particular, the ultimate goal, as indicated in Chapter 2 Section 2.5, is to produce a segmentation of the sinogram domain $\mathcal{Y}_T$ into the region of support of the sinogram $\mathcal{G}$ and its complement $\bar{\mathcal{G}}$. The desired segmentation follows directly from knowledge of the convex support of the object — which is the focus of these three chapters — and plays an important role in the full reconstruction algorithms of Chapters 3, 7, and 8.

In this chapter we develop algorithms for reconstructing 2-D convex sets given support line measurements for which the angles are known precisely but the lateral displacements are noisy. As illustrated in Fig. 4.1, perfect measurement of a *projection* — i.e. of a full set of integrals along the parallel lines $L(t,\theta)$ with θ fixed — provides us knowledge of the two extreme lines at this angle that just graze the set on either side. These are known as *support lines*. Note that knowledge of these support lines in this case is completely equivalent to knowledge of the *silhouette* at this angle [90,89], i.e. to a function that is 1 if $L(t,\theta)$ intersects the object and 0 otherwise. Given such support lines from many different angles, it is possible to reconstruct a convex 2-D polyhedron, which contains the object, by intersecting all of the halfplanes defined by the measurements (since each support line also comes with information on which side of the line the object lies). When the projections are

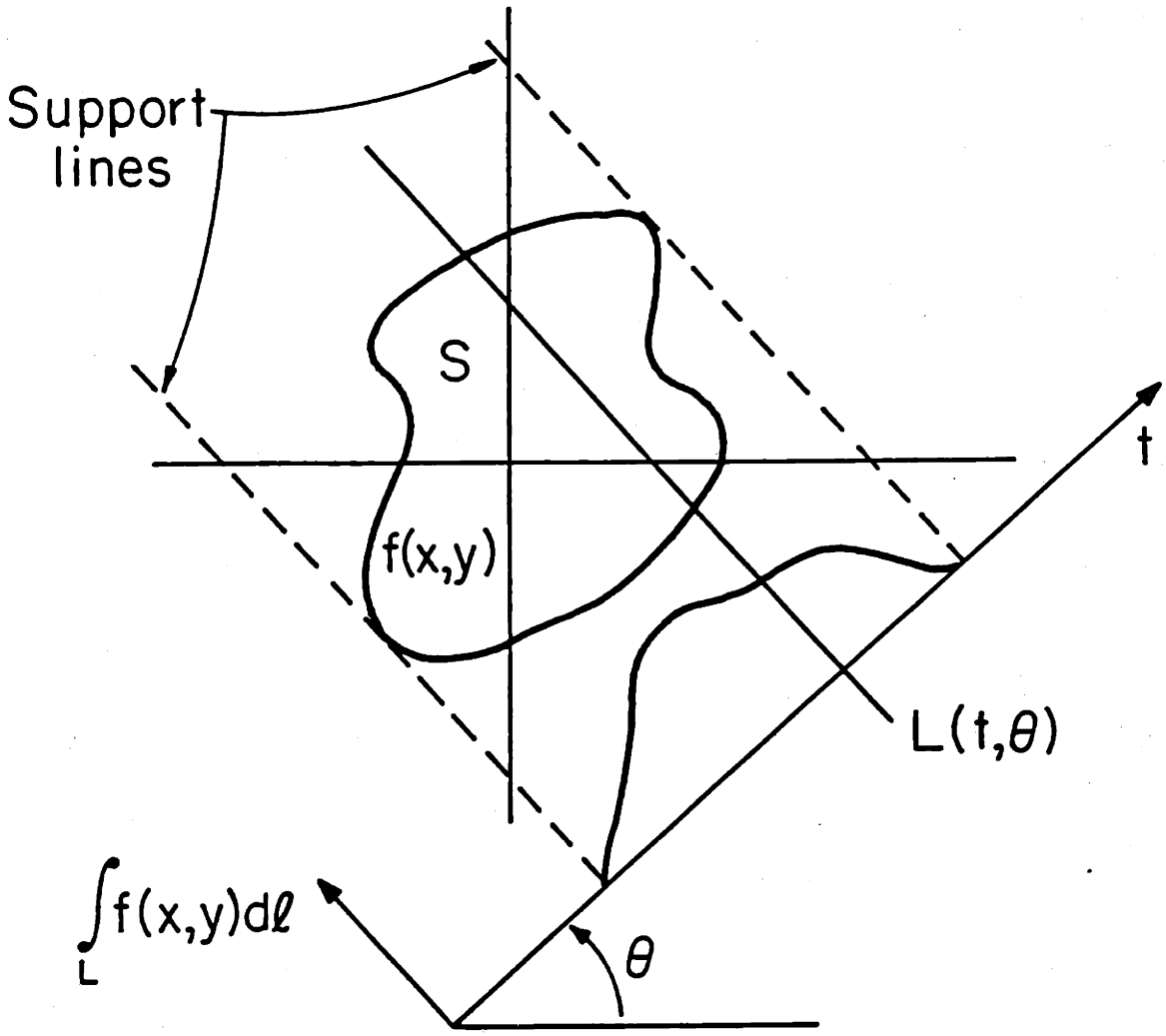


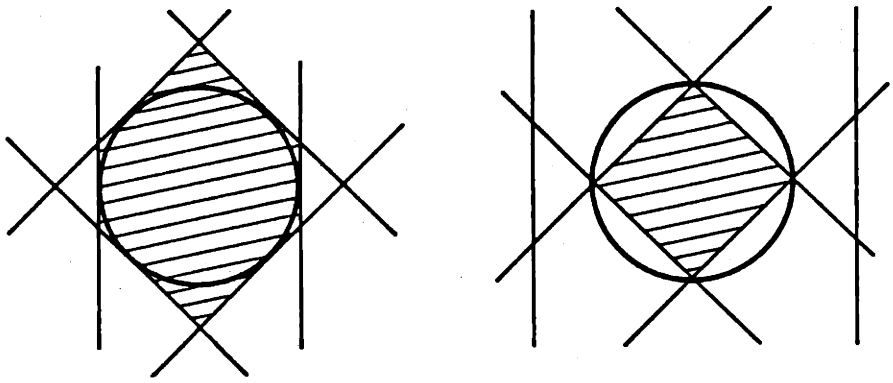
Figure 4.1: The geometry of computed tomography.

noisy, however, such as is the case in low-dose CT, then the estimates of the lateral positions of the support lines will also be noisy. In this case, the set of measured lines may be inconsistent — that is, taken together, there may be no set S which has all of the measured lines as support lines.

The consistency conditions on support lines, which will be discussed in detail later, form the basis of the algorithms presented in this chapter. These algorithms use the consistency requirements, along with known noise statistics and prior information, to reconstruct a convex set which is in a specific sense the optimal estimate based on all the available information. It is also worth noting that the reconstruction problem considered in this chapter is also of interest in a number of other applications. For example, in tactile sensing [78], a parallel plate robot jaw may provide two support line measurements as it clamps down on a "thick 2-D object" which is completely enclosed by the jaw. The jaw may then clamp down from different angles yielding a finite set of support line measurements, as in the CT example above. Other applications include robot vision [37] and chemical component analysis [38].

The problem described in this chapter is fundamentally a problem in computational geometry [67], [29]. In contrast to most work in this field which assumes perfect measurements of information such as points, lines, and sets, and focuses on issues such as algorithm complexity, we focus explicitly on an estimation/optimization theoretic perspective so that we may deal with uncertain measurements and, where appropriate, incorporate prior knowledge. As we will see, the incorporation of measurement error statistics, prior knowledge, and the fundamental constraint on support lines can lead to optimization-based algorithms of considerable efficiency. Indeed the algorithms presented here are implemented with linear programming and quadratic programming methods, both useful tools in computational geometry.

The support line measurements we consider in this chapter have known angles evenly spaced over 2π. In addition, we assume that a support line measurement consists not only of a lateral position, but also indicates on which side of the line the object lies. A natural first guess at a reconstruction then would be to intersect the halfplanes determined by each of the support lines. To see why this intersection



(a) (b)

Figure 4.2: A circle with (a) six true support line, and (b) six noisy measurements.

method might not be a desirable reconstruction and also to give some insight into the fundamental support constraint, consider Fig. 4.2. Fig. 4.2a shows a set of six perfect support line measurements corresponding to the unit circle. The reconstruction resulting from the intersection method is the shaded hexagonal region which is obviously the best reconstruction given these measurements. Suppose now, however, that there are measurement uncertainties and in particular that all six lines have the lateral measurement errors indicated in Fig. 4.2b. In this case, the intersection method produces the diamond shaped estimate indicated by the shaded region. Note that the two vertical lines on either side do not touch the diamond, and in fact, it should be apparent that given the other four measurements as indicated, there is *no* set that has these six lines as support lines. This demonstrates, geometrically, what is meant by inconsistency. Now consider what the diamond estimate implies about the noise model. What this estimate is telling us is that the two vertical lines (the outermost lines) are in error, and that the remaining four lines (the innermost lines) are perfect. Obviously, this does not correspond to any

reasonable noise model, in general. The algorithms developed in this chapter, in contrast to the intersection method, use explicit noise models to develop optimum methods given the model.

The chapter is organized as follows. In Section 4.2, we define the support vector and describe the fundamental support line constraints, i.e., the consistency conditions. In Section 4.3 we define the set of all consistent support vectors, called the support cone, and elaborate on the geometry of the support cone and of objects represented by points in this cone. Section 4.4 presents the noise models and algorithms that use the geometry of the support cone to advantage, and Section 4.5 contains experimental results. We give concluding remarks in Section 4.6, including a brief discussion of how more elaborate models of prior shape information can be included.

4.2 Support Line Constraints

4.2.1 Support Lines and Support Functions

Fig. 4.3 shows what is meant by the support line $L_S(\theta)$ of a set S. It is the line orthogonal to the unit normal ω which just "grazes" S in the positive ω direction. The quantity $h(\theta)$ is the value of the largest possible projection of any point in S onto the ω-axis. One can see that S lies completely in a particular one of the two halfplanes determined by $L_S(\theta)$. We may now define the above quantities precisely. The *support line* at angle θ for the closed and bounded 2-D set S is given by

$$L_S(\theta) = \{x \in \mathbb{R}^2 \mid x^{\mathrm{T}}\omega = h(\theta)\} \tag{4.1}$$

where $\omega = [\cos\theta \;\; \sin\theta]^{\mathrm{T}}$ and

$$h(\theta) = \sup_{x \in S} x^{\mathrm{T}}\omega \;\; . \tag{4.2}$$

The function $h(\theta)$ is called the *support function* of the set S; for any particular value of θ we call $h(\theta)$ the *support value* at angle θ.

The support function $h(\theta)$ has important and well-known properties which are analogous to properties we shall be developing for the support vector defined below

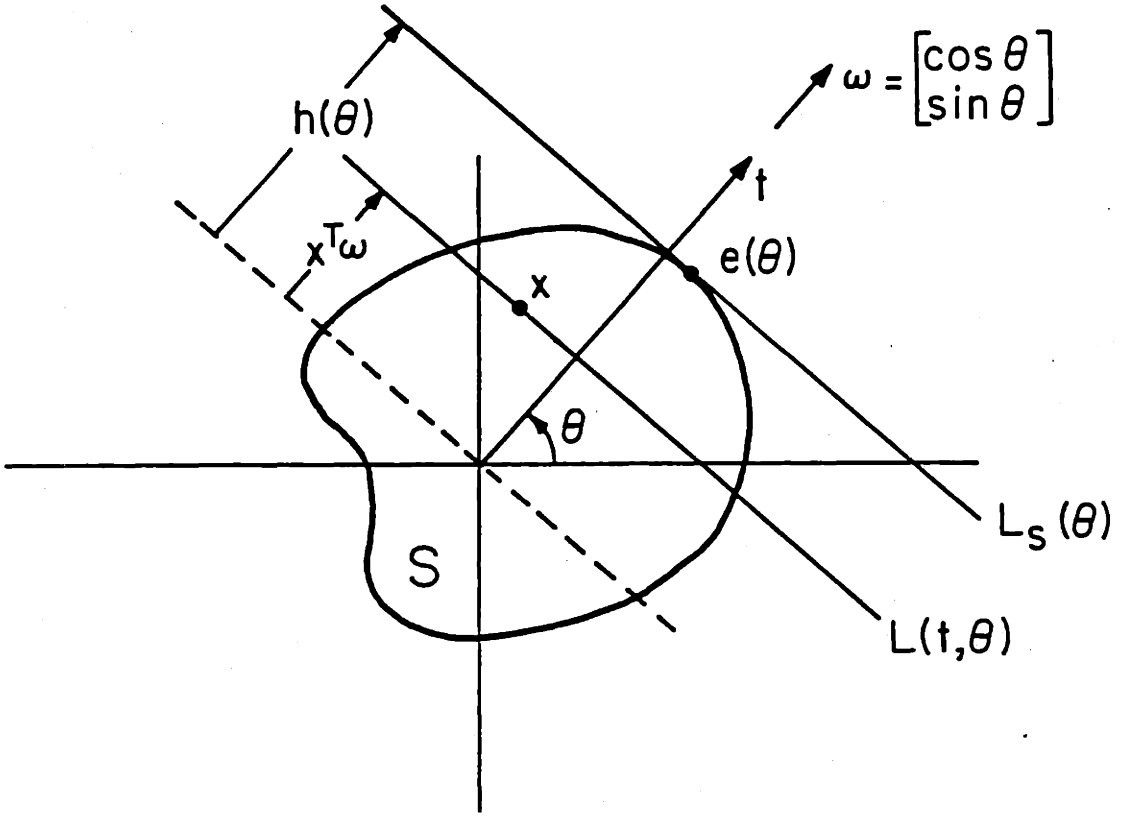


Figure 4.3: The geometry of support lines.

(see [77], [42], and [80]). For example, $h(\theta)$ uniquely determines the convex hull of S, hul(S). It is also true that if $h(\theta)$ is twice differentiable then S itself must be convex, and the boundary of S must be continuous and smooth (i.e., it has continuously turning normals). In this case, the curvature of the boundary of S at the boundary point $e(\theta) = L_S(\theta) \cap S$ (see Fig. 4.3) is given by $h''(\theta) + h(\theta)$. Then, since S is convex, the curvature of its boundary must be non-negative, which leads to the conclusion that support functions which are twice differentiable must satisfy the constraint

$$h''(\theta) + h(\theta) \geq 0 \quad . \tag{4.3}$$

The constraint we derive below is analogous to (4.3), but is more fundamental since it applies to *any* set in the plane, not just convex sets with smooth boundaries. We shall also develop an analog to the radius of curvature which will be exploited by algorithms designed to incorporate prior knowledge.

4.2.2 Support Vectors and Constraints

We shall require some additional notation in this section. From this point on, we consider a finite number M of angles $\theta_i = 2\pi(i-1)/M$, $i = 1, \ldots, M$, spaced evenly over $[0, 2\pi)$, and associated sets of lines L_i, orthogonal to the corresponding unit vector $\omega_i = [\cos\theta_i \ \sin\theta_i]^{\mathrm{T}}$. In what follows the index i is always interpreted modulo M. The line L_i is defined by its lateral displacement h_i, via

$$L_i = \{u \in \mathbb{R}^2 \mid u^{\mathrm{T}}\omega_i = h_i\} \tag{4.4}$$

The most important quantity in this chapter is the vector made by organizing the M lateral displacement values of the M lines under consideration as a vector $h = [h_1 \ h_2 \ \ldots \ h_M]^{\mathrm{T}}$. We call the vector h a *support vector* if the lines L_i, for $i = 1, \ldots, M$ are support lines for some set $S \in \mathbb{R}^2$, i.e. if $h_i = h(\theta_i)$ where $h(\theta)$ is the support function of some set S. In this case we refer to the h_i as *support values.*

Before proceeding to the basic theorem of this chapter, let us characterize, in terms of the quantities defined above, the estimate produced by the intersection method introduced in Section 4.1. Given measurements h_i, $i = 1, \ldots, M$ of the

lateral displacements of M lines, the intersection method simply produces the set of all points $u \in \mathbb{R}^2$ which satisfy $u^{\mathrm{T}}\omega_i \leq h_i$ for all $i = 1, \ldots, M$, i.e.[1]

$$S_B = \{u \in \mathbb{R}^2 \mid u^{\mathrm{T}}[\omega_1\, \omega_2 \ldots \omega_M] \leq [h_1\, h_2 \ldots h_M]\} \ . \tag{4.5}$$

The two shaded regions in Fig. 4.2 correspond to S_B for two different vectors h. In Fig. 4.2a, h is a support vector since the lines actually support S_B, however in Fig. 4.2b, h is not a support vector because there is no set which the given lines support. We now proceed to state the basic theorem of this chapter, which characterizes precisely the consistency constraints satisfied by support vectors.

Theorem 4.1 (The Support Theorem)

A vector $h \in \mathbb{R}^M$ $(M \geq 5)$ is a support vector if and only if

$$h^{\mathrm{T}}C \leq [0 \ldots 0] \tag{4.6}$$

where C is an M by M matrix given by

$$C = \begin{bmatrix} 1 & -k & 0 & & & -k \\ -k & 1 & -k & \ldots & \ldots & 0 \\ 0 & -k & 1 & & & \vdots \\ \vdots & 0 & -k & \ldots & \ldots & 0 \\ 0 & \vdots & & & & -k \\ -k & 0 & 0 & & & 1 \end{bmatrix} \tag{4.7}$$

and $k = 1/(2\cos(2\pi/M))$. □

It is important to point out the similarity between the continuous support function constraint of (4.3) and the the discrete support vector constraint of (4.6). The quantity $-h^{\mathrm{T}}C$, which has *non-negative* entries, is analogous to the quantity $h''(\theta) + h(\theta)$, which is also non-negative. It can be shown, in fact, that in the limit as $M \to \infty$ the expression $-h^{\mathrm{T}}C \geq 0$ goes to $h''(\theta) + h(\theta) \geq 0$ [70]. As a further

[1] A vector inequality such as $x^{\mathrm{T}} \leq y^{\mathrm{T}}$ where $x, y \in \mathbb{R}^n$ implies that $x_i \leq y_i$ for $i = 1, \ldots, n$, where x_i and y_i are the i^{th} elements of the vectors x and y respectively.

extension of the analogy, we shall reveal in a subsequent section that the entries of the vector $-h^T C$ can be directly interpreted from the geometry as a type of discrete radius of curvature. This interpretation allows us to propose methods for incorporating prior shape information related to boundary smoothness in the algorithms of Section 4.3.

Before proceeding with the proof, we give a brief indication of the geometric intuition behind it. First, consider the situation depicted in Fig. 4.4, in which we have shown two lines L_{i-1} and L_{i+1}. A third line L_i is parallel to the dashed line in the figure, and we seek constraints on the lateral displacement of this line so that the 3 lines L_{i-1}, L_i, and L_{i+1} could possibly be support lines of some set. If L_{i-1} and L_{i+1} are support lines of a set S, then S is contained in the set D_i illustrated in the figure. Now suppose that the line L_i were located to the left of (and parallel to) the dotted line. Then it is possible to construct a set $S \subset D_i$ which touches each of the three lines L_{i-1}, L_i, and L_{i+1} — these lines are consistent. However, if L_i were measured to the right of the dotted line, then it is impossible to construct such a set — these lines are inconsistent. When stated in mathematical notation and applied to all lines L_i, $i = 1, \ldots, M$, this relationship yields precisely the vector constraint in (4.6).

The above observation leads to the necessity of (4.6), but in order to establish the sufficiency of (4.6) we need to define a new set $S_\nu \in \mathbb{R}^2$ which may be thought of as another choice of reconstruction, different than S_B. As shown in Figs. 4.5, 4.6, and 4.7, S_ν is formed from the convex hull of the points of intersection of lines L_i and L_{i+1} for $i = 1, \ldots, M$. Formally, we have that

$$S_\nu = \text{hul}(\nu_1, \nu_2, \ldots, \nu_M) \tag{4.8}$$

where the ν_i's are given by

$$\nu_i = L_i \cap L_{i+1} \tag{4.9}$$

and hul(.) denotes the convex hull. We refer to the points ν_i as *vertex points* rather than vertices because, as one can see from Fig. 4.6, they need not be distinct points. In Fig. 4.5 the support line L_1 is located to the right of the point $L_2 \cap L_5$, and from our discussion on Fig. 4.4, we know that these lines do not satisfy (4.6). Note that

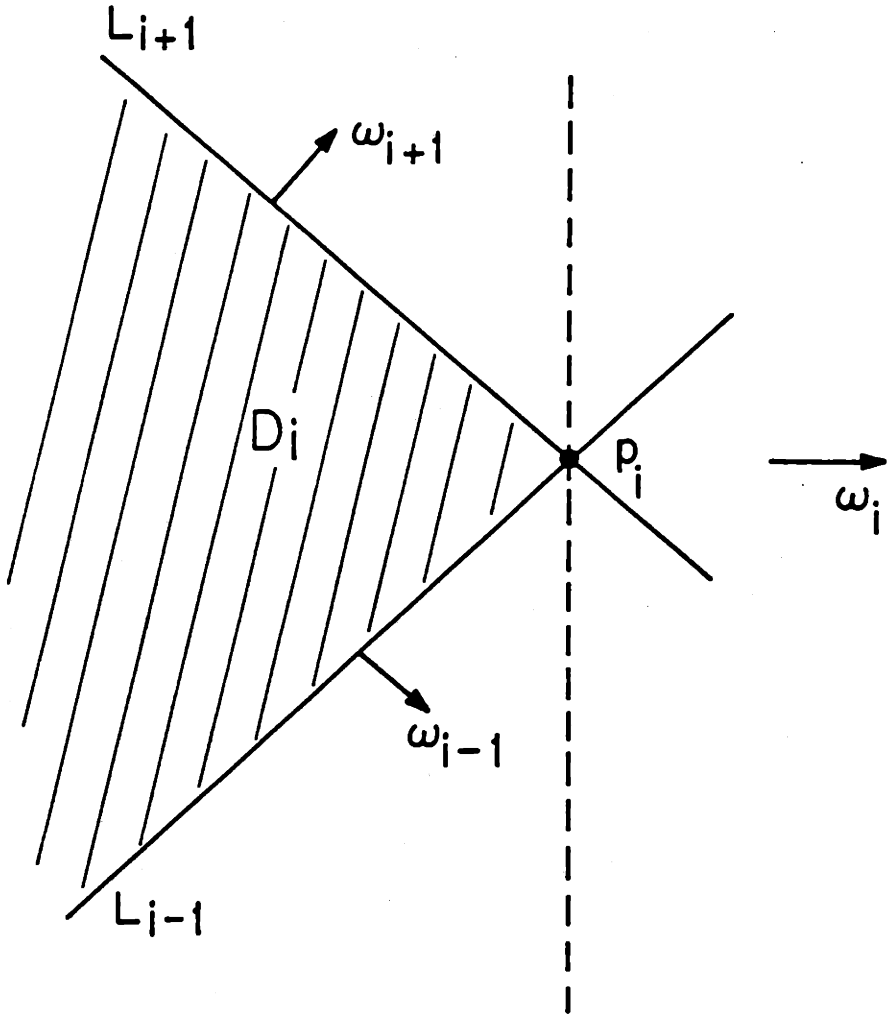


Figure 4.4: For consistency, line L_i must lie to the left of the dotted line.

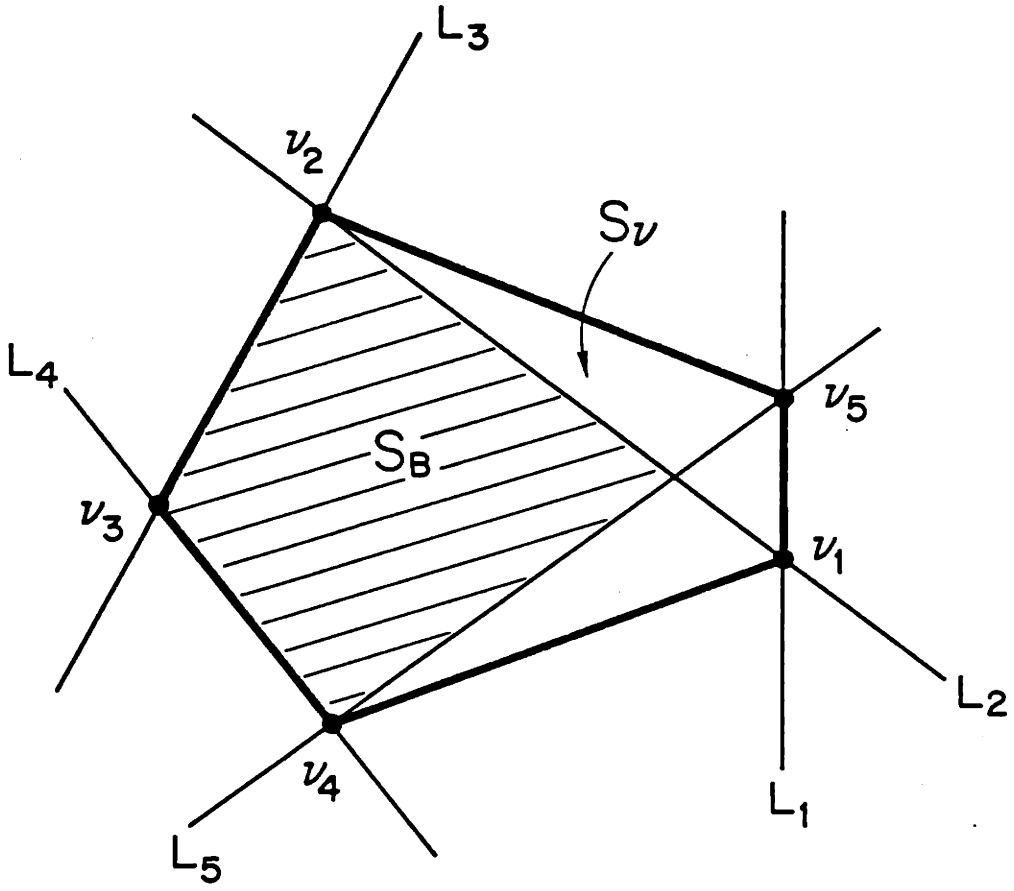


Figure 4.5: Inconsistent lines, the sets S_B and S_ν, and the vertex points ν_i.

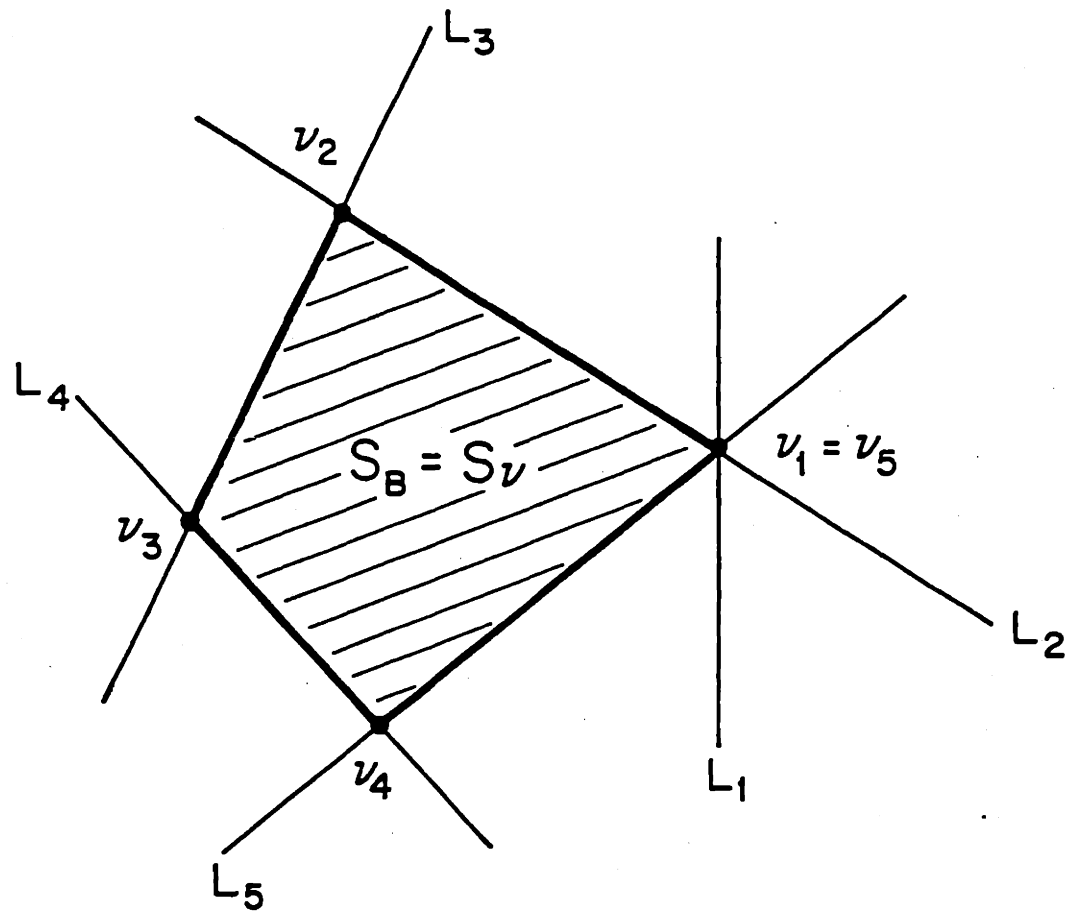


Figure 4.6: Consistent lines, the sets S_B and S_ν, and the vertex points ν_i.

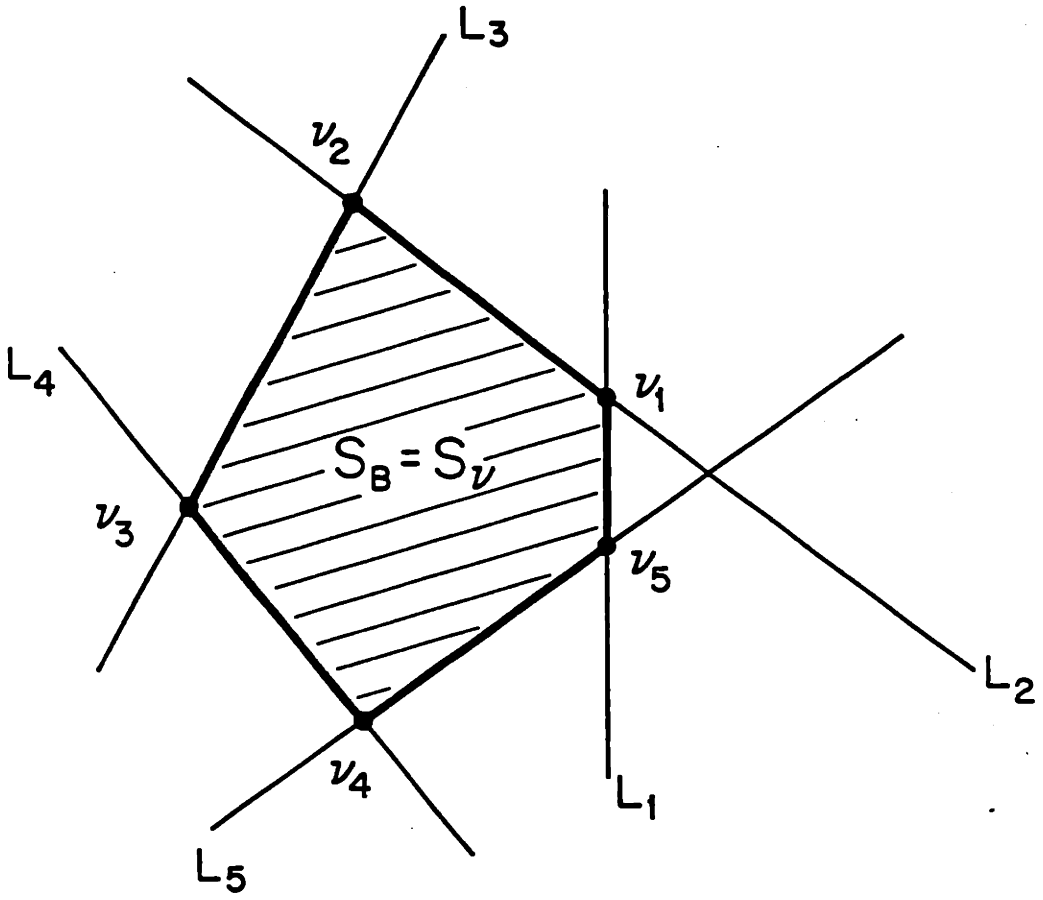


Figure 4.7: Consistent lines, the sets S_B and S_ν, and the vertex points ν_i.

in this case $S_B \neq S_\nu$. However, in Figs. 4.6 and 4.7 the lines do satisfy (4.6) and $S_B = S_\nu$. Indeed what we show in the proof is that (4.6) implies that $S_B = S_\nu$ and h is the support vector to this set.

Proof of Theorem 1:

First, we show the necessity of condition (4.6). By hypothesis, h is a support vector of some set S. Now consider the set D_i defined by the two support lines L_{i-1} and L_{i+1} as shown in Fig. 4.4. Note that by hypothesis $M \geq 5$, which implies that $\theta_{i+1} - \theta_{i-1} < \pi$. This in turn implies that the two lines L_{i-1} and L_{i+1} have a finite intersection point p_i (see Fig. 4.4), and that ω_i may be written as a positive combination of ω_{i-1} and ω_{i+1}. These two facts are necessary and in fact easily allows us to conclude that the support value at angle θ_i for the set D_i is $p_i^{\mathrm{T}}\omega_i$. Then, since $S \subset D_i$ we must have that $h_i \leq p_i^{\mathrm{T}}\omega_i$. With some algebraic manipulation (see Appendix 4.A), this inequality may be shown to be equivalent to the condition given by the i^{th} column of (4.6).

To prove the sufficiency of (4.6) we must show that a vector h which satisfies (4.6) is a support vector for some set. In Appendix 4.B we show that (4.6) implies that $S_B = S_\nu = S$, and what remains then is to show that S has h as its support vector. To see this, first note from the definition of S_B in (4.5), that we must have

$$\sup_{x \in S} x^{\mathrm{T}}\omega_i \leq h_i.$$

On the other hand, $\nu_i \in S_\nu = S_B = S$ and $\nu_i^{\mathrm{T}}\omega_i = h_i$. Consequently, h_i is the support value at this angle. □

The immediate use of the support theorem is as a test of consistency. Given a test vector h we may determine whether h specifies a consistent set of support lines by evaluating $h^{\mathrm{T}}C$ and seeing whether the elements of the resultant row vector are all non-positive. From an estimation viewpoint, we see that if we are trying to estimate a support vector h from a set of noisy measurements, then we must make sure that our estimate $\hat{h}$ satisfies $\hat{h}^{\mathrm{T}}C \leq 0$. In the following section we examine the geometry of these constraints in more detail.

4.3 Object and Support Cone Geometry

4.3.1 Geometry of the Support Cone

The convex polyhedral cone given by

$$C = \{h \in \mathbb{R}^M \mid h^T C \leq [0 \ldots 0]\} \tag{4.10}$$

consists of all M-dimensional support vectors. We call C the *support cone.*[2] The matrix C is circulant and, therefore, its eigenvalues are given by the discrete Fourier transform of the first row [3]. After simplification (see Appendix 4.A), the eigenvalues are found to be

$$\lambda_k = 1 - \frac{\cos(2\pi(k-1)/M)}{\cos(2\pi/M)} \quad k = 1, \ldots, M.$$

We now recognize that exactly two eigenvalues are identically zero: $\lambda_2 = \lambda_M = 0$. Hence, C is singular, and a basis for the nullspace $\mathcal{N}$ (and also of the left nullspace since C is symmetric), is found to be

$$\begin{aligned} n_1 &= \left[\, 1 \quad \cos\theta_o \quad \cos 2\theta_o \quad \ldots \quad \cos(M-1)\theta_o \,\right]^T \\ n_2 &= \left[\, 0 \quad \sin\theta_o \quad \sin 2\theta_o \quad \ldots \quad \sin(M-1)\theta_o \,\right]^T \end{aligned} \tag{4.11}$$

where $\theta_0 = 2\pi/M$.

The geometrical consequence of C being singular is that the support cone $\mathcal{C}$ is not a proper cone; i.e., there is a linear subspace (of dimension 2) contained entirely in $\mathcal{C}$. Therefore, the support cone is composed of the Cartesian product of a proper cone, $\mathcal{C}_p = \{h \in \mathcal{C} \mid h^T n_1 = 0,\ h^T n_2 = 0\}$, and $\mathcal{N}$, the nullspace of C. Accordingly, any support vector may be written as the sum of two orthogonal components, h_p and h_n, as

$$h = h_p + h_n \tag{4.12}$$

where $h_p \in \mathcal{C}_p$ and $h_n \in \mathcal{N}$. We will see in the following section that the nullspace component of a support vector h may be interpreted as a simple *shift* of the set in the plane that corresponds to h.

[2] $\mathcal{C}$ is a cone because it obeys the usual property of cones: if h is in $\mathcal{C}$ then αh ($\alpha > 0$) is also in $\mathcal{C}$. It is a polyhedron because it is the intersection of a finite number of closed half spaces in $\mathbb{R}^M$.

4.3.2 Object Geometry

Given a (consistent) support vector h, there are, in general, an entire family of sets which have h as their support vector. The largest of these sets, which is uniquely determined by h is the polygonal set S_B defined in (4.5). We call S_B the *basic object* of support vector h. Two examples of basic objects for $M = 5$ are shown in Figs. 4.6 and 4.7. Note that for M small, S_B may not be a good approximation to the true set S, but as M gets larger, S_B becomes an increasingly better approximation to $\mathrm{hul}(S)$.

Suppose we were to add a nullvector h_n to support vector h. What happens to the basic object? We show here that it is simply *shifted* (or translated) in the plane. We start by noting that any nullvector may be written as

$$h_n = Nv \tag{4.13}$$

where

$$N = [n_1 \; n_2] \tag{4.14}$$

(see (4.11)) and v is a two-dimensional vector. Next, we notice that S_B may be written as

$$S_B = \{u \in \mathbb{R}^2 \mid u^{\mathrm{T}} N^{\mathrm{T}} \leq h^{\mathrm{T}}\} \quad .$$

Now suppose that w is an element of S_B; then, clearly, w satisfies

$$h \geq Nw \quad . \tag{4.15}$$

Now we may add, component by component, equations (4.13) and (4.15) (preserving the inequality) yielding

$$h + h_n \geq N(w + v) \quad .$$

Finally, we now see that $w + v$ must be an element of the basic object corresponding to $h + h_n$, i.e., the new basic object is just a shifted version of S_B. Clearly, the reverse holds as well: shifting S_B by v corresponds to adding the nullvector Nv to h.

The extreme points of the basic object, which we have termed vertex points, are given by the points $\nu_1, \ldots, \nu_M$ in (4.9) (see Figs. 4.6 and 4.7). An explicit equation

for the vertex point ν_i is easily found using the definition of L_i and L_{i+1} and solving a system of two linear equations (see Appendix 4.A). We find that

$$\nu_i^{\mathrm{T}} = \frac{1}{\sin\theta_o}\begin{bmatrix} h_i & h_{i+1} \end{bmatrix}\begin{bmatrix} \sin\theta_{i+1} & -\cos\theta_{i+1} \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \quad i = 1,\ldots,M \ . \tag{4.16}$$

where $\theta_o = \theta_{i+1} - \theta_i = 2\pi/M$. The "shift" property given above relates to the *relative* position of two identically-shaped and oriented basic objects. It turns out that a useful definition of the *absolute* position of a basic object is the average position of its vertex points, denoted $\bar{\nu}$. The relationship between the support vector h and $\bar{\nu}$ is found to be (see Appendix 4.A)

$$\bar{\nu} = \begin{bmatrix} \bar{\nu}_x \\ \bar{\nu}_y \end{bmatrix} = \frac{1}{M}\sum_{i=1}^{M} \nu_i = \frac{2}{M} N^{\mathrm{T}} h \ . \tag{4.17}$$

We shall see in Section 4.4 that (4.17) can be used as a constraint on estimated support vectors if the position of the true object is known *a priori*. Note, in particular, that when h has no nullspace component, i.e., h is in C_p, then $N^{\mathrm{T}}h = 0$ and, therefore, $\bar{\nu} = 0$ — the basic object is centered on the origin.

Now we develop the idea of "discrete radius of curvature" to characterize the smoothness of the boundaries of basic objects. Suppose that in Fig. 4.4, the line L_i were to pass through the intersection point p_i of L_i and L_{i+1}. Then the boundary of S_B is "sharp" at that point. As L_i moves toward the left of p_i, the boundary is made "smoother". Now consider the more detailed drawing in Fig. 4.8. As the boundary is traced along the i^{th} face from ν_{i-1} to ν_i, the outward unit normal to the boundary changes in angle by $\theta_0 = \theta_i - \theta_{i-1}$ over a distance f_i. In analogy to the usual radius of curvature, which is defined as the rate of change of arclength with respect to the angle the unit normal makes to the x-axis, we define the i^{th} *discrete radius of curvature* as

$$r_i = \frac{f_i}{\theta_0} \ . \tag{4.18}$$

It can be shown from the geometry (see Appendix 4.A) that the distance from p_i to L_i is given by $\rho_i = -h^{\mathrm{T}}c_i$, where c_i is the i^{th} column of C. Then, by simple trigonometry, we have that

$$f_i = \frac{2\rho_i}{\tan\theta_0} \tag{4.19}$$

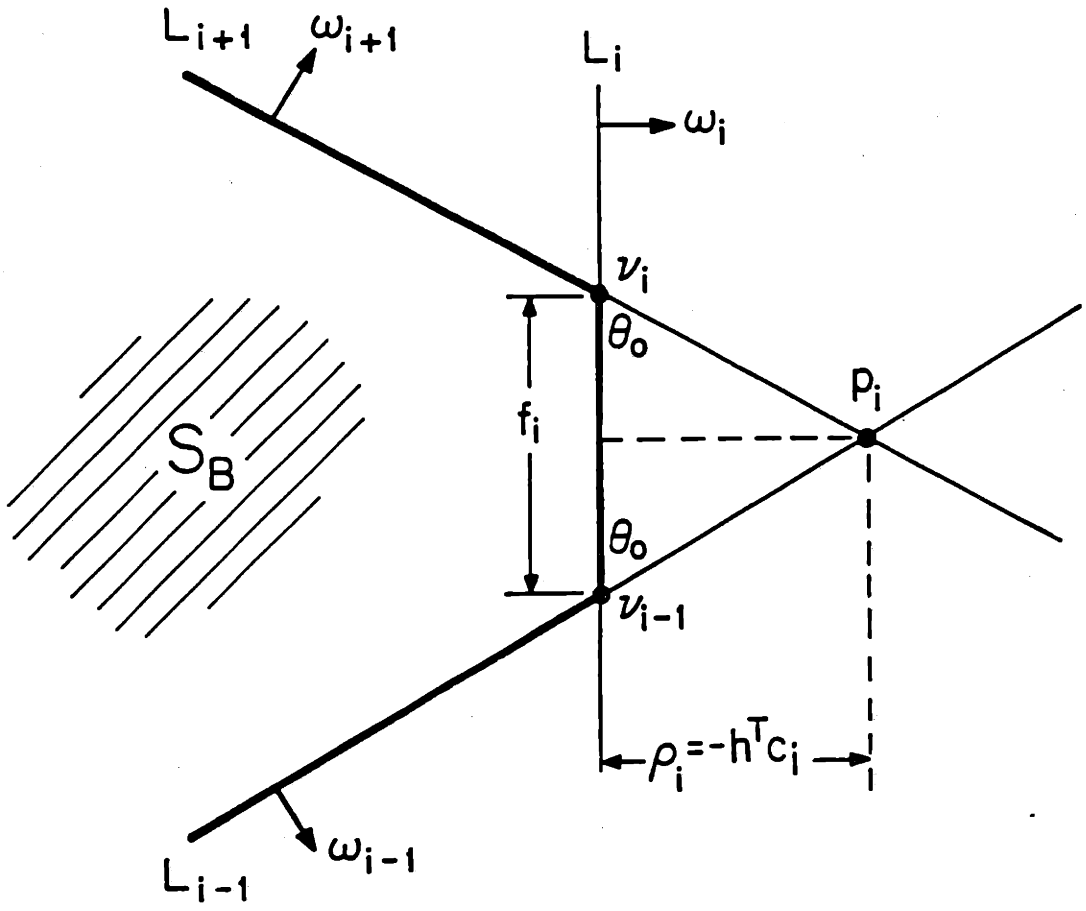


Figure 4.8: Three support lines and a face of S_B.

and, hence

$$\rho_i = \frac{1}{2} r_i \theta_0 \tan \theta_0 \tag{4.20}$$

Hence, the vector $\rho = -h^T C$ has elements that are proportional to the discrete radii of curvature, r_i. The elements of ρ which are small correspond to "sharp" corners; the larger elements correspond to "smoother" boundaries. We use this idea in Section 4.4 to incorporate prior knowledge about object shape.

This completes the discussion of geometry of the support cone and basic objects. Using the constraints established in Section 4.2 and the geometrical ideas established in this section, we proceed to develop algorithms for estimating support vectors (and hence the basic objects) given noisy observations. The geometrical ideas play a role both in the development of prior information to be included in the statement of the algorithms, and in the execution and analysis of the actual computational methods.

The algorithms we develop in the following section are constrained optimization algorithms because the support vectors to be estimated are constrained to lie in the support cone. Fortunately, the constraints are linear inequalities, which are simple enough to allow efficient computational methods. A further constraint which may be imposed if the position of the object is known a priori, is a linear equality constraint, which is even simpler. The algorithms are designed to illustrate how to incorporate these constraints along with prior information and noise models to reconstruct convex sets. We have elected to demonstrate only the simplest formulations necessary to accomplish this goal. As a result, the algorithms use the very efficient computational methods of linear programming (LP) and quadratic programming (QP). In Section 4.6, we discuss possible extensions which include more sophisticated models of prior information and that will undoubtedly lead to somewhat more complex algorithms.

4.4 Estimation Algorithms

We now present three estimation algorithms based on the ideas developed in Sections 4.2 and 4.3. We assume that the measured support values are given by

$$y_i = h_i + n_i, \quad i = 1, ..., M \tag{4.21}$$

where h_i are the true support values which we are estimating and n_i are samples of either 1) independent white Gaussian noise with zero mean and variance σ^2, or 2) uniformly distributed noise over the range $[-\gamma, \gamma]$. Because of the noise, it is likely that the measurement vector $y = [y_1 \ \ldots, \ y_M]^T$ is not a feasible support vector. Therefore, the first objective of the following algorithms is to obtain a feasible support vector from the measurements. The second objective is to use prior information to guide the estimates toward "preferable" values. The development begins with the Closest algorithm, which uses a minimum of prior knowledge in a maximum likelihood (ML) formulation, and concludes with the Close-Min algorithm, which uses prior shape information in a formulation much like maximum *à posteriori* (MAP) estimation. The algorithms also tend to increase in complexity as we proceed, but are each solved by efficient quadratic or linear programming methods.

4.4.1 The Closest Algorithm

Here, we assume the Gaussian noise model given above. In the absence of any prior probabilistic knowledge we may form the maximum likelihood estimate of h given the measurement vector y and subject to $h \in C$ as (see, for example, [91])

$$\hat{h}_C = \hat{h}_{ML} = \underset{h:\, h^T C \leq 0}{\operatorname{argmax}} \ -\frac{1}{2}(y-h)^T(y-h) \ . \tag{4.22}$$

We see that this estimate is the support vector h in C which is *closest* (in the Euclidean metric) to the observation y. If y is in C then $\hat{h}_C = y$, otherwise the solution may be found by (efficient) quadratic programming (QP) methods (see, for example, [48] and [28]).

4.4.2 The Mini-Max Algorithm

The Mini-Max algorithm incorporates the following prior knowledge: objects of interest tend to have smooth boundaries. To cause objects to have smooth boundaries we define the Mini-Max estimate to *maximize the minimum discrete radius of curvature.* As the problem is stated, however, the solution is unbounded, since basic objects circumscribing circles of ever increasing radii have unbounded discrete radii of curvature. This problem is partially solved by incorporating the uniform noise model. In this case, since the noise is bounded by $\pm\gamma$, each element of the true solution cannot be farther than γ away from the corresponding element of the observation. Formally, we write that the true vector, and therefore the estimate, must be an element of the hypercube

$$\mathcal{B} = \{h \in \mathbb{R}^M \mid y - [\gamma\,\gamma\,\ldots\,\gamma]^{\mathrm{T}} \leq h \leq y + [\gamma\,\gamma\,\ldots\,\gamma]^{\mathrm{T}}\} \tag{4.23}$$

Finally, recognizing that the estimate must also be in the support cone, and recalling the proportionality of $\rho_i = -h^{\mathrm{T}}c_i$ to the discrete radius of curvature r_i (see (4.20)), we define the Mini-Max estimate as

$$\hat{h}_{MM} = \underset{h:\ h\in C\cap\mathcal{B}}{\operatorname{argmax}}\ \min\{-h^{\mathrm{T}}c_1,\ -h^{\mathrm{T}}c_2,\ \ldots,\ -h^{\mathrm{T}}c_M\} \tag{4.24}$$

where $c_1, \ldots, c_M$ are the columns of C.

The solution to (4.24) may be found by linear programming (LP) techniques (see [56], for example). To show that this is so, we define a new scalar variable μ which satisfies

$$\mu \leq -h^{\mathrm{T}}c_i \quad i = 1, \ldots, M\ . \tag{4.25}$$

Now consider the two augmented vectors

$$u = \begin{bmatrix} h \\ \\ \mu \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix}\ . \tag{4.26}$$

We now notice that the solution to (4.24) may be found by maximizing $u^{\mathrm{T}}b$, subject to the original constraints and the new constraints given in (4.25). The new

objective function is clearly linear in u; and both sets of constraints are linear in u. Therefore, the augmented problem is an LP and may be solved by any LP code, or a QP code with the Hessian matrix set to zero.

Unfortunately, as is often true of LP's, the solution to (4.24) may not be unique. We may see a potential non-uniqueness by observing that adding a vector from the nullspace of C does not change the value of the objective function. Therefore, providing that the constraints are still met, there may be a family of shifted objects, each one corresponding to an optimal solution to (4.24). The Mini-Max estimate is also tied to the observations only by the hypercube $\mathcal{B}$, and as γ (and therefore the size of $\mathcal{B}$) increases, the influence of the measurements on the solution may decrease dramatically. For example, we expect that the basic object corresponding to the estimate resulting from this objective function will be as largest possible given the bounds, and as near to circular as possible so as to maximize the minimum discrete radius of curvature. Thus, even if the true object is quite eccentric, and the observation is just barely infeasible, the Mini-Max estimate may resemble a circle if the bound γ is large. We shall see examples of both types of behavior in Section 4.5. In addition, these observations provided part of the motivation for the next algorithm.

4.4.3 The Close-Min Algorithm

The Close-Min algorithm is designed to combine the Closest and Mini-Max algorithms to produce an estimate which attempts to match the observations, as in the Closest algorithm, yet also incorporate prior knowledge, as in the Mini-Max algorithm. The concept is simple: we define a new cost function which is a convex combination of the two objective functions. We note that this method resembles MAP estimation where the Closest objective function plays the role of the logarithm of the measurement density (assuming the Gaussian model), and the Mini-Max objective function plays the role of the logarithm of the prior density. The trade-off between these two objective functions is controlled by the parameter α which has a value between 0 and 1. This provides the means for weighting prior information and that available from the measurements as is done in optimal MAP estimation.

The Close-Min estimate is defined as

$$\hat{h}_{CM} = \underset{h:\ h \in C \cap B}{\operatorname{argmax}}\ \alpha f_C(h) + (1-\alpha) f_M(h) \tag{4.27}$$

where $0 \leq \alpha \leq 1$ and

$$\begin{aligned} f_C(h) &= -\frac{1}{2}(y-h)^{\mathrm{T}}(y-h) \\ f_M(h) &= \min\{-h^{\mathrm{T}}c_1,\ -h^{\mathrm{T}}c_2, \ldots, -h^{\mathrm{T}}c_M\} \end{aligned}$$

are the objective functions corresponding to the Closest and Mini-Max algorithms, respectively. The solution to (4.27) may be found using QP after augmenting h as in (4.26). Note that provided $\alpha \neq 0$, the constraint B may be removed and the solution will be unique.

4.4.4 Shift Corrected Algorithms

As we suggested previously, prior positional information may be included in the estimation process. Suppose one knows that the true object is centered at $\bar{\nu}$, that is, that the average position of its vertex points is $\bar{\nu}$. Then the estimate should also be centered at $\bar{\nu}$. From Equation (4.17) we see that this may be assured provided that we enforce the following linear constraint

$$N^{\mathrm{T}}h = \frac{M}{2}\bar{\nu} \tag{4.28}$$

Since this is a linear equation, (4.28) may be incorporated into the three algorithms as an additional linear constraint causing no essential change in the nature of the solution method. The effect of this added prior knowledge can be quite dramatic, however, as we shall see in the following section.

4.5 Experimental Results

To show the behavior of the three algorithms, we use noise-corrupted measurements of a 10-dimensional support vector corresponding to either 1) a circle with radius 1/2, centered on the origin or, 2) an ellipse, also centered on the origin, with major

axes in the x-direction with radius 3/4 and y-direction with radius 1/3. The measurements are given by (4.21) where n_i are independent random variables, uniform over the range $[-\gamma, \gamma]$, with several values of γ. To plot the data (for either the feasible support vectors or infeasible observations) we simply connect the vertex points $\{\nu_1, \nu_2, \ldots, \nu_M, \nu_1\}$ in sequence, producing a *vertex plot.* For a (feasible) support vector, this plot produces an outline of the basic object; however, for a (infeasible) measurement, the plot crosses itself, clearly demonstrating the infeasibility. We refer to a point where a vertex plot crosses itself as a *point of inconsistency.*

Figs. 4.9a and 4.10a show both the true basic object corresponding to the circle (dashed line) and the vertex plot for the measured vector (solid line), where $\gamma = 0.2$ and $\gamma = 0.4$, respectively. Figs. 4.11a and 4.12a show the corresponding figures for the ellipse. The shaded regions shown in the (a) panels of Figs. 4.9–4.12 are estimates produced by the intersection method which is described in Sections 4.1 and 4.2. One can see that, in each case, there is at least one measured line which does not support the shaded region, which clearly demonstrates the infeasibility of the measurements. It is important to point out that the set constructed from the raw measurements using the intersection method is a bad estimate of the true set, in general. This is because, as mentioned before, the construction of this set essentially ignores the support lines that are farthest out. In contrast, each of the algorithms proposed here uses *all* of the measurements to "pull" the inner support lines out, if necessary.

In panels (b)–(d) of Figs. 4.9–4.12, the shaded regions correspond to the estimated basic objects produced by the three algorithms using the measurements shown in the respective (a) panels. The results of the Closest algorithm are shown in the (b) panels, the Mini-Max algorithm in the (c) panels, and the Close-Min algorithm in the (d) panels. For comparison, we have also included the outline of the true basic object (dashed line) in each of these panels. The most important observation to make here is that the Closest estimates strongly resemble the measurements, the Mini-Max estimates strongly resemble our prior expectation (large circular objects) and the Close-Min estimates "blend" these two outcomes. Note that we have chosen $\alpha = 0.5$ for the Close-Min experiments; clearly, there are a range of different

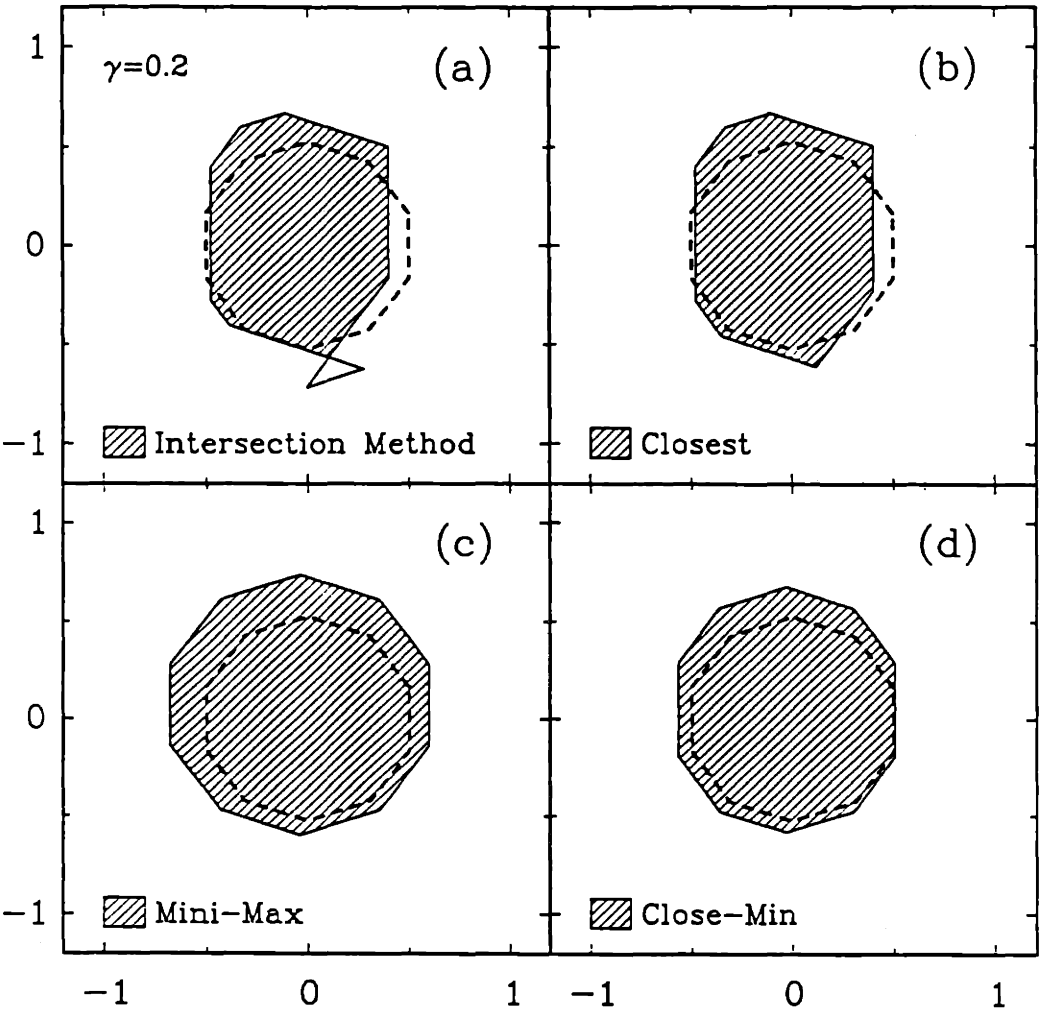


Figure 4.9: (a) The true object (circle), the measured support vector ($\gamma = 0.2$), and the reconstruction obtained using the intersection method. (b) Closest, (c) Mini-Max, and (d) Close-Min estimates.

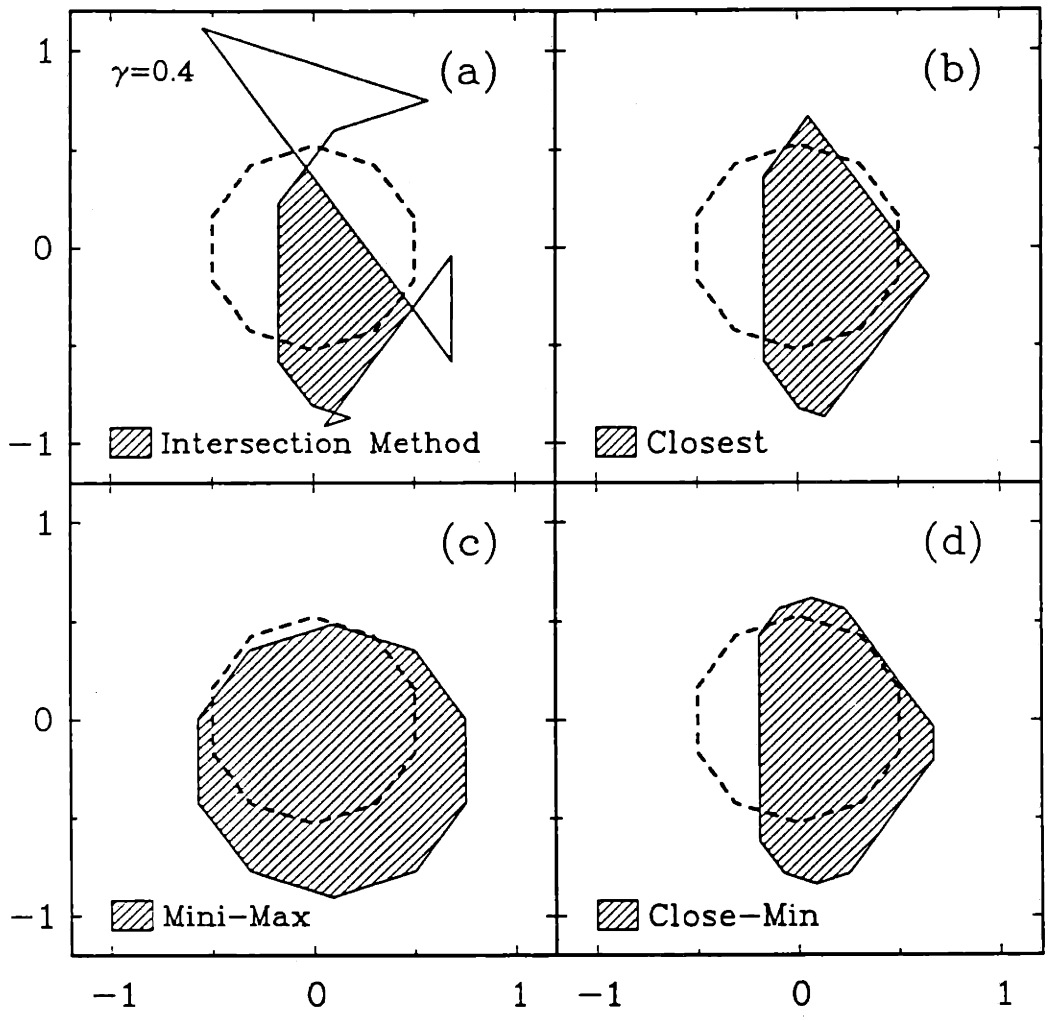


Figure 4.10: (a) The true object (circle), the measured support vector ($\gamma = 0.4$), and the reconstruction obtained using the intersection method. (b) Closest, (c) Mini-Max, and (d) Close-Min estimates.

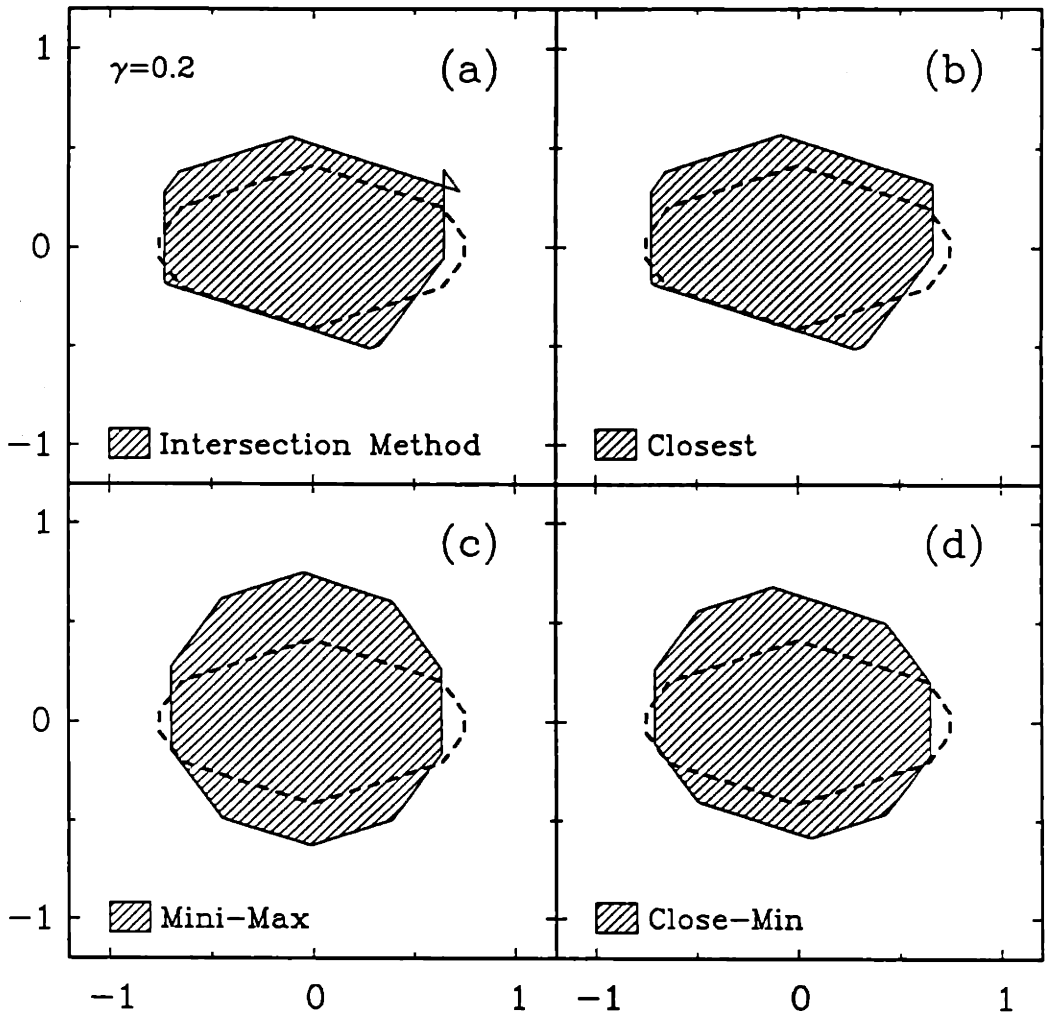


Figure 4.11: (a) The true object (ellipse), the measured support vector ($\gamma = 0.2$), and the reconstruction obtained using the intersection method. (b) Closest, (c) Mini-Max, and (d) Close-Min estimates.

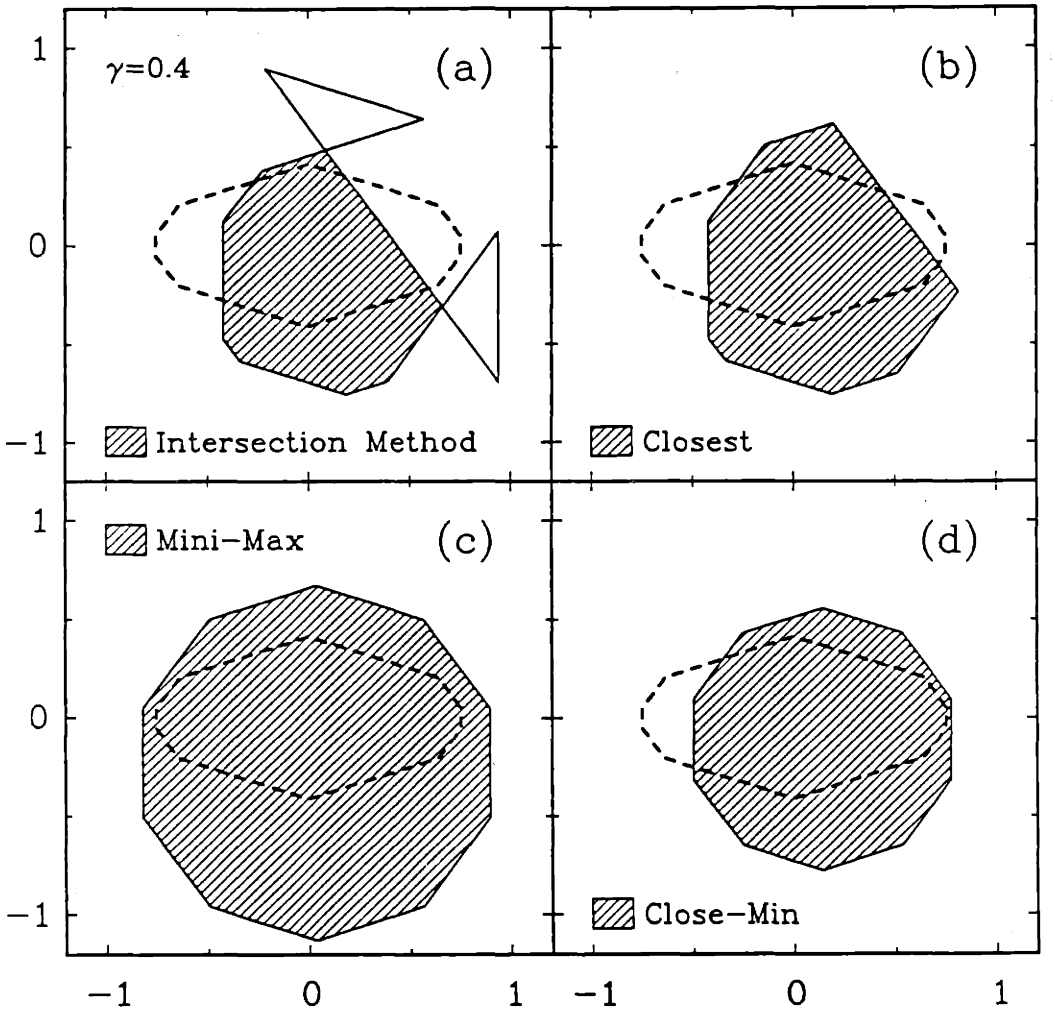


Figure 4.12: (a) The true object (ellipse), the measured support vector ($\gamma = 0.4$), and the reconstruction obtained using the intersection method. (b) Closest, (c) Mini-Max, and (d) Close-Min estimates.

estimates corresponding to different α's which should yield figures ranging between the Closest and Mini-Max solutions.

Let us examine the results in more detail. The Closest estimates show the following behavior: the lines are moved just enough in order to correct the points of inconsistency. Note that, around a point of inconsistency, the inner lines are "pulled" out and the outer lines are "pushed" in. This is in accordance with the Closest criteria which, in words, is to adjust the lateral positions of the lines in order to make them consistent, but in such a way that minimizes the sum of the squares of the lateral movements. For example, in Fig. 4.9b we see that three lines were moved to correct the single point of inconsistency. Note that it is possible to move only one line to fix such a point, but clearly that move yields a larger squared difference between observation and support vector. Because of this behavior, the Closest estimate always produces a basic object which is larger than the intersection method (provided that the measurement is infeasible). Then, for almost all noise models, we expect that the Closest estimate is better than the intersection method, since it is not as biased toward small figures.

To clarify some of the behavior of the Mini-Max estimates, it is useful to examine the estimates together with the bounds imposed by the hypercube $\mathcal{B}$ of (4.23). Fig. 4.13 shows the vertex plots for the Mini-Max estimate (solid line), the inner bound $y_a = y - [\gamma\,\gamma\,\ldots\,\gamma]^{\mathrm{T}}$ (dotted line), and the outer bound $y_b = y + [\gamma\,\gamma\,\ldots\,\gamma]^{\mathrm{T}}$ (dashed line) for the example shown in Fig. 4.9. First, this figure demonstrates how the Mini-Max estimate, in effort to maximize the minimum discrete radius of curvature, produces a figure which is as large as possible given the bounds, yet is also nearly circular (that is, nearly a regular polygon). Second, it is clear from the figure that the estimated basic object may be shifted down a short distance and still remain within the bounds. Since, as we have already pointed out, adding a nullvector to the estimated support vector does not affect the value of the Mini-Max objective function, any feasible shifted version of the solution is also optimal. Therefore, in this example, the solution is not unique. In the shift-corrected algorithms discussed below, this component of non-uniqueness is eliminated by imposing a known object position. As we shall see, this simple correction has dramatic effects on the Mini-

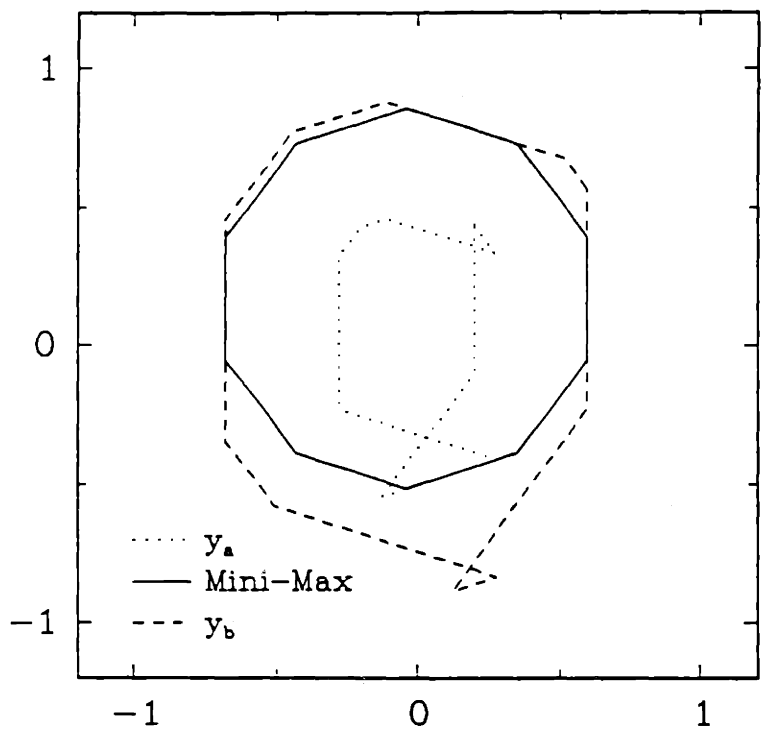


Figure 4.13: The observation bounds and the Mini-Max estimate.

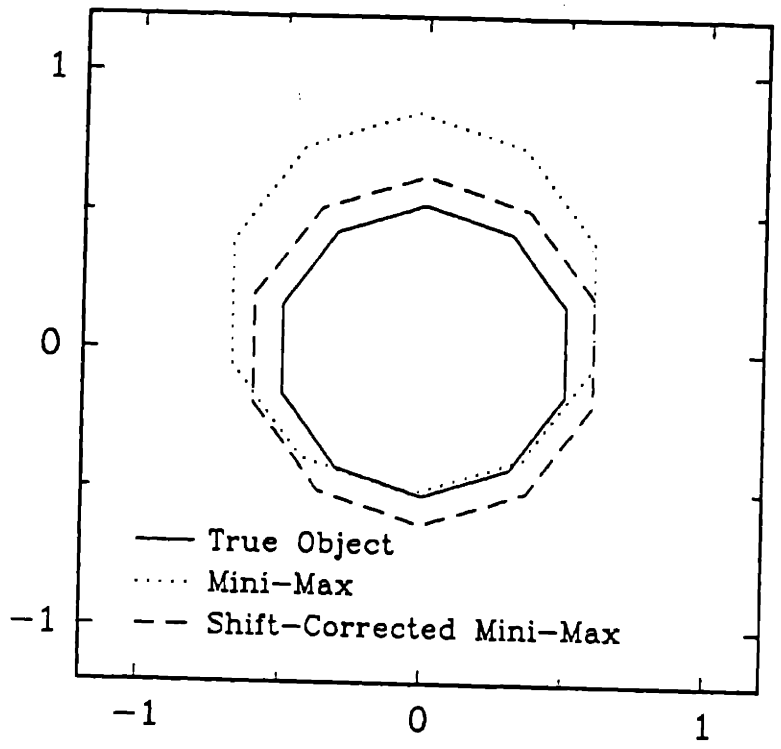


Figure 4.14: Shift-corrected Mini-Max estimate.

Max estimates.

The Close-Min algorithm produced the "blended" estimates that we expected. In particular, where the Closest algorithm corrected the points of inconsistency, it invariably left a sharp corner on the boundary. The Close-Min algorithm produced estimates which appear quite similar to the results of the Closest algorithm but which have smoothed these corners.

Finally, we present one experiment which demonstrates the results of shift correction applied to the Mini-Max algorithm. Fig. 4.14 shows three vertex plots corresponding to the true support vector (solid line), the Mini-Max estimate from Fig. 4.9c (dotted line), and the Mini-Max shift-corrected (for $\bar{v} = 0$) estimate (dashed line). We see that the shift correction *does not simply shift the original Mini-Max solution down.* To understand this we recall Fig. 4.13. We saw that due to non-uniqueness we could shift the solution vertically over a finite range. But,

evidently, none of these shifted positions causes the sum of the vertex points to be exactly zero. To allow this to occur, the shift-corrected algorithm was forced to shrink the estimate as well. Clearly, prior information about the position of the object may have a very strong influence on the performance of the algorithms.

4.6 Conclusions

In this chapter we have introduced several important ideas related to the reconstruction of convex sets from support line measurements. The primary contribution of this chapter is in the formulation of the problem as a constrained optimization problem which includes the fundamental support vector constraint, prior information, and uncertainty in the measurements. We have shown how knowledge of

1. Fundamental geometric constraints,
2. Object shape and position, and
3. Underlying measurement noise models,

may lead directly to optimization-based or probabilistic-based algorithm formulations. We have shown how these methods produce better reconstructions, which are more consistent with the available information, than the conventional intersection method, which does not use any of this information.

The algorithms we have proposed in this chapter are of the very simplest type, however, they serve the purpose of illustration of the fundamental ideas, and they are implemented using particularly efficient codes. The Closest algorithm gives the constrained maximum likelihood estimate assuming the noise is Gaussian. It requires the minimum amount of prior knowledge about the set to be reconstructed, and is implemented in a straightforward manner using quadratic programming techniques. The Mini-Max algorithm gives one method to produce smoother boundaries which results in fast linear programming codes. However, the Mini-Max solution is not necessarily unique and tends to produce large, nearly circular objects. The Close-Min algorithm blends the preceding two objective functions to produce estimates that balance the prior information and the information contained in the

measurements. Finally, we have shown that prior knowledge of object location can lead to considerable improvement for the resulting shift-corrected algorithms. Note that object location is one quantity that can typically be estimated with great accuracy in CT applications.

Many extensions of this work are possible, both in the inclusion of additional constraints or in the development of more elaborate objective functions. Among the possible constraints one might consider including is a known object area. The area of a basic object is a quadratic function of h, however, which leads to inherently more complicated computational methods. A simpler extension of the constraints may arise if one has only partial information about the position of the object in the plane. For example, if the position were bounded, then instead of having two linear *equality* constraints (corresponding to the x and y position) as in the shift-corrected algorithms, one would have four linear *inequality* constraints.

A potentially important extension of the form of the objective function involves the development of explicit prior probabilities on support vectors. For example, if one interprets the Close-Min algorithm as an explicit MAP formulation, one finds that the implied prior distribution on h strongly favors large objects. This, in general, is not desirable. One would prefer to specify a prior distribution which permits separate control of size and smoothness, for example, and perhaps also makes explicit such quantities as eccentricity and orientation. Once specifying such prior distributions, the algorithms may be formulated precisely using MAP techniques with the additional knowledge of the measurement noise statistics. We begin this development in Chapter 5.

Another extension of these methods may be made to account for situations where one has missing measurements. This application is particularly important to the CT problem studied in this thesis, where one has limited-angle or sparse-angle observations. For example, suppose one has M measurements but wishes to reconstruct a support vector of dimension $2M$. One may think of this as an interpolation or extrapolation procedure, and provided there is some prior shape information, this may be accomplished with relatively simple additions to the current algorithms. We begin this development also in Chapter 5 (see also [70]).

4.A Formulas

We collect here for convenience several formulas and brief derivations that are often referred to in the text. Here $\theta_0 = 2\pi/M$.

The point p_i of intersection of L_{i-1} and L_{i+1} (see Fig. 4.4):

$$\begin{aligned} p_i^{\mathrm{T}} &= \begin{bmatrix} h_{i-1} & h_{i+1} \end{bmatrix} \begin{bmatrix} \omega_{i-1} & \omega_{i+1} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} h_{i-1} & h_{i+1} \end{bmatrix} \begin{bmatrix} \cos\theta_{i-1} & \cos\theta_{i+1} \\ \sin\theta_{i-1} & \sin\theta_{i+1} \end{bmatrix}^{-1} \\ &= \frac{1}{\sin 2\theta_0} \begin{bmatrix} h_{i-1} & h_{i+1} \end{bmatrix} \begin{bmatrix} \sin\theta_{i+1} & -\cos\theta_{i+1} \\ -\sin\theta_{i-1} & \cos\theta_{i-1} \end{bmatrix} \end{aligned}$$

Also

$$\begin{aligned} p_i^{\mathrm{T}}\omega_i &= \frac{1}{\sin 2\theta_0} \begin{bmatrix} h_{i-1} & h_{i+1} \end{bmatrix} \begin{bmatrix} \sin\theta_{i+1} & -\cos\theta_{i+1} \\ -\sin\theta_{i-1} & \cos\theta_{i-1} \end{bmatrix} \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} \\ &= \frac{1}{\sin 2\theta_0} \begin{bmatrix} h_{i-1} & h_{i+1} \end{bmatrix} \begin{bmatrix} \sin\theta_0 \\ \sin\theta_0 \end{bmatrix} \\ &= \frac{1}{2\cos\theta_0} (h_{i-1} + h_{i+1}) \end{aligned}$$

Since $h_i \leq p_i^{\mathrm{T}}\omega_i$, we see that this result yields the necessary result in Theorem 1.

Discrete radius of curvature:

$$\begin{aligned}\rho_i &\equiv -h^{\mathrm{T}}c_i \\ &= \frac{1}{2\cos\theta_0}(h_{i-1}+h_{i+1}) - h_i \\ &= p_i^{\mathrm{T}}\omega_i - h_i\end{aligned}$$

Vertex points:

$$\begin{aligned}\nu_i^{\mathrm{T}} &= \begin{bmatrix} h_i & h_{i+1} \end{bmatrix}\begin{bmatrix} \omega_i & \omega_{i+1} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} h_i & h_{i+1} \end{bmatrix}\begin{bmatrix} \cos\theta_i & \cos\theta_{i+1} \\ \sin\theta_i & \sin\theta_{i+1} \end{bmatrix}^{-1} \\ &= \frac{1}{\sin\theta_0}\begin{bmatrix} h_i & h_{i+1} \end{bmatrix}\begin{bmatrix} \sin\theta_{i+1} & -\cos\theta_{i+1} \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}\end{aligned}$$

Eigenvalues of the constraint matrix:

$$\begin{aligned}\lambda_k &= \sum_{n=1}^{M} c_{1n}e^{-j2\pi(k-1)(n-1)/M} \quad k = 1,\ldots,M \\ &= 1 + \frac{-1}{2\cos 2\pi/M}e^{-j2\pi(k-1)/M} + \frac{-1}{2\cos 2\pi/M}e^{-j2\pi(k-1)(M-1)/M} \\ &= 1 - \frac{\cos 2\pi(k-1)/M}{\cos 2\pi/M}\end{aligned}$$

The x and y coordinates of the center of gravity of the vertex points:

$$
\begin{aligned}
\bar{v}_x &= \frac{1}{M}\sum_{i=1}^{M}\frac{1}{\sin\theta_0}\begin{bmatrix} h_i & h_{i+1}\end{bmatrix}\begin{bmatrix}\sin\theta_{i+1}\\ -\sin\theta_i\end{bmatrix} \\
&= \frac{1}{M\sin\theta_0}(h^{\mathrm{T}}\begin{bmatrix}\sin\theta_2\\ \sin\theta_3\\ \vdots\\ \sin\theta_M\\ \sin\theta_1\end{bmatrix} - h^{\mathrm{T}}\begin{bmatrix}\sin\theta_M\\ \sin\theta_1\\ \vdots\\ \sin\theta_{M-2}\\ \sin\theta_{M-1}\end{bmatrix}) \\
&= \frac{1}{M\sin\theta_0}h^{\mathrm{T}}\begin{bmatrix}\sin(\theta_1+\theta_0)-\sin(\theta_1-\theta_0)\\ \vdots\\ \sin(\theta_M+\theta_0)-\sin(\theta_M-\theta_0)\end{bmatrix} \\
&= \frac{2}{M}h^{\mathrm{T}}\begin{bmatrix}\cos\theta_1\\ \vdots\\ \cos\theta_M\end{bmatrix} \\
&= \frac{2}{M}h^{\mathrm{T}}n_1
\end{aligned}
$$

$$
\begin{aligned}
\bar{\nu}_y &= \frac{1}{M}\sum_{i=1}^{M}\frac{1}{\sin\theta_0}\begin{bmatrix} h_i & h_{i+1}\end{bmatrix}\begin{bmatrix} -\cos\theta_{i+1} \\ \cos\theta_i \end{bmatrix} \\
&= \frac{1}{M\sin\theta_0}\left(-h^{\mathrm{T}}\begin{bmatrix} \cos\theta_2 \\ \cos\theta_3 \\ \vdots \\ \cos\theta_M \\ \cos\theta_1 \end{bmatrix} + h^{\mathrm{T}}\begin{bmatrix} \cos\theta_M \\ \cos\theta_1 \\ \vdots \\ \cos\theta_{M-2} \\ \cos\theta_{M-1} \end{bmatrix}\right) \\
&= \frac{1}{M\sin\theta_0}h^{\mathrm{T}}\begin{bmatrix} \cos(\theta_1-\theta_0)-\cos(\theta_1+\theta_0) \\ \vdots \\ \cos(\theta_M-\theta_0)-\cos(\theta_M+\theta_0) \end{bmatrix} \\
&= \frac{2}{M}h^{\mathrm{T}}\begin{bmatrix} \sin\theta_1 \\ \vdots \\ \sin\theta_M \end{bmatrix} \\
&= \frac{2}{M}h^{\mathrm{T}}n_2
\end{aligned}
$$

4.B Proof of Theorem 1 (cont.)

To complete the proof, we must show that (4.6) implies $S_B = S_\nu$. This is done in two stages. First we show that $S_B \subset S_\nu$, then that $S_\nu \subset S_B$. Since S_B is a bounded (convex) polytope (proof omitted), it may be written as

$$S_B = \mathrm{hul}(e_1, e_2, \ldots, e_p) \tag{4.29}$$

where e_i are the extreme points of S_B. Consider one particular extreme point of S_B, e_j; it must satisfy with equality at least two inequalities in (4.5). Let one of

those inequalities be indexed by k. Then we have

$$e_j^{\mathrm{T}}\omega_k = h_k \ , \tag{4.30}$$

i.e., e_j lies on the line L_k. Two of the ν_i's also lie on L_k: ν_{k-1} and ν_k. Now suppose e_j could be written as the convex combination of ν_{k-1} and ν_k. Then any extreme point of S_B could be written as the convex combination of two points in S_ν. And since both S_B and S_ν are convex, then we must have that $S_B \subset S_\nu$, proving this stage of the theorem.

We now show that e_j can indeed be written as the convex combination of ν_{k-1} and ν_k. Here, there are two possibilities: $\nu_{k-1} = \nu_k$ and $\nu_{k-1} \neq \nu_k$. Each of these cases require some development.

In the case where $\nu_{k-1} = \nu_k$, we show that $e_j = \nu_{k-1} = \nu_k$. First, since e_j and ν_k are on the line perpendicular to ω_k, we may write e_k as

$$e_j = \nu_k + \beta\omega_k^{\perp} \ , \tag{4.31}$$

where

$$\omega_k^{\perp} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \omega_k$$

is the perpendicular to ω_k. Taking inner products of both sides of (4.31) with ω_{k-1} and using the fact that e_j is in S_B we may write

$$\omega_{k-1}^{\mathrm{T}} e_j = h_{k-1} + \beta\omega_{k-1}^{\mathrm{T}}\omega_k^{\perp} \leq h_{k-1}$$

and, similarly, for ω_{k+1}

$$\omega_{k+1}^{\mathrm{T}} e_j = h_{k+1} + \beta\omega_{k+1}^{\mathrm{T}}\omega_k^{\perp} \leq h_{k+1}$$

Hence,

$$\beta\omega_{k-1}^{\mathrm{T}}\omega_k^{\perp} \leq 0 \quad \text{and} \quad \beta\omega_{k+1}\omega_k^{\perp} \leq 0 \ .$$

After simplifying the above expressions using the definitions of ω_{k-1}, ω_{k+1}, and $\omega_k^{\perp}$, we are led to the contradictory equations

$$\beta(-\sin\theta_0) \leq 0 \quad \text{and} \quad \beta(\sin\theta_0) \leq 0 \ ,$$

hence, β must be zero, and therefore $e_j = \nu_k = \nu_{k-1}$, as required.

In the case where $\nu_{k-1} \neq \nu_k$ we first need an auxiliary result relating the unit vectors ω_{k-1}, ω_k, and ω_{k+1}. From the geometry it is easy to verify that

$$\omega_k = \frac{1}{2\cos\theta_0}(\omega_{k-1} + \omega_{k+1}) \tag{4.32}$$

where $\theta_0 = 2\pi/M$. Next, since e_j, ν_{k-1}, and ν_k all lie on the same line L_k, and ν_{k-1} and ν_k are distinct points, we may express e_j as a linear combination of ν_{k-1} and ν_k using the single parameter α as

$$e_j = \alpha\nu_{k-1} + (1-\alpha)\nu_k \tag{4.33}$$

Taking the inner product of both sides of (4.33) with ω_{k-1} we have

$$\begin{aligned} e_j^{\mathrm{T}}\omega_{k-1} &= \alpha\nu_{k-1}^{\mathrm{T}}\omega_{k-1} + (1-\alpha)\nu_k^{\mathrm{T}}\omega_{k-1} \\ &= \alpha h_{k-1} + (1-\alpha)\nu_k^{\mathrm{T}}\omega_{k-1} \\ &\leq h_{k-1} \ . \end{aligned} \tag{4.34}$$

The last inequality results from the fact that e_j is, by definition, in S_B. Now we eliminate ω_{k-1} from (4.34) using (4.32) yielding

$$\alpha h_{k-1} + (1-\alpha)\nu_k^{\mathrm{T}}(2\cos\theta_0\,\omega_k - \omega_{k+1}) \leq h_{k-1}$$

which may be further reduced to

$$(1-\alpha)(2\cos\theta_0\,h_k - h_{k-1} - h_{k+1}) \leq 0 \ . \tag{4.35}$$

Since from (4.6) the quantity $2\cos\theta_0\,h_k - h_{k-1} - h_{k+1}$ must be non-positive we immediately recognize that $\alpha \leq 1$.

Taking the inner product of both sides of (4.33) with ω_{k+1} and using a similar sequence of steps leading to (4.35) one may show that

$$\alpha(2\cos\theta_0\,h_k - h_{k-1} - h_{k+1}) \leq 0 \tag{4.36}$$

from which we conclude that $\alpha \geq 0$. Hence, we have that $0 \leq \alpha \leq 1$ and, therefore, that e_j is, in fact, a *convex combination* of ν_{k-1} and ν_k. This completes the proof that $S_B \subset S_\nu$.

Now we begin the proof that $S_\nu \subset S_B$. In what follows, we show that $\nu_i \in S_B$ for each $i = 1, \ldots, M$. Since S_B is convex this is sufficient to prove that S_ν is contained in S_B. Accordingly, we intend to show that

$$\nu_i^{\mathrm{T}}[\omega_1\, \omega_2\, \ldots\, \omega_M] \leq [h_1\, h_2\, \ldots, h_M] \tag{4.37}$$

for all $i = 1, \ldots, M$. Substituting expressions for ν_i and ω_j $j = 1, \ldots, M$ into (4.37) and simplifying yields

$$\frac{1}{\sin\theta_0}[q_{i1}\, q_{i2}\, \ldots\, q_{iM}] \leq [h_1\, h_2\, \ldots\, h_M] \tag{4.38}$$

where $q_{ij} = h_i \sin(\theta_{i+1} - \theta_j) - h_{i+1}\sin(\theta_i - \theta_j)$. Our task is to show that (4.38) is true given $h^{\mathrm{T}}C \leq 0$.

Equation (4.38) is true if each term is separately true. Hence, we must show that

$$\frac{1}{\sin\theta_0}\left(h_i \sin(\theta_{i+1} - \theta_j) - h_{i+1}\sin(\theta_i - \theta_j)\right) \leq h_j \tag{4.39}$$

for $i = 1, \ldots, M$ (each ν_i) and $j = 1, \ldots, M$ (each term in (4.37)). Because of the rotational symmetry of the problem we may, without loss of generality, choose $j = 1$ and prove that (4.39) is true for $i = 1, \ldots, M$. Since $\theta_i = (i-1)2\pi/M = (i-1)\theta_0$, then for $j = 1$ we may simplify (4.39) to

$$\frac{1}{\sin\theta_0}\left(h_i \sin i\theta_0 - h_{i+1}\sin(i-1)\theta_0\right) \leq h_1 \tag{4.40}$$

Denoting the left-hand side of (4.40) by E_i we have for $i = 1$ that

$$E_1 = \frac{1}{\sin\theta_0}\left(h_1 \sin\theta_0 - 0\right) = h_1$$

which satisfies (4.40) trivially. The general expression E_i for $i = 2, \ldots, M$ may be related to E_1 using the relation $h^{\mathrm{T}}C \leq 0$ as follows. From (4.40) we have that

$$E_i = \frac{1}{\sin\theta_0}\left(h_i \sin i\theta_0 - h_{i+1}\sin(i-1)\theta_0\right) \quad .$$

Using the formula $\sin ia = 2\sin(i-1)a\cos a - \sin(i-2)a$, this becomes

$$\begin{aligned} E_i &= \frac{1}{\sin\theta_0}\left[h_i\left(2\sin(i-1)\theta_0\cos\theta_0 - \sin(i-2)\theta_0\right) - h_{i+1}\sin(i-1)\theta_0\right] \\ &= \frac{1}{\sin\theta_0}\left[(2h_i\cos\theta_0 - h_{i+1})\sin(i-1)\theta_0 - h_i\sin(i-2)\theta_0\right] \end{aligned} \tag{4.41}$$

Now we notice that the i^{th} constraint in $h^{\mathrm{T}}C \leq 0$ may be written as $2\cos\theta_i\, h_i - h_{i+1} \leq h_{i-1}$. Using this inequality in (4.41) yields

$$E_i \leq \frac{1}{\sin\theta_0}\left[h_{i-1}\sin(i-1)\theta_0 - h_i \sin(i-2)\theta_0\right]$$

which may be reduced to $E_i \leq E_{i-1}$. This is the result that we sought. Now we may conclude that

$$E_M \leq E_{M-1} \leq \ldots \leq E_2 \leq E_1 = h_1$$

which concludes the proof of sufficiency and, hence, the theorem. □

Chapter 5

MAP ESTIMATION OF SUPPORT VECTORS

5.1 Introduction

This chapter continues the development of ideas involving support estimation by developing MAP estimation algorithms that arise from explicit specification of prior probabilities on support vectors. We begin by showing that the Close-Min algorithm of Chapter 4 has a precise interpretation as an MAP estimator. The prior distribution which is implied by the algorithm, which we will call the *Close-Min prior*, shows quite clearly that larger objects have higher probability, which is in general an undesirable property. At this point, in order to more clearly define what we mean by the *size* of basic objects we introduce a new decomposition of support vectors, which we call the *Size/Shape/Shift* (SSS) decomposition, which yields a type of polar representation for support vectors in the proper cone C_p. We will see that one natural interpretation of the *size* of a basic object is its *circumference.*

In the following section we consider a class of priors which are scale-invariant in the sense that objects of precisely the same shape but of different size have exactly the same prior probability. The resultant estimation algorithms, which we call *Scale-Invariant algorithms*, are MAP algorithms in which the contribution of prior knowledge involves shape only — the observations provide all of the position and size information. The first Scale-Invariant algorithm is a slight variation on

the Close-Min algorithm, designed simply to eliminate the size dependency. The second and third algorithms also favor circular objects, but the priors which yield this effect are somewhat more natural than the Close-Min prior, and the resultant algorithms are slightly more efficient as well.

One important outcome of our investigation of the Scale-Invariant algorithms is the development of a new optimization procedure, required since none of the formulations can be solved directly using LP or QP, as was the case for the algorithms of Chapter 4. We use the SSS decomposition to write the objective function in terms of the size, shape, and shift of the unknown support vector and show how the shift component may be solved for directly. To determine the size and shape components we implement a line search [68] over the (scalar) size component, where for each size the optimum shape vector is found by solving a QP. The line search converges to the optimum size and therefore yields the jointly optimum size and shape components.

Section 5.4 considers the use of prior knowledge concerning the eccentricity and orientation of the true object. We develop three *Ellipse-Based algorithms*, each of which uses this type of prior knowledge in a different way. The first algorithm finds the closest ellipse to a given (possibly infeasible) support vector, so that under a certain noise model this algorithm may be used to find the ML estimates of the ellipse parameters, assuming the true object to be exactly an ellipse. The second algorithm assumes that the true object is nearly elliptic in shape and that we have prior knowledge of the underlying ellipse parameters. Finally, the third algorithm assumes that the true object is nearly elliptic in shape, but that we do *not* have any prior knowledge of the underlying ellipse parameters. This algorithm estimates *jointly* the support vector and the ellipse parameters.

Section 5.5 presents the results of many simulations designed to show the behavior of the Scale-Invariant algorithms and of the Ellipse-Based algorithms. We conclude the chapter in Section 5.6 with a general discussion of the algorithms developed in this chapter and of the experimental results.

5.2 The Close-Min Algorithm as an MAP Estimator

In this section we interpret the Close-Min algorithm as an MAP estimator which has a certain implied prior probability density function (PDF) on support vectors. We call this implied PDF the *Close-Min prior* and denote it by $p_{CM}(h)$. To help explain the behavior of the Close-Min algorithm, we develop the Size/Shape/Shift (SSS) decomposition of a support vector and analyze $p_{CM}(h)$ in terms of this decomposition. Finally, we describe a method to choose a certain constant so that the Close-Min prior is approximately independent of M, the dimension of the support vector. At the conclusion of this section we will have a clearer understanding of the reasons for the behavior of the Close-Min algorithm in the experiments of Chapter 4, and we will see how to make improvements by specifying α — the constant which trades off the relative contributions of the prior knowledge and the observations — in a more consistent manner. Also, we will see more clearly how to change the prior to better suit our prior knowledge about shapes of objects; and this leads into the next section on scale-invariant priors and algorithms.

5.2.1 The Implied Close-Min Prior

The Close-Min estimate is given by (see Chapter 4)

$$\hat{h}_{CM} = \underset{h:\, h \in B \cap C}{\operatorname{argmax}} \; \alpha f_C(h) + (1-\alpha) f_M(h) \tag{5.1}$$

where $0 \leq \alpha \leq 1$ and

$$f_C(h) = -\frac{1}{2}(y-h)^{\mathrm{T}}(y-h) \tag{5.2}$$

$$f_M(h) = \min\{-h^{\mathrm{T}}c_1, \ldots, -h^{\mathrm{T}}c_M\} \tag{5.3}$$

where c_i are the columns of C. The MAP estimate of h given the measurements y maximizes [91]

$$l(h) = \ln p(y|h) + \ln p(h)\,, \tag{5.4}$$

and assuming that $y = h + n$, where the elements of n are zero-mean, independent Gaussian random variables with variance σ^2, then

$$\ln p(y|h) = -\frac{1}{2\sigma^2}(y-h)^{\mathrm{T}}(y-h) - \frac{M}{2}\ln 2\pi\sigma^2 \ . \tag{5.5}$$

Since the estimate in (5.1) is not changed by adding constants to the objective function or by multiplying the objective function by a positive constant, we may use (5.2)-(5.5) to manipulate (5.1) into the following form

$$\hat{h}_{CM} = \underset{h:\, h\in\mathcal{B}\cap\mathcal{C}}{\operatorname{argmax}} \ \ln p(y|h) + \frac{1-\alpha}{\alpha\sigma^2}\min\{-h^{\mathrm{T}}c_1,\ldots,-h^{\mathrm{T}}c_M\} - \ln z \tag{5.6}$$

where z is a constant which does not depend on h. Note that we also have made the assumption that $\alpha \neq 0$. Comparing (5.6) with (5.4) allows us to identify the logarithm of the prior as

$$\ln p_{CM}(h) = \frac{1-\alpha}{\alpha\sigma^2}\min\{-h^{\mathrm{T}}c_1,\ldots,-h^{\mathrm{T}}c_M\} - \ln z \tag{5.7}$$

and hence the prior itself as

$$p_{CM}(h) = \frac{1}{z}\exp\left(\frac{1-\alpha}{\alpha\sigma^2}\min\{-h^{\mathrm{T}}c_1,\ldots,-h^{\mathrm{T}}c_M\}\right)$$

where z is now seen to be the constant required to make $p_{CM}(h)$ integrate to one.[1]

Since $p_{CM}(h)$ is a prior probability on h, it should be independent of the observation noise variance. Hence, the expression $(1-\alpha)/\alpha\sigma^2$ must evaluate to a constant, independent of σ^2. Making the following equivalence,

$$\frac{1-\alpha}{\alpha\sigma^2} = \frac{1}{\tau}$$

where τ is a real constant (which may, however, depend upon M as is discussed in Section 5.2.5), we see that for the Close-Min algorithm to be interpreted precisely as an MAP estimator we must have that $\alpha = \alpha^*$ where

$$\alpha^* = \frac{\tau}{\tau+\sigma^2} \ . \tag{5.8}$$

[1]The exponential term in $p_{CM}(h)$ is not integrable over the full cone $\mathcal{C}$ (see (4.10) in Chapter 4). But since $\mathcal{B}$ is bounded, it is integrable over the intersection $\mathcal{B}\cap\mathcal{C}$ which specifies the region of feasible support vectors for the Close-Min estimate of (5.1). Note that although we defined $\mathcal{B}$ in the previous chapter as a hypercube centered on the observation vector y, we need not be this restrictive here since we have also assumed that α is not zero. It is sufficient to think of $\mathcal{B}$ as simply a large M-dimensional ball centered on the origin which limits the size of h.

Now we see that choosing the convexity parameter α — for example, in the experiments of Chapter 4 — has the effect of determining the form of the implied prior probability for a given noise variance σ^2. A better way to determine the correct tradeoff between the prior knowledge and the observations would be to specify τ rather than α, and then set α to α^*, a quantity which also depends on the noise variance. This makes the Close-Min algorithm the optimal MAP estimator for the Close-Min prior which may now be written as

$$p_{CM}(h) = \frac{1}{z} \exp\left(\frac{1}{\tau} \min\{-h^{\mathrm{T}}c_1, \ldots, -h^{\mathrm{T}}c_M\}\right) . \tag{5.9}$$

In Section 5.2.5 we show how to select τ as a function of M so that $p_{CM}(h)$ is approximately independent of M, but until that discussion we will assume that τ does not depend on M.

5.2.2 Characterization of the Close-Min Prior

In this section and in Section 5.2.4 we explore several parametric classes of support vectors in order to learn what the Close-Min prior implies about the expected shape, size, orientation, and position of basic objects. This helps to explain some of the behavior of the Close-Min algorithm in the experiments of Chapter 4, and will aid in developing algorithms in subsequent sections.

The Axial Class

In order to see how the Close-Min prior of (5.9) favors larger objects consider the class of support vectors parametrized by a non-negative real number t given by

$$h_t = te$$

where $e = [1\,1\,\ldots\,1]^{\mathrm{T}}$. Each such vector yields the support lines for the circle of radius t centered at the origin. Evaluating the prior of (5.9) for h_t yields

$$p_{CM}(h_t) = \frac{1}{z} \exp\left(\frac{\gamma}{\tau} t\right)$$

where

$$\begin{aligned} \gamma &= -e^{\mathrm{T}} c_i \\ &= \frac{1}{\cos\theta_0} - 1 \\ &> 0 \end{aligned} \tag{5.10}$$

We see that $p_{CM}(h_t)$ is a *growing exponential* with t — larger circles are very much more likely. We may further see that as $M \to \infty$, $\theta_0 \to 0$, which means that $\gamma \to 0$, and therefore, this strong size dependence disappears as M gets large.[2] We think of t as the *size* of the support vector — a concept to be examined in greater detail in the following section — we see that this begins to explain why the Close-Min algorithm favors larger objects.

Before continuing with the characterization of $p_{CM}(h)$ we now explore a new decomposition of a support vector h which allows us to define precisely what we mean by the size of h.

5.2.3 The Size/Shape/Shift Decomposition

In Chapter 4 we showed that any support vector h may be uniquely decomposed as $h = h_p + h_n$ where h_p is in the proper cone $\mathcal{C}_p$ and h_n is in the nullspace $\mathcal{N}$ of $\mathcal{C}$. Let us refer to vectors in $\mathcal{C}_p$ as *proper support vectors* and those in $\mathcal{N}$ as *shift vectors.* In Chapter 4 we found that proper support vectors have basic objects which are centered at the origin and that adding a non-zero shift vector to h simply shifts the basic object in the plane. We now show that any proper support vector h_p may be written as $h_p = tq$ where t is a real number satisfying $t \geq 0$ and q is a proper support vector which also satisfies $q^{\mathrm{T}} e = M$ where $e = [1\ 1\ \ldots\ 1]^{\mathrm{T}}$. We will show that two support vectors whose t components are identical have basic objects with the same *circumference.* This is what we will mean by the *size* of h in the sequel. The remaining quantity, the vector q, contains the information related to the *shape* of the basic object, about which we shall have more to say in a later section.

[2] This is true unless τ depends on M in such a fashion as to eliminate this effect — a subject discussed in Section 5.2.5.

To show that h_p may be decomposed as described above, consider the truncated cone

$$T = \left\{ h \in \mathbb{R}^M \mid h^{\mathrm{T}} A \leq 0, \;\; h^{\mathrm{T}} \sum_{j=1}^{M+4} -a_j = \mu \right\}$$

where $A = [C \;\; -N \; N]$, a_j are the columns of A and $\mu = M(\cos^{-1}\theta_0 - 1)$ (see Chapter 4 for definitions for the matrices C and N). The polyhedron T does in fact truncate the proper cone $\mathcal{C}_p$ since T has no ray points.[3] To see this note that for non-zero $v \in \mathbb{R}^M$ we must have $v^{\mathrm{T}} A \neq 0$ since the rows of A are linearly independent. Then for v also satisfying $v^{\mathrm{T}} A \leq 0$ we must have $v^{\mathrm{T}} a_j < 0$ for some j. Hence, $v^{\mathrm{T}} \sum_{j=1}^{M+4} -a_j > 0$.

Now consider a vector $h \neq 0$ in $\mathcal{C}_p$. From the above argument we may conclude that $h^{\mathrm{T}} \sum_{j=1}^{M+4} -a_j > 0$, and therefore for some $\xi > 0$, that $\xi h^{\mathrm{T}} \sum_{j=1}^{M+4} -a_j = \mu$ since $\mu > 0$. Now let us simplify the summation. First note that the last four columns of A sum to zero and therefore we need only sum over the range $j = 1, \ldots, M$ to get the same result. By inspection, we see that the rows of C sum to the same value $-\gamma$, where γ was defined in (5.10). Hence, we have that

$$\sum_{j=1}^{M+4} -a_j = \gamma e$$

and, therefore, given any vector $h \neq 0$ in $\mathcal{C}_p$ there exists a $\xi > 0$ such that $\xi h^{\mathrm{T}} = M$, from which the desired result follows. Clearly, if $h = 0$ then it may be written as $h = tq$ where $t = 0$.

This result, combined with the nullspace result of Chapter 4, gives the Size/Shape/Shift (SSS) decomposition property of a support vector which may be stated as follows.

Theorem 5.1 (Size/Shape/Shift (SSS) Decomposition)

A support vector h may be written as

$$h = tq + h_n \tag{5.11}$$

where $t \geq 0$, $q \in \mathcal{C}_p$, $q^{\mathrm{T}} e = M$, and $h_n \in \mathcal{N}$.

Proof *Follows from the above discussion.* □

[3] A *ray point* of a polyhedron $D = \{h \mid h^{\mathrm{T}} A \leq c\}$ is a point $v \neq 0$ such that $v^{\mathrm{T}} A \leq 0$.

Given an arbitrary vector h we may find the components of the SSS decomposition as follows. To find the nullspace component h_n we use the fact that $h_n = Nv$ for some $v \in \mathbb{R}^2$. Then we may write $h = h_p + Nv$, which when premultiplied by N^T on both sides yields $N^T h = N^T N v$. It can be shown that $N^T N = (M/2)I$ where I is the 2 by 2 identity matrix, which leaves us with $v = (2/M)N^T h$ and therefore that

$$h_n = \frac{2}{M} N N^T h \ . \tag{5.12}$$

which is, of course, just the projection of h onto the nullspace of C. Clearly, then we may form h_p as

$$\begin{aligned} h_p &= h - h_n \\ &= h - \frac{2}{M} N N^T h \\ &= (I - \frac{2}{M} N N^T) h \ . \end{aligned} \tag{5.13}$$

To find an expression for t we take the inner product of both sides of (5.11) with e, yielding $h^T e = t q^T e$ since $h_n^T e = 0$. Since $q^T e = M$ we have that

$$t = \frac{1}{M} h^T e \tag{5.14}$$

which is, remarkably, simply the average value of the components of h. In Appendix 5.A we show that the t-coordinate of a support vector is proportional to the circumference of its basic object. Indeed, denoting the circumference by P we find that

$$P = \frac{2M\gamma}{\tan\theta_0} t$$

where γ is given in (5.10). Thus, we think of the t-coordinate of a support vector as being proportional to the *size* of the basic object and, in particular, by size we mean its circumference. Finally, provided that $t \neq 0$ we may find q as

$$q = h_p / t \ . \tag{5.15}$$

In the case where $t = 0$ we see that the SSS decomposition is not unique since any q may be chosen to satisfy the equality in (5.11). But when $t \neq 0$ (5.11) is a unique

representation of the support vector h. Note that when $t = 0$ the basic object is a point, which is a degenerate object since it has no area and hence no interior points. The shape of the object (which is contained in the vector q) therefore has no meaning in such a case.

5.2.4 Characterization of the Close-Min Prior (cont.)

Having established the SSS decomposition of support vectors we may now examine the Close-Min prior of (5.9) from a new perspective: what is the effect of varying the different SSS components (i.e. t, q, and h_n) independently? Using the SSS decomposition of (5.11) we may rewrite the Close-Min prior of (5.9) as

$$p_{CM}(h) = \frac{1}{z} \exp\left(\frac{t}{\tau} \min\{-q^{\mathrm{T}}c_1, \ldots, -q^{\mathrm{T}}c_M\}\right) \quad . \tag{5.16}$$

Note that the nullspace component disappears from the expression because h_n is orthogonal to the columns of C. Therefore, any shifted version of a centered basic object has the same probability — i.e. this prior is *shift invariant*. Suppose we apply a circular rotation to the elements of h so that $[h_1\ h_2\ \ldots\ h_M]^{\mathrm{T}}$ becomes $[h_M\ h_1\ \ldots\ h_{M-1}]^{\mathrm{T}}$. Note that this transformation has the effect of rotating the basic object counterclockwise by θ_0 radians. Clearly, the new support vector has the same value of t since the average of the elements of the vector is unchanged, and the value of the minimum in the exponent is also unchanged. Therefore, since the probability is unchanged we say that this prior is *rotationally invariant* as well.

The Generalized Axial Class

We continue our characterization of the Close-Min prior by evaluating $p_{CM}(h)$ for support vectors in a slightly more general class than the axial class given earlier. Suppose we choose q_0 — a support *shape* vector — so that $-q_0^{\mathrm{T}}c_j = \min\{-q_0^{\mathrm{T}}c_1, \ldots, -q_0^{\mathrm{T}}c_M\}$; in other words, the jth discrete radius of curvature of the basic object to q_0 is smallest. Now consider the class of support vectors $h_t = tq_0$, $t \geq 0$, which generalizes the earlier expression for the axial class given by $h_t = te$.

For this class, (5.16) may be written as

$$p_{CM}(h_t) = \frac{1}{z}\exp\left(\frac{t}{\tau}(-q_0^{\mathrm{T}}c_j)\right) \quad . \tag{5.17}$$

Now, provided that $-q_0^{\mathrm{T}}c_j \neq 0$ then, as before, we have a growing exponential with t. However, the probability corresponding to a class of off-axis support vectors — that is, support vectors not equal to a constant multiple of e — does not rise as rapidly as that corresponding to the axial class since $-q_0^{\mathrm{T}}c_j < -q_0^{\mathrm{T}}e$. In fact, in the extreme case when $-q_0^{\mathrm{T}}c_j = 0$, $p_{CM}(h_t)$ is not dependent on t at all — an interesting fact about which we shall have more to say below. Then, for all but this special case, the Close-Min prior prescribes a higher probability — with exponential growth — for support vectors whose basic objects have identical shape but larger size (circumference). The first Scale-Invariant prior described in Section 5.3 is a slight modification to the Close-Min prior which eliminates this size-dependency feature.

Shape Classes

With the SSS decomposition we established that the q-component of a support vector contains the information which corresponds to the shape of the basic object. We now examine the Close-Min prior of (5.16) for constant t but varying q using a particularly simple class of support vectors. Let x be a support vector in the proper cone $\mathcal{C}_p$ such that $x^{\mathrm{T}}e = M$ and $-x^{\mathrm{T}}c_j = 0$, for some j; i.e. x lies on the (relative) boundary of $\mathcal{C}_p$ in the plane $\{x \mid x^{\mathrm{T}}e = M\}$. Now consider the class of support vectors consisting of the line segment between x and e given by

$$q_s = (1-s)x + se \quad 0 \leq s \leq 1 \ .$$

Note that q_s satisfies $q_s^{\mathrm{T}}e = M$ for any $s \in [0,1]$, so that these vectors correspond to support vectors whose t coordinate is 1 — therefore, their basic objects have the same circumference $P = 2M\gamma/\tan\theta_0$. Evaluating the prior of (5.16) for support vectors $h_s = q_s$ yields

$$p_{CM}(h_s) = \frac{1}{z}\exp\left(\frac{1}{\tau}\min\{-\left[(1-s)x + se\right]^{\mathrm{T}}c_i : \ i = 1,\ldots,M\}\right) \ ,$$

Now, by design we have that $-x^{\mathrm{T}}c_j = \min\{-x^{\mathrm{T}}c_1, \ldots, -x^{\mathrm{T}}c_M\} = 0$, and since $-e^{\mathrm{T}}c_i = \gamma$ for any i, we see that for $0 \leq s \leq 1$, we may simplify the above expression to

$$p_{CM}(h_s) = \frac{1}{z} \exp\left(\frac{\gamma}{\tau} s\right) \ .$$

We see that between any point x on the (relative) boundary of C_p and the central axis point e, the probability rises exponentially — in fact, it rises to a cusp at e since this applies from any point x. Note that as $M \to \infty$ the ratio γ/τ goes to zero, so that this shape dependency disappears for large M.[4]

The Boundary Class

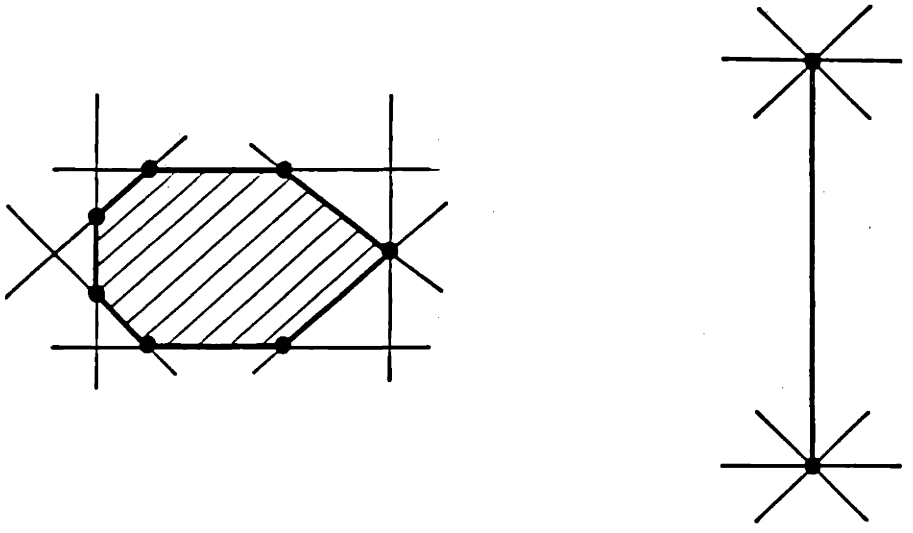
We mentioned above that support vectors which satisfy $-q^{\mathrm{T}}c_j = 0$ for some j — and are therefore on the relative boundary of C_p — have a prior probability which is independent of t. The implication of this property is rather startling: *all support vectors whose basic object has at least one discrete radius of curvature value that is zero are equally likely* à priori.

Let us examine this result in more detail by considering the two basic objects shown in Fig. 5.1. Here we have the basic objects corresponding to two six-dimensional support vectors, each having at least one discrete radius of curvature equal to zero and both having the same circumference.[5] In (a) only one discrete radius of curvature (r_1, or equivalently, ρ_1) is zero, while in (b) four are zero — yet the Close-Min prior gives equal probability to these two figures. This is yet another feature of the Close-Min prior which is not desirable in general. In subsequent sections we shall discuss modifications to the Close-Min prior which change the prior so that objects which are highly eccentric such as the line segment in (b) will have lower prior probability than less eccentric objects such as in (a).

---

[4] As in previous examples, this is true provided τ does not depend on M. We provide a further discussion in Section 5.2.5.

[5] The circumference of the line segment depicted in Fig. 5.1b is twice its length.



(a) (b)

Figure 5.1: Two basic objects with at least one discrete radius of curvature equal to zero.

5.2.5 Selection of τ: Dimension Independence

The pdf $p_{CM}(h)$ is a prior on *support vectors h of dimension M*, each of which has a corresponding basic object, and as such it clearly implies a prior on the shape, size, etc., of basic objects. However, in order for $p_{CM}(h)$ to specify probabilities more directly on the *shape* (or size, shift, etc.) of basic objects we must counter the effect of changing M, which we have seen in the previous sections to have an effect on the relative prior probability of support vectors and their basic objects. We now develop a method to choose τ as a function of M so that $p_{CM}(h)$ specifies expected shapes which are (roughly) independent of M — we call this property *dimension independence.*

In Section 5.2.1 we saw that, for the axial class $h_t = te$, as $M \to \infty$ the strong size dependency of the Close-Min prior disappears. In the shape class of Section 5.2.3, the discrimination of shape also disappears as $M \to \infty$. The reason is simply that the function $\min\{-h^{\mathrm{T}}c_1, \ldots, -h^{\mathrm{T}}c_M\}$ yields a value which is not exactly equal to the minimum discrete radius of curvature, but one that is only proportional to it, and in fact, the constant of proportionality depends on M. We recall that the true discrete radius of curvature r_i (see (4.18)) is a quantity directly analogous to the continuous radius of curvature, while $\rho_i = -h^{\mathrm{T}}c_i$ is related to r_i via the equation

$$\rho_i \equiv -h^{\mathrm{T}}c_i = \frac{1}{2} r_i \theta_0 \tan\theta_0$$

where $\theta_0 = 2\pi/M$. Hence, to make $p_{CM}(h)$ of (5.16) be directly dependent on r_i we must choose

$$\tau = \bar{\tau}\left(\frac{\theta_0 \tan\theta_0}{2}\right) \tag{5.18}$$

where $\bar{\tau}$ is a new underlying parameter in the Close-Min prior — a real number independent of M and σ^2. We may now rewrite the Close-Min prior as

$$p_{CM}(h) = \frac{1}{z}\exp\left(\frac{2}{\bar{\tau}\theta_0 \tan\theta_0}\min\{-h^{\mathrm{T}}c_1, \ldots, -h^{\mathrm{T}}c_M\}\right). \tag{5.19}$$

Let us verify that this change produces the desired result for each of the classes of the previous sections. In each case we see that the quantity γ/τ contains the dependency on M. Since τ is now viewed as a function of M we must re-evaluate

this ratio as a function of M. We have

$$\frac{\gamma}{\tau} = \frac{1}{\bar{\tau}} \cdot \frac{\cos^{-1}\theta_0 - 1}{\frac{1}{2}\theta_0 \tan\theta_0} \tag{5.20}$$

which we recognize as $1/\bar{\tau}$ times the discrete radius of curvature of the support vector of length M which supports a unit circle. Although this quantity is *not* independent of M we may show that $\lim_{M\to\infty} \gamma/\tau = 1/\bar{\tau}$, which is a desirable property since this shows that the shape and size discrimination does not disappear as M gets large. We see this from (5.20) and the following manipulations:

$$\begin{aligned}
\lim_{M\to\infty} \frac{\cos^{-1}\theta_0 - 1}{\frac{1}{2}\theta_0 \tan\theta_0} &= \lim_{M\to\infty} \frac{2 - 2\cos\theta_0}{\theta_0 \sin\theta_0} \\
&= \lim_{M\to\infty} \frac{2\sin\theta_0}{\sin\theta_0 \div \theta_0 \cos\theta_0} \\
&= \lim_{M\to\infty} \frac{2}{1 \div \frac{\theta_0}{\tan\theta_0}} \\
&= \lim_{M\to\infty} \frac{2}{1 \div \frac{\theta_0}{\sin\theta_0} \cdot \frac{\cos\theta_0}{1}} \\
&= 1 \ .
\end{aligned}$$

Note that for $M = 5$, the smallest value that fits into our theory, we have that $\gamma/\tau = 1.15632/\bar{\tau}$, a ratio not much different than that when $M \to \infty$ — i.e. for practical purposes this choice of τ makes the Close-Min prior independent of the support vector dimension M.

5.2.6 Summary

We now have a fairly complete description of the implied Close-Min prior $p_{CM}(h)$. The probability is constant for support vectors on the relative boundary of C_p but rises to a maximum (for a given t) at the central axis in an exponential fashion. The probability rises exponentially as h is increased in size, where the rate of increase is smaller for support vectors nearer the relative boundary of C_p. We have also observed that adding a shift vector h_n does not change the probability, and finally, that applying a circular rotation to the elements of h does not change the probability.

In Section 5.2.5 we saw that the various dependencies on the value of M can be effectively eliminated by choosing τ so that it depends on M in a particular fashion. With τ so chosen, we may now view $p_{CM}(h)$ as a prior on objects — specifying indirectly their shape, size, position, and orientation — rather than simply a prior on support vectors h. This is an important concept which plays a large role in the development of priors (and corresponding MAP estimation algorithms) in the remainder of this chapter.

We now summarize the Close-Min MAP methods. In order to use the Close-Min algorithm as an MAP estimator we begin by selecting $\bar{\tau}$, the (only) underlying parameter of the Close-Min prior. Then with M given by the geometry of the problem we determine τ using equation (5.18). Also, we assume that the noise variance σ^2 is known so that we may now calculate α^*. Finally, we set $\alpha = \alpha^*$ and use the Close-Min algorithm of Chapter 4 to find $\hat{h}_{CM}$ given observations y.

5.3 Scale-Invariant Algorithms

In this section we develop three prior probability density functions on support vectors which depend upon shape alone. Each prior makes circular objects more likely *a priori*, but this effect is accomplished in different ways. For each prior, we develop an algorithm to find the MAP estimate given noisy observations; experimental results are given in Section 5.5.

The first prior incorporates a simple modification to the Close-Min prior to eliminate size dependency, and it leads to what we will call the Scale-Invariant Close-Min (SICM) algorithm. The objective function of the resultant MAP estimation problem is non-quadratic, and therefore requires a more complicated algorithm than either the LP or QP methods of Chapter 4. We solve the problem by developing a line search algorithm which uses quadratic interpolation and successive iterations to find the optimum (scalar) size, where at each size, we solve a QP to find the corresponding optimum shape vector.

The second prior addresses the undesirable property of the Close-Min prior that gives the same prior probability to any two support vectors which are on

the relative boundary of C_p, therefore implying an equal prior probability on all basic objects which have one or more radii of curvature equal to zero. We propose a prior that gives more eccentric figures lower prior probability and, keeping the scale-invariance property of the previous prior, this leads to the Scale-Invariant Closest (SIC) algorithm. The complexity of the algorithm is the same as the SICM algorithm, and is solved by essentially the same method.

The third prior places higher probability on support vectors whose basic objects have larger area to circumference ratio — while still maintaining scale-invariance. The resultant MAP problem is called the Scale-Invariant Maximum Area (SIMA) algorithm; it is solved using a line search algorithm similar to that used for the previous two estimates.

5.3.1 The Scale-Invariant Close-Min Algorithm

The Close-Min prior of (5.9) depends on the size of the support vector since t appears in the exponent when the SSS decomposition is applied (see (5.16)). To eliminate this dependency we simply divide the exponent by $h^{\mathrm{T}}e/M$ yielding

$$p_{SICM}(h) = \frac{1}{z}\exp\left(\frac{1}{\tau}\frac{\min\{-h^{\mathrm{T}}c_1,\ldots,-h^{\mathrm{T}}c_M\}}{h^{\mathrm{T}}e/M}\right) \tag{5.21}$$

which leads to the MAP problem

$$\hat{h}_{SICM} = \underset{h\in C}{\operatorname{argmax}}\ \frac{1}{\sigma^2}f_C(h) + \frac{M}{\tau}f_M(h)/h^{\mathrm{T}}e \tag{5.22}$$

where $f_C(h)$ and $f_M(h)$ are given in (5.2) and (5.3), respectively. To see that (5.21) is indeed independent of size we simply substitute $h = tq + h_n$ into the expression yielding

$$p_{SICM}(h) = \frac{1}{z}\exp\left(\frac{1}{\tau}\min\{-q^{\mathrm{T}}c_1,\ldots,-q^{\mathrm{T}}c_M\}\right)\ . \tag{5.23}$$

which clearly shows a dependency on shape q only.

To find an efficient solution to (5.22) we decompose h into its SSS components and simplify the result. First, writing $h = h_p + h_n$ and using the expressions for $f_C(h)$ and $f_M(h)$ we write the objective function of (5.22) as

$$f_{SICM}(h) \ = \ -\frac{1}{2\sigma^2}(y_n - h_n)^{\mathrm{T}}(y_n - h_n) - \frac{1}{2}(y_p - h_p)^{\mathrm{T}}(y_p - h_p)$$

$$+\frac{1}{\tau}\frac{M}{h_p^{\mathrm{T}}e}\min\{-h_p^{\mathrm{T}}c_1,\ldots,-h_p^{\mathrm{T}}c_M\} \tag{5.24}$$

where y_n and y_p are orthogonal vectors summing to y, and y_n is in the nullspace of C. (Note that the notation y_p does not mean to imply that y_p is in the proper cone.) Since h_n is unconstrained and its value does not affect the last two terms, the first term on the right-hand side of in (5.24) is maximized (to the value zero) by setting $\hat{h}_n = y_n$. Now since $h_p = tq$ by the SSS decomposition, we see that h_p may be found by maximizing

$$\begin{aligned} F(t,q) &= -\frac{1}{2\sigma^2}(y_p - tq)^{\mathrm{T}}(y_p - tq) + \frac{1}{\tau}\frac{Mt}{h_p^{\mathrm{T}}e}\min\{-q^{\mathrm{T}}c_1,\ldots,-q^{\mathrm{T}}c_M\} \\ &= -\frac{1}{2\sigma^2}(y_p - tq)^{\mathrm{T}}(y_p - tq) + \frac{1}{\tau}\min\{-q^{\mathrm{T}}c_1,\ldots,-q^{\mathrm{T}}c_M\} \end{aligned} \tag{5.25}$$

with respect to (feasible) t and q.

To further simplify the problem statement we now use the same technique of augmentation appearing in Chapter 4 that allowed us to convert the Mini-Max problem into an LP and to solve the Close-Min problem with a QP. Utilizing an extra real unknown ς, we find that $\hat{h}_p = \hat{t}\hat{q}$ may be found by solving

$$\begin{aligned} \underset{t,q,\varsigma}{\text{minimize}} \quad & \frac{1}{2\sigma^2}(y_p - tq)^{\mathrm{T}}(y_p - tq) - \frac{1}{\tau}\varsigma \\ \text{subject to} \quad & t \geq 0, \\ & q^{\mathrm{T}}C \leq 0, \qquad q^{\mathrm{T}}N = 0, \\ & q^{\mathrm{T}}e = M, \text{ and} \\ & \varsigma \leq -q^{\mathrm{T}}c_i, \quad i = 1,\ldots,M. \end{aligned} \tag{5.26}$$

The full Scale-Invariant Close-Min estimate is then given by

$$\hat{h}_{SICM} = \hat{t}\hat{q} + \hat{h}_n \ . \tag{5.27}$$

Although, the constraints in (5.26) are linear with respect to the unknowns, the objective function is, unfortunately, fourth order. Therefore, the QP algorithm used to solve the Close-Min algorithm of Chapter 4 cannot be used directly. However, we see that if t were known, then the optimum q (and ς) could be found directly using

a QP. This property suggests a type of line search algorithm which searches for the optimum size by choosing various values of t, finding the optimum q for that t, and seeking the t that minimizes (5.25) for q determined by (5.26). We summarize the algorithm below.

Algorithm 5.1 (Scale-Invariant Close-Min)

1. *Define the function of t to be minimized:*

$$F(t) = \frac{1}{2\sigma^2}(y_p - tq_\star)^{\mathrm{T}}(y_p - tq_\star) - \frac{1}{\tau}\min\{-q_\star^{\mathrm{T}}c_1, \ldots, -q_\star^{\mathrm{T}}c_M\}$$

   where $q_\star$ is the solution of (5.26) for fixed t.

2. *Bracket a minimum — that is, find t_a, t_b, and t_c so that $t_a < t_b < t_c$ and $F(t_a) > F(t_b)$ and $F(t_c) > F(t_b)$ using the golden section method [68].*

3. *Refine the minimum to within a prespecified tolerance using quadratic interpolation and successive iterations via Brent's method [6,68].*

Experimental results for this algorithm are given in Section 5.5.

5.3.2 The Scale-Invariant Closest Algorithm

We return now to one of the peculiarities of the Close-Min prior that was characterized by the fact that all support vectors on the relative boundary of $\mathcal{C}_p$ have the same prior probability. The implication of this fact, as demonstrated by Fig. 5.1, is that basic objects of drastically different eccentricities may have the same prior probability. In this section we develop a new prior which corrects this deficiency using the following heuristic: proper support vectors satisfying $h^{\mathrm{T}}e = M$ — i.e., the shape vectors q of the SSS decomposition — tend to have more eccentric basic objects the farther they are from e. We will use $\|q - e\|$ as a measure of eccentricity in this section; an alternate measure will be used in Section 5.3.3 and a more precise development of the concept of eccentricity will be done in Section 5.4.

Given the above comments, we specify the Scale-Invariant Closest (SIC) prior to favor circular objects as in the Close-Min and Scale-Invariant Close-Min priors, but

to place the dependency on $\|q-e\|^2$ rather than on the minimum discrete radius of curvature. Accordingly, we have

$$p_{SIC}(h) = \frac{1}{z}\exp\left(-\frac{1}{\tau}(q-e)^{\mathrm{T}}(q-e)\right) \tag{5.28}$$

This leads directly to the MAP estimate given by

$$\begin{aligned}\hat{h}_{SIC} &= \operatorname*{argmax}_{h\in C} -\frac{1}{2\sigma^2}(y-h)^{\mathrm{T}}(y-h) - \frac{1}{\tau}(q-e)^{\mathrm{T}}(q-e) \\ &= \operatorname*{argmax}_{h\in C} -\frac{1}{2\sigma^2}(y-h)^{\mathrm{T}}(y-h) - \frac{1}{\tau}q^{\mathrm{T}}q \\ &= \operatorname*{argmin}_{h\in C} \frac{1}{2\sigma^2}(y-h)^{\mathrm{T}}(y-h) + \frac{1}{\tau}q^{\mathrm{T}}q \end{aligned} \tag{5.29}$$

where it is understood that q is the shape component of h according to the SSS decomposition. The second equality in (5.29) follows because $q^{\mathrm{T}}e = M$ according to the decomposition, and any constant term in the objective function may be eliminated since it does not affect the outcome of the maximization. We interpret (5.29) as follows. As in each of the previous algorithms, the estimator attempts to keep the distance between the observation and the estimate small. But in this case it is simultaneously trying to keep the magnitude of q as small as possible, which keeps it close to e.

To compute $\hat{h}_{SIC}$ given an observation y, we use an algorithm similar to that of the previous section. The nullvector component of the solution is, as before, given by y_n and the proper component is found using the following algorithm.

Algorithm 5.2 (Scale-Invariant Closest)

1. *Define the function of t to be minimized:*

$$F(t) = \frac{1}{2\sigma^2}(y_p - tq_*)^{\mathrm{T}}(y_p - tq_*) + \frac{1}{\tau}q_*^{\mathrm{T}}q_*$$

where q_* is the solution of

$$\begin{aligned} \operatorname*{minimize}_{q} \quad & \frac{1}{2\sigma^2}(y_p - tq)^{\mathrm{T}}(y_p - tq) + \frac{1}{\tau}q^{\mathrm{T}}q \\ \text{subject to} \quad & q^{\mathrm{T}}C \le 0\,, \text{and} \\ & q^{\mathrm{T}}e = M\,. \end{aligned}$$

for fixed t.

2. *Bracket a minimum — that is, find t_a, t_b, and t_c so that $t_a < t_b < t_c$ and $F(t_a) > F(t_b)$ and $F(t_c) > F(t_b)$ — using the golden section method [68].*
3. *Refine the minimum to within a prespecified tolerance using quadratic interpolation and successive iterations via Brent's method [6,68].*

Experimental results are given in Section 5.5.

5.3.3 The Scale-Invariant Maximum Area Algorithm

It is well-known that the circle is the figure which maximizes the ratio of the area to the square of the circumference[58]. Also, this ratio grows smaller for more eccentric figures. It is reasonable (if one has reason to expect round objects) therefore to specify a prior probability which gives greater probability to support vectors whose basic objects have larger area to (circumference)^2 ratio. Since all basic objects corresponding to shape vectors q have the same circumference $P = 2M\gamma/\tan\theta_0$, then specifying a prior in terms of the area of q implicitly yields an expression depending on the above ratio. Denoting the area of q by $S(q)$, we define the Scale-Invariant Maximum Area (SIMA) prior to be

$$p_{SIMA}(h) = \frac{1}{z}\exp\left(\frac{1}{\tau}S(q)\right) \quad . \tag{5.30}$$

Since $p_{SIMA}(h)$ is a growing exponential with $S(q)$, we expect that the most eccentric basic objects, such as the line segment in Fig. 5.1 will have smaller areas and therefore lower probabilities, while the most nearly circular basic objects will have larger areas and therefore larger prior probability. In Appendix 5.A we find that $S(q) = -q^{\mathrm{T}}Cq/\tan\theta_0$, so that we may now write the SIMA prior as

$$p_{SIMA}(h) = \frac{1}{z}\exp\left(\frac{-1}{\tau\tan\theta_0}q^{\mathrm{T}}Cq\right) \tag{5.31}$$

and the corresponding MAP estimate as

$$\hat{h}_{SIMA} = \underset{h:\, h\in C}{\operatorname{argmax}} \ \frac{-1}{2\sigma^2}(y-h)^{\mathrm{T}}(y-h) + \frac{-1}{\tau\tan\theta_0}q^{\mathrm{T}}Cq \ . \tag{5.32}$$

As in the previous two scale-invariant algorithms, the objective function for the above optimization problem is non-quadratic and therefore cannot be solved directly

via a QP. The method we use to compute this estimate is almost identical to the SIC estimate: the nullspace component is given by $\hat{h}_n = y_n$ and the proper component is given by the following algorithm.

Algorithm 5.3 (Scale-Invariant Maximum Area)

1. *Define the function of t to be minimized:*

$$F(t) = \frac{1}{2\sigma^2}(y_p - tq_*)^{\mathrm{T}}(y_p - tq_*) + \frac{1}{\tau \tan\theta_0} q_*^{\mathrm{T}} C q_*$$

where q_* is the solution of

$$\begin{array}{ll} \underset{q}{\text{minimize}} & F(q,t) = \frac{1}{2\sigma^2}(y_p - tq)^{\mathrm{T}}(y_p - tq) + \frac{1}{\tau \tan\theta_0} q^{\mathrm{T}} C q \\ \text{subject to} & q^{\mathrm{T}} C \leq 0 \ \textit{and} \\ & q^{\mathrm{T}} e = M\,. \end{array}$$

for fixed t.

2. *Bracket a minimum — that is, find t_a, t_b, and t_c so that $t_a < t_b < t_c$ and $F(t_a) > F(t_b)$ and $F(t_c) > F(t_b)$ — using the golden section method [68].*
3. *Refine the minimum to within a prespecified tolerance using quadratic interpolation and successive iterations via Brent's method [6,68].*

Experimental results are given in Section 5.5.

5.4 Ellipse-Based Estimation

5.4.1 Introduction

The prior probabilities on support vectors that we have specified so far — Close-Min, SI Close-Min, SI Closest, and SI Max-Area — tend to yield larger probabilities for circular objects. In this section we examine several estimation formulations based on the prior knowledge that *the objects we seek are more elliptical in shape.* Thus since the circle is the simplest type of ellipse, what we are doing is to extend the

scope of our prior shape knowledge to include something about the *eccentricity* and *orientation* of the object. This type of information is very common in medical CT scanning. For example, the skull is often very nearly elliptic, and its orientation is usually known quite accurately. It is possible that in such situations, this additional knowledge — particularly in the limited- or sparse-angle cases — could lead to greatly improved reconstructions.

The situation we are most interested in in this section is the situation where we know that the object is likely to be *near* the shape of an ellipse, but not (necessarily) *exactly* that of an ellipse. In Section 5.4.2, we develop a parametric expression for the support vector of an ellipse using the SSS decomposition. Then, we examine three estimation algorithms which use this expression in different ways to derive some knowledge about the elliptic nature of the unknown set. For example, if we knew the object to be *exactly* an ellipse, then we might consider a formulation which directly estimates the ellipse parameters: position, size, orientation and eccentricity. This problem was studied in great detail for the general tomographic reconstruction problem by Rossi and Willsky [74,75]. We present in Section 5.4.3 an ML estimation algorithm which estimates the ellipse parameters from noisy support vector observations; we call this algorithm the Closest Ellipse (CE) algorithm. This algorithm may also be used, for example, after any support vector estimation procedure to determine the parameters of the ellipse whose support vector is nearest to the estimate. The second algorithm makes a simple modification to the SI Closest algorithm so that the prior probability on the support vector has its highest probability at a prespecified ellipse rather than at a circle. This algorithm therefore assumes some specific knowledge of the ellipse parameters *a priori*. Finally, we explore an algorithm which seeks to *jointly* estimate the feasible support vector and the ellipse parameters; this algorithms uses only the prior knowledge that the true object is nearly elliptical.

5.4.2 Support Vectors of Ellipses

One set of parameters which completely describes an ellipse are its position, size, orientation and eccentricity. To quantify these terms let us denote the length of the

longer semiaxis by a and the shorter semiaxis by b, as shown in Fig. 5.2a. Then the *eccentricity* ϵ is given by [86]

$$\epsilon = \frac{\sqrt{a^2 - b^2}}{a} \tag{5.33}$$

so that $0 \leq \epsilon \leq 1$ where $\epsilon = 0$ corresponds to a circle, and the ellipse becomes more elongated as $\epsilon \to 1$. The *position* $v \in \mathbb{R}^2$ of an ellipse is simply the position in the plane of the centroid of the ellipse, and the *orientation* ϕ is the angle that the longest semiaxis makes with the x-axis. Both of these quantities are depicted in Fig. 5.2b.

We might think of the *size* of an ellipse as simply a (or b, since either one uniquely determines the other using ϵ and the fact that $a \geq b$), or perhaps its area or circumference. However, in order to maintain consistency with the developments of the earlier sections in this chapter, we will use the quantity t of the SSS decomposition to mean *size* of the ellipse. In fact, we have already shown that knowledge of t is equivalent to knowing the circumference of the *basic object* associated with a support vector. Thus, we acknowledge the fact that although we know the object to be an ellipse, our observations consist only of a finite set of (noisy) support lines to the ellipse.

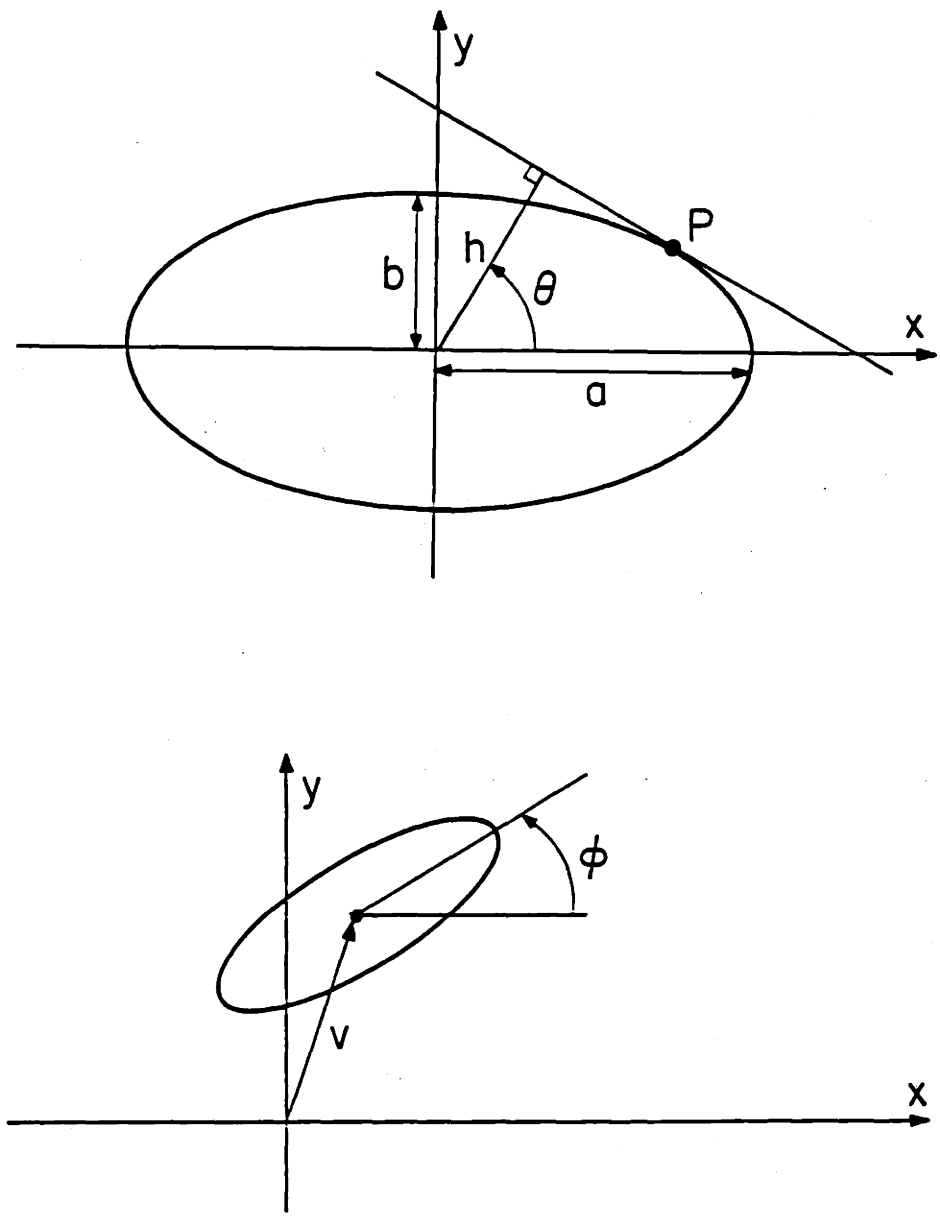
We now proceed to find an expression for the support vector $h(v, t, \epsilon, \phi)$ which corresponds to the ellipse parameterized by position v, size t, eccentricity ϵ, and orientation ϕ. In Appendix 5.C we find that the support *function* of an ellipse may be written as a function of the parameters a, b, v, and ϕ as

$$h(\theta) = \sqrt{a^2 \cos^2(\theta - \phi) + b^2 \sin^2(\theta - \phi)} + [\cos\theta \;\; \sin\theta]v \,. \tag{5.34}$$

The support vector is just a sampled version of the support function, so we have that the i^{th} support value is given by

$$\begin{aligned} h_i &= \sqrt{a^2 \cos^2(\theta_i - \phi) + b^2 \sin^2(\theta_i - \phi)} + [\cos\theta_i \;\; \sin\theta_i]v \\ &= b\sqrt{\frac{1}{1-\epsilon^2}\cos^2(\theta_i - \phi) + \sin^2(\theta_i - \phi)} + [\cos\theta_i \;\; \sin\theta_i]v \,, \end{aligned} \tag{5.35}$$

where to get the second expression we used the definition of eccentricity in (5.33). However, this does not give us the support vector in terms of v, t, ϵ, and ϕ as



(b)

Figure 5.2: Ellipse parameters.

required — we must therefore make some additional manipulations. The size t of a basic object is equal to the *average* of the elements of a support vector (see Section 5.2), so, for the above support vector we have that

$$t = \frac{b}{M} \sum_{j=1}^{M} \sqrt{\frac{1}{1-\varepsilon^2} \cos^2(\theta_j - \phi) + \sin^2(\theta_j - \phi)} \,. \tag{5.36}$$

Now we may solve (5.36) for b and substitute this back into (5.35) to yield

$$h_i = \frac{Mt\sqrt{\frac{1}{1-\varepsilon^2} \cos^2(\theta_i - \phi) + \sin^2(\theta_i - \phi)}}{\sum_{j=1}^{M} \sqrt{\frac{1}{1-\varepsilon^2} \cos^2(\theta_j - \phi) + \sin^2(\theta_j - \phi)}} + [\cos\theta_i \;\; \sin\theta_i]v \,, \tag{5.37}$$

which is the desired expression for the elements of $h(v, t, \varepsilon, \phi)$.

5.4.3 Closest Ellipse (CE) Algorithm

In this section we develop an algorithm to determine the *closest* ellipse support vector to an arbitrary vector. The Closest Ellipse (CE) algorithm serves two purposes. First, in some instances one might desire to learn something about the elliptical nature of an arbitrary support vector. This algorithm finds the closest ellipse support vector to this support vector by finding the ellipse parameters v, t, ε, and ϕ. For example, this algorithm might be used after using the Closest algorithm or any one of the SI algorithms which estimates a support vector from a set of noisy measurements. Second, if the arbitrary vector is a noisy observation of a true ellipse support vector, and the elements of the noise vector are independent zero-mean Gaussian random variables, then the CE algorithm actually finds the ML estimates of the ellipse parameters directly.

The mathematical statement of this problem is simple: find v, t, ε, and ϕ to minimize $\|z - h(v, t, \varepsilon, \phi)\|^2$ where $z \in C$ is an arbitrary vector. Using the SSS decomposition we may write $z = t_z q_z + N v_z$, and it is not difficult to see that $\hat{v} = v_z$. What remains is a constrained optimization problem in three dimensions:

$$\begin{aligned} \underset{t,\varepsilon,\phi}{\text{minimize}} \quad & f(t, \varepsilon, \phi) = \|z - h(v_z, t, \varepsilon, \phi)\|^2 \\ \text{subject to} \quad & t \geq 0 \,, \\ & 0 \geq \varepsilon \geq 1 \,. \end{aligned} \tag{5.38}$$

Because of the complex form of $h(v_z, t, \varepsilon, \phi)$, this problem is highly nonlinear and must be solved iteratively. Since the objective function is differentiable and the interesting solutions do not lie on the constraints, we use the conjugate gradient method to solve this problem [56]. We show the gradient calculations in Appendix 5.D.

5.4.4 The ESIC Algorithm

If we knew *a priori* that the true object shape is nearly elliptical with eccentricity $\bar{\varepsilon}$ and orientation $\bar{\phi}$, then our reconstruction algorithm should favor shapes which resemble the ellipse with these parameters. This type of knowledge, as we shall see in Section 5.5, can greatly improve reconstructions, particularly in the limited- and sparse-angle cases, where prior knowledge plays an important role. The algorithm we present here parallels the Scale-Invariant Closest algorithm of Section 5.3.2, and is solved in almost identical fashion.

We begin by specifying the Ellipse-based Scale-Invariant Closest (ESIC) prior as

$$p_{ESIC}(q) = \frac{1}{z} \exp\left\{ -\frac{1}{\tau} \left\| q - h(0, 1, \varepsilon, \phi) \right\|^2 \right\}, \tag{5.39}$$

where $\|x\|^2$ denotes $x^T x$. This prior is identical in form to the SI Closest prior except that the largest probabilities are concentrated around the ellipse whose support *shape* vector[6] is $h(0, 1, \bar{\varepsilon}, \bar{\phi})$ rather than the shape vector e. Now let us assume that our observations of the true support vector are given by $y = h + n$ and that n has elements which are independent zero-mean Gaussian random variables with variance σ^2. Then the MAP estimate $\hat{h}_{ESIC}$ may be formulated and solved in nearly identical fashion to the SIC MAP problem of Section 5.3.2. The resulting algorithm given below performs a line search in t, solving a QP at each stage, until the jointly optimum (t, q) pair is determined.

[6]The fact that the t-coordinate is 1 and the v-coordinate is 0, makes this a shape vector (see Section 5.2).

Algorithm 5.4 (Ellipse-based Scale-Invariant Closest)

1. *Define the function of t to be minimized:*

$$F(t) = \frac{1}{2\sigma^2}\|y_p - tq_*\|^2 + \frac{1}{\tau}\left\|q_* - h(0,1,\bar{\varepsilon},\bar{\phi})\right\|^2$$

where q_* is the solution of

$$\begin{array}{ll} \underset{q}{\text{minimize}} & \frac{1}{2\sigma^2}\|y_p - tq)\|^2 + \frac{1}{\tau}\left\|q - h(0,1,\bar{\varepsilon},\bar{\phi})\right\|^2 \\ \text{subject to} & q^{\mathrm{T}}C \leq 0\,, \text{and} \\ & q^{\mathrm{T}}e = M\,. \end{array}$$

for fixed t.

2. *Bracket a minimum — that is, find t_a, t_b, and t_c so that $t_a < t_b < t_c$ and $F(t_a) > F(t_b)$ and $F(t_c) > F(t_b)$ — using the golden section method [68].*

3. *Refine the minimum to within a prespecified tolerance using quadratic interpolation and successive iterations via Brent's method [6,68].*

As in the SI problems of Section 5.3, there is a corresponding algorithm for the case of partial observations (refer to Appendix 5.B). Some experimental results are presented in Section 5.5.

5.4.5 Joint Support Vector/Ellipse Parameter Estimation

Now we extend the CE and ESIC algorithms presented in the previous two sections to estimate *jointly* a support vector and ellipse parameters. Here, the prior knowledge we utilize is that the true support vector is likely to be near to the shape of an ellipse, *but we do not know which ellipse* a priori. We write this problem formally as

$$\begin{array}{ll} \underset{h,v,t,\varepsilon,\phi}{\text{minimize}} & \alpha\|h - y\|^2 + (1-\alpha)\|h - h(v,t,\varepsilon,\phi)\|^2 \qquad 0 < \alpha \leq 1 \quad (5.40) \\ \text{subject to} & t \geq 0\,, \\ & 0 \leq \varepsilon \leq 1\,, \quad \text{and} \\ & h^{\mathrm{T}}C \leq 0\,. \qquad (5.41) \end{array}$$

We note that if $\alpha = 1$ then the objective function of (5.41) is independent of the ellipse parameters[7] v, t, ε, and ϕ, and the optimum h is found using the Closest algorithm of Chapter 4. Alternatively, as $\alpha \to 0$, the optimum ellipse parameters approach their ML estimates — found by a variant of the CE algorithm as discussed in Section 5.4.2 — and the optimum h approaches the ellipse support vector corresponding to the ML estimates of the ellipse parameters. These two extremes provide some insight as to the solution in the general case when $0 < \alpha < 1$. For example, the optimum h cannot be closer to y than $\hat{h}_C$, the Closest estimate, since then it would be infeasible; also, it cannot be farther away from y than $h(\hat{v}_{ML}, \hat{t}_{ML}, \hat{\varepsilon}_{ML}, \hat{\phi}_{ML})$.

The joint estimation problem of (5.41) is an optimization problem of greater nonlinearity than any encountered in this thesis so far. However, we have already gained a great deal of insight into the structure of this problem when we studied the Scale-Invariant algorithms in Section 5.3 and the Closest Ellipse algorithm in Section 5.4.3. For example, we see that (in a manner similar to the behavior of the SI algorithms and the parameter t) the globally optimum h may determined for fixed v, t, ε, and ϕ using a quadratic program (QP). Therefore, once we have determined the optimum ellipse parameters, the optimum support vector follows automatically, and these quantities are therefore jointly optimum. However, the opposite viewpoint has merits as well: for fixed h, the globally optimum ellipse parameters may be determined by the CE algorithm, so for optimum h we may immediately determine the jointly optimum ellipse parameters. This situation suggests a type of coordinate descent algorithm in which we alternate between finding the optimum h for fixed ellipse parameters and then finding the optimum ellipse parameters for fixed h. We summarize the algorithm below.

Algorithm 5.5 (JE Algorithm)

1. *Find the ML estimates of the ellipse parameters and set* $v^0 = \hat{v}_{ML}$, $t^0 = \hat{t}_{ML}$, $\varepsilon^0 = \hat{\varepsilon}_{ML}$, *and* $\phi^0 = \hat{\phi}_{ML}$.
2. *Solve (5.41) for* h, *keeping the ellipse parameters fixed to* $v = v^0$, $t = t^0$, $\varepsilon = \varepsilon^0$, *and* $\phi = \phi^0$. *Let the solution of this QP be denoted* h^0. *Set* $k = 0$.

[7] Here, v and t are *not* SSS coordinates of h.

3. *Find closest ellipse to* h^k *using the CE algorithm, yielding* v^{k+1}, t^{k+1}, ε^{k+1}, *and* ϕ^{k+1}.

4. *Solve (5.41) for h, keeping the ellipse parameters fixed to* $v = v^{k+1}$, $t = t^{k+1}$, $\varepsilon = \varepsilon^{k+1}$, *and* $\phi = \phi^{k+1}$. *This yields* h^{k+1}.

5. *If* $\|h^{k+1} - h^k\|^2 + \|v^{k+1} - v^k\|^2 + \|t^{k+1} - t^k\|^2 + \|\varepsilon^{k+1} - \varepsilon^k\|^2 + \|\phi^{k+1} - \phi^k\|^2 < \epsilon$ *then we are done and* $\hat{h} = h^k$, $\hat{v} = v^k$, $\hat{t} = t^k$, $\hat{\varepsilon} = \varepsilon^k$, *and* $\hat{\phi} = \phi^k$. *Otherwise, set* $k \leftarrow k + 1$ *and go to Step 3.*

Ordinarily, the convergence of this type of algorithm — a primal feasible directions method (see [56]) — is *not* guaranteed due to the possibility of *jamming*, but in this case jamming will not occur since the constraints are not coupled between h and the ellipse parameters. For example, in Step 3 the value of h does not affect the constraints on the ellipse parameters, and similarly, in Step 4 the value of the ellipse parameters does not change the constraints on h. Because of this condition, each step (3 and 4) causes the objective function to decrease (or at least not increase) and produce a globally optimal solution that is unconstrained by the fixed variables; therefore, convergence to the jointly optimal solution is guaranteed. Examples are shown in Section 5.5.

5.5 Experimental Results

In this section we present the results of several simulations designed to demonstrate the important features of the Scale-Invariant (SI) and Ellipse-Based (EB) algorithms. The first set of three experiments are concerned with the Scale-Invariant algorithms. First, we show the effect of varying the parameter τ (or $\bar{\tau}$ where appropriate) to show how the influence of prior knowledge can be varied. Second, we compare the different SI algorithms for a fixed τ on the same observations. Third, we show the behavior of the SI algorithms for limited- and sparse-angle cases. The second set of three experiments examine the behavior of three different approaches to Ellipse-Based estimation. First, we examine the results of the Closest Ellipse (CE) algorithm which estimates the closest ellipse support vector to a given arbi-

trary vector. Second, we examine the case in which a fixed ellipse *shape vector* is used in the SI Closest algorithm in place of the vector e — this is the ESIC algorithm. Finally, we examine the results of the JE algorithm which *jointly* estimates the support vector and the unknown ellipse parameters.

5.5.1 Scale-Invariant Experiments

Varying τ

Fig. 5.3 shows a sequence of trials which compare the SI Close-Min algorithm for different values of $\bar{\tau}$ (denoted by τ in the figure). The true support vector has 30 observed angles (i.e., $M = 30$), and has as its basic object the ellipse shown using the dashed line in each of the panels (a)–(f). Its vertex plot actually circumscribes a true ellipse centered at the origin with eccentricity 0.9 and orientation 45 degrees. The observations are created by adding independent zero-mean Gaussian noise samples with variance 0.01 to each element of the true support vector. The *vertex plot* (see Chapter 4) for the observations and the reconstruction using the intersection method is shown in panel (a). The remaining panels show reconstructions using the SI Close-Min (SICM) algorithm for (b) $\tau = 0.01$, (c) $\tau = 0.05$, (d) $\tau = 0.1$, (e) $\tau = 0.2$, and (f) $\tau = 0.5$. Figs. 5.4 and 5.5 show analogous plots for the SI Closest (SIC) and SI Max-Area (SIMA) algorithms, respectively.

In each of the three figures (Figs. 5.3, 5.4, and 5.5), panel (b) shows the result in which the prior knowledge of shape has the most influence. Therefore, these estimates are the most *circular* with respect to the remaining panels, but each is circular in a different way. In particular, the SICM algorithm produces a figure which contains two semi-circles connected by two line segments. Thus the discrete radii of curvature for this estimate are identical except for *two* — those corresponding to the two line segments — which are much larger. On the other hand, the SIC and SIMA estimates do appear to be more circular, except that the SIC estimate appears to have sharper corners on the boundary.

The reasons for the different behavior of the three algorithms lies in the way each SI prior specifies its probability on shape. The SI Close-Min prior defines

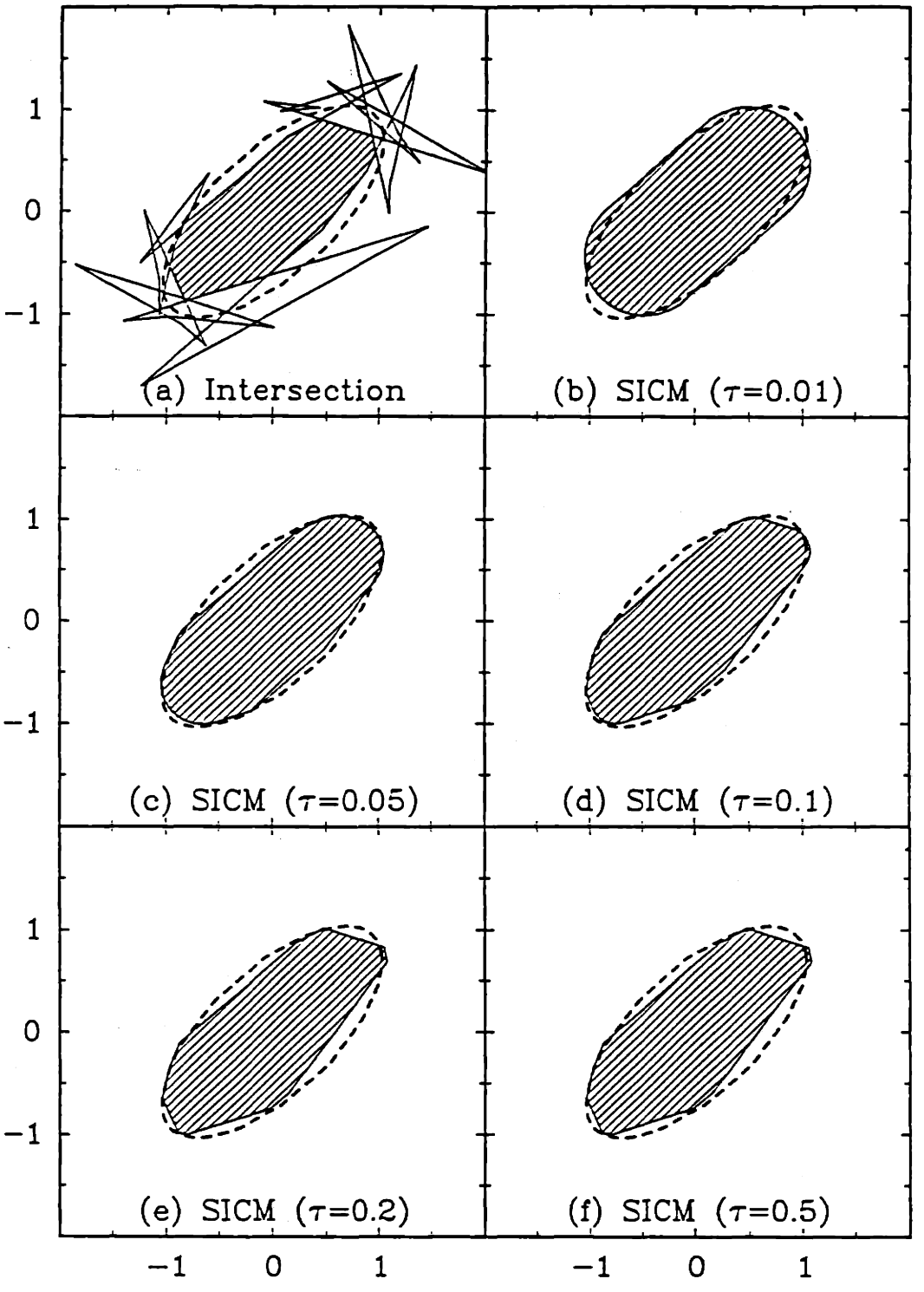


Figure 5.3: Comparison of the Scale-Invariant Close-Min algorithm for different values of τ.

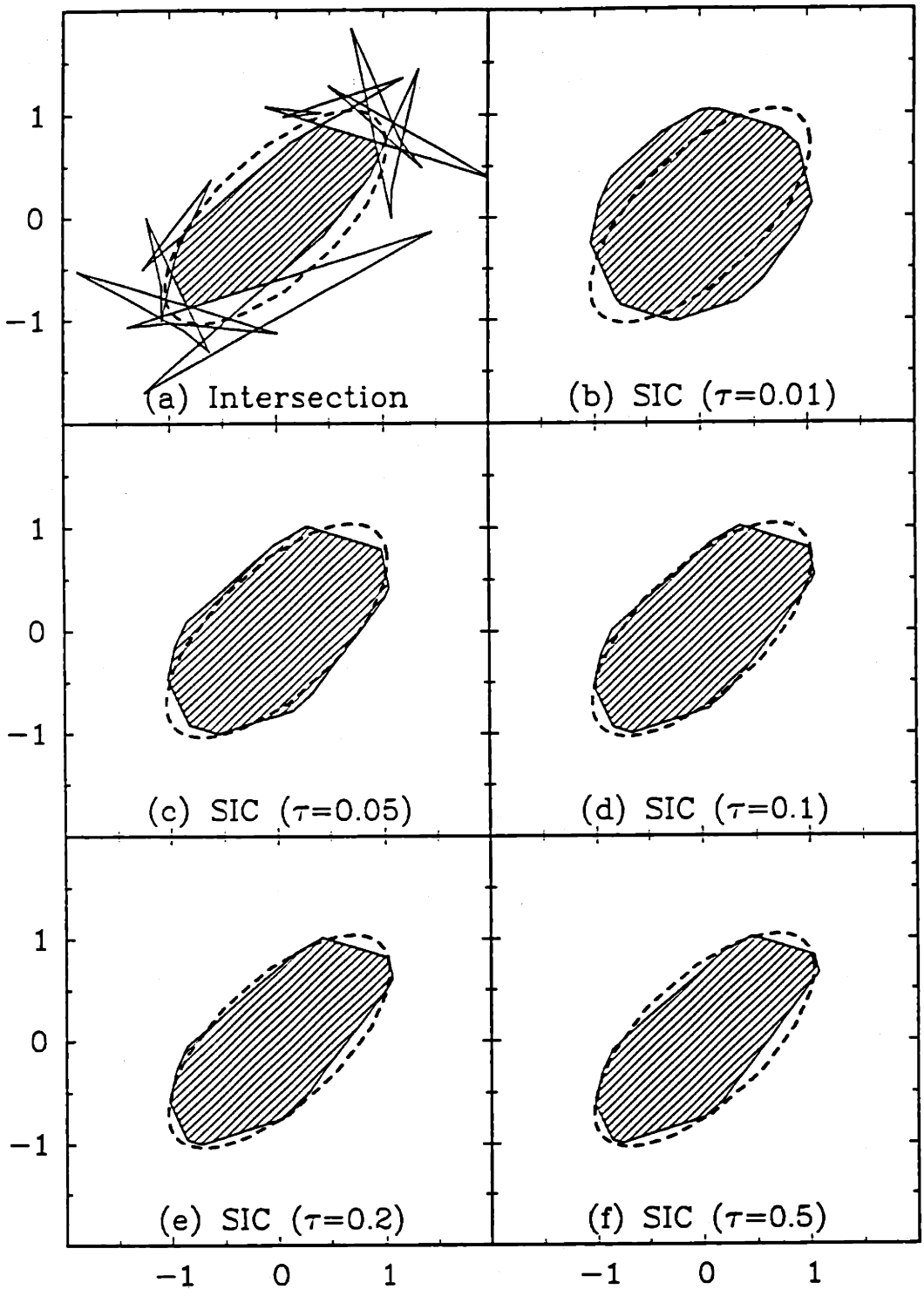


Figure 5.4: Comparison of the Scale-Invariant Closest algorithm for different values of τ.

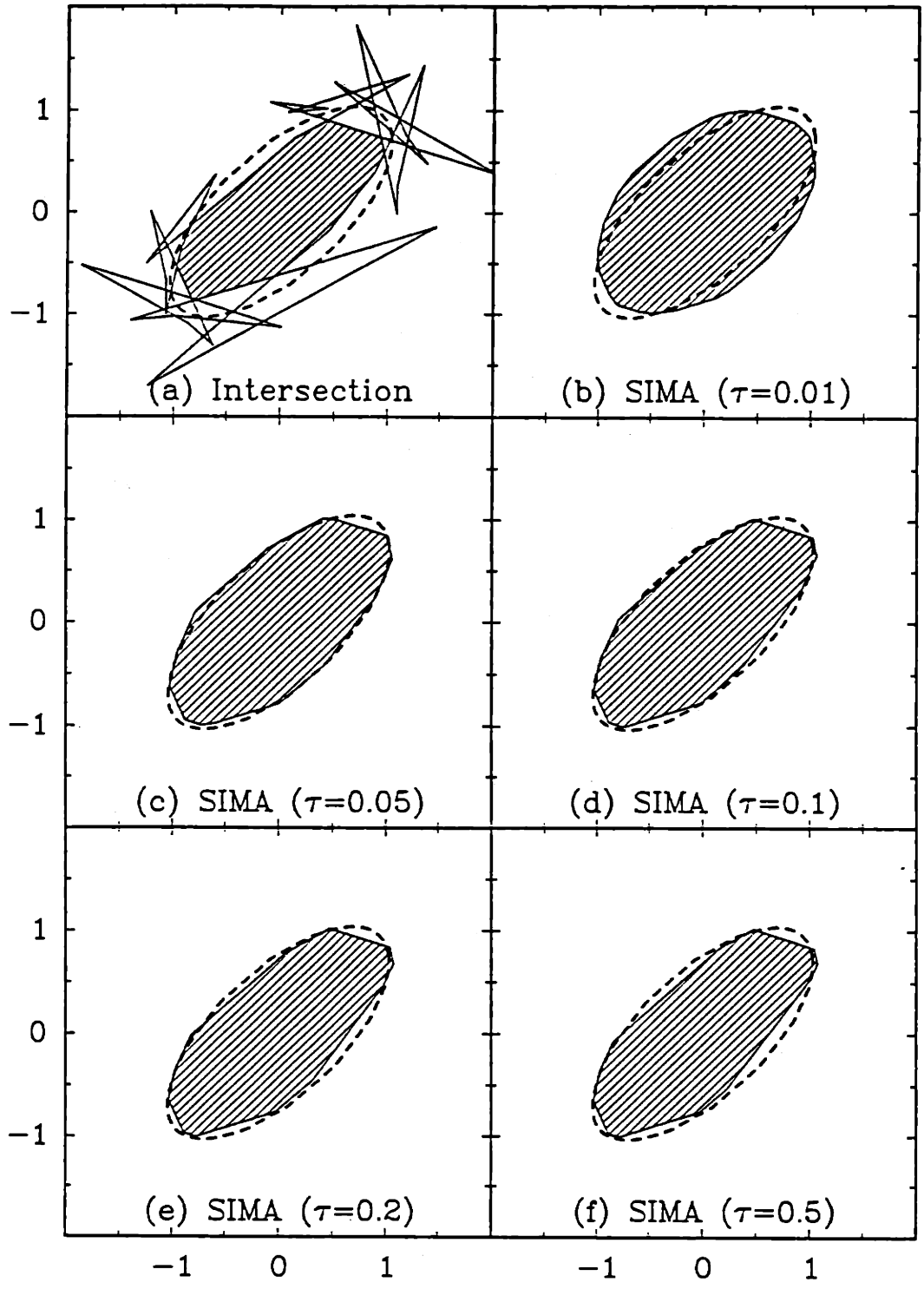


Figure 5.5: Comparison of the Scale-Invariant Max-Area algorithm for different values of τ.

the prior probability of a support vector through the discrete radii of curvature of its basic object. Therefore, it focuses on the *boundary* of the basic object and, in particular, it places higher probability on those objects which have the larger minimum discrete radius of curvature. Therefore, for a given circumference (i.e., objects with the same t coordinate) the most likely object is a circle, and the most likely object which is also elongated is one which has two semicircles connected by two line segments — objects with even one sharp corner are much less likely. The SI Max-Area prior gives larger probability to objects with larger area to circumference ratio. Therefore, given that these objects have the same circumference, a circle is more probable than an ellipse, and an ellipse more probable than a square, and a square more probable than a rectangle, for example. Although the focus is not explicitly on the boundary, the net effect is that the objects with more circular boundaries have higher probability. The SI Closest algorithm gives higher prior probability to shape vectors which are closest to e, and since the basic object of e is a regular polygon that circumscribes a circle, the most likely objects are circular in overall shape. Finally, it is important to note that as τ is made very small, all the SI estimates become circles regardless of the observations, since in each case, a circle has the highest prior probability.

The last panel (f) of each of the three figures show the result in which the *least* amount of prior knowledge is used. In fact, although it is not shown, each of these estimates are identical to the estimate produced by the Closest algorithm of Chapter 4. In the remainder of the experiments in this chapter we will use $\tau = 0.1$, which is shown in each of the (d) panels. The reason for using this value of τ is that we are primarily interested in producing smoother boundaries than that of the Closest algorithm (cf. panel (f)), not in producing circular objects. This choice of τ has this effect, as shown in each of the three (d) panels, and it also produces estimates that have the correct overall elliptical shapes.

Different SNR's

Figs. 5.6, 5.7, and 5.8 compare the SI algorithms, together with the Intersection, Closest, and Close-Min methods, using the same observations. The noise standard

deviation (also indicated in the (a) panels) is 0.1 for Fig. 5.6, 0.2 for Fig. 5.7, and 0.3 for Fig. 5.8; the true support vector is identical to that used in the previous section. The parameters that control the influence of prior knowledge are given by $\bar{\tau} = 0.1$ for the Close-Min and SI Close-Min algorithms, and $\tau = 0.1$ for the SI Closest and SI Max-Area algorithms. Also, note that the starting random number seed for the three examples was identical so that the initial (unit variance) noise sequence was identical — only the multiplying constant was different.

The main point to be observed from this set of figures (Figs. 5.6, 5.7, and 5.8) is that as the noise gets larger, the effect of prior knowledge increases. This is really just a simple property which is common to all MAP estimators; it reflects the fact that σ^2 is in the denominator of the term in the objective function which involves the measurements. But this property has a striking effect on *all* of the MAP estimators (including the Close-Min algorithm). For example, the results in Fig. 5.8 (panels (d)–(f)) show objects which are very similar to the objects shown in the (b) panels of Figs. 5.3–5.5, respectively. In other words, when there is a large amount of noise, each algorithm produces an object which resembles those objects which have higher probability for that particular prior. In contrast, the Closest estimate (panel (b)) has an appearance which is remarkably similar in all three figures. This situation leads us to make the following comment: as the level of noise increases, the influence of prior knowledge also increases, but in this case its effect becomes *too strong*. The reason for this effect is that the support cone constraint is a *fundamental* mathematical constraint on support vectors which is not imposed simply because the prior probability is zero outside the cone. Therefore, the effect of the constraint is felt even for the ML part of the objective function; it has the effect of reducing the effective variance of the measurement noise independently of the effects of the prior knowledge. Therefore, the normal tradeoff between prior knowledge and measurements in the MAP formulation is not appropriate, and because of this situation, it is reasonable to consider making τ a function of the noise variance σ^2 so that, in particular, τ increases with σ^2.

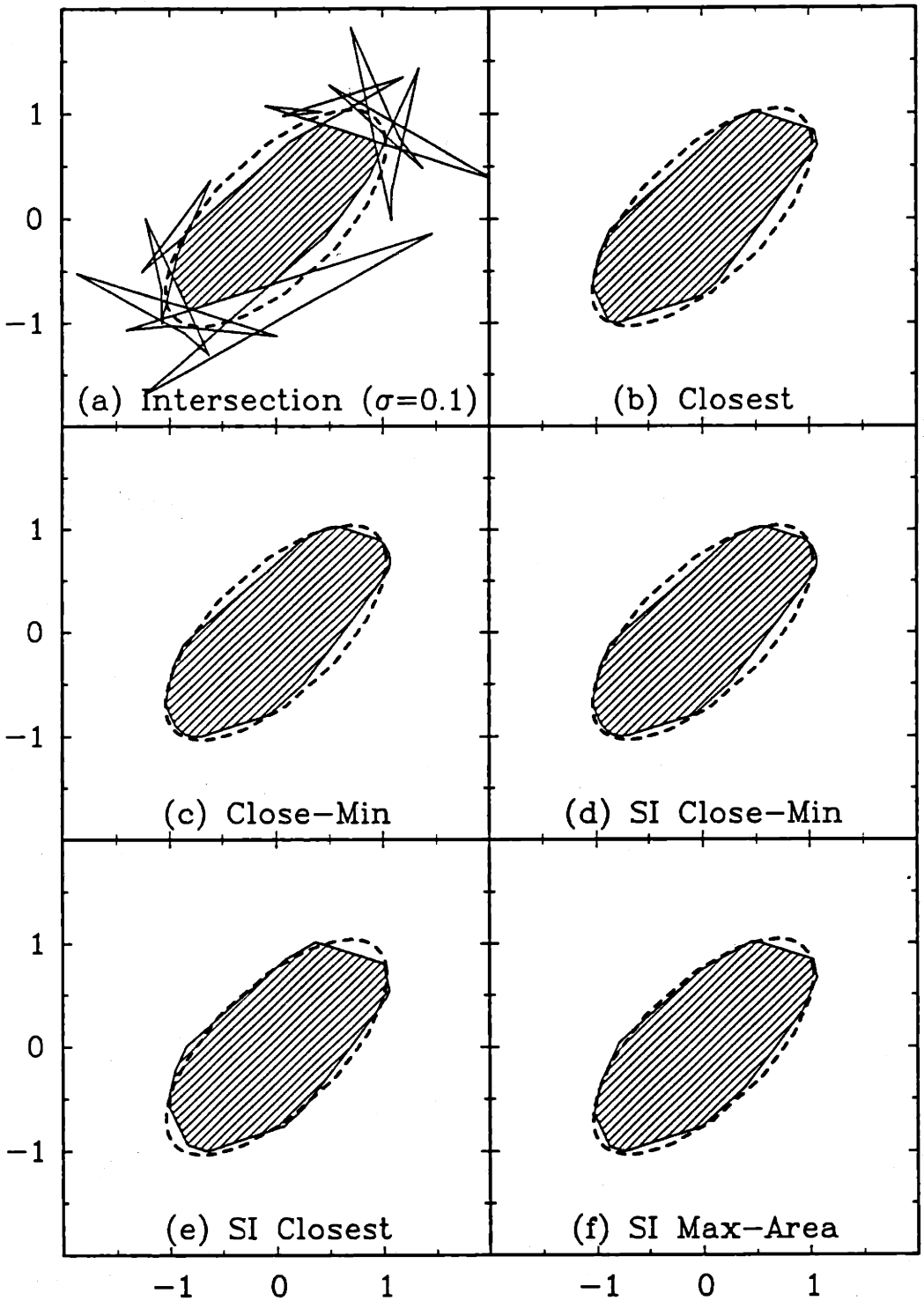


Figure 5.6: Comparison of support vector estimation algorithms for $M = 30$, $\sigma = 0.1$, and $\tau = 0.1$.

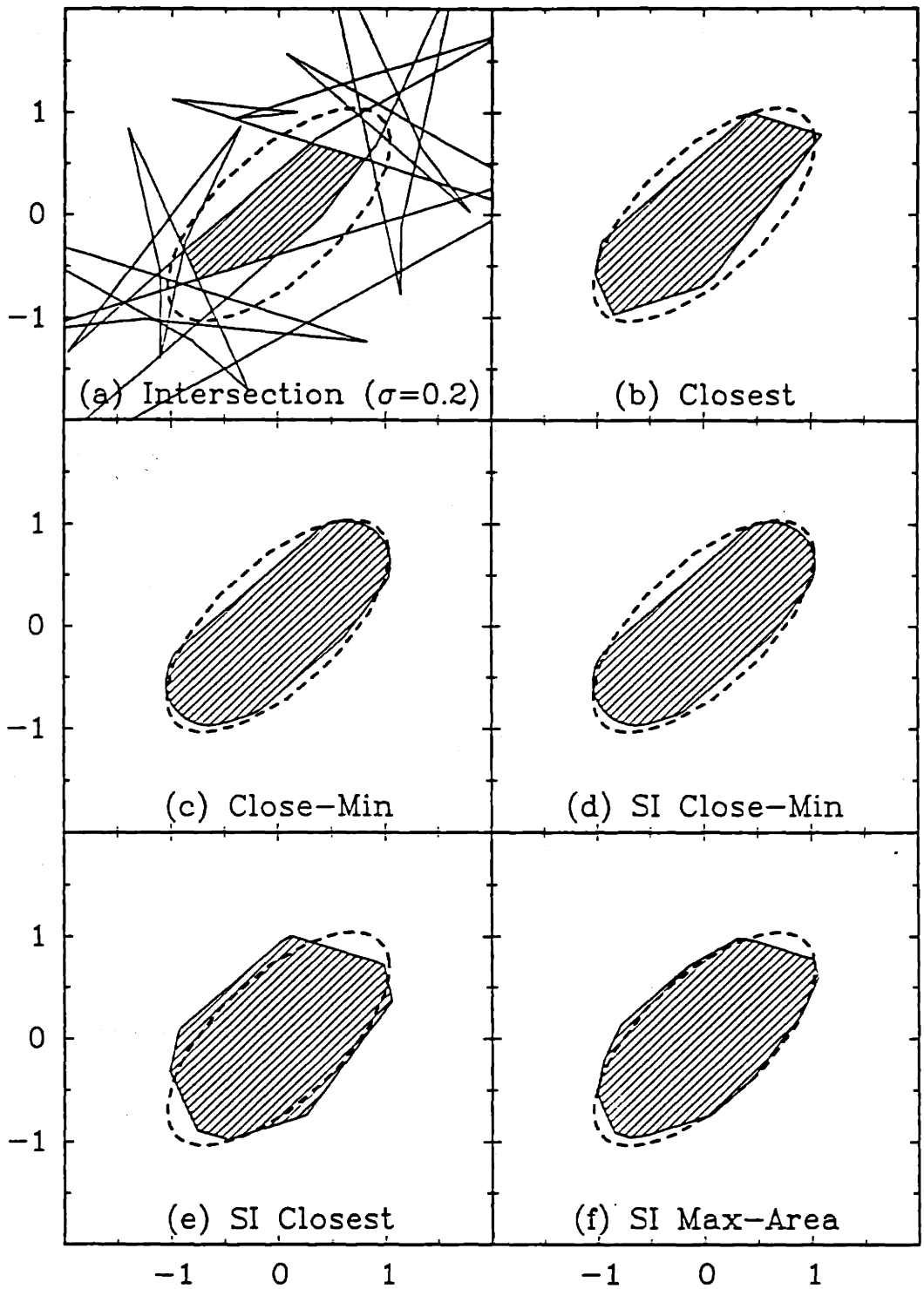


Figure 5.7: Comparison of support vector estimation algorithms for $M = 30$, $\sigma = 0.2$, and $\tau = 0.1$.

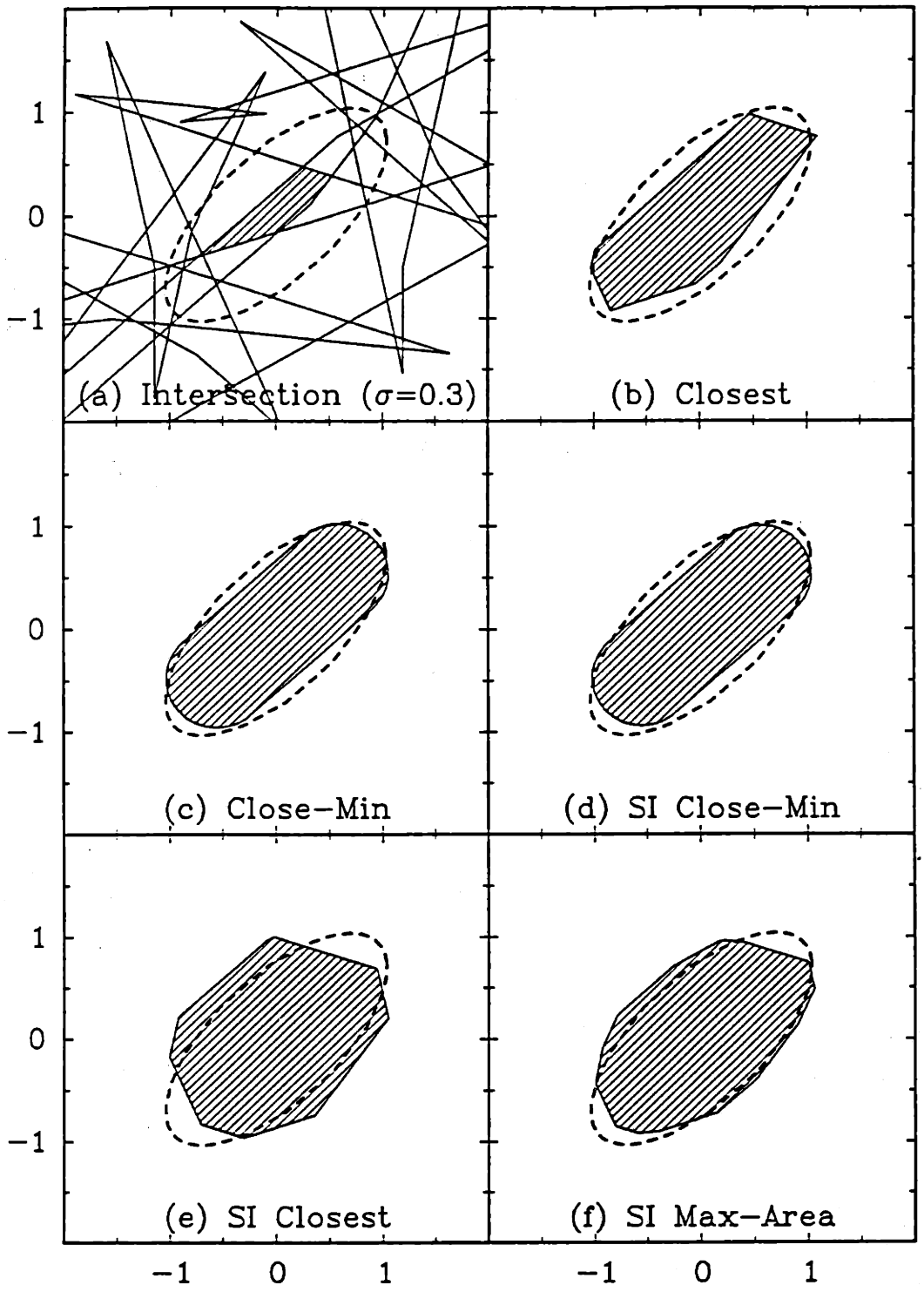


Figure 5.8: Comparison of support vector estimation algorithms for $M = 30$, $\sigma = 0.3$, and $\tau = 0.1$.

Sparse- and Limited-Angle Studies

We show in Figs. 5.9, 5.10, and 5.11, the behavior of, respectively, the SI Close-Min, SI Closest, and SI Max-Area algorithms for several sparse- and limited-angle cases. In this case, the true support vector has 60 elements ($M = 60$), and its basic object, shown using the dashed lines, circumscribes the same ellipse as in the previous two sections (eccentricity=0.9, orientation=45 degrees, centered at the origin). The noise standard deviation is $\sigma = 0.1$, and the vertex plot corresponding to the observations together with the estimate due to the intersection method is shown in the (a) panel of each figure. In the (b) panels of the three figures we show the reconstructions obtained by the three SI algorithms in the case where all the elements of the noisy support vector are observed. In (c), the algorithms used only every other element (including y_1), and in (d) the algorithms used only every third element (including y_1). Therefore, since $M = 60$ is even, and relating this situation to the computed tomography setting, these cases correspond to having observed (b) 30, (c) 15, and (d) 10 evenly spaced projections. The limited-angle studies are shown in panels (e) and (f). Panel (e) corresponds to situation in which we have available *projections* over 0–135 degrees; therefore elements 1–22 and 31–52 of the noisy support vector are observed. Panel (f) assumes that projections are available over only 0–90 degrees; hence, elements 1–15 and 31-45 are observed.

The sparse- and limited-angle studies shown in Figs. 5.9, 5.10, and 5.11 perform an *interpolation* using prior knowledge while simultaneously producing a feasible support vector and keeping the support values at observed angles close to the observations. The effect is that with fewer samples available (as in panels (d) and (f)) the result is an estimate which is more *circular* than desired. As in previous experiments, we may observe here that each algorithm has higher prior probability on objects that are circular in different ways. The most striking example is contained in panel (f) of each figure. Here, the support value observations which would ordinarily provide information about the narrow dimension of the true ellipse are missing. Remarkably, the SI Close-Min algorithm produces an estimate which is much better than either the SI Closest or the SI Max-Area. This is because the object which has two semicircles connected by two line segments has a *high* prior

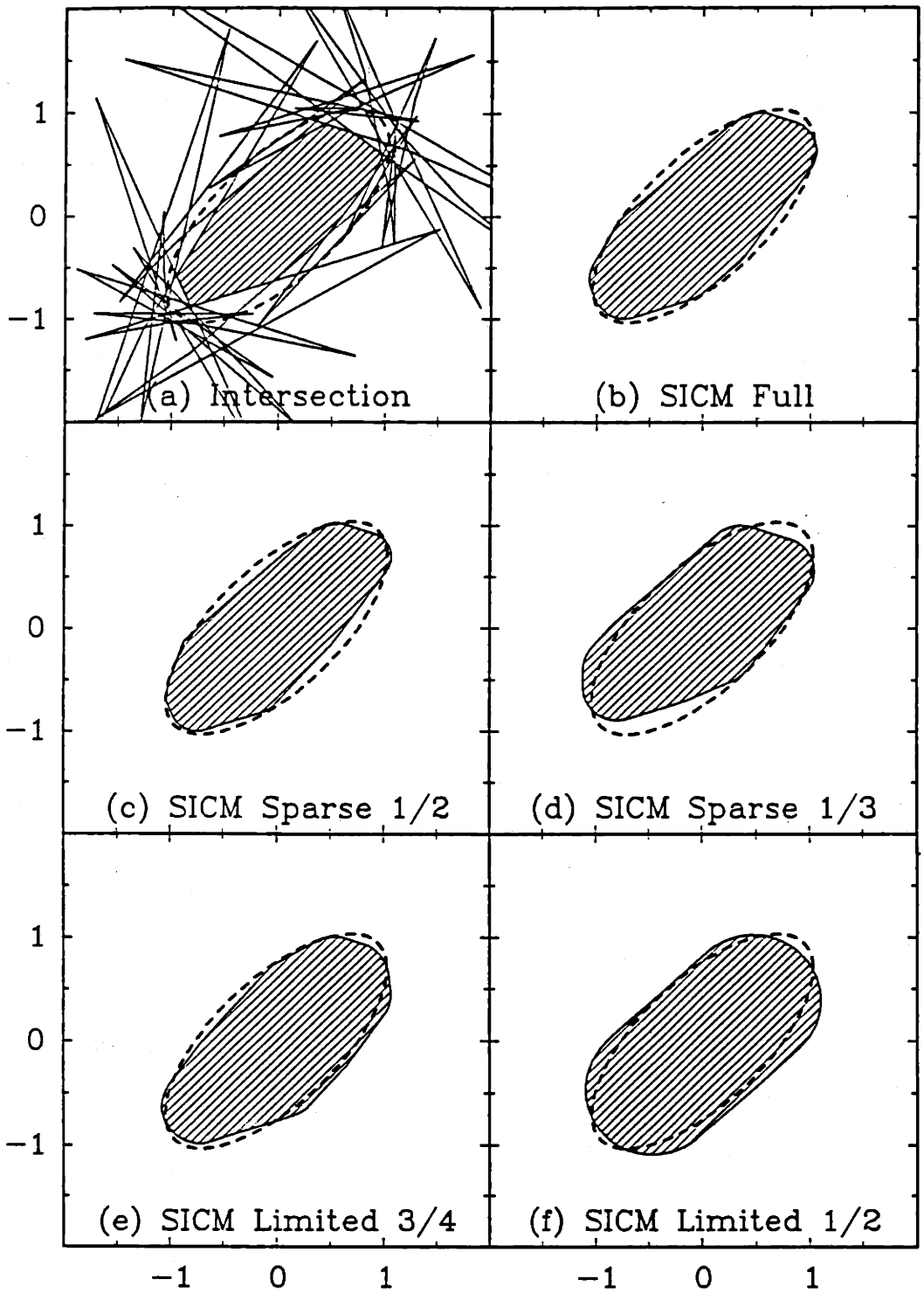


Figure 5.9: SI Close-Min algorithm for sparse- and limited-angle cases ($M = 60$, $\sigma = 0.1$, $\tau = 0.1$).

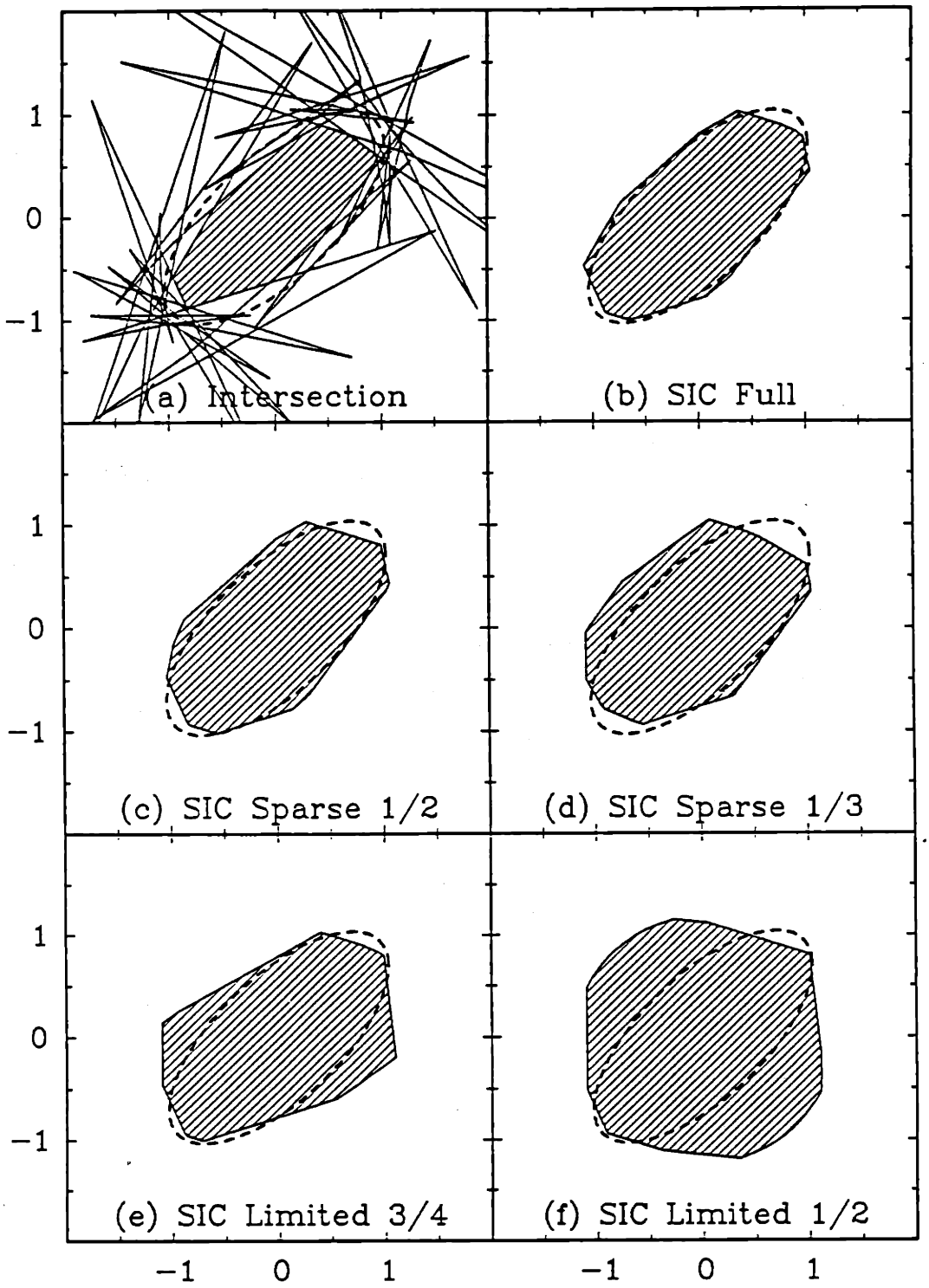


Figure 5.10: SI Closest algorithm for sparse- and limited-angle cases ($M = 60$, $\sigma = 0.1$, $\tau = 0.1$).

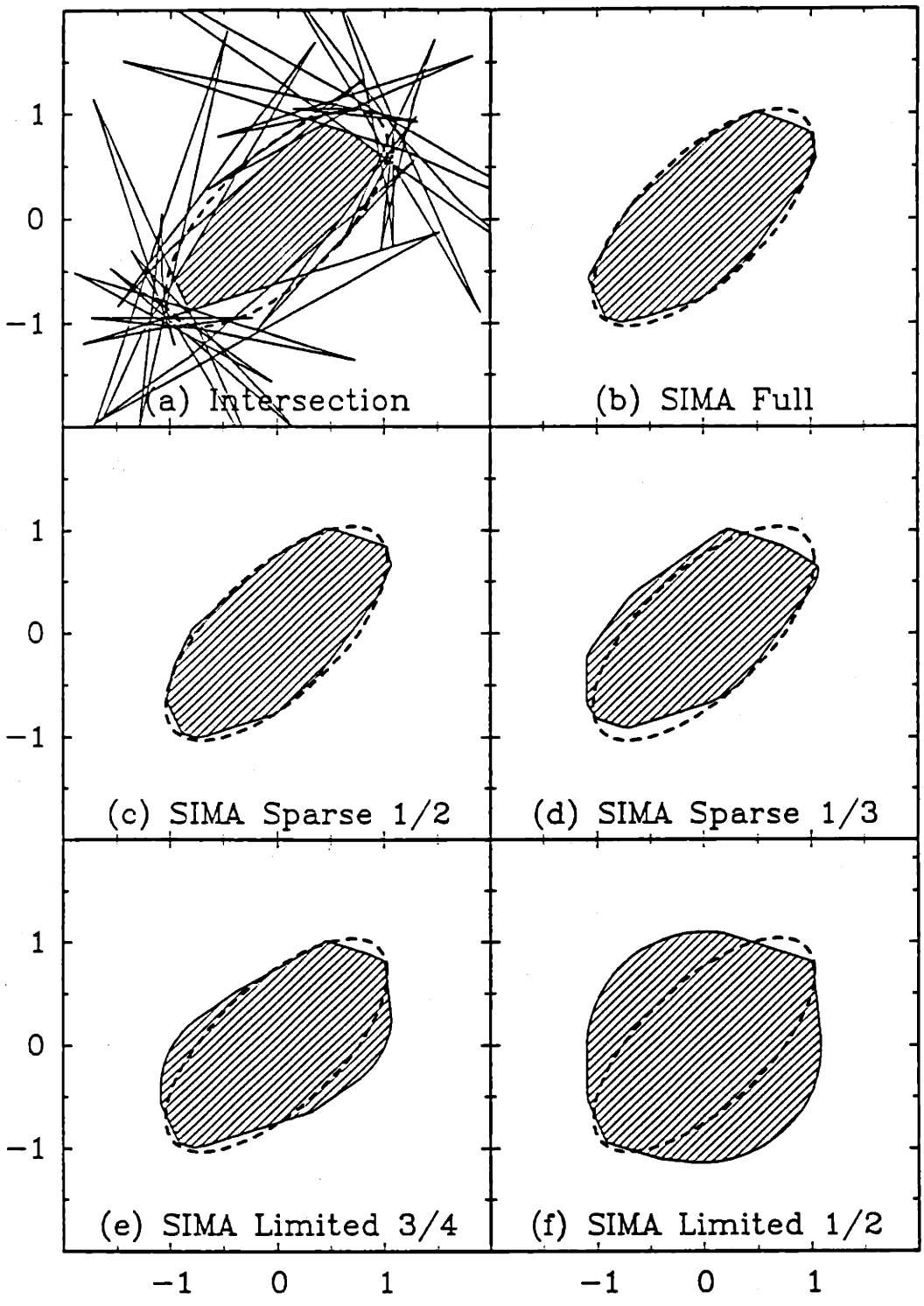


Figure 5.11: SI Max-Area algorithm for sparse- and limited-angle cases ($M = 60$, $\sigma = 0.1$, $\tau = 0.1$).

probability for the Close-Min (or SI Close-Min) prior, and in this case each of the semicircles matches the available measurements quite well. In contrast, this type of elongated object has a *low* prior probability for either the SIC or SIMA algorithm, and so it is an unfavorable estimate. We will see in the Ellipse-Based experiments in the next section that when SIC prior has its probability concentrated on the correct ellipse, rather than the circle, then very nice estimates may be obtained even in this limited-angle case.

5.5.2 Ellipse-Based Experiments

In this section we present results of experiments involving the Ellipse-Based (EB) algorithms of Section 5.4.

Closest Ellipse (CE)

Figs. 5.12, 5.13, and 5.14 summarize the results of our CE simulations. In Fig. 5.12 we show six different trials in which the support vector z (of dimension $M = 30$) has its basic object depicted by the cross-hatched region, and the estimated *closest ellipse* by the dashed line. In addition, we provide the estimated ellipse parameters in each panel above the objects, and the final value of the CE objective function f in the lower right portion of each panel. Actually, the different vectors z correspond to Closest estimates from different noisy observations ($\sigma = 0.2$) of an ellipse with parameters $v = (0,0)$, $t = 1.0$, $\varepsilon = 0.9$, and $\phi = 45.0°$. Therefore, this result also corresponds to an optimization procedure in which we start with a set of noisy observations of a support vector, find the Closest estimate, and from this support vector, determine the nearest ellipse. Fig. 5.13 shows the true ellipse using a dashed line, and the estimates from Fig. 5.12 by the crosshatched region. Finally, in Fig. 5.14 we show the same type of comparison as in Fig. 5.12, except that we estimated the ellipse parameters directly from the measurements rather than obtaining $\hat{h}_C$ as an intermediate support vector. Therefore, the ellipse parameters in this figure are the ML estimates, and in a certain sense, the cross-hatched ellipses in these panels are the ML estimates of the true ellipses depicted by the dashed lines

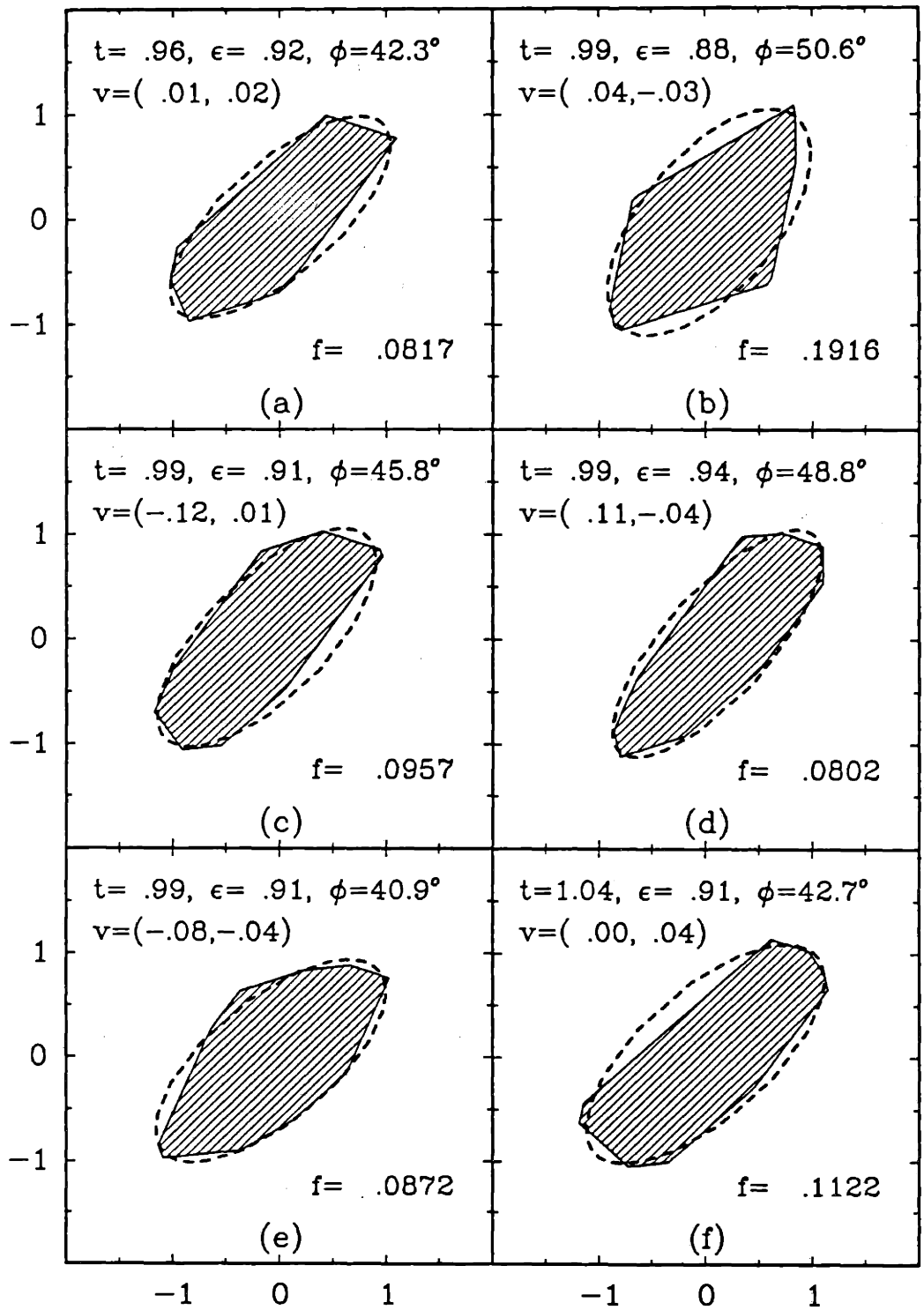


Figure 5.12: Results of six different trials of the Closest Ellipse algorithm.

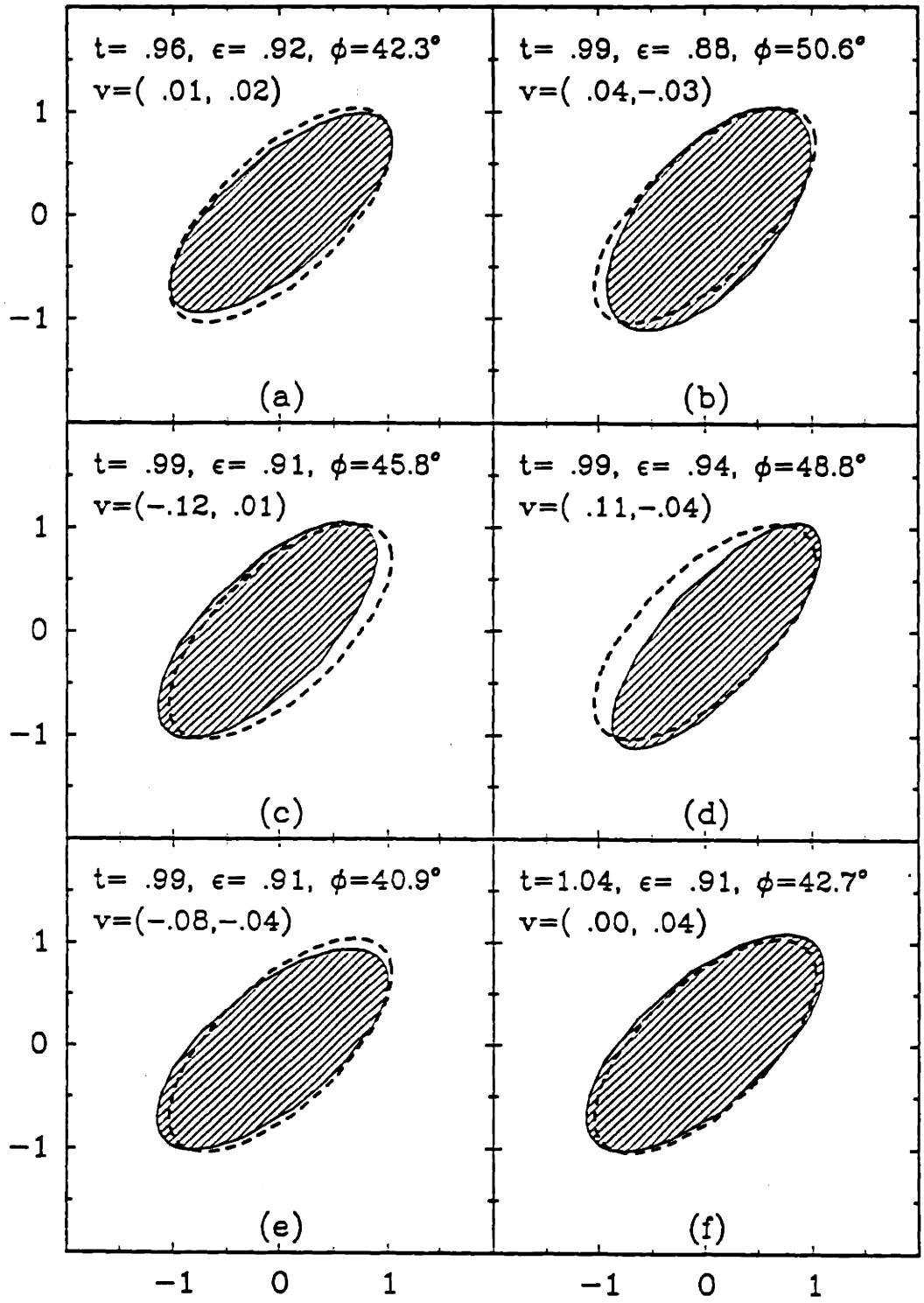


Figure 5.13: Results of ellipse estimation using the Closest algorithm followed the Closest Ellipse algorithm.

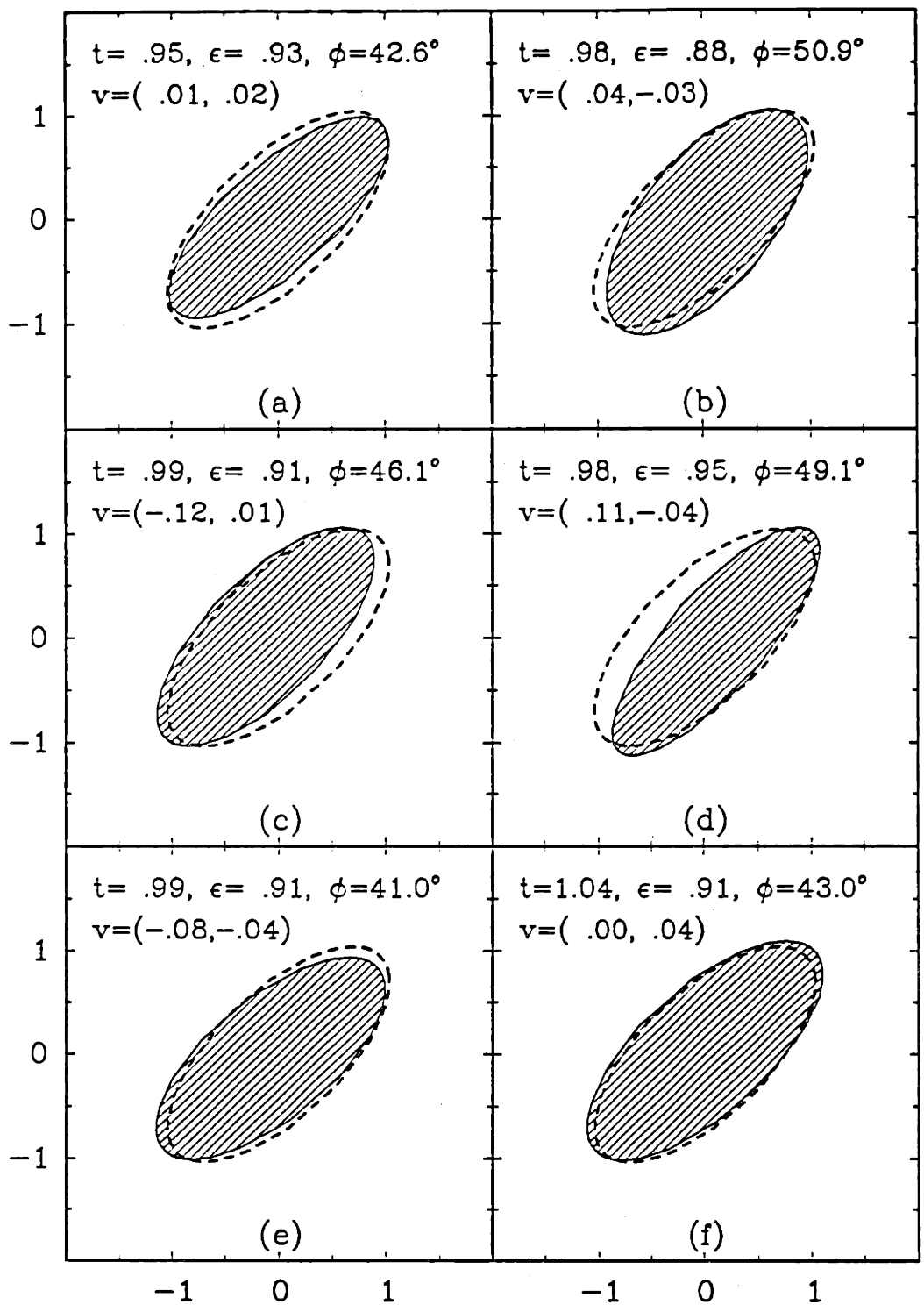


Figure 5.14: Ellipse estimation using the Closest algorithm to yield the ML estimates of the ellipse parameters.

in each of the corresponding panels.

We learn from Fig. 5.12 that the estimated ellipses match well what might be our expectation of the best ellipse to fit the cross-hatched object. Also, the value of the objective function for the optimal ellipse correlates to a visual impression of how good the fit appears to be. For example, the fit in panel (a) appears to be much better than the fit in panel (b), and indeed, f in panel (a) is less than half the value of f in panel (b). A comparison of Figs. 5.13 and 5.14 shows that there is not much difference — at least in this sample — between direct ML estimation of the ellipse parameters and the two-step procedure which first estimates the Closest support vector, and then finds the closest ellipse to that (intermediate) support vector. For example, in all panels except (a) and (b), the orientation estimate (indicated in the panels by ϕ) is closer to the true value of 45.0° in Fig. 5.14 than in Fig. 5.13. However, the estimated eccentricity is equal in four respective panels and is closer to the true value in two of the panels in Fig. 5.13. The two-step procedure also performs better in three of six panels and exactly the same in the other three.

Ellipse-Based SI Closest (ESIC)

Figs. 5.15 and 5.16 show two sets of experiments in which the ESIC algorithm of Section 5.4.3 is applied to the same sparse- and limited-angle cases presented in Section 5.5.1. Fig. 5.15 corresponds to the case in which $\tau = 0.1$, $\bar{\varepsilon} = 0.6$, and $\bar{\phi} = 45.0°$. The true ellipse has $\varepsilon = 0.9$ and $\phi = 45.0°$, so that only the eccentricity does not have the correct value. Fig. 5.16 uses the correct values: $\bar{\varepsilon} = 0.9$ and $\bar{\phi} = 45.0$ degrees. These results should be considered to be a continuation of the SIC result shown in Fig. 5.10 since the SIC algorithm is but a special case of the ESIC algorithm in which $\bar{\varepsilon} = 0.0$ and the value of $\bar{\phi}$ is irrelevant.

What we see in this sequence of three figures (Figs. 5.10, 5.15, and 5.16) is evidence that as the prior knowledge of the true object becomes more accurate, the reconstructions improve. In particular, where a lot of the data is missing — e.g., panels (d) and (f) in these figures — the improvement is quite noticeable, because it is in the ranges of the missing data where the prior knowledge is relied upon most heavily.

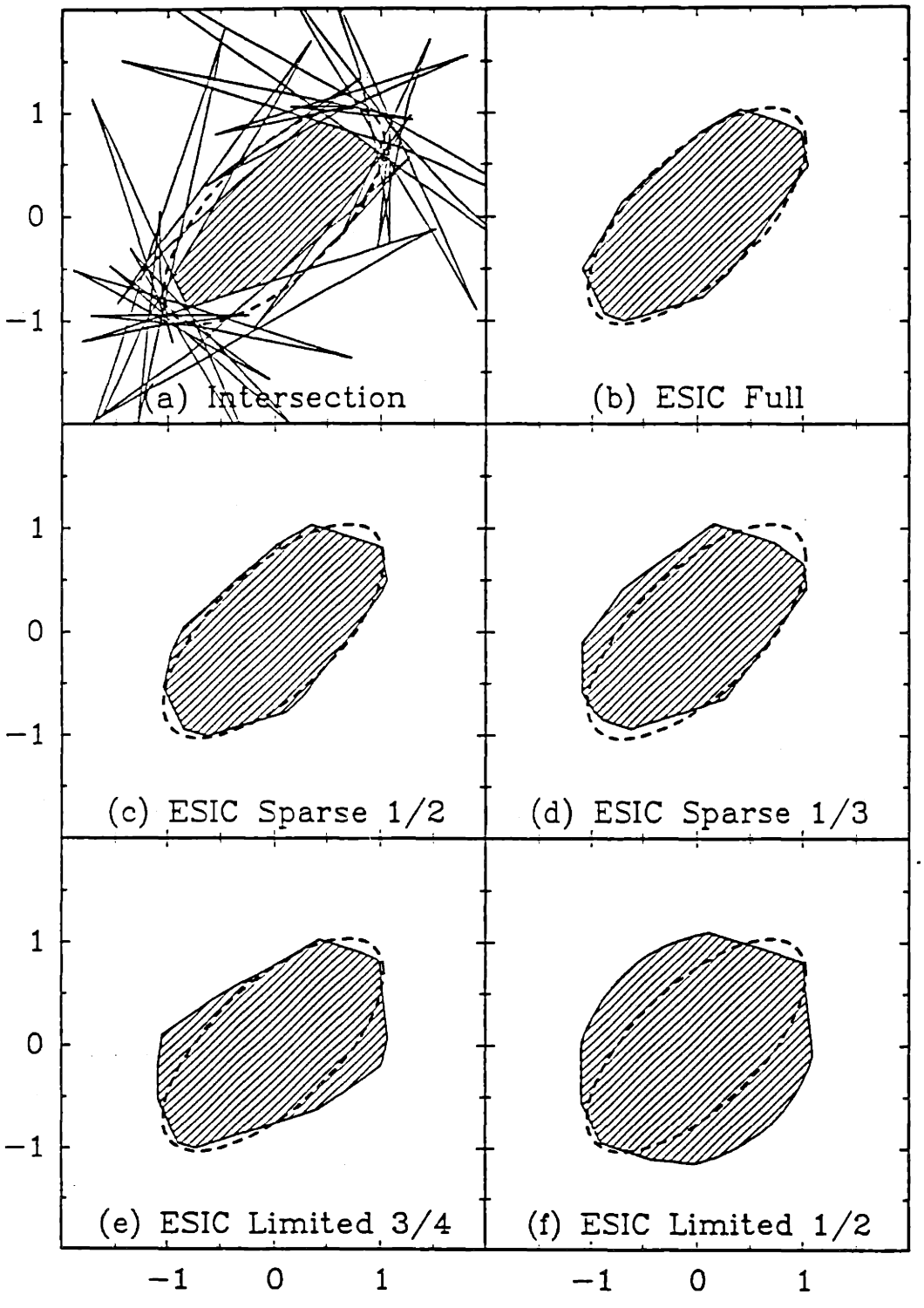


Figure 5.15: Sparse- and limited-angle examples using the ESIC algorithm ($M = 60$, $\sigma = 0.1$, $\tau = 0.1$, $\bar{\varepsilon} = 0.6$, and $\bar{\phi} = 45.0°$)

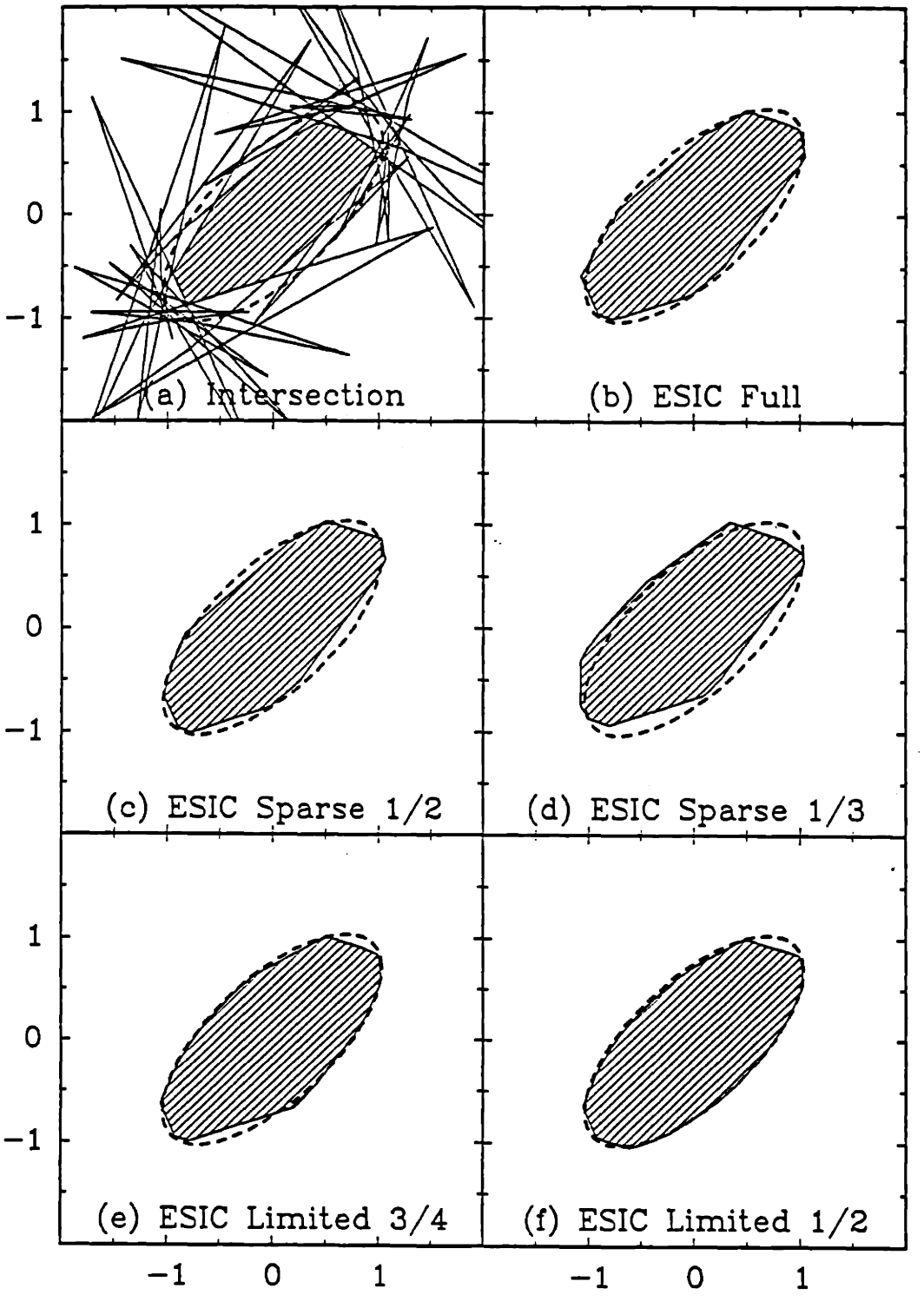


Figure 5.16: Sparse- and Limited-angle examples using the ESIC algorithm ($M = 60$, $\sigma = 0.1$, $r = 0.1$, $\bar{\varepsilon} = 0.9$, and $\bar{\phi} = 45.0°$)

Joint Ellipse/Support Vector (JE)

We applied the JE algorithm to the same noisy observations that led to the Closest estimates shown in Fig. 5.12 and that underly each of the ML estimates shown in Fig. 5.14. The results for $\alpha = 0.5$ are shown in Fig. 5.17. Here we depict six independent trials showing both the basic object of the estimated support vector h (the cross-hatched region) and the outline of the estimated ellipse (dashed curve). Also, the estimated ellipse parameters — which generated the dashed curve — are given in the top portion of each panel.

We observe several properties of the JE algorithm from these results. First, the estimated ellipse parameters indicated in Fig. 5.17 are nearly identical to the results of both the two-step procedure shown in Figs. 5.12 and 5.13 and the direct ML procedure shown in Fig. 5.14. This evidence leads us to suggest that estimating the ellipse parameters is a robust procedure, although we have only these few samples and no analytical confirmation. The estimated support vectors shown in the panels of Fig. 5.17 should be compared to the Closest estimates given in the respective panels of Fig. 5.12. We see here that the JE algorithm produces an estimate which is similar to the Closest estimate in overall form, but that the sharp portions of the boundary are smoother. This corresponds well to the original prior knowledge that the true object is likely to be elliptical in overall shape. Although we do not show this result, running the JE algorithm on the same data with $\alpha = 0.3$ yields support vector estimates which are even more elliptic in shape.

Since the JE algorithm attempts to estimate a support vector which is close to the shape of an ellipse, we would expect to show improved reconstructions for the limited- and sparse-angle cases shown in Figs. 5.9–5.11. These figures showed the results of the SI algorithms, all of which favor circles. The results of the JE algorithm for $\alpha = 0.5$, applied to the same noisy data and the same limited- and sparse-angle geometry, is shown in Fig. 5.18. As in the aforementioned series of figures, each panel shows the boundary of the true object using dashed lines and the estimate by the crosshatched region. In addition, since the JE algorithm estimates the ellipse parameters along with the support vector, we have indicated in the top portion of each panel of Fig. 5.18 the optimum estimates of the ellipse parameters.

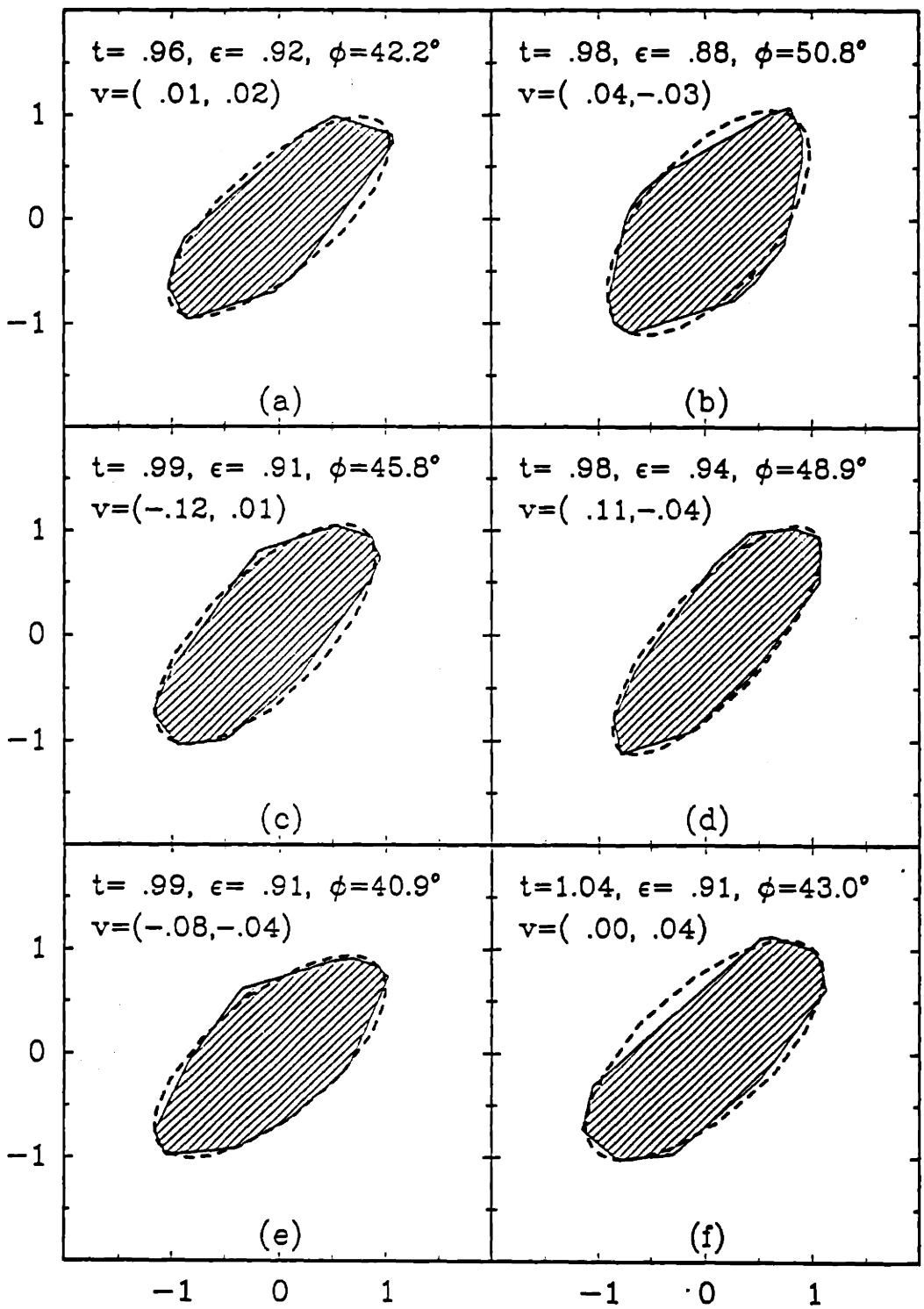


Figure 5.17: Results of the JE algorithm ($\alpha = 0.5$) using the same underlying noisy support vector observations (not shown) as in Figs. 5.12–5.14.

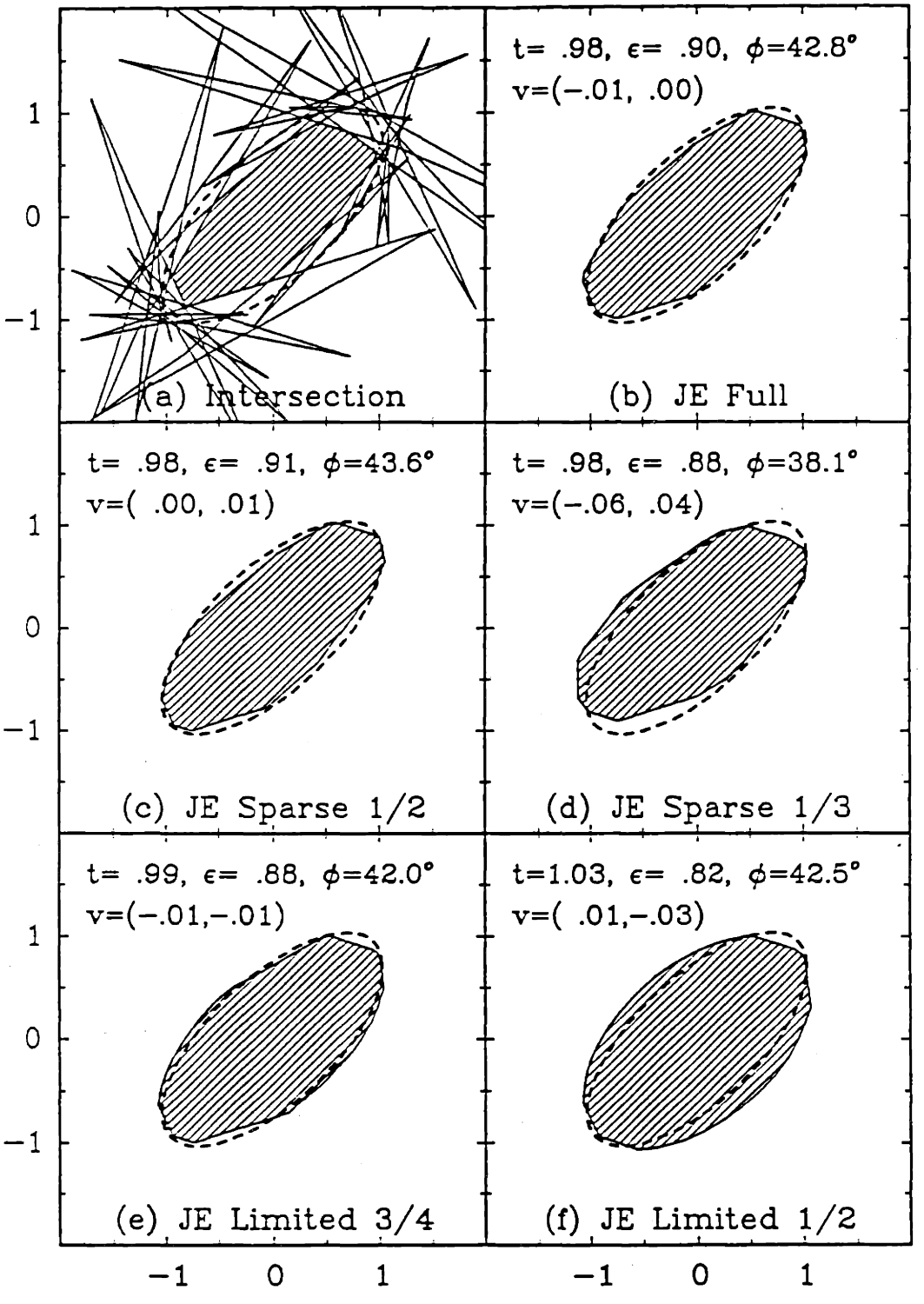


Figure 5.18: Results of the JE algorithm for various sparse- and limited-angle cases.

The performance of the JE algorithm does indeed improve over that of any of the SI algorithms, in these cases. In particular, the reconstructions in panels (d), (e), and (f) of Fig. 5.18 are much improved over the reconstructions in the respective panels of Figs. 5.9–5.11. It is also useful to compare these results with those of the ESIC algorithm as shown in Figs. 5.15 and 5.16. In this case, the overall performance of the JE algorithm lies somewhere in between the two ESIC results. However, it is important to remember that the ESIC algorithm required both the eccentricity and the orientation to be indicated *explicitly* to the algorithm, and the best results (Fig. 5.16) followed when these parameters happened to be exactly correct. In contrast, the JE algorithm used only the knowledge that the true object is likely to be nearly elliptic in shape, and any knowledge of the true ellipse parameters is not assumed *a priori*.

5.6 Discussion

In this chapter we have extended the results of Chapter 4 by developing support vector estimation methods that incorporate prior shape information using probabilistic methods. We began by characterizing the Close-Min algorithm to see how it may be interpreted precisely as an MAP estimator. This led us to discover the SSS decomposition, a decomposition of support vectors which allows us to relate certain aspects of the geometry of support vectors to that of object characteristics in 2-D. In particular, the size, shape, and shift (position) of objects are identified with the t, q, and h_n components of h.

This geometric description together with the overall probabilistic approach, allowed us to develop three Scale-Invariant algorithms by directly specifying their prior probabilities on the shape vectors q, and then using MAP methods to yield the form of the algorithm. Each of these priors uses the prior knowledge that the true objects tend to be circular, but the manner in which this knowledge is specified is different in each case. These new formulations require an added level of complexity to the specific method of solution because of the decomposition. The core of the algorithm is still a quadratic program, but there is an outer line search to find

the optimal size t. The addition of prior knowledge also allows us to solve sparse- and limited-angle problems, where the unobserved support values are automatically interpolated by the algorithm.

The final section of this chapter considers adding the additional knowledge that the true object has exactly or nearly the shape of an ellipse. We explore three different algorithms which have different degrees of knowledge on this subject. The first method assumes that the object is exactly an ellipse, and determines the ML estimates of the ellipse parameters. The second method assumes only that the object is nearly the shape of an ellipse for which we have prior knowledge of the eccentricity and orientation. The third algorithm assumes only that the true object is nearly elliptical, but we do not have any prior information about any of the ellipse parameters, so that the ellipse parameters and support vector are jointly estimated.

In Chapter 7 we show how the results of this chapter may be used to assist in the full computed tomography reconstruction problem with which this thesis is primarily concerned. In this general problem, the presence of prior knowledge is vital since we are considering problems in which there may be some projections which are missing. We will see that the presence of a good support estimate can significantly improve the reconstructions and that the algorithms developed in this chapter can indeed provide such estimates.

5.A The Circumference and Area of Basic Objects

In this appendix we derive expressions for the circumference and area of a basic object as a function of its support vector h.

5.A.1 The Circumference of a Basic Object

We now derive an expression for the circumference $P(h)$ of a basic object as a function of its support vector h. In particular, we show that the quantity $t = h^{\mathrm{T}}e/M$ is proportional to $P(h)$.

The circumference of a basic object is given by the sum of the face lengths f_i. Hence, referring to equations (4.19) and (4.20), and Fig. 4.8, we may make the following manipulations

$$P(h) = \sum_{i=1}^{M} f_i = \sum_{i=1}^{M} \frac{2\rho_i}{\tan\theta_0} = \frac{2}{\tan\theta_0}\rho^{\mathrm{T}}e \ . \tag{5.42}$$

where $e = [1\ 1\ \ldots\ 1]^{\mathrm{T}}$. But since $\rho \equiv -h^{\mathrm{T}}C$ we may simplify this expression even further as as follóws:

$$P(h) \ = \ \frac{-2}{\tan\theta_0}h^{\mathrm{T}}Ce = \frac{2}{\tan\theta_0}h^{\mathrm{T}}\gamma e = \frac{2\gamma}{\tan\theta_0}h^{\mathrm{T}}e\,,$$

where the second equality follows because the sum of the elements of any row of C is equal to γ (see (5.10)). We now easily see that

$$P(h) = \frac{2M\gamma}{\tan\theta_0}\frac{h^{\mathrm{T}}e}{M} = \frac{2M\gamma}{\tan\theta_0}t \tag{5.43}$$

which is the desired result.

5.A.2 The Area of a Basic Object

We now derive an exact expression for $S(h)$, the area of the basic object corresponding to support vector h. Without loss of generality let us assume that $h \in C_p$. We know this to be completely general since the formula for the area of any basic object

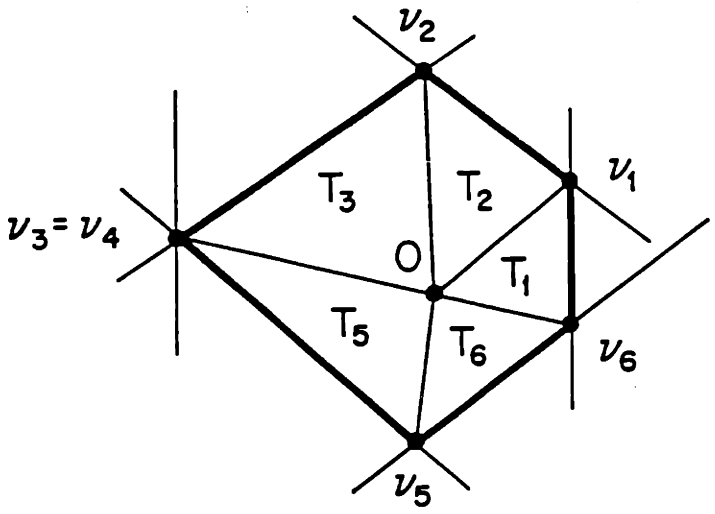


Figure 5.19: Triangulated basic object with $M = 6$.

must be independent of the nullspace component of a support vector. Now consider Fig. 5.19, which depicts a basic object for $M = 6$. The area of S_B is given by the sum of the areas of the M triangles given by $T_i = \text{hul}(O, \nu_{i-1}, \nu_i)$, $i = 1, \ldots, M$ where O stands for the origin. Hence, denoting the area of triangle T_i by A_i we have,

$$S(h) = \sum_{i=1}^{M} A_i \ .$$

Note that a triangle may be degenerate when $\nu_{i-1} = \nu_i$ (so that it becomes a line segment), and have zero area (as shown in Fig. 5.19 where $\nu_3 = \nu_4$ and therefore $A_4 = 0$). The area of a triangle is given by the usual formula — one half base times height — where the base may be taken to be the length f_i of the i^{th} face (see (4.19)), and the height is h_i, the i^{th} support value. Hence, we have

$$\begin{aligned} S(h) &= \sum_{i=1}^{M} \frac{1}{2} f_i h_i \\ &= \sum_{i=1}^{M} \frac{1}{\tan\theta_0} \rho_i h_i \end{aligned}$$

$$= \sum_{i=1}^{M} \frac{1}{\tan\theta_0} \left(-h^{\mathrm{T}} c_i\right) h_i$$
$$= \frac{-1}{\tan\theta_0} h^{\mathrm{T}} C h \tag{5.44}$$

where the second equality results from substitution of (4.19) and the third follows from the definition of ρ_i. It is easy to show that $S(h)$ remains unchanged when a nullvector is added to h so that this formula is valid in the general case.

In developing the formula for $S(h)$ given above we have made three implicit assumptions which must be proven. These are:

1. The origin is contained in the basic object to a proper support vector.

2. If $h \in C_p$ then $h \geq 0$.

3. The triangular regions T_i with non-zero area do not overlap.

It is apparent that with these facts in hand, the method given above for finding the area to S_B is correct. We now show that these facts hold using the following propositions. First, we show that the origin is contained in the basic object to a proper support vector.

Proposition 5.2 *Let $h \in \mathbb{R}^M$ be a proper support vector with basic object S_B. Then the origin is an element of S_B.*

Proof *Suppose the origin is* not *in S_B. Then, since S_B is convex, there exists a vector $w \in \mathbb{R}^2$ such that $w^{\mathrm{T}} u < 0$ for all points u in S_B.*[8] *Now, by assumption $h \in C_p$, so we must have that $v = \sum_{i=1}^{M} \nu_i = 0$, where $\{\nu_i\}$ are the vertex points of S_B. Since $\nu_i \in S_B$, for all i, then $w^{\mathrm{T}} \nu_i < 0$ and hence, $\sum_{i=1}^{M} w^{\mathrm{T}} \nu_i < 0$, which is a contradiction.* □

We also need the fact that $h \geq 0$ when $h \in C_p$.

[8] In other words, there exists a line which separates O from S_B, a fact which follows from the *separating hyperplane theorem* (see [2] Thm. 2.3.4).

Proposition 5.3 *Let $h \in \mathbb{R}^M$ be a support vector and S_B its basic object. Then the origin is in S_B if and only if $h \geq 0$.*

Proof *An element u of S_B satisfies $u^T[\omega_1 \ \ldots \ \omega_M] \leq [h_1 \ \ldots \ h_M]$. Therefore, if $u = 0$ then $h \geq 0$. Furthermore, if $h \geq 0$ then $u = 0$ satisfies the condition to be in S_B.* □

Finally, we must show that the triangular regions with non-zero area used to calculate the area of S_B don't overlap. This follows from the following proposition.

Proposition 5.4 *Let $h \in \mathbb{R}^M$ be a support vector whose basic object S_B contains the origin, and let $\{\nu_i, i = 1, \ldots, M\}$ be the vertex points of S_B. Then the determinant of the matrix $[\nu_i \ \nu_{i+1}]$ for each $i = 1, \ldots, M$ is non-negative.*

Proof *The proof is largely a matter of algebra. From Chapter 4 we have*

$$\begin{aligned} \nu_i^T &= \frac{1}{\sin\theta_0}[h_{i-1}h_i]\begin{bmatrix} \sin\theta_0 & -\cos\theta_0 \\ -\sin\theta_{i-1} & \cos\theta_{i-1} \end{bmatrix} \\ &= \frac{1}{\sin\theta_0}\begin{bmatrix} h_{i-1}\sin\theta_i - h_i\sin\theta_{i-1} & -h_{i-1}\cos\theta_i + h_i\cos\theta_{i-1} \end{bmatrix} \end{aligned}$$

so that we may form the required determinant as

$$\begin{vmatrix} \nu_i & \nu_{i+1} \end{vmatrix} = \frac{1}{\sin^2\theta_0}\begin{vmatrix} h_{i-1}\sin\theta_i - h_i\sin\theta_{i-1} & h_i\sin\theta_{i+1} - h_{i+1}\sin\theta_i \\ -h_{i-1}\cos\theta_i + h_i\cos\theta_{i-1} & -h_i\cos\theta_{i+1} + h_{i+1}\cos\theta_i \end{vmatrix}$$

which simplifies further to

$$\begin{aligned} \begin{vmatrix} \nu_i & \nu_{i+1} \end{vmatrix} &= \frac{1}{\sin^2\theta_0}(h_{i-1}h_i\sin\theta_0 - h_ih_i\sin 2\theta_0 + h_ih_{i+1}\sin\theta_0) \\ &= \frac{1}{\sin^2\theta_0}h_i\sin\theta_0\,(h_{i-1} - 2h_i\cos\theta_0 + h_{i+1}) \\ &= \frac{1}{\sin^2\theta_0}2h_i\sin\theta_0\cos\theta_0\,(h_{i-1}/2\cos\theta_0 - h_i + h_{i+1}/2\cos\theta_0) \ . \end{aligned}$$

We may now see that each of the terms on the right-hand side of the above expression is non-negative. First, we know by a previous proposition that $h_i \geq 0 \ \forall i$, since S_B contains the origin. Furthermore, since $M \geq 5$, then $0 < \theta_0 < \pi/2$, hence, $\sin\theta_0 \geq 0$ and $\cos\theta_0 \geq 0$. Finally, since h is a support vector, the expression in the parentheses is non-negative since it is just the negative of $c_i^T h$, where c_i is a column of C. □

To see why this proposition shows the non-overlapping property of the triangles T_i used in the development of the area of S_B, we consider the meaning of the determinant of $[\nu_i \ \nu_{i+1}]$. If this determinant is zero then we know that the vectors ν_i and ν_{i+1} must be linearly dependent — hence the area of the "triangle" $T_i = \text{hul}(O, \nu_i, \nu_{i+1})$ is automatically zero, and is not important in the sum. However, if this matrix has a positive determinant — which is the only other possibility according to this proposition — then the ordered pair (ν_i, ν_{i+1}) is right-handed (cf. [82]) which implies that, in polar coordinates, ν_{i+1} has a larger angle than that of ν_i. Hence, the triangles do not overlap.

Note that this last proposition also provides another way to find an expression for the area of S_B. Since the area of T_i is just $\frac{1}{2}|\nu_i \ \nu_{i+1}|$ (cf. [82]), we must have

$$S(h) = \sum_{i=1}^{M} \frac{1}{2} \left| \begin{array}{cc} \nu_i & \nu_{i+1} \end{array} \right| .$$

and indeed, it is possible to show — after quite a bit of algebra — that the above expression is equivalent to that given in (5.44).

5.B Sparse Scale-Invariant Algorithms

In this appendix we extend the results of Section 5.3 to the case where we have fewer than M (noisy) observations of a support vector but we desire an estimate of the full M-dimensional vector. In this case, the ML solution is not unique, and we must use a formulation that includes prior knowledge such as the Scale-Invariant MAP formulations of Section 5.3. As we shall see, the main complication in this case is that the optimal shift vector cannot be solved independently of the size and shape components — but this problem is not too difficult to overcome. The resultant algorithms retain the line search over t and the embedded QP parts, as in the original SI algorithms, but they require pre- and post-processing to account for the shift estimate. We will refer to the algorithms developed in this appendix as the *Sparse Scale-Invariant* (SSI) algorithms.

We may write the observation equation when there are missing or sparse obser-

vations as

$$\tilde{y} = Sh + n \tag{5.45}$$

where h is the true M-dimensional support vector, $\tilde{y}$ is a K-dimensional observation vector (where $K < M$), n is a zero-mean jointly Gaussian vector with covariance $\sigma^2 I$, and S is a $K \times M$ matrix which "selects" the elements of h in the following way. Suppose the first K elements of h are observed; then we have

$$S = [I_k \;\; 0] \tag{5.46}$$

where I_k denotes the $K \times K$ identity matrix. Alternate elements of h may be selected by permuting the columns of S. Therefore, we assume in what follows that S is a matrix obtained by permuting the columns of the above matrix.

The log likelihood of h for the above observation model is given by

$$l(h) = -\frac{1}{2\sigma^2}(\tilde{y} - Sh)^{\mathrm{T}}(\tilde{y} - Sh) - \frac{1}{2}\ln\left|2\pi\sigma^2 I_k\right| \tag{5.47}$$

which may be written as

$$l(h) = \frac{1}{2\sigma^2}(y - h)^{\mathrm{T}} D(y - h) - \frac{1}{2}\ln\left|2\pi\sigma^2 I_k\right| \tag{5.48}$$

where

$$D = S^{\mathrm{T}}S, \text{ and} \tag{5.49}$$

$$y = S^{\mathrm{T}}\tilde{y}. \tag{5.50}$$

We may now form expressions for the Sparse Scale-Invariant (SSI) algorithms by adding the natural logarithm of the prior probability to $l(h)$. Since the SI priors only depend on the shape vector q we may write a *generic* SSI problem as

$$\underset{h \in C}{\text{maximize}} \quad -\frac{1}{2\sigma^2}(y - h)^{\mathrm{T}} D(y - h) + f(q) \tag{5.51}$$

where we have dropped the constant term $-\frac{1}{2}\ln|2\pi\sigma^2 I_k|$ and where $f(q)$ stands for the natural logarithm of either $p_{SICM}(h)$, $p_{SIC}(h)$, or $p_{SIMA}(h)$.

To solve (5.51) we must expand h using the SSS decomposition in order that $f(q)$ may be combined with the $l(h)$ part. Here is where the essential difference

between the sparse case and the full-view case appears. In the sparse case we cannot completely separate the solution of h_n from h_p since they are now coupled through cross terms of the form $h_p^T D h_n$. However, in what follows we show that the optimum shift vector may be calculated directly from the optimum size and shape vector, and we may use this knowledge to simplify the form of the SSI optimization problem.

To see how to find the optimum shift vector component, we focus on the expansion of $(y-h)^T D(y-h)$ using the SSS decomposition. First, we use the fact that $h = h_p + h_n$ and that $h_n = Nv$ for some v to make the following manipulations

$$\begin{aligned}
&(y-h)^T D(y-h) \\
&\quad = y^T Dy - 2y^T Dh + h^T Dh \\
&\quad = y^T Dy - 2y^T Dh_p - 2y^T DNv + h_p^T Dh_p + 2h_p^T DNv + v^T N^T DNv\,.
\end{aligned}$$

Now, any SI objective function has a prior which does not depend on v, therefore we may determine the necessary conditions for v to be a minimum by taking the (vector) partial derivative of (5.52) with respect to v and setting it equal to zero. We get

$$-2N^T D^T y + 2N^T D^T h_p + 2N^T DNv^* = 0$$

or

$$v^* = (N^T DN)^{-1} N^T D(y - h_p)\,. \tag{5.52}$$

Since for any choice of h_p, (5.52) yields the optimum v, we may substitute this expression back into (5.52) and simplify. After some algebra we find

$$(y-h)^T D(y-h)\Big|_{v=v^*} = y^T(D-Q)y - 2y^T(D-Q)h_p + h_p^T(D-Q)h_p \tag{5.53}$$

where

$$Q = DN(N^T DN)^{-1} N^T D \tag{5.54}$$

Making the substitution $h_p = tq$ and adding the natural logarithm of the appropriate prior we obtain the following SSI formulations:

SICM:

$$\underset{t,q}{\text{minimize}} \ \frac{t^2}{2\sigma^2} q^{\mathrm{T}}(D-Q)q - \frac{t}{\sigma^2} y^{\mathrm{T}}(D-Q)q \tag{5.55}$$

$$-\frac{2}{\bar{\tau}\theta_0 \tan\theta_0} \min\{-q^{\mathrm{T}}c_1, \ldots, -q^{\mathrm{T}}c_M\} \tag{5.56}$$

SIC:

$$\underset{t,q}{\text{minimize}} \ \frac{t^2}{2\sigma^2} q^{\mathrm{T}}(D-Q)q - \frac{t}{\sigma^2} y^{\mathrm{T}}(D-Q)q + \frac{1}{\tau} q^{\mathrm{T}}q \tag{5.57}$$

SIMA:

$$\underset{t,q}{\text{minimize}} \ \frac{t^2}{2\sigma^2} q^{\mathrm{T}}(D-Q)q - \frac{t}{\sigma^2} y^{\mathrm{T}}(D-Q)q + \frac{1}{\tau \tan\theta_0} q^{\mathrm{T}}Cq \tag{5.58}$$

Each of the three optimization problems may be solved using a line search approach similar to the SI algorithms of Section 5.3. We may see this by noting that in each case the optimization over q given a fixed t is just a QP. Therefore, once the optimum size $\hat{t}$ is found by searching the non-negative real line, the optimum shape vector $\hat{q}$ is known (because it is part of the internal steps needed to calculate $F(t)$, the objective function which $\hat{t}$ minimizes). An additional step is required for the SSI algorithms however: we must calculate the optimal shift vector $\hat{h}_n$ using $\hat{t}$ and $\hat{q}$. The *generic* SSI algorithm is summarized below.

Algorithm 5.6 (Generic Sparse Scale-Invariant)

1. *Define the function of t to be minimized:*

$$F(t) = \frac{1}{2\sigma^2} q_*^{\mathrm{T}}(D-Q)q_* - \frac{t}{\sigma^2} y^{\mathrm{T}}(D-Q)q_* - f(q_*)$$

where D and Q are given in (5.49) and (5.54) respectfully, $f(q)$ is the natural logarithm of the SI particular prior, and q_* is the solution of

$$\begin{array}{ll} \underset{q}{\text{minimize}} & \frac{1}{2\sigma^2} q^{\mathrm{T}}(D-Q)q - \frac{t}{\sigma^2} y^{\mathrm{T}}(D-Q)q - f(q) \\ \text{subject to} & q^{\mathrm{T}}C \leq 0 \text{ and} \\ & q^{\mathrm{T}}e = M\,. \end{array}$$

for fixed t.

2. *Bracket a minimum — that is, find t_a, t_b, and t_c so that $t_a < t_b < t_c$ and $F(t_a) > F(t_b)$ and $F(t_c) > F(t_b)$ — using the golden section method [68].*

3. *Refine the minimum to within a prespecified tolerance using quadratic interpolation and successive iterations via Brent's method [6,68].*

4. *Given the optimum size $\hat{t}$ and the optimum shape vector for that size $\hat{q} = q_*$, calculate the optimum shift vector*

$$\hat{h}_n = Nv^*$$

where v^* is given by (5.52) for $h_p = \hat{t}\hat{q}$.

5. *The SSI support estimate is given by*

$$\hat{h} = \hat{t}\hat{q} + \hat{h}_n$$

5.C Derivation of the Support Function of an Ellipse

Points (x, y) on the boundary of the ellipse shown in Figure 5.2 satisfy the equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{5.59}$$

where a and b are the x and y semiaxes of the ellipse, respectively. Since an ellipse is convex and has a continuously turning boundary normal, we know that each

support line intersects the boundary at precisely one point — e.g., the point P in the figure. Therefore, P is on the line whose points satisfy

$$x\cos\theta + y\sin\theta = h \tag{5.60}$$

where θ is the angle (measured counterclockwise from the x-axis) of the unit outward normal to the ellipse boundary at the point P and h is the shortest distance from the origin to the tangent line at P. We recognize h to be the support distance distance at angle θ. We seek in this appendix an expression for h as a function of θ.

The simplest approach begins by solving for the (x, y) pair that satisfies both (5.59) and (5.60). We have from (5.60) that

$$x^2 = \left(\frac{h - y\sin\theta}{\cos\theta}\right)^2$$

which when substituted into (5.59) yields (after some manipulation)

$$\underbrace{\left(\frac{\sin^2\theta}{a^2\cos^2\theta} + \frac{1}{b^2}\right)}_{p} y^2 + \underbrace{\left(\frac{-2h\sin\theta}{a^2\cos^2\theta}\right)}_{q} y + \underbrace{\left(\frac{h^2}{a^2\cos^2\theta} - 1\right)}_{r} = 0\,. \tag{5.61}$$

The solution to this equation is given by

$$y = \frac{-q \pm \sqrt{q^2 - 4pr}}{2p}\,, \tag{5.62}$$

but we do not need to solve this to accomplish the goals of this section. Instead we observe that there are three possible outcomes: 1) y has two complex (i.e., real part plus imaginary part) solutions, 2) y has two real solutions, and 3) y has a single real solution. The first possibility arises when the line $L(h, \theta)$ does not intersect the ellipse, the second arises when the line intersects the ellipse at two points, and the third possibility occurs when the line intersects the ellipse at just one point. Only the third possibility causes the line to be a support line to the ellipse — this is the situation we require.

To assure that the line $L(h, \theta)$ intersects the ellipse at just one point we must have that (refer to equation (5.62))

$$q^2 - 4pr = 0\,, \tag{5.63}$$

which when making the required substitutions from (5.61) becomes

$$\frac{4h^2\sin^2\theta}{a^4\cos^4\theta} - 4\left(\frac{\sin^2\theta}{a^2\cos^2\theta} - \frac{1}{b}\right)\left(\frac{h^2}{a^2\cos^2\theta} - 1\right) = 0\,.$$

After simplification, the above expression yields

$$h^2 = a^2\cos^2\theta + b^2\sin^2\theta\,,$$

from which we get

$$h(\theta) = \sqrt{a^2\cos^2\theta + b^2\sin^2\theta}\,, \tag{5.64}$$

the support function to the ellipse of Fig. 5.2.

The most general expression for the support function to an ellipse must include orientation and position as well as the lengths of the two semiaxes. We may produce any ellipse by rotating the ellipse of Fig. 5.2 in the counterclockwise direction by ϕ radians, and then shifting the resulting figure so that it is centered at the point $v \in \mathbb{R}^2$. The general expression for the support vector is then given by

$$h(\theta) = \sqrt{a^2\cos^2(\theta-\phi) + b^2\sin^2(\theta-\phi)} + [\cos\theta \;\; \sin\theta]v\,, \tag{5.65}$$

where we have used the continuous analog of the shift theorem to produce the last term in the equation. The support *vector* to an ellipse is found by sampling $h(\theta)$ at the support angles $\{\theta_i\}$ and arranging the samples in the required vector form.

5.D Gradient Calculations for the CE Algorithm

To solve the Closest Ellipse (CE) estimation problem of (5.39) we require the gradient of the objective function $f(t,\varepsilon,\phi) = \|z - h(v_z,t,\varepsilon,\phi)\|^2$; hence, we require the partial derivatives of f with respect to t, ε, and ϕ. First, we write $f(t,\varepsilon,\phi) = \sum_{i=1}^{M}(z_i - h_i(v_z,t,\varepsilon,\phi))^2$, and using the chain rule for derivatives, the required partials of f may be formally written as

$$\frac{\partial}{\partial t}f(t,\varepsilon,\phi) \;=\; \sum_{i=1}^{M} 2\,(z_i - h_i(v_z,t,\varepsilon,\phi))\left(-\frac{\partial}{\partial t}h_i(v_z,t,\varepsilon,\phi)\right)\,,$$

$$\begin{aligned}
\frac{\partial}{\partial \varepsilon} f(t,\varepsilon,\phi) &= \sum_{i=1}^{M} 2\left(z_i - h_i(v_z,t,\varepsilon,\phi)\right)\left(-\frac{\partial}{\partial \varepsilon} h_i(v_z,t,\varepsilon,\phi)\right), \\
\frac{\partial}{\partial \phi} f(t,\varepsilon,\phi) &= \sum_{i=1}^{M} 2\left(z_i - h_i(v_z,t,\varepsilon,\phi)\right)\left(-\frac{\partial}{\partial \phi} h_i(v_z,t,\varepsilon,\phi)\right).
\end{aligned}$$

From (5.37) we see that $h_i(v_z,t,\varepsilon,\phi)$ may be written as

$$h_i = \frac{Mtp_i(\varepsilon,\phi)}{\sum_{j=1}^{M} p_j(\varepsilon,\phi)} + Nv_z \tag{5.66}$$

where

$$p_i(\varepsilon,\phi) = \sqrt{\frac{1}{1-\varepsilon^2}\cos^2(\theta_i - \phi) + \sin^2(\theta_i - \phi)}\,. \tag{5.67}$$

The partials of h_i with respect to t, ε, and ϕ are found as follows

$$\begin{aligned}
\frac{\partial}{\partial t} h_i(v_z,t,\varepsilon,\phi) &= \frac{Mp_i(\varepsilon,\phi)}{\sum_{j=1}^{M} p_j(\varepsilon,\phi)}, \\
\frac{\partial}{\partial \varepsilon} h_i(v_z,t,\varepsilon,\phi) &= \frac{\partial}{\partial \varepsilon}\frac{Mtp_i(\varepsilon,\phi)}{\sum_{j=1}^{M} p_j(\varepsilon,\phi)} \\
&= \frac{\left(\sum_{j=1}^{M} p_j(\varepsilon,\phi)\right) Mt\frac{\partial}{\partial \varepsilon} p_i(\varepsilon,\phi) - Mtp_i(\varepsilon,\phi)\left(\sum_{j=1}^{M}\frac{\partial}{\partial \varepsilon} p_j(\varepsilon,\phi)\right)}{\left(\sum_{j=1}^{M} p_j(\varepsilon,\phi)\right)^2}, \\
\frac{\partial}{\partial \phi} h_i(v_z,t,\varepsilon,\phi) &= \frac{\partial}{\partial \phi}\frac{Mtp_i(\varepsilon,\phi)}{\sum_{j=1}^{M} p_j(\varepsilon,\phi)} \\
&= \frac{\left(\sum_{j=1}^{M} p_j(\varepsilon,\phi)\right) Mt\frac{\partial}{\partial \phi} p_i(\varepsilon,\phi) - Mtp_i(\varepsilon,\phi)\left(\sum_{j=1}^{M}\frac{\partial}{\partial \phi} p_j(\varepsilon,\phi)\right)}{\left(\sum_{j=1}^{M} p_j(\varepsilon,\phi)\right)^2}.
\end{aligned}$$

The expression for $\partial h_i/\partial t$ requires no further work, however, to simplify the latter two equations we require the partial derivatives of $p_i(\varepsilon,\phi)$. After some work we may determine the following two expressions:

$$\begin{aligned}
\frac{\partial}{\partial \varepsilon} p_i(\varepsilon,\phi) &= \frac{\varepsilon\cos^2(\theta_i - \phi)}{(1-\varepsilon^2)^2 p_i(\varepsilon,\phi)}, \\
\frac{\partial}{\partial \phi} p_i(\varepsilon,\phi) &= \frac{\varepsilon^2 \sin 2(\theta_i - \phi)}{2(1-\varepsilon^2) p_i(\varepsilon,\phi)}.
\end{aligned}$$

Finally, putting all the terms together and making some simple cancellations yields

$$\frac{\partial}{\partial t} f(t,\varepsilon,\phi) = \frac{-2M}{\sum_{j=1}^{M} p_j(\varepsilon,\phi)} \sum_{i=1}^{M} \left(z_i - h_i(v_z,t,\varepsilon,\phi) - N v_z\right) p_i(\varepsilon,\phi)\,, \tag{5.68}$$

$$\begin{aligned}\frac{\partial}{\partial \varepsilon} f(t,\varepsilon,\phi) = {} & \frac{-2Mt\varepsilon}{(1-\varepsilon^2)^2 \left(\sum_{j=1}^{M} p_j(\varepsilon,\phi)\right)^2} \sum_{i=1}^{M} \left(z_i - h_i(v_z,t,\varepsilon,\phi) - N v_z\right) \\ & \cdot \left(\left(\sum_{j=1}^{M} p_j(\varepsilon,\phi) \right) \frac{\cos^2(\theta_i - \phi)}{p_i(\varepsilon,\phi)} - p_i(\varepsilon,\phi) \sum_{j=1}^{M} \frac{\cos^2(\theta_j - \phi)}{p_j(\varepsilon,\phi)} \right)\,,\end{aligned} \tag{5.69}$$

$$\begin{aligned}\frac{\partial}{\partial \phi} f(t,\varepsilon,\phi) = {} & \frac{-2Mt\varepsilon^2}{2(1-\varepsilon^2) \left(\sum_{j=1}^{M} p_j(\varepsilon,\phi)\right)^2} \sum_{i=1}^{M} \left(z_i - h_i(v_z,t,\varepsilon,\phi) - N v_z\right) \\ & \cdot \left(\left(\sum_{j=1}^{M} p_j(\varepsilon,\phi) \right) \frac{\sin 2(\theta_i - \phi)}{p_i(\varepsilon,\phi)} - p_i(\varepsilon,\phi) \sum_{j=1}^{M} \frac{\sin 2(\theta_j - \phi)}{p_j(\varepsilon,\phi)} \right)\,,\end{aligned} \tag{5.70}$$

which is the desired result.

Chapter 6

ESTIMATING SUPPORT VALUES FROM PROJECTIONS

6.1 Overview

In this chapter we examine two methods to estimate support values from projections. Using the notation from Section 2.5 and referring to Fig. 2.2 we see that the objective is to estimate $t_-(\theta)$, the t-coordinate at which the projection $g(t,\theta)$ becomes non-zero (as t is increased from $-T$), and $t_+(\theta)$ $(\geq t_-(\theta))$, the t-coordinate at which $g(t,\theta)$ becomes zero again. The first method we discuss in Section 6.2 uses Kalman filtering and generalized likelihood ratios to estimate points of slope discontinuity — these points are called *knots*. The knot closest to $-T$ becomes the estimate of $t_-(\theta)$ and the knot closest to T is the estimate of $t_+(\theta)$. The focus of this method is *local* in the sense that only a small interval around a support estimate is important in determing the estimate — the remainder of the projection is largely unimportant. The second method we develop in Section 6.3 is more of a global approach since it uses the (known) mass and center of mass of the projection to estimate support values which are as close to the origin as possible so that the estimated projection (subject to the proposed support constraint) is consistent with the known mass. We motivate the method from the point of view of Akaike's model order estimation methods, but develop the concept a bit further to show that what is actually being done is MAP estimation of the support values using a certain implied prior. We

then generalize the method by choosing more appropriate priors, given what we already know about the support values.

To see how this chapter fits into the whole, we should observe that given the support values $t_-(\theta)$ and $t_+(\theta)$ for all angles $\theta \in [0, \pi)$ we may construct the support function of the object support set $\mathcal{F}$ (refer to Section 2.5) as follows

$$h(\theta) = \begin{cases} t_+(\theta), & 0 \leq \theta < \pi \\ -t_-(\theta - \pi), & \pi \leq \theta < 2\pi \end{cases} . \tag{6.1}$$

Since the projections are noisy and since we have only a finite number of them in the discrete sinogram, we see that estimating the support values for each projection yields a noisy observation of the true *support vector*, a concept we explored in depth in Chapters 4 and 5. Also, in the sparse- and limited-angle cases, we may obtain support value estimates only for a subset of the projections in the full sinogram. Therefore, since we obtain noisy and partial observations, the methods in this chapter do not, in general, yield the desired segmentation of $\mathcal{Y}_T$ into $\mathcal{G}$ and $\bar{\mathcal{G}}$ (see Section 2.5). In general, we must use one of the support vector estimation algorithms of Chapters 4 and 5 in order to produce a feasible support vector and thus a feasible segmentation of $\mathcal{Y}_T$ from the support value estimates made using the methods of this chapter. In Chapter 7 we present the full hierarchical reconstruction algorithm, and show several results of this sequence of steps. In this chapter, however, we concentrate entirely on producing two support estimates from a single projection, and on obtaining estimates of the magnitudes of the estimation errors.

6.2 Knot Location Method

In this section we model the projection $g(t)$ as a continuous piecewise-linear waveform as shown in Fig. 6.1.[1] Such a function is called a linear *spline*; it is composed of a set of linear functions which connect a series of points called *knots*, so that the resultant function is continuous but its slope has an abrupt change at each of the knots. Referring again to Fig. 6.1, we see that the objective is to determine

[1] In this section and in subsequent sections we drop the θ index from our usual notation indicating projections.

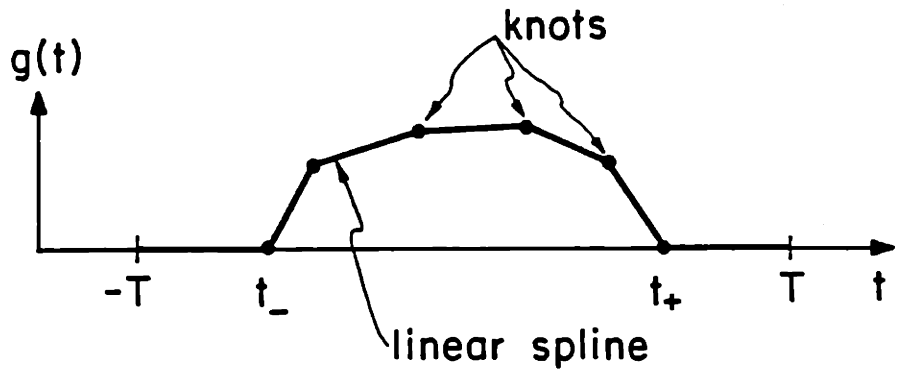


Figure 6.1: A projection modeled as a linear spline with knots.

the locations of the two knots which correspond to the points t_- and t_+. To do this we use the generalized likelihood ratio (GLR) techniques developed by Willsky and Jones [94] for detecting abrupt changes in dynamic systems, and later applied to spline estimation by Mier-Muth and Willsky [62]. The basic approach is to run a Kalman filter, starting at $t = -T$, which assumes an underlying signal model corresponding to a *linear ramp* waveform (initialized with with slope=0 to begin with). Examining the innovations of the Kalman filter, which should be zero-mean white Gaussian noise, provides the basis for estimating the point at which the true signal deviates from the assumed model — i.e. the point at which the slope changes suddenly. The first such point yields the point t_-; the point t_+ is found by running the filter backwards starting at $t = T$ and finding the first knot in the backwards direction.

Before discussing this method in more detail we motivate this approach by examining the noise-free projection of the function which is 1 on the disk D_a and zero elsewhere as shown in Fig. 6.2. The figure depicts both the projection $g(t)$ and the absolute value of its derivative with respect to t, $|\dot{g}(t)|$. We see that the projection is continuous (although this need not be the case in general) and that at each of the

support values the derivative has a discontinuity. Indeed, in this example, $\dot{g}(t)$ experiences an infinite discontinuity at both t_- and t_+. The focus of our efforts in this section is therefore to identify the positions of discontinuties in the derivative of the projection. Since we observe the projections in additive noise, it is not appropriate (especially, in the low SNR cases) to actually take finite differences to approximate the derivative since this tends to accentuate the effects of the noise, producing spurious estimates. Instead, the Kalman filter provides some noise reduction and the basis for the GLR methods.

We now present the approach taken to find a single knot assuming that the true signal has a single slope discontinuity [62]. A more general theory (for abrupt changes in several states or for the case of multiple knots, for example) may be found in [94] and [62]. We assume the true signal (projection) may be modeled by the following state equation

$$\dot{x}(t) = Ax(t) + \alpha f \delta(t - \tau) \tag{6.2}$$

where $\delta(\cdot)$ is the Dirac delta function, τ is the location of the (single) knot, α is the size of the discontinuity,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \text{ and } f = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{6.3}$$

and $x_1(t) = g(t)$ and $x_2(t) = \dot{g}(t)$. Now assume that the projections are sampled (i.e., the waveform is discrete) with n_d samples over the interval $[-T, T]$, and that the samples are indexed by i, the first sample being given by $i = 1$. Assuming that the slope discontinuity takes place at one of the sample points, the discrete state equation is given by

$$x(i+1) = \Phi x(i) + \alpha f \delta(i + 1 - k) \tag{6.4}$$

where $\delta(\cdot)$ denotes the discrete impulse function, α is the height of the discontinuity, f is as given above, k is the discrete position of the knot ($1 \leq k \leq n_d$), and

$$\Phi = \exp(A\Delta_t) = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix} \tag{6.5}$$

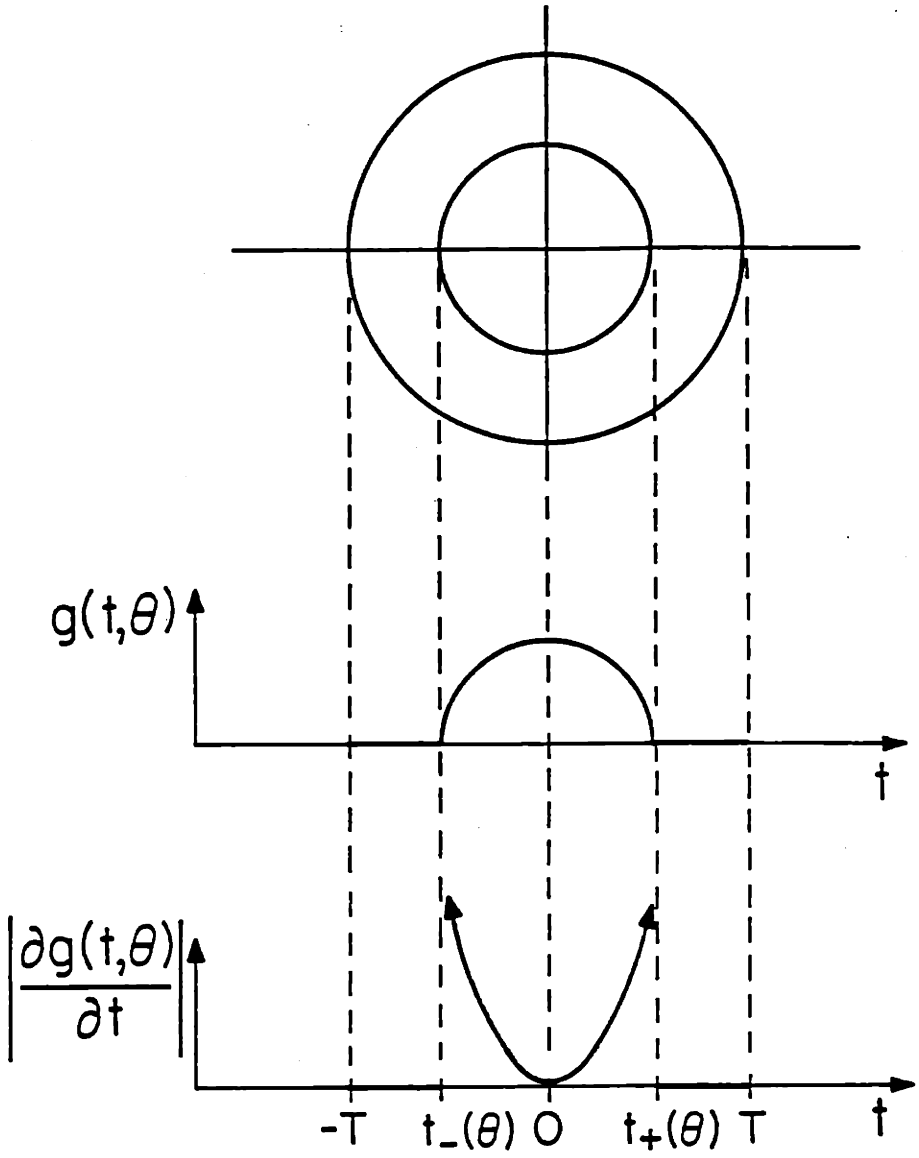


Figure 6.2: A disk object, its projection, and the absolute value of the derivative of the projection.

where $\Delta_t = 2T/n_d$. The actual observations are noisy samples of the projection; hence, we have the following observation equation

$$y(i) = h^{\mathrm{T}} x(i) + v(i) \tag{6.6}$$

where $h^{\mathrm{T}} = [1 \;\; 0]$, and $v(i)$ are zero-mean white jointly Gaussian random vectors with covariance R. The first step in the knot location algorithm is to run the following Kalman filter on the data:

$$\begin{aligned} \hat{x}(i|i-1) &= \Phi \hat{x}(i-1|i-1) && (6.7)\\ \hat{x}(i|i) &= \hat{x}(i|i-1) + K(i)\gamma(i) && (6.8)\\ \gamma(i) &= y(i) - h^{\mathrm{T}} \hat{x}(i|i-1) && (6.9) \end{aligned}$$

where $\hat{x}(i|l)$ is the best estimate of $x(i)$ given $y(1), \ldots, y(l)$, and $\gamma(i)$ is the innovations sequence, and $K(i)$ is the Kalman filter gain. Assuming that there is no jump (slope discontinuity), the innovations are zero-mean, white, jointly Gaussian random vectors with covariance $V(i)$, which, together with the filter gain $K(i)$, may be computed as follows:

$$\begin{aligned} P(i|i) &= [I - K(i)h^{\mathrm{T}}]P(i|i-1) && (6.10)\\ P(i+1|i) &= \Phi P(i|i)\Phi^{\mathrm{T}} && (6.11)\\ V(i) &= h^{\mathrm{T}} P(i|i-1)h + R && (6.12)\\ K(i) &= P(i|i-1)hV^{-1}(i) && (6.13) \end{aligned}$$

where $P(i|l)$ denotes the error covariance of the estimate $\hat{x}(i|l)$. Because a projection is zero outside the disk of radius T and since we may take T to be as large as necessary, we initialize the Kalman filter as follows:

$$\begin{aligned} \hat{x}(0|0) &= \hat{x}(1|0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} && (6.14)\\ P(0|0) &= P(1|0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. && (6.15)\\ & && (6.16) \end{aligned}$$

In order to determine whether a jump has occurred we examine the innovations sequence, which will deviate from the statistics given above if a jump takes place. In particular, given our single jump model of (6.4) it can be shown that the true innovations sequence takes the form [94]

$$\gamma(i) = G(i,k)f\alpha + \tilde{\gamma}(i) \tag{6.17}$$

where $\tilde{\gamma}(i)$ is the innovations if there is no jump and $G(i,k)$ is the *jump signature matrix* given by

$$G(i,k) = h^{\mathrm{T}}\left[\Phi^{i-k} - \Phi F(i-1,k)\right] \tag{6.18}$$

$$F(i,k) = K(i)G(i,k) + \Phi F(i-1,k) \tag{6.19}$$

where $G(i,k)$ and $F(i,k)$ are both 0 for $i < k$ and $F(i,i) = K(i)h^{\mathrm{T}}$.

Equation (6.17) is the key to the GLR knot-location method. Through this equation, we see how to form the ML estimate of α assuming a jump occurred at time k in the filter's past for each current time index i of the Kalman filter. Actually, to reduce the required computation, at each point i we look for possible jumps only over a trailing window $W(i)$ of length N in the filter's past given by

$$W(i) = \{i-1, i-2, \ldots, i-N\}. \tag{6.20}$$

Then using the ML estimate of α for each $k \in W(i)$, we form the GLR for the hypothesis that a jump actually occured at k. If the GLR exceeds a preset threshold then a jump is deemed to have occured. The above calculations are given by the following equations [62] each evaluated for all $k \in W(i)$

$$C(i,k) = \sum_{j=i}^{i} G^{\mathrm{T}}(j,k)V^{-1}(j)G(j,k), \tag{6.21}$$

$$d(i,k) = \sum_{j=k}^{i} G^{\mathrm{T}}(j,k)V^{-1}(j)\gamma(j), \tag{6.22}$$

$$\hat{\alpha}(i,k) = \frac{f^{\mathrm{T}}d(i,k)}{f^{\mathrm{T}}C(i,k)f}, \tag{6.23}$$

$$l(i,k) = \frac{(f^{\mathrm{T}}d(i,k))^2}{f^{\mathrm{T}}C(i,k)f}, \tag{6.24}$$

where $\hat{\alpha}(i,k)$ is the ML estimate of α assuming that a jump occurred at time k, and $l(i,k)$ is the logarithm of the generalized likelihood ratio for this event. The best estimate of the *location* of a jump is then given by

$$\hat{k}(i) = \underset{k \in W(i)}{\text{argmax}}\ l(i,k)\,. \tag{6.25}$$

Then, to decide whether a jump has actually taken place we use the following threshold rule

$$l(i,\hat{k}(i)) \begin{array}{ll} > \varepsilon & \text{Jump} \\ \leq \varepsilon & \text{No Jump} \end{array}\,. \tag{6.26}$$

We describe an adaptive method to choose ε in Chapter 7.

6.3 Support Width Penalty Methods

6.3.1 Introduction

In this section we develop a method to determine the support values of a projection which uses knowledge of the mass, center of mass, and positivity of the projections. The approach is depicted in Fig. 6.3. Here we show a noisy observation $y(t)$ of a projection and two *candidate* support values $\tilde{t}_-$ and $\tilde{t}_+$. The estimate $\hat{g}(t)$ of the projection is the ML estimate of the true projection given the measurement y and assuming that the projection is positive, has a center of mass at the origin, has a total mass equal to m, and has support values $\tilde{t}_-$ and $\tilde{t}_+$. We now make the following observation: as the two candidate support estimates approach the origin, the estimate $\hat{g}(t)$ becomes a poorer estimate of the true projection because its height must grow too large in order to accomodate the mass constraint. Such an estimate also has a low likelihood. Suppose, on the other hand, we moved both candidate support values toward the two bounds $-T$ and T. In this case, the likelihood will increase, and in fact, setting $\tilde{t}_- = -T$ and $\tilde{t}_+ = T$ will, in general, produce an estimate $\hat{g}(t)$ which maximizes the likelihood function.

However, maximizing the likelihood function cannot be the sole objective here since we also desire estimates of the support values t_- and t_+ which are supposed

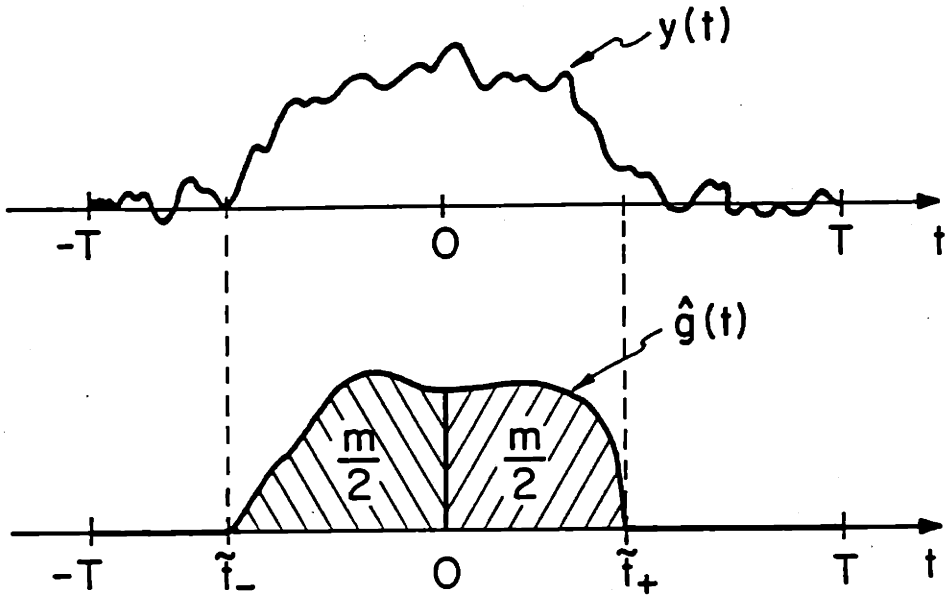


Figure 6.3: A noisy projection and a constrained estimate.

to tightly bound the region of support of the true projection. Therefore, we add a penalty function to the likelihood function which has a larger value for smaller support widths, and seek a *joint* estimate of the true projection and the two support values. The form of the penalty function is derived using the model identification methods due to Akaike [1] which serve as an extension to the methods of maximum likelihood. We also show in a later section how these methods may be interpreted as a joint MAP estimation of the projection and the support values when a certain implied prior on the support values is assumed.

This method is not a local edge-finding algorithm; indeed, the characteristics (e.g., value and slope) of the observation at an estimated support value are not intrinsically an important factor in determining that estimate. Thus, this method is quite different in character than the knot location methods of the previous section. We compare the performance of the two in Section 6.5.

6.3.2 Formulation

Using the known center of mass (taken to be at $t = 0$, without loss of generality) we divide each projection in two *half-projections* and treat each half independently. The discrete version of a half-projection will be denoted by the vector $s = [g(0)\ g(\Delta_t)\ g(2\Delta_t)\ \ g(N\Delta_t)]^{\mathrm{T}}$, where for n_d (odd) samples per projection (using the notation of Chapter 3) we have that $N = (n_d - 1)/2$ and $\Delta_t = 2T/n_d$. An observation of a discrete half-projection is given by $y = s + n$ where the elements of n are zero-mean, independent Gaussian random variables.

We now define a series of hypotheses H_k, $k = 0, \ldots, N$, where H_k corresponds to the hypothesis that $s_{k+1} = s_{k+2} = \cdots = s_N = 0$. The index k corresponds to the quantity $\tilde{t}_+$ (or $\tilde{t}_-$ for a left-side half-projection) and in order to use the methods of Akaike, we consider k to be an unknown model parameter. Now we form the log likelihood function as

$$l(s) = -\frac{1}{2\sigma^2}(y - s)^{\mathrm{T}}(y - s) - \frac{1}{2}\ln\left|2\pi\sigma^2 I\right| \tag{6.27}$$

and define $\hat{s}_k$ to be the maximum likelihood estimate of s assuming that hypothesis H_k is true. Note that $\hat{s}_k$ corresponds to $\hat{g}(t)$ given above. We find $\hat{s}_k$ by solving the following QP

$$\begin{aligned} \text{minimize} \quad & (y - s)^{\mathrm{T}}(y - s) \\ \text{subject to} \quad & s_{k+1} = s_{k+2} = \cdots = s_N = 0, \qquad \text{(Support)} \\ & s_i \geq 0 \ \ \forall i, \qquad \text{(Positivity)} \\ & \frac{d}{2}s_0 + \sum_{i=1}^{k} d s_i = \frac{m}{2}. \qquad \text{(Mass)} \end{aligned} \tag{6.28}$$

In Akaike's method, the penalty function which is added to the likelihood function is derived from the number of degrees of freedom f_k (also called the number of free parameters) associated to each hypothesis. Since (6.28) fixes $s_{k+1} = s_{k+2} = \cdots = s_N = 0$, then the number of free parameters for hypothesis H_k can be no larger than $k + 1$, the number of remaining elements of s to be be determined. However, the mass constraint removes one degree of freedom so that the number

of free parameters for H_k is exactly k, i.e., $f_k = k$. The estimate of the model parameter minimizes the Akaike Information Criterion (AIC) given by

$$AIC(k) = -2l(\hat{s}_k) + 2f_k - (N+1)\,. \tag{6.29}$$

Hence we have

$$\begin{aligned} \hat{k} &= \underset{0\le k\le N}{\operatorname{argmin}}\, AIC(k) \\ &= \underset{0\le k\le N}{\operatorname{argmin}}\, \frac{1}{\sigma^2}(y-\hat{s}_k)^{\mathrm{T}}(y-\hat{s}_k) + 2k\,. \end{aligned} \tag{6.30}$$

6.3.3 Interpretation as an MAP Estimate

Let us think of the support index k as a parameter of the unknown half-projection s. Given k one knows that $s_{k+1} = \cdots = s_N = 0$, and in addition the half-projection must satisfy $s^{\mathrm{T}}a = b$ — which represents the mass constraint — and $s \geq 0$. We write the observation vector as

$$y = s(k) + n$$

where the half-projection explicitly shows the dependency on the unknown support index k. It is convenient to partition $s(k)$ as follows

$$s(k) = \begin{bmatrix} s^k \\ z^k \end{bmatrix}$$

where $z^k = 0$. For the following development we assume that the unknown half-projection has a prior probability density of the form

$$p(s|k) = \frac{1}{z}\exp\left(-U(s)\right) \quad s \in \Omega_k \tag{6.31}$$

where

$$\Omega_k = \left\{ s \mid s \geq 0,\, s^{\mathrm{T}}a = b,\, z^k = 0 \right\}. \tag{6.32}$$

Using the observation equation we may write

$$p(y|s,k) = \left|2\pi\sigma^2 I\right|^{-1/2} \exp\left(-\frac{1}{2\sigma^2}(y-s(k))^{\mathrm{T}}(y-s(k))\right), \tag{6.33}$$

and Bayes' rule may be used as follows

$$p(y,s|k) = p(y|s,k)\,p(s|k) \tag{6.34}$$

$$= \frac{|2\pi\sigma^2 I|^{-1/2}}{z} \exp\left(-\frac{1}{2\sigma^2}(y-s(k))^{\mathrm{T}}(y-s(k)) - U(s(k))\right). \tag{6.35}$$

Now suppose that k has a known prior distribution of the form

$$p(k) = \frac{1}{v}\exp(-\alpha k)\,. \tag{6.36}$$

Then we have

$$p(y,s,k) = p(y,s|k)\,p(k) \tag{6.37}$$

$$= \frac{|2\pi\sigma^2 I|^{1/2}}{zv} \exp\left(-\frac{1}{2\sigma^2}(y-s(k))^{\mathrm{T}}(y-s(k)) - U(s(k)) - \alpha k\right). \tag{6.38}$$

Now we observe that for an observation $y = Y$ the *joint* MAP estimate of s and k is given by

$$(\hat{s},\hat{k})_{MAP} = \underset{s,k}{\operatorname{argmax}}\ \ln p(y=Y,s,k)$$

$$= \underset{s,k}{\operatorname{argmax}} -\frac{1}{2\sigma^2}(Y-s(k))^{\mathrm{T}}(Y-s(k)) - U(s(k)) - \alpha k\,. \tag{6.39}$$

To solve the above MAP problem we might simply consider each k, find the best $s(k)$, and after enumerating all k, find the pair which maximizes the objective function together. One sees by comparing (6.29) with (6.39) that the solution of the joint MAP problem above yields the Akaike estimate of k only if $U(s(k))$ is constant for all k and $s(k)$, and if $\alpha = 2$. The advantage of the present formulation is that it reveals how changes may be made to modify the performance of the estimator for different experimental situations; we take up this matter in the following section.

6.3.4 Alternate Formulations

In this section we propose several modifications to the support width penalty methods given above to account for deficiencies we have noted in the experiments and to account for cases where additional prior knowledge is available. These changes are based on the MAP interpretation given in the previous section.

We have found in the experiments (see Section 6.5) that the standard Akaike-based support width penalty method produces support estimates which tend to be too small (i.e., too close to the origin). A simple modification to the standard Akaike estimator is simply to vary the constant α in (6.39) but to leave the term $U(s(k))$ constant over all k and $s(k)$. For example, if $\alpha < 2$ then there is less tendency *a priori* for k to be small. In fact, if α becomes too small, then the increase in the likelihood of $s(k)$ as k is increased overwhelms the αk term in the objective function and the contribution of the penalty term becomes negligible. Of course, α may be made larger than 2, but we have found in the experiments that this tends to produce support estimates that are much too small.

The MAP methods outlines above suggest a way to incorporate the prior knowledge that $s(k)$ tends to be smooth over the region of support — this is an idea that follows naturally from our development of the sinogram MRF in Chapter 3. To include this prior knowledge we simply define $U(s)$ to produce a smoothing effect. We may do this by modeling $s(k)$ as a 1-D MRF with nearest-neighbor quadratic "affinities". Accordingly, we may define the energy function as

$$U(s(k)) = \sum_{i=1}^{k} b_v (s_i - s_{i-1})^2$$

which specifies affinities between the values of $s(k)$. The real constant b_v is non-negative and is set larger for increasing similarity between the adjacent half-projection values.[2]

A third modification to the standard methods that one might make is to change the form of the prior on k completely. For example, suppose that we had prior information that suggested that the support index is near l, for $0 < l < N$. Then a reasonable prior for k might be

$$p(k) = \frac{1}{z} \exp\left(-\frac{1}{2\xi^2}(k-l)^2\right) \quad 0 \le k \le N$$

where z is chosen to make $p(k)$ sum to 1 and ξ is a real constant chosen to reflect our confidence in l. For example, in the problem of support estimation for the set

[2] Technically, when such smoothing is incorporated, we should consider the projection as a whole rather than as two half-projections. The solution would then involve a two-dimensional search over the two candidate support values rather than the one-dimensional search for a single support value.

of projections comprising a sinogram, one might derive an initial support estimate — yielding l — by linearly interpolating the estimates from the two adjacent projections. The development of the joint MAP estimate to solve the above problem is similar to that given above and the only difference is in the final form of the Akaike criterion is to replace αk by $\frac{1}{2\xi}(k-l)^2$.

Of course, one may combine the modifications described above to obtain a variety of different estimators.

6.3.5 Computational Issues

All of the support-width penalty algorithms have the following underlying problem to solve

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2\sigma^2}(Y-s)^{\mathrm{T}}(Y-s) \\ \text{subject to} \quad & s \geq 0 \\ & s^{\mathrm{T}}a = b \\ & s_{k+1} = \cdots = s_N = 0 \end{aligned}$$

for some k. This problem is clearly a QP, but because of the last constraint many of the values are fixed (at zero), and the number of free elements of s may be considerably less than the dimension of s. In fact, using the notation given earlier we see that an equivalent problem to solve is

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2\sigma^2}(Y^k-s^k)^{\mathrm{T}}(Y^k-s^k) \\ \text{subject to} \quad & s^k \geq 0 \\ & (s^k)^{\mathrm{T}}a^k = b\,. \end{aligned}$$

Denoting the solution to this problem as $\hat{s}^k$ we see that the solution to the former problem is simply

$$\hat{s} = \begin{bmatrix} \hat{s}^k \\ 0 \end{bmatrix}.$$

6.4 Performance of the Support Estimation Algorithms

6.4.1 Introduction

How well do the support estimation algorithms of Sections 6.2 and 6.3 work? It is important in our hierarchical approach to reconstruction to be able to assess the performance of each stage so that this information may be fed into the next stage of the algorithm — this issue is considered in greater detail in Chapter 7. In particular, we know that the estimates produced by the support estimation algorithms of this chapter must be further processed by those of Chapters 4 and/or 5 so that — at the very least — the segmentation of the sinogram is feasible. Also, in order that the MAP algorithms correctly trade off the measurements with the *a priori* information, the error variance of the initial support estimates must be known. But the performance of the initial algorithms depends not only on the variance of the additive noise but also on the characteristics of the underlying projection, particularly at or around the true support value. And these characteristics can vary widely from projection to projection even for the same object. For example, the two projections of the ellipse shown in Fig. 6.4 are taken ninety degrees apart, and they show quite different profiles. One would expect that, given the same additive noise to each projection, the initial support estimates of the rightmost projection (shown vertically) would be much better than those computed from the bottom projection.

In this section we examine methods to obtain estimates of the error variance arising from each of the two initial support estimation algorithms described in Sections 6.2 and 6.3. Since there is no shape information concerning the object available *a priori* these estimates are made completely on the basis of the statistics available during the processing itself.

6.4.2 The Knot-Location Algorithm

Mier-Muth and Willsky [62,61] have examined the performance of the knot-location algorithm in great depth. In the ideal situation where there is a single knot, a sample

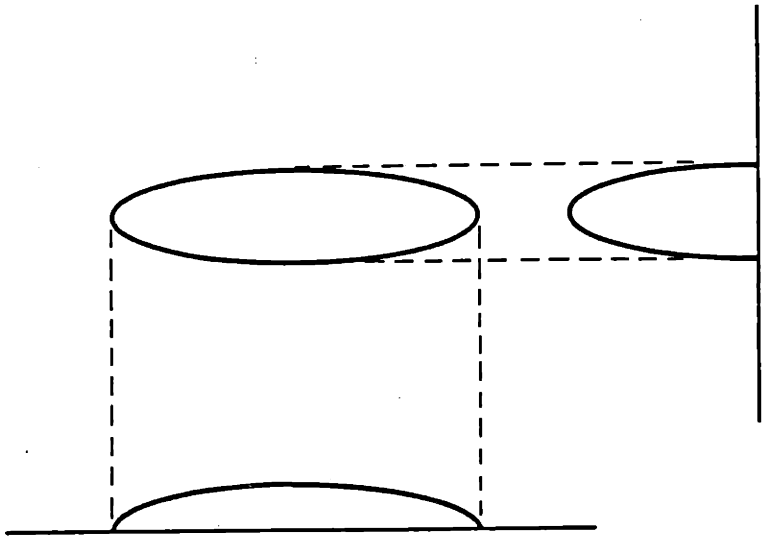


Figure 6.4: Two projections of an ellipse object.

of the log likelihood function $l(i,k)$ is a chi-squared random variable with one degree of freedom, whose expected value is

$$E\{l(i,k)\} = 1 + \delta^2(i,k,k_T) \tag{6.40}$$

where δ^2 is the noncentrality parameter of the distribution and k_T is the true knot location. The function $E\{l(i,k)\}$ when viewed as a function of k is known as the *generalized ambiguity function* and often plays a role in analyzing the performance of a given estimation scheme [92]. They further show that among other things δ^2 has its maximum value at $k = k_T$ for any fixed value of i (for $i > k$) which means that the ML estimate is unbiased. The Cramer-Rao lower bound on the error variance can be determined from the second derivative of the generalized ambiguity function evaluated at the true knot location. If we had an expression for the Cramer-Rao bound, we might use our estimates (of slope and of the knot location) to evaluate the bound, and use this as an estimate of the error variance.[3] Instead, the approach we take is to fit a downturned quadratic centered at $\hat{k}$ to the log likelihood function

[3] Mier-Muth and Willsky did not derive an expression for the Cramer-Rao bound, however.

which was evaluated over the window $W(i)$. Thus, although any given likelihood function is but a sample of the ensemble — and therefore does not yield the expected value — the act of fitting this quadratic is much like determining the Cramer-Rao bound using the true ambiguity function.

To see how to make this quadratic fit we let $\hat{k} \in W(i)$ be our estimate of the knot location (made when the Kalman filter has progessed to index i). We wish to fit a downturned quadratic of the form

$$\hat{l}(k) = -a(k - \hat{k})^2 + c \tag{6.41}$$

to the data $l(i,k)$ so that, in particular, we may determine a. To make this fit, we minimize

$$\sum_{k=1}^{N} (\hat{l}(k) - l(i,k))^2 = \sum_{k=1}^{N} \left(-a(k - \hat{k})^2 + c - l(i,k)\right)^2$$

with respect to a and c. Differentiating with respect to a and setting the result to zero yields

$$\frac{\partial}{\partial a} \sum_{k=1}^{N} \left(-a(k - \hat{k})^2 + c - l(i,k)\right)^2 = \sum_{k=1}^{N} 2\left(-a(k - \hat{k})^2 + c - l(i,k)\right)\left[-(k - \hat{k})^2\right] = 0$$

from which we may determine

$$a = \frac{\sum_{k=1}^{N} (c - l(i,k))(k - \hat{k})^2}{\sum_{k=1}^{N} (k - \hat{k})^4} \tag{6.42}$$

and

$$c = l(i, \hat{k})\,. \tag{6.43}$$

The estimate of error variance for the support estimate is given by

$$\text{Initial Support Error Variance} = \frac{1}{2|a|}\,. \tag{6.44}$$

6.4.3 Performance of the Support-Width Penalty Algorithm

In order to implement the support-width penalty method within the hierarchical algorithm to be introduce in Chapter 7, we need an on-line estimate of its performance. So, together with the actual support value estimate, $\hat{k}$, we desire an estimate of the variance $\hat{\sigma}_k^2$ of this estimate.

Following the results of Section 6.3, we may interpret the support-width penalty estimate $\hat{k}$ as the maximum of the *a posteriori* density $p(k|y)$ where y represents the measurements of a half-projection s. Although we do not have $p(k|y)$, we may take the normalized exponential of -AIC to be a sample $\tilde{p}(k|y)$ of this probability density function:

$$\tilde{p}(k|y) = \frac{1}{z}\exp\left(-AIC(k)\right) \tag{6.45}$$

where

$$z = \sum_{k=0}^{N}\exp\left(-AIC(k)\right) .$$

Then we form our estimate of the error variance as

$$\begin{aligned}\hat{\sigma}_k^2 &= \sum_{k=0}^{N}(k-\hat{k})^2\tilde{p}(k|y) \\ &= \sum_{k=0}^{N}(k-\hat{k})^2\frac{1}{z}\exp\left(-AIC(k)\right) .\end{aligned} \tag{6.46}$$

6.5 Experimental Results

The results presented in this section are designed to demonstrate the general behavior of the knot-location (KL) and the support-width penalty (SP) algorithms. More detailed results regarding the knot-location algorithm are given within the context of the full hierarchical algorithm in Chapter 7.

Figs. 6.5 and 6.6 show the results of several simulations using the knot-location and support-width penalty algorithms on several noisy projections. Both figures show noisy projections of an ellipse centered at the origin, with major semiaxis radius of 0.806 and minor semiaxis radius of 0.242. Fig. 6.5 shows the *narrowest* projection — so that the support values are -0.242 and and 0.242 — with different noise variances. Fig. 6.6 shows the *widest* projection — so that the support values are -0.806 and 0.806 — also with different noise variances. The positions of the true support values are indicated by the vertical dotted lines in each of the panels. As a point of interest, the ellipse is exactly the same dimensions and orientation as the M I T ellipse introduced in Section 3.6, but the letters in the interior are not present. Also, the indicated noise standard deviation in each of the panels

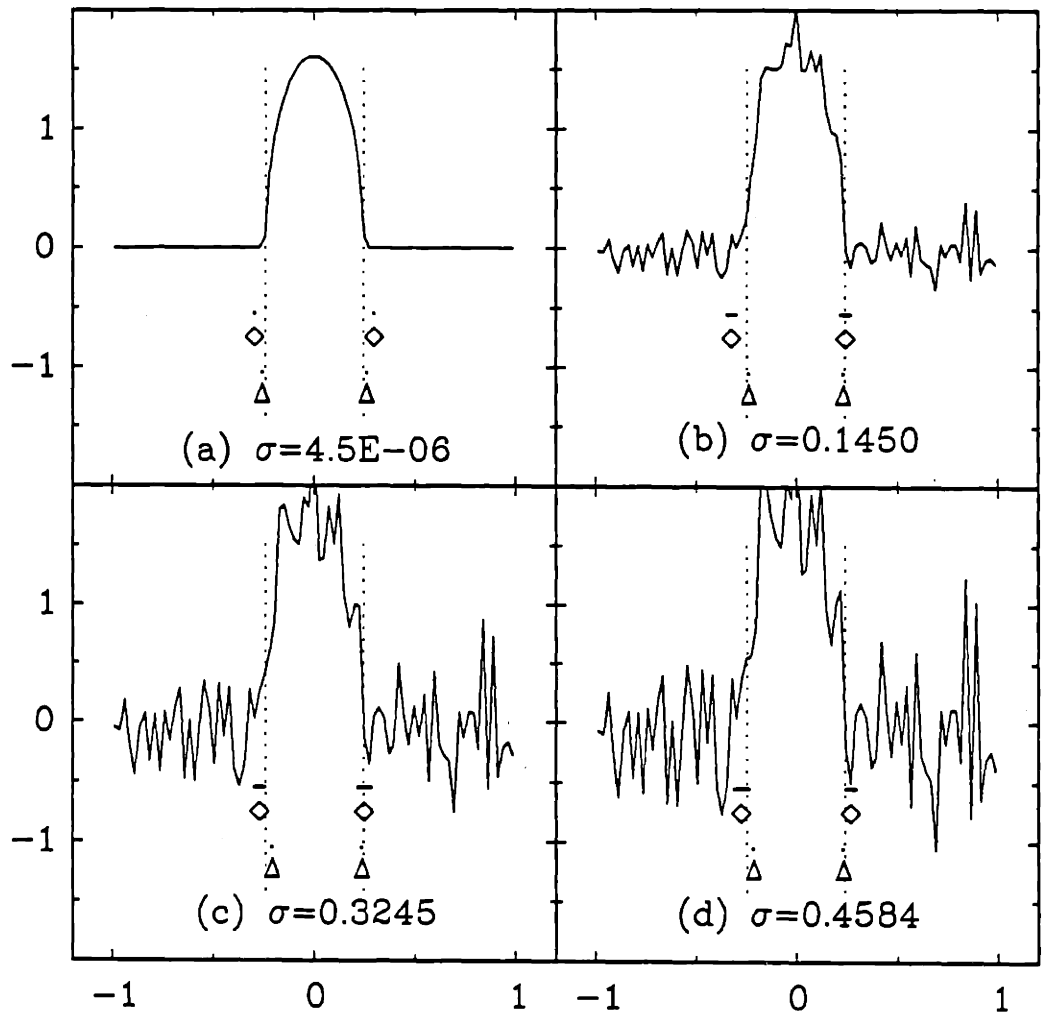


Figure 6.5: Support value estimates using the Knot-location (diamond markers) and Support-width penalty (triangular markers) algorithms: head-on noisy projections of an ellipse.

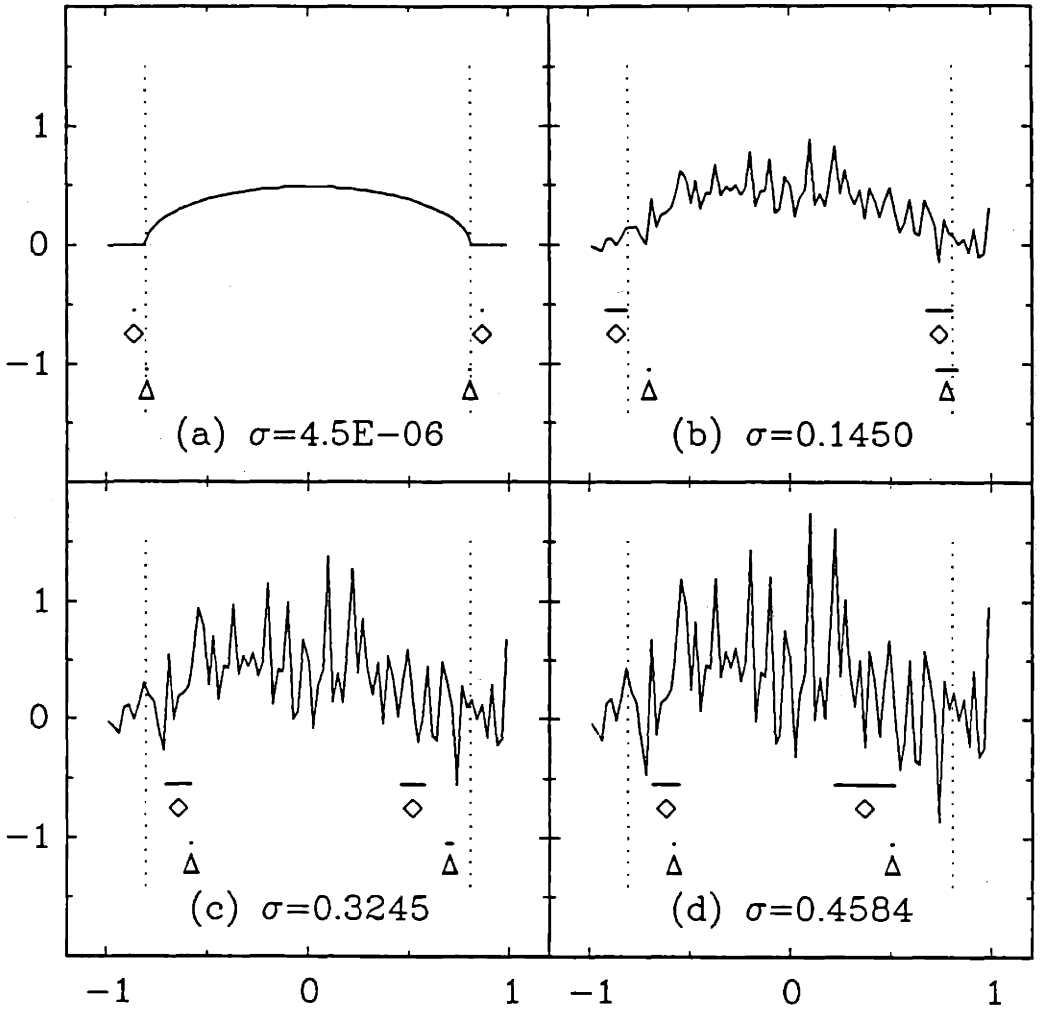


Figure 6.6: Support value estimates using the Knot-location (diamond markers) and Support-width penalty (triangular markers) algorithms: broad-side noisy projections of an ellipse.

corresponds to that of noise added to *full* sinograms so as to acheive the SNR (defined in Section 3.6) of (a) 100.0dB, (b) 10.0dB, (c) 3.0dB, and (d) 0.0dB. The same underlying unit variance noise sequence was used for all four projections in each figure, which accounts for the similarity in the noise structure.

The support value estimates due to the knot-location algorithm are indicated by the *diamond* markers in each of the panels in Figs. 6.5 and 6.6, and the error bar directly above each of these symbols has a length of two standard deviations. The corresponding results for the support-width penalty method are indicated by the *triangular* markers and bars directly above them. In several cases the error bars are very short and appear merely as points. We should also note that the threshold values used in the knot-location algorithm for knot detection, were computed on-line by methods to be described in Chapter 7. Also, the mass used for the SP algorithm was computed approximately for each projection independently, whereas in the full hierarchical sinogram reconstruction algorithm, a better estimate would be available since we can average over all the observed projections (see Section 7.2).

There are several observations to make regarding these results. First, in Fig. 6.5a, we should note that although the observations are nearly perfect, the KL algorithm produces estimate which are too far out. This effect is *entirely* due to the choice of threshold — in this case it is too low — and can be corrected by improvements in the methods to be described in Section 7.3. To some a lesser extent this effect is observed in all the panels in Fig. 6.5 and also in Fig. 6.6a. The opposite effect is be observed for the nearly all of the SP estimates: there is a strong tendency for the SP estimates to be near the origin. The fact that the error bars over the SP markers (triangles) tend to be small, indicates that this tendency is a *bias* in the estimates, and that this is a fundamental property of this estimator. In fact, this is one of the reasons for considering the alternate formulations (based on MAP principles) of Section 6.3.4. In constrast, the error bars over the KL markers (diamonds) tend to be larger, and to more reasonably reflect the error in the estimates. However, Figs. 6.6c and 6.6d also indicate a tendency for the KL estimates to be too near the origin, and without the necessary error variance to account for this property. This again reflects a bias in the estimates which may, however, be compensated for by

better threshold selection algorithms.

6.6 Discussion

We have presented two different support value estimation algorithms in this chapter. The Knot-location (KL) method is primarily a local edge-finder while the Support-width Penalty (SP) method is based primarily on the global mass constraint property. Based on the results presented in Section 6.5, we have elected to use the KL method in the full hierarchical algorithm presented in Chapter 7. The reason for this choice is that we require a good performance indicator so that the support vector estimation algorithms can properly trade off the measurement information with the prior information. While the SP algorithm is an interesting approach, and has solid ties to the constraint-based algorithms that are used in a large portion of this thesis, it seems that further work would be required in order to make it useful as a component in our full hierarchical algorithm.

Chapter 7

HIERARCHICAL RECONSTRUCTION ALGORITHM

7.1 Introduction

In this chapter we present a full reconstruction algorithm which is implemented by cascading several of the methods developed in earlier chapters. A block diagram of the full hierarchical algorithm is shown in Fig. 7.1. This figure shows the (sparse- or limited-angle) observations y as input to blocks A, B, and E. Blocks B and E estimate the *center of mass* and *mass*, respectively, using algorithms which will be presented in Section 7.2. Using the center of mass estimate, block A shifts each observed projection so the the new center of mass is at the origin. This set of shifted observations, denoted $\tilde{y}$, feeds into the *support value* estimation stage of block C, implemented using the algorithms presented in Chapter 6. Block D, which is described in Section 7.3, determines suitable threshold values for the knot-location algorithm of Block C. The initial estimate of a (possibly only partially observed) *support vector* h_1 is processed in block F using the methods of Chapters 4 and 5 to produce a feasible support vector estimate h_2. Block G now has the necessary inputs: 1) the shift corrected observations, 2) the estimated mass, and 3) the estimated support vector h_2 which defines a feasible segmentation of the sinogram domain into an estimate of the sinogram support and its complement. Block G produces an estimate of the *full sinogram* (which may have been interpolated or ex-

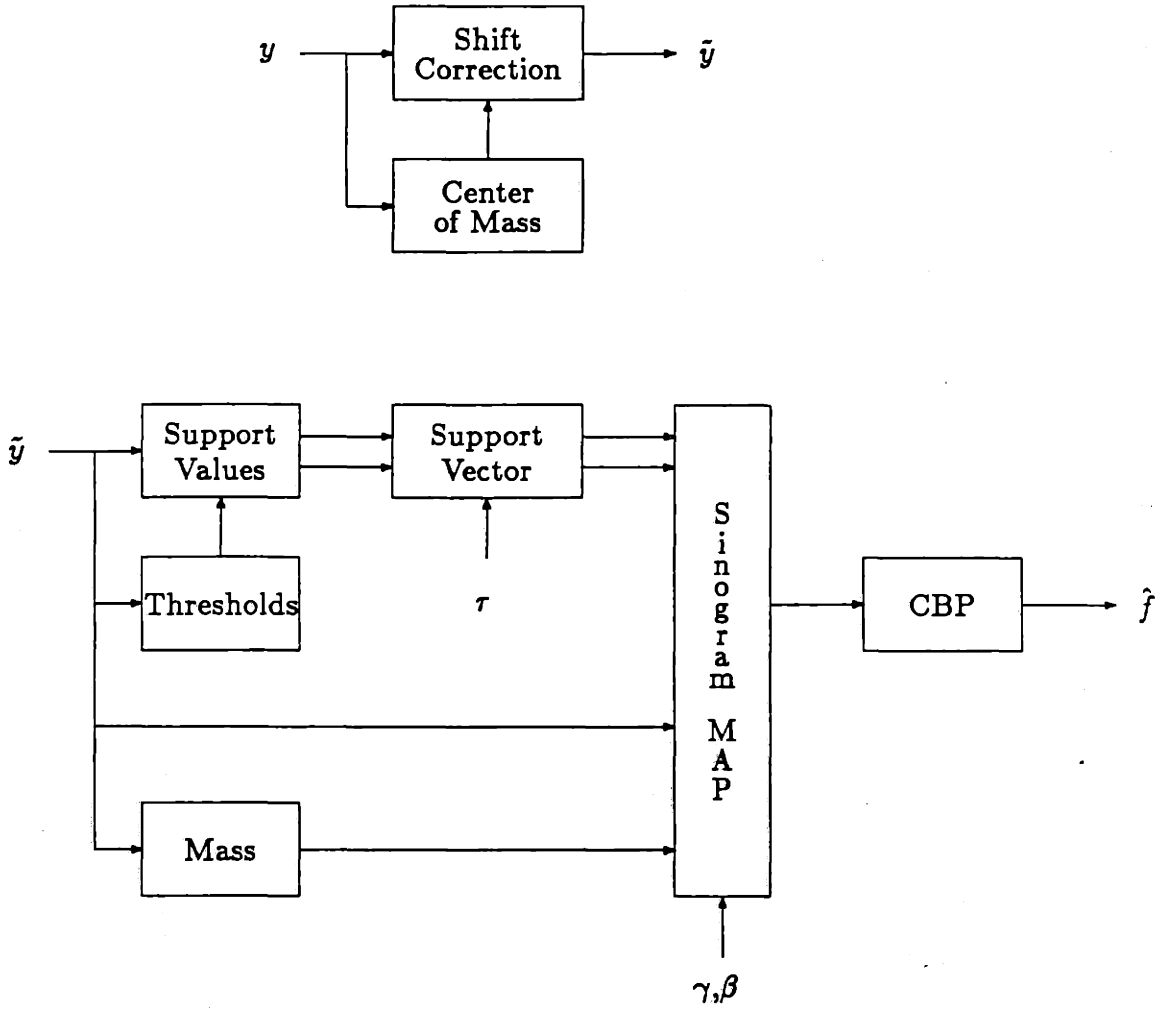


Figure 7.1: Block diagram of the full hierarchical algorithm.

trapolated in the process) using an optimization technique of Chapter 3. Finally, an estimate $\hat{f}$ of the object density is produced by convolution backprojection (CBP) in block H.

Besides passing forward the primary data estimates, many of the blocks also pass forward information containing the *reliability* of its estimates. For example, block F requires an estimate of the error variance for each of the support value estimates produced by block C. Also, we assume that each block knows the variance σ^2 of the uncertainty in each line integral measurement comprising the observations y.

This chapter is organized as follows. In Section 7.2 we discuss the mass, center of mass, and shift-correction algorithm of blocks E, B, and A, respectively. Section 7.3 develops the algorithms used in block D to generate suitable threshold values for block C. In Section 7.4 we develop methods to estimate the reliability of the entire support estimation branch — these methods generate κ for input to block G. In Section 7.5 we present numerous experimental results on a variety of different input geometries and SNR levels and finally, Section 7.6 discusses these results.

7.2 Mass and Center of Mass Estimation

7.2.1 Mass Estimation

As developed in Chapter 2 and utilized as a constraint in Chapter 3, the first order constraint on the Radon transform imposed by the consistency conditions is the mass constraint given by

$$\int_{-\infty}^{\infty} g(t,\theta)\, dt = m \quad \forall\theta\,, \tag{7.1}$$

and may be approximated by the summation

$$\frac{2T}{n_d} \sum_{i=1}^{n_d} g(t_i,\theta_j) = m \quad \forall j\,. \tag{7.2}$$

Now, we have assumed that the observations are given by a finite number of noisy samples of the true Radon transform where each sample is given by $y(t_i,\theta_j) = g(t_i,\theta_j) + n(t_i,\theta_j)$ where $n(t_i,\theta_j)$ is a zero-mean Gaussian random variable, independent for different values of i and j and with variance σ^2. From each observed

projection we form the *observed mass* m_j as follows

$$\begin{aligned} m_j &= \frac{2T}{n_d}\sum_{i=1}^{n_d} y(t_i,\theta_j) \\ &= \frac{2T}{n_d}\sum_{i=1}^{n_d} g(t_i,\theta_j) + n(t_i,\theta_j) \\ &= m + \frac{2T}{n_d}\sum_{i=1}^{n_d} n(t_i,\theta_j) \end{aligned} \tag{7.3}$$

The quantity $n_j = \frac{2T}{n_d}\sum_{i=1}^{n_d} n(t_i,\theta_j)$ is a zero-mean Gaussian random variable with variance $(2T/n_d)^2 n_d \sigma^2$. Therfore, m_j is an observation of m in additive Gaussian noise, and since $\mathrm{E}\{n_j n_k\} = 0$ for $j \neq k$, the ML estimate (also the MMSE estimate) of m is just

$$\hat{m} = \frac{1}{J}\sum_{j\in J} m_j = \frac{2T}{Jn_d}\sum_{j\in J}\sum_{i=1}^{n_d} y(t_i,\theta_j)\,, \tag{7.4}$$

where J is the set of indices corresponding to the angular positions of the observed projections and J is the number of elements in J.

7.2.2 Center of Mass

The center of mass $c \in \mathbb{R}^2$ of the object $f(x)$ is given by

$$c = \frac{1}{m}\int_{x\in\mathbb{R}^2} f(x)x\,dx\,. \tag{7.5}$$

It was shown in Chapter 2 that the center of mass of a projection $c(\theta)$ is related to c by

$$c(\theta) = c\cdot\omega \tag{7.6}$$

where $\omega = [\cos\theta \;\; \sin\theta]^{\mathrm{T}}$. Proceeding as in the previous section, we approximate the center of mass of a projection by

$$c_j = \frac{1}{m}\frac{2T}{n_d}\sum_{i=1}^{n_d} t_i g(t_i,\theta_j)\,. \tag{7.7}$$

Assuming that the observations are given as in the previous section we may calculate for each observed projection the following statistic

$$\begin{aligned} \tilde{c}_j = \frac{1}{\hat{m}}\frac{2T}{n_d}\sum_{i=1}^{n_d} t_i y(t_i,\theta_j) &\approx \frac{1}{m}\frac{2T}{n_d}\sum_{i=1}^{n_d} t_i g(t_i,\theta_i) + \frac{1}{m}\frac{2T}{n_d}\sum_{i=1}^{n_d} t_i n(t_i,\theta_j) \\ &= c_j + n_j \end{aligned} \tag{7.8}$$

where $n_j = \frac{1}{m}\frac{2T}{n_d}\sum_{i=1}^{n_d} t_i n(t_i, \theta_j)$ in this case. The expression is approximate because we have used the mass estimate $\hat{m}$ instead of the true mass to calculate $\tilde{c}_j$. Proceeding as before, we identify the n_j as additive zero-mean Gaussian random variables with variance $(2T/mn_d)^2 \sum_{i=1}^{n_d} t_i^2 \sigma^2$ and $\mathrm{E}\{n_j n_k\} = 0$ for $j \neq k$, so we may write the following center of mass observation equation

$$\tilde{c}_j \approx c \cdot \omega_j + n_j \,. \tag{7.9}$$

Then, given more than two angular observations, the ML solution (equal to the MMSE solution) is given by

$$\hat{c} = (A^{\mathrm{T}} A)^{-1} A^{\mathrm{T}} b \tag{7.10}$$

where

$$A = \begin{bmatrix} \cos\theta_{j1} & \sin\theta_{j1} \\ \vdots & \vdots \\ \cos\theta_{jJ} & \sin\theta_{jJ} \end{bmatrix} \tag{7.11}$$

and

$$b = \begin{bmatrix} \tilde{c}_{j1} \\ \vdots \\ \tilde{c}_{jJ} \end{bmatrix} . \tag{7.12}$$

7.2.3 Shift Correction

The estimate $\hat{c}$ of the center of mass of the object is used to correct each projection so that it corresponds (approximately) to the projection one would measure if the object were centered at the origin. The shift-corrected projection $\tilde{y}(t, \theta)$ is ideally given by

$$\tilde{y}(t, \theta) = y(t - \hat{c} \cdot \omega, \theta) \,. \tag{7.13}$$

However, since y is discretely sampled in t, we would accomplish the shift-correction using interpolation, or by using a (discrete) Fourier transform of y, followed by multiplication by a linear phase term, and then inverse transforming. In our simulation studies, however, the true objects were known to be centered at the origin, and therefore we did not implement the shift correction phase of the algorithm.

7.3 Threshold for Knot-Location

In Chapter 6, Section 2, we described a GLR technique for support value estimation. Once the maximum likelihood over a trailing window is determined, the decision as to whether a knot has occurred depends on the value of threshold ϵ. In this section we describe a heuristic method to choose a threshold value for each projection which leads to good experimental results.

Since the mass of each projection is the same, we might expect that a projection with a small support width would rise rapidly at the support values in order to include the required mass. In contrast, a projection with a larger support width might rise less rapidly. This effect is demonstrated quite well for the ellipse of Fig. 6.3. To get an initial idea as to the width of a projection we calculate the approximate second moment of a shift-corrected projection as follows

$$m_2(\theta_j) = \frac{1}{\hat{m}} \sum_{i=1}^{n_d} t_i^2 \max\{0, \tilde{y}(t_i, \theta_j)\}\,, \tag{7.14}$$

where $\hat{m}$ is the estimated mass. The $\max\{\}$ function is included since it possible that elements of $\tilde{y}$ are negative, and therefore that $m_2(\theta)$ might otherwise be negative. The second moment is roughly equivalent to a variance calculation and the quantity $p(\theta) = \sqrt{m_2(\theta)}$ is analogous to a standard deviation. This gives us an approximate measure of the width of the projection.[1]

We now prescribe a threshold value which is a function of this width measure p. If p is large, then the projection is wide and the slope change at the support value is probably small. Therefore, we want to specify a GLR threshold ϵ that is relatively small. Using similar reasoning we conclude that for small p, the threshold ϵ should be large. After some experimentation we have chosen the function $\epsilon(p)$ depicted in Fig. 7.3. Since p is a measure of the width of the entire projection, ϵ is used as the threshold value for both the forward and backward stages of the knot-location algorithm.

[1] In fact, since the object is known to be centered at the origin, the quantity $p(\theta)/2$ may be considered to be a coarse estimate of $h(\theta)$ (and $h(\theta + \pi)$) itself — but this is not how we use this quantity.

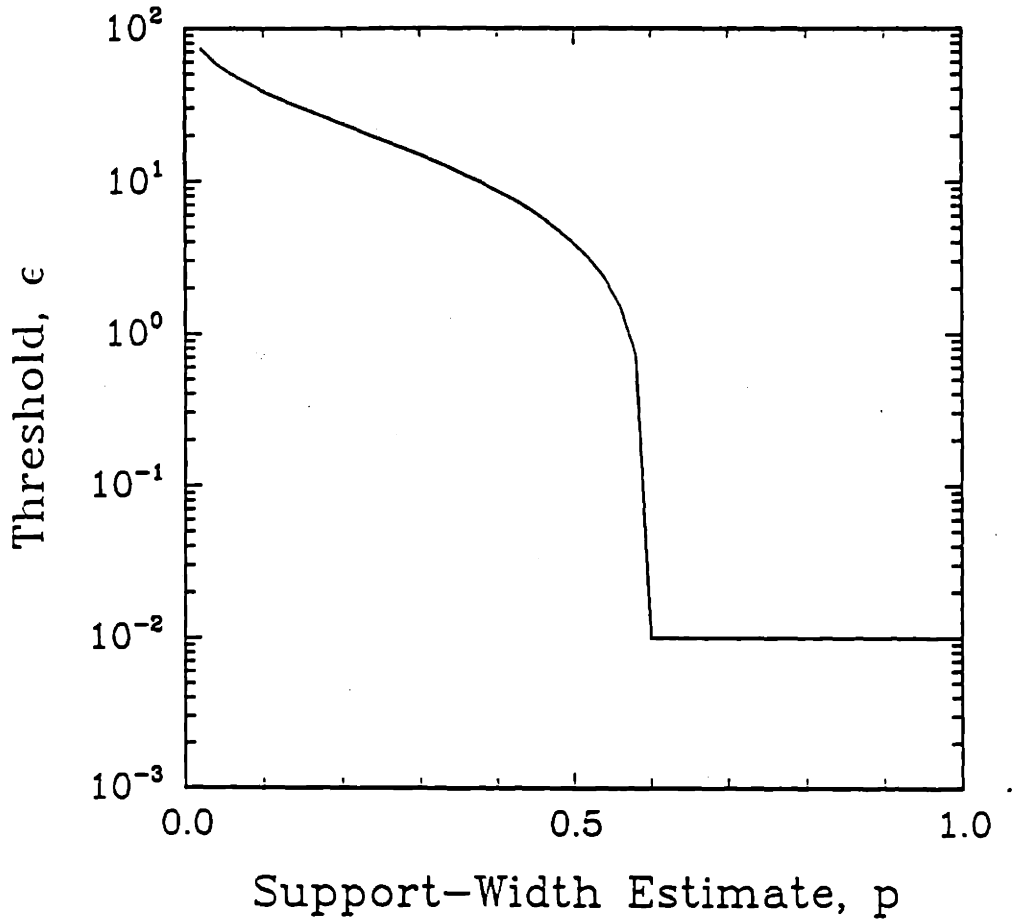


Figure 7.2: Threshold selection curve.

7.4 Overall Support Performance

The coefficient κ is used in the sinogram MAP algorithms to indicate the confidence in the given sinogram segmentation into its region of support $\mathcal{G}$ and its complement $\bar{\mathcal{G}}$. A larger value indicates a higher degree of confidence, so a very large value of κ would be used if the true segmentation were known and a smaller value for estimated segmentations. In this section we discuss several aspects involved in the selection of κ from the perspective of the hierarchical algorithm of this chapter.

7.4.1 Spatially Varying κ

Our ability to determine the true segmentation of the sinogram domain $\mathcal{Y}_T$ depends on several factors. First, the SNR of each projection affects the performance of the support value algorithm — a larger SNR means that, at least for observed projections, the support value estimate will be more accurate. A second factor comes from the geometric arrangement of the projections. For example, in the limited-angle case, there may be a large gap in the observed angles, and in that range the support estimates may be poor, regardless of the SNR. A third factor which must be considered is that the size of the object itself can have a strong influence on the performance of the support value estimation. For example, a small object with the same mass as a large object would, in general, allow better support value estimation than that of the large object. A fourth influence on the overall support estimation performance is the presence and accuracy of prior knowledge — e.g., prior shape information. For example, with accurate prior knowledge the missing projections may have their support values estimated with greater accuracy. A fifth factor is the size of the smoothing coefficients β and γ of the sinogram MRF. As these coefficients increase, κ must increase as well in order to maintain the same effect near the boundaries of the segmentation. A final factor might simply be the total number of observed projections — the more observations, the better we would expect to be able to determine a good segmentation.

Some of the above factors are embodied in the performance indicators that we derived in Section 6.4 for the knot-location and support-width penalty *support value*

estimation algorithms. For example, these performance measures contain in them information related to the SNR and support-width of the observed projections. It would seem reasonable to use this information to specify a κ that varies in t. For example, one might want κ to be small near the estimated support value and increase with increasing t. The rate of increase might be related to the performance measure mentioned above. Since each projection has a (potentially) different error associated to it, clearly κ would also vary in θ. For example, when there are missing projections we might want the value of κ to be much smaller for the missing angles, depending upon our confidence in our knowledge about prior shape.

Referring to our formulation of Chapter 3, we may think of κ as spatially varying in the following sense: the value of κ is zero at the $t = 0$ axis, but as t is increased (or decreased), it becomes non-zero at the support value, and remains constant out to $t = T$ (or $t = -T$). In order to use such a (piecewise-constant) κ function but also to incorporate some of the information concerning the reliability of the support value estimates, we may consider changing the *region* over which κ is non-zero. In Section 7.4.2, we discuss a method which moves the points at which κ becomes non-zero towards the $\pm T$ boundaries by an amount which is a multiple of the error standard deviation of the support estimate.

7.4.2 Modification of the Segmentation

Up until now, we have thought of the segmentation of the sinogram as having been produced by an estimated support vector. Here, however, we propose to alter this segmentation of the sinogram domain using the current support value estimate *and* its error variance estimate. We do this by *increasing* each support value by adding some fraction of its own error standard deviation. In this way, for large estimated errors, the segmentation boundary will be far away from the estimated support value, and its effect on the estimated sinogram values near the boundary will be reduced (see Fig. 7.3). Suppose $h(\theta)$ is the estimated support function and $\xi(\theta)$ is the estimated error variance of the support estimate at angle θ. We define this modified *segmentation* of y_T (which is *not* our best estimate of the support of the

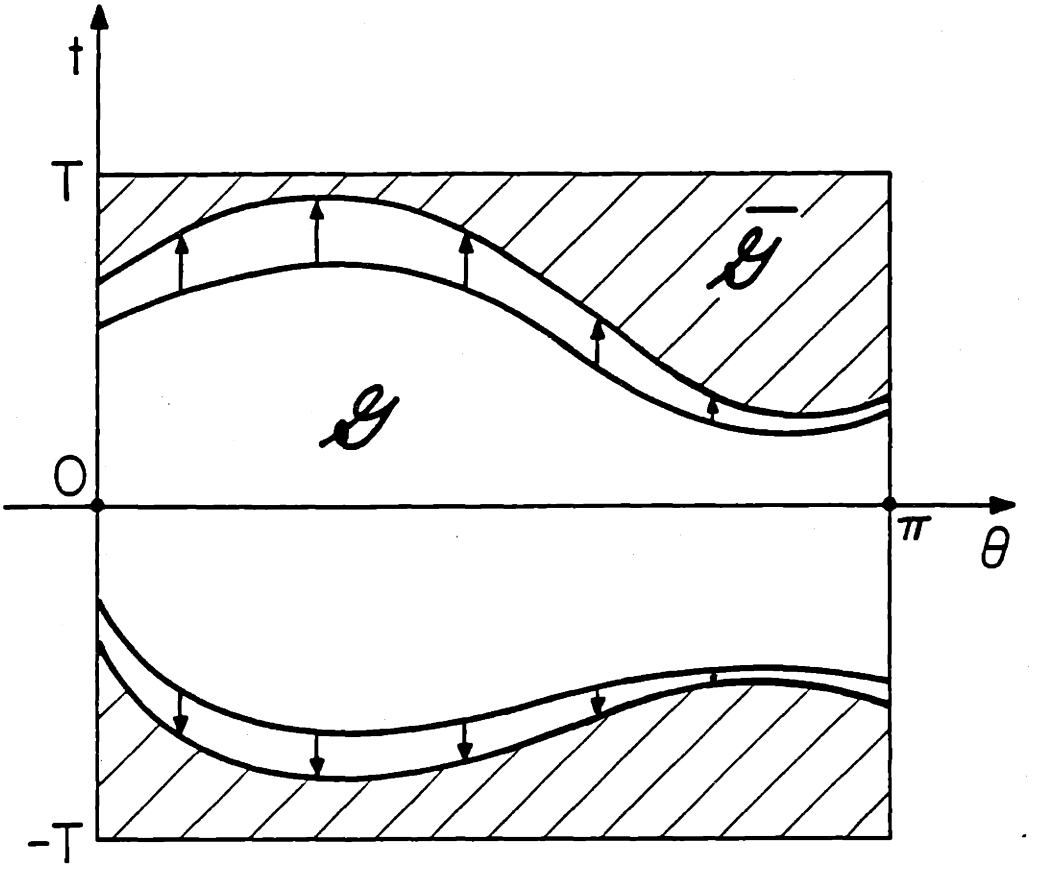


Figure 7.3: Adjusted support segmentation method.

Radon transform) to be

$$\mathcal{G} = \{(t,\theta) \in \mathcal{Y}_T \mid h(\theta) + \alpha\sqrt{\xi(\theta)} \leq t \leq -h(\theta+\pi) - \alpha\sqrt{\xi(\theta+\pi)}\}. \qquad (7.15)$$

We see that α specifies the number of standard deviations away from the estimated object boundary to begin the segmentation boundary in the sinogram domain. We show several experiments in Section 7.5 for various α's and SNR's.

7.5 Experimental Results

In this section we present results which show the overall behavior of the full hierarchical algorithm. The first set of experiments show the behavior of the Knot-location and Closest estimate algorithms for several full-view sinograms of different noise levels. Then we show the performance of the segmentation performed by these two algorithms in tandem on several several limited- and sparse-angle cases and show the resultant full hierarchical reconstructions of those objects as well. Finally, we introduce an object that consists of two disks with a gap in between them, and show the results of the full hierarchical algorithm for two limited-angle studies.

7.5.1 Full-View Segmentation

Fig. 7.4 shows in each panel a noisy sinogram together with the full set of Knot-location support value estimates (shown using thin white curves) and the Closest support vector estimate (shown using thick white curves). The SNR of the sinogram in each panel is given by (a) 100.0dB, (b) 10.0dB, (c) 3.0dB, and (d) 0.0dB. In Fig. 7.4a, the experiment in which the data has the highest SNR, the Knot-location and Closest estimates do not correspond as well as might be expected to the position of the true boundary. This is due to an effect we noticed in the results of Chapter 6, where we observed that the Knot-location algorithm tends to detect the position of an edge too soon in the case of high SNR data. We concluded in Chapter 6 that this is an effect due to the threshold selection algorithm, and may be improved upon by better algorithm selection. The final segmentation in Fig. 7.4b (indicated by the thick white lines), however, closely matches the true sinogram support, and shows a

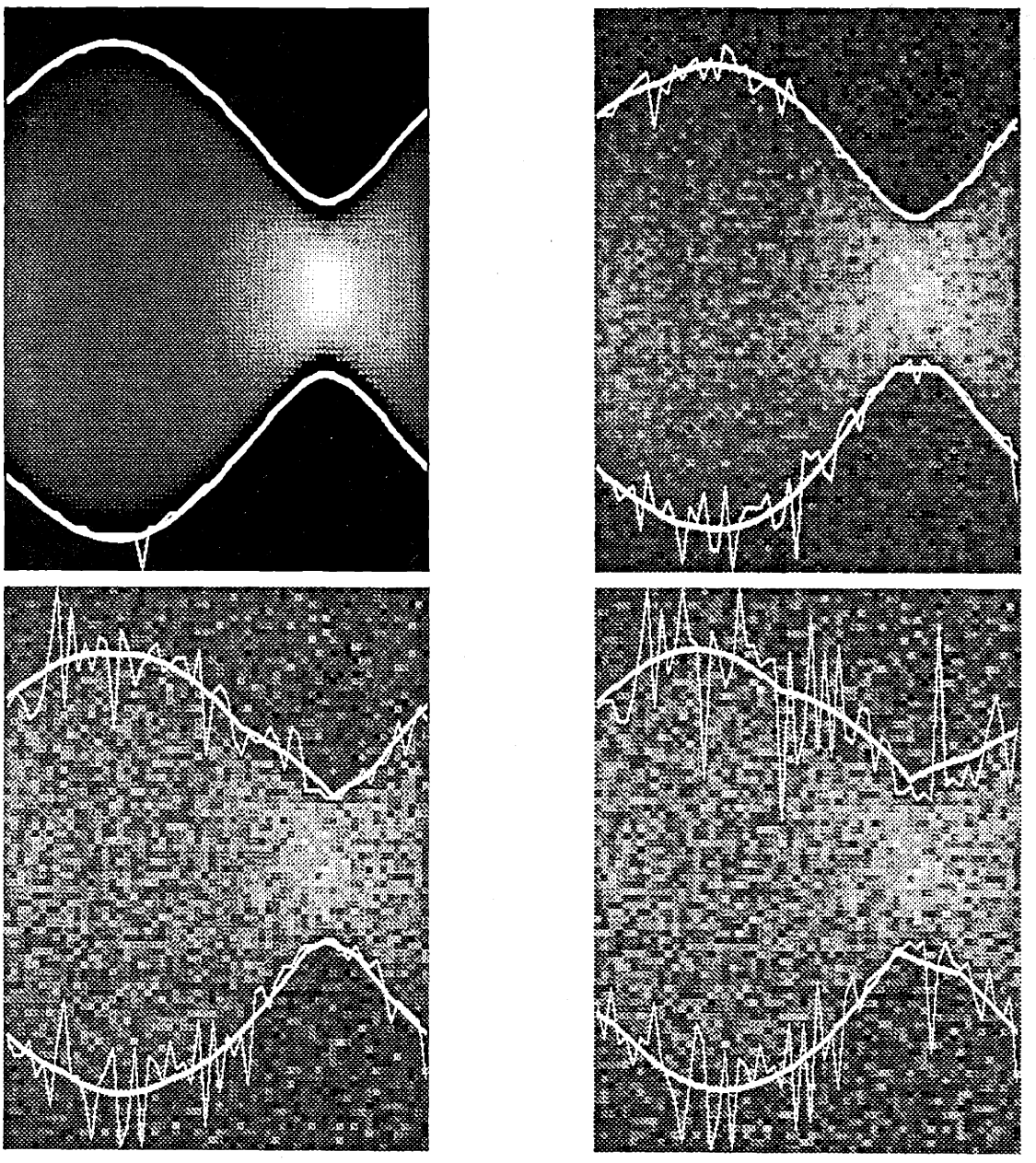


Figure 7.4: Knot-location followed by Closest support vector estimation for (a) 100.0dB, (b) 10.0dB, (c) 3.0dB, and (d) 0.0dB.

clear improvement over the Knot-location algorithm used alone. As the noise level increases in Fig. 7.4c and again in Fig. 7.4d, the performance of the Knot-location algorithm gets noticeably worse but, because of the Closest support vector algorithm — which uses the fundamental support vector constraint developed in Chapter 4 — the final segmentation does not reflect the same qualitative degradation.

The results we present in Fig. 7.4 are equivalent in many ways to the results presented for the Closest algorithm in Chapter 4 — they show support vector estimation from sets of noisy support vector observations. However, in these experiments the focus is very much on *sinogram segmentation* rather than estimation of convex sets (even though these ideas are equivalent). In particular, the noise which gives rise to the lateral displacement error (of the support line measurements) is not generated by a known probability distribution, but is actually generated by the combined effect of the additive sinogram noise and the Knot-location algorithm itself. Also, the number of support line measurements is 120 (twice the number of views) and is, therefore, yields a larger problem than any of the examples of Chapters 4 and 5. The issues of feasibility and prior shape knowledge are identical, however, and we may solve the second stage of the segmentation problem — support vector estimation — using exactly the methods formulated in Chapters 4 and 5.

7.5.2 Utilization of Support Information

As suggested in Section 7.4.3, we use the support estimation information in one of two different ways: 1) use the feasible support vector estimate as the final segmentation or 2) adjust each feasible support value by adding some multiple of the estimated error standard deviation. In Fig. 7.5a we show the sinogram estimate obtained from the LR algorithm using the full noisy (3.0dB) sinogram, $\gamma = 0.05$, $\beta = 0.01$, $\kappa = 5.0$, and the final segmentation shown in Fig. 7.4c (superposed on the figure). The corresponding reconstruction using CBP is shown in Fig. 7.5b. Fig. 7.5c shows the sinogram estimate obtained using the same data and parameters, except that $\kappa = 10000$ and the segmentation (superposed on the figure) has been adjusted outward (toward the $\pm T$ bounds) by one standard deviation as de-

scribed in Section 7.4.3. The corresponding reconstruction using CBP is shown in Fig. 7.5d.

The reconstructions shown in Fig. 7.5 (panels (b) and (d)) show three major differences — although even these differences are subtle in these examples. First, the object in (d) has greater contrast between the object and its background than in (b); this primarily reflects the fact that κ (the LR support coefficient) is much larger in (d) than in (b). Second, there is less contrast in the *interior* of the ellipse in panel (d) than in (b). Third, there is a slight two-tiered effect on certain parts of the boundary of the object in (d), where the effects of two boundaries — the segmentation boundary and the true boundary — are apparent. All three of these effects were observed in Chapter 3, Sections 3.6.3–4, and may be viewed as fundamental properties of the algorithm. In particular, when κ is very large (e.g. 10000) one can expect an increase in overall contrast between the object and its background, but this is at the expense of loss of contrast in the interior.

What one might risk when using the original feasible support for the segmentation (rather than the adjusted support) is the possibility that the support estimate will be too far in, and thereby cut off part of the object in the reconstruction. But it appears that as long as κ is small enough (e.g. 5.0), there is some increase in overall contrast and little of the two-tiered effects, or loss of interior contrast. Therefore, in the results that follow — which include limited-angle and sparse-angle cases as well as results for a different object — we do not adjust the support, and we use $\kappa = 5.0$.

7.5.3 Limited- and Sparse-Angle Cases

In the situations involving limited-data, some type of prior shape knowledge must be utilized in order to estimate support values for the missing projections. In the experiments in this section we use the Knot-location support value estimation algorithm followed by the Scale-Invariant Maximum Area (SIMA) support vector estimation algorithm to determine the segmentation of the sinogram. Also, the LR parameters that we use are $\gamma = 0.05$, $\beta = 0.005$, values that were shown to yield good results in Chapter 3.

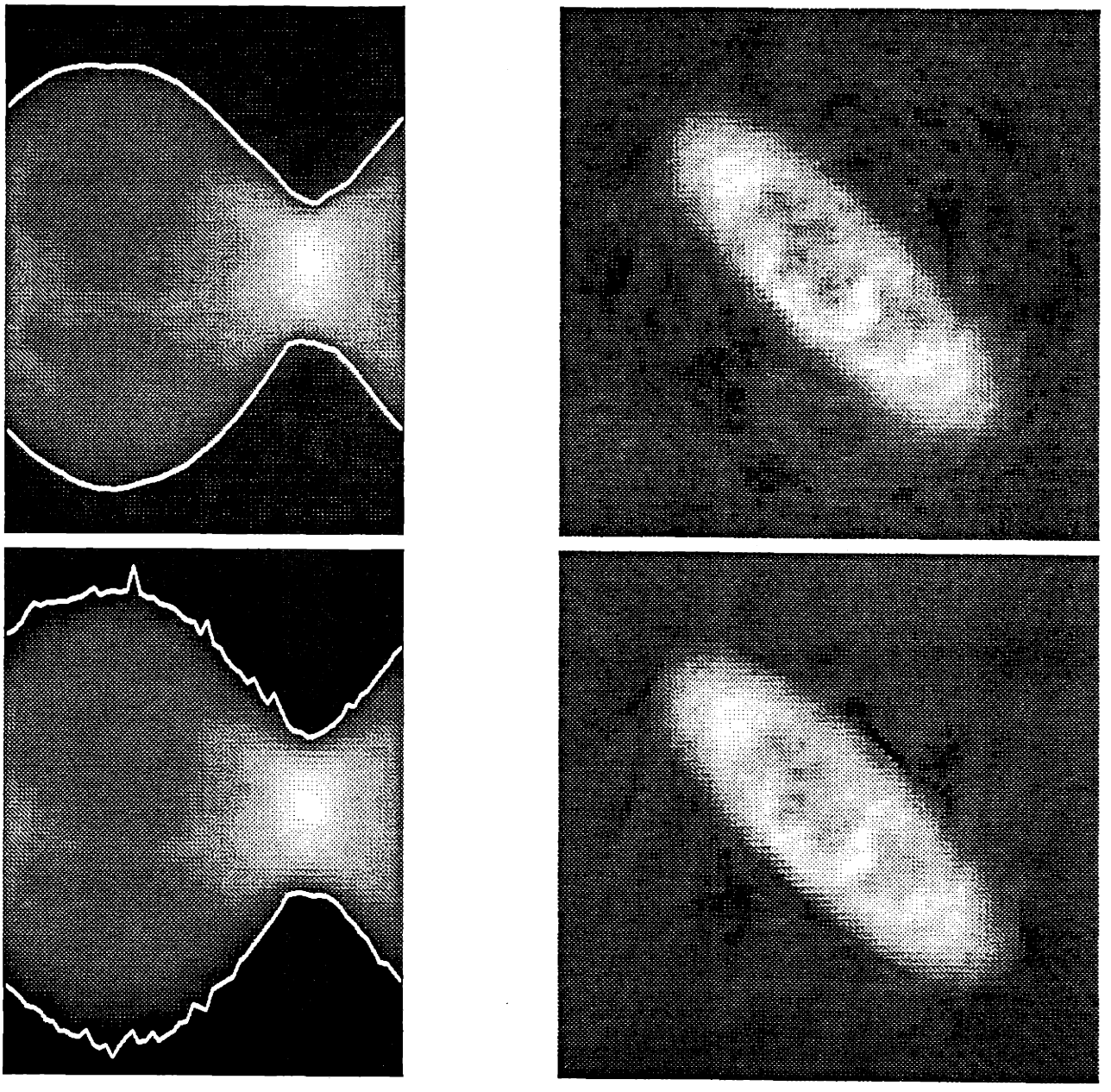


Figure 7.5: Full hierarchical sinogram estimates and object reconstructions for two segmentations.

Fig. 7.6 shows the segmentation and estimated sinograms for the 10.0dB M I T ellipse sinogram with limited- and sparse-angle observations. The top two panels show the two limited-angle cases introduced in Chapter 3: in Fig. 7.6a, the left 40 projections are observed, and in Fig. 7.6b the right 40 projections are observed. The bottom two panels show the two sparse angle cases: Fig. 7.6c used 15 projections and Fig. 7.6d used only 10 projections. The CBP reconstructions for these cases are shown in the respective panels of Fig. 7.7.

We recall that although fewer projections are observed than are actually in the sinogram, the support estimation procedure obtains estimates for *all* of the projections. The estimates obtained for the missing projections — these may be thought of as *interpolated* support values — strongly rely on the prior knowledge provided by the SIMA prior, which we have examined in great detail in Chapter 5. Here, as in Chapter 5, when we do not observe the narrow view of the ellipse (Fig. 7.6a) , the support estimate is not as good because the interpolation (done automatically by the SIMA algorithm) uses the prior knowledge that the objects are more likely to be circular. As we shall see in Section 7.5.5, the Ellipse-Based algorithms developed in Chapter 5 yield better results in this case. In both sparse-angle cases, the support estimation is quite good, and the corresponding reconstructions are also quite good.

The results of these simulations should be compared to the (unprocessed) CBP reconstructions shown in Fig. 3.5 and to the LR processed reconstructions for perfectly known and unknown support shown in Figs. 3.12 and 3.13. We see that the results presented here are *better* in each case than the corresponding results in Chapter 3. We may therefore conclude that *some* support information is better than none, *even if it is slightly in error* (as is true in the results of this Chapter). Also, we may conclude, as we saw in Chapter 3 and in a previous result in this chapter, that a modest sized κ leads to an improved balance between overall contrast and the internal contrast.

7.5.4 Two-Disk Object

Up to this point in the thesis, we have exclusively used the M I T ellipse (or the same ellipse with the internal letters absent) for our sinogram MAP estimation studies.

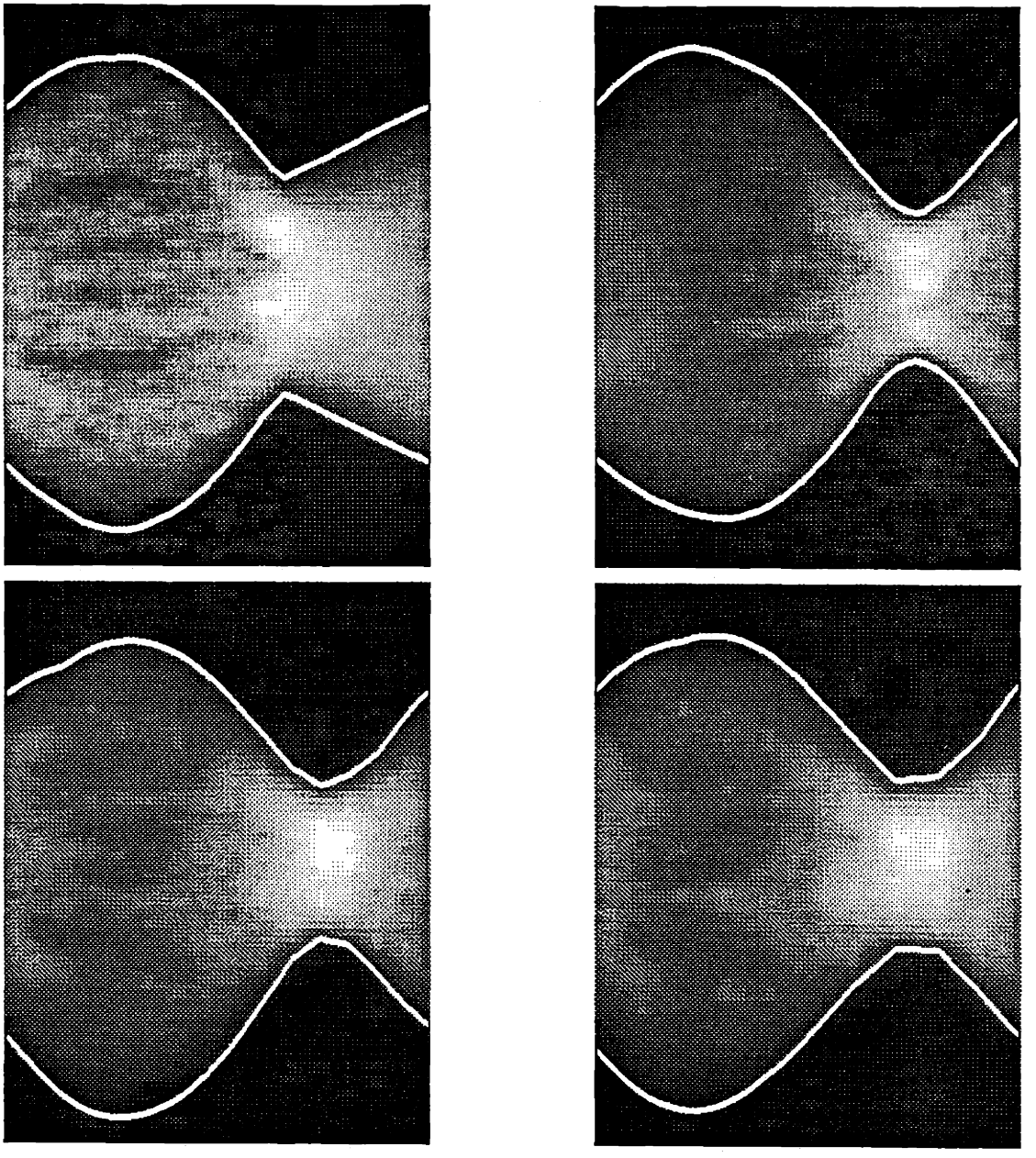


Figure 7.6: Estimates from 10.0dB M I T sinogram using observations of (a) the left 40 projections, (b) the right 40 projections, (c) 15 evenly spaced projections, and (d) 10 evenly spaced projections.

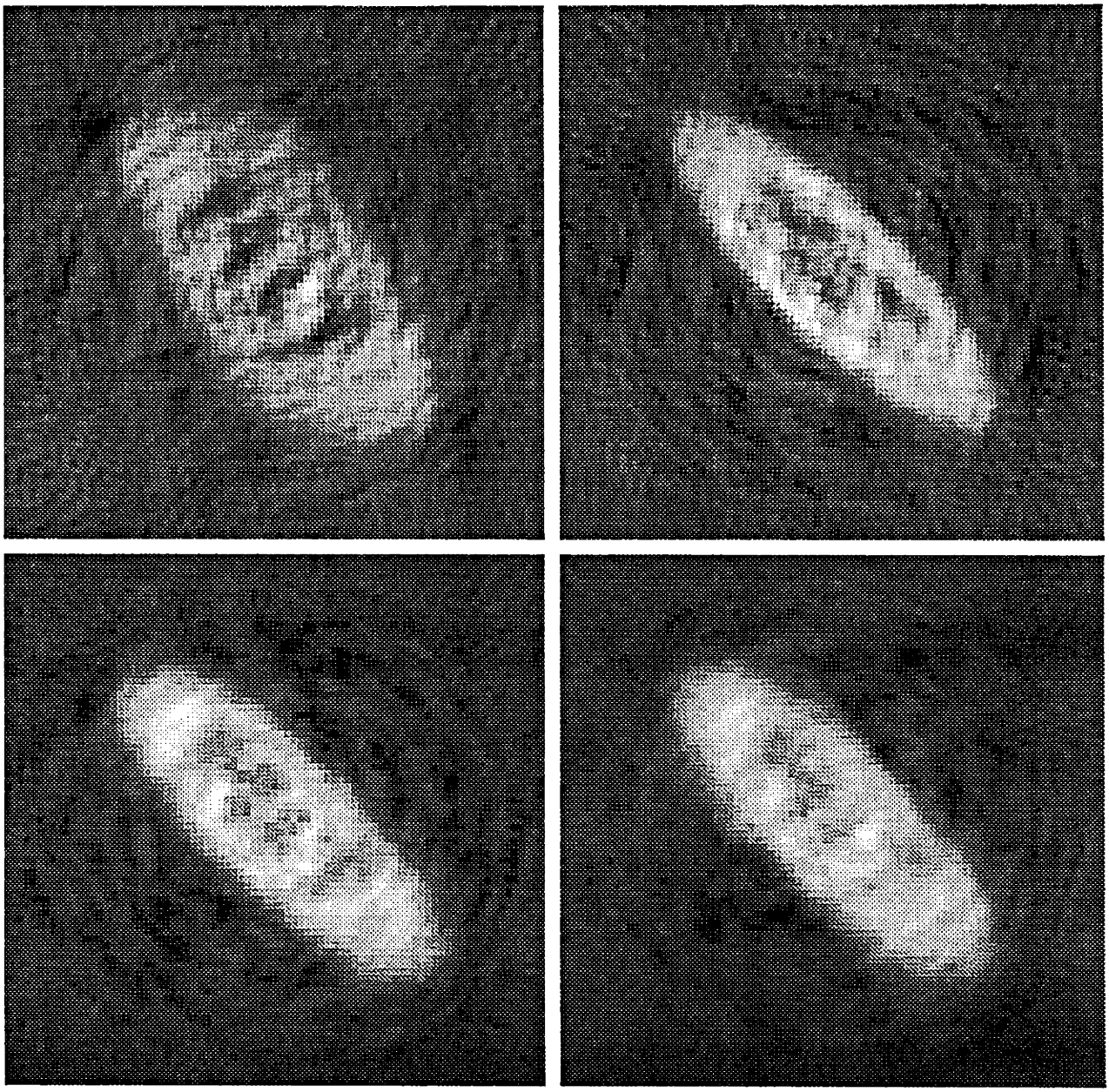


Figure 7.7: Reconstructions using CBP from respective panels in Fig. 7.6.

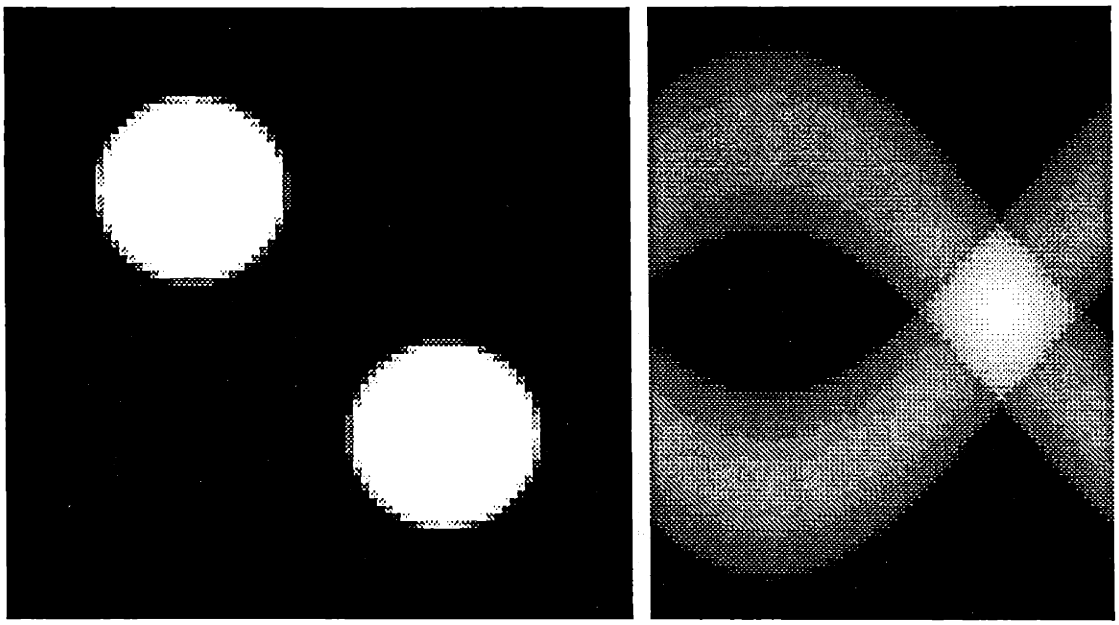


Figure 7.8: Two disk object and noise-free sinogram.

In this set of experiments we introduce a new object which consists of two isolated disks rotated off the vertical axis by 45.0 degrees as shown in Fig. 7.8a. Fig. 7.8b shows the noise-free sinogram corresponding to this object. In several ways, this figure is similar to the M I T ellipse: it is oriented in a similar manner and it may be viewed broadside (where in this case it allows us to see that there are two distinct objects — although we do not explicitly use this fact in our processing) and head-on. The results that are shown in this section result from the full hierarchical estimation algorithm using the Knot-location support value estimation procedure, the SIMA support vector algorithm, and the LR algorithm with $\kappa = 5.0$, $\gamma = 0.05$, and $\beta = 0.005$.

Fig. 7.9 shows the results of CBP applied directly to the (limited- and sparse-angle) measurements (SNR=10.0dB) in the two panels on the left, and the respective full hierarchical processed reconstructions on the right. The top two panels correspond to the limited-angle situation in which the left 40 projections are observed; the bottom two panels used only the right 40 projections. We have also superposed

the SIMA estimates of the convex support using white curves on the right-hand reconstructions.

As in the limited-angle studies with the M I T ellipse, the performance of the algorithm differs substantially between the two different limited-angle cases. In particular, where the majority of the views are obtained from the left side of the sinogram (top two panels in Fig. 7.9), the narrow dimension of the two disks is not observed, and the resultant support estimate is too wide. Once again, this reflects the fact that the most likely shapes (using the SIMA prior) are circular. The unfortunate consequence of this property is to produce a narrow band (or swath) of energy in the reconstruction, which follows just inside the estimated boundary (see Fig. 7.9b). It is an artifact that can just barely be discerned in the direct CBP reconstruction shown in Fig. 7.9a, so that it definitely results from our enhancement efforts. The reason for this bright band lies primarily in the mass constraint, which together with the support constraint, causes most of the mass to be in the region of estimated support, and also because of the horizontal smoothing coefficient, which causes rotation swirling.

This swirling effect is also noticeable in Fig. 7.9d, which shows the processed reconstruction for the case in which the right 40 projections are observed. This phenomena is at a low energy level in this panel, however, and does not take away from the fact that this reconstruction is significantly better than the direct CBP reconstruction shown in Fig. 7.9c. The processed reconstruction shows an excellent convex support estimate, which largely reflects the fact that the missing projections correspond to the broadside views, and their support values have a circular curvature at this point. The only artifact which is present in this reconstruction is a very slight tendency for the disks to be elongated towards each other in the center. This arises from the vertical smoothing and the fact that many of the projections from the broadside views of the two disks were missing, and therefore had to be interpolated.

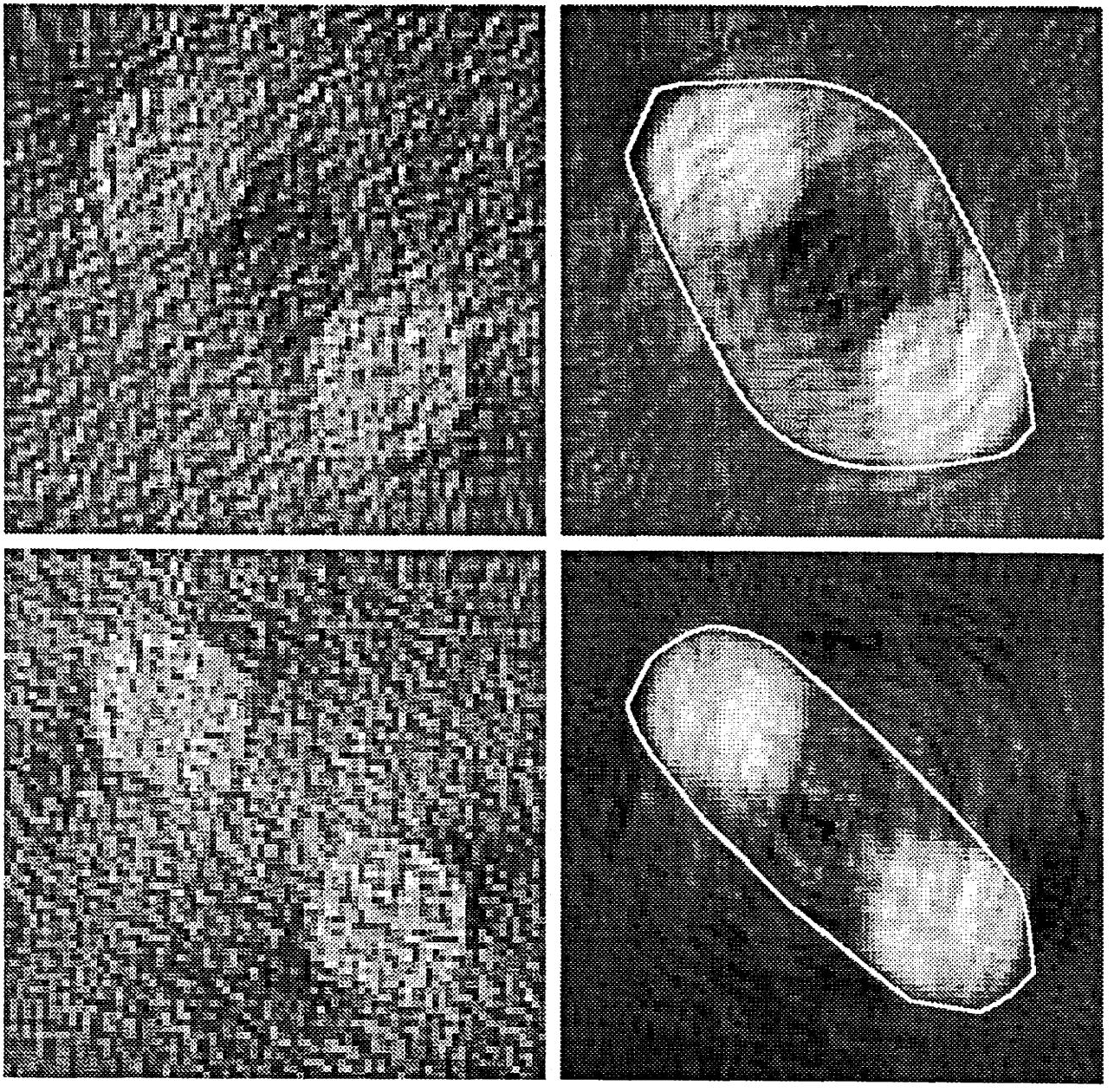


Figure 7.9: Two limited-angle cases showing CBP reconstruction (left) and full hierarchical processing followed by CBP (right).

7.5.5 Ellipse-Based Support Estimation

We have now seen two examples of limited-angle observations in which the narrow views were missing, and this resulted in poor support estimation and correspondingly poor reconstructions. Fig. 7.10 shows several results in which the Ellipse-Based support vector estimation of Chapter 5, Section 5.4 are used instead of the SIMA algorithm to estimate the sinogram support. In this figure, the top two panels are reconstructions of the M I T ellipse, from the left 40 projections of the 10.0dB SNR noisy sinogram — these are the same observations which led to Figs. 7.6a and 7.7a. The bottom two figures are reconstructions of the two-disk object used in the previous section. The observations used are the left 40 projections of the 10.0dB SNR noisy sinogram used in producing the results in Fig. 7.9, panels (a) and (b). The left panels of Fig. 7.10 used the JE support vector estimation algorithm with $\alpha = 0.5$, while the right panels use the ESIC support vector estimation algorithm with $\bar{\varepsilon} = 0.9$ and $\bar{\phi} = -45.0°$. (Note that the true value of $\bar{\varepsilon}$ for the M I T ellipse is 0.95, so that our input slightly underestimates the true eccentricity, in this case.) The support estimates are superposed over each reconstruction using white curves.

Each of these reconstructions is an improvement over the former efforts in this chapter. In particular, the width of the support estimates are much narrower than those obtained using the SIMA algorithm in the previous examples, and this has the effect of concentrating most of the energy within a region that more closely approximates the true region of convex support. Also, since the available views of the M I T ellipse and the two-disk object allow us to see the details of, respectively, the internal letters and the gap between the two objects, the clarity of these features remains good in these reconstructions. This demonstration is perhaps the clearest example in this thesis of how support information — estimated in a hierarchical fashion — can lead to very much improved reconstructions over conventional methods.

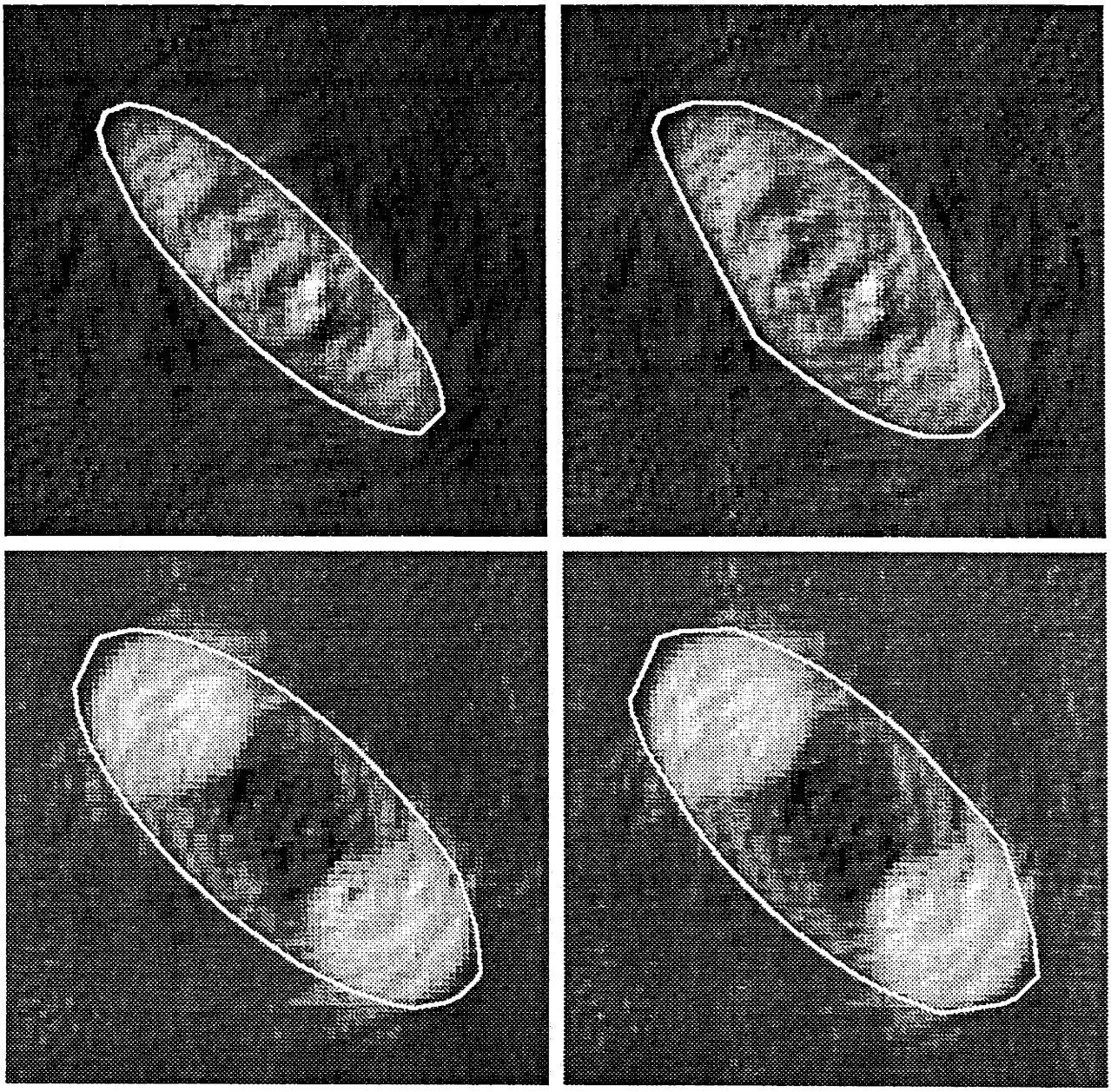


Figure 7.10: Full hierarchical results for limited-angle studies using the JE support estimation algorithm on the left and the ESIC support estimaion algorithm on the right.

7.6 Discussion

The ideas developed in this chapter relate directly to the task of combining the various algorithms that we have developed in the preceding chapters into one hierarchical reconstruction algorithm. The result of this combination is an algorithm which attempts first to determine some geometric features of the object — in particular, the mass, center of mass, and convex support — and along with these, some indication as to the reliability of these estimates. In the case of the mass and center of mass, the algorithm assumes implicitly — since it uses this information to constrain the feasible sinograms — that these quantities are estimated perfectly. In the case of the convex support of the object, the algorithm uses an on-line estimate of the error variance in one of two ways to affect the influence that this estimate has on the subsequent stages in the hierarchy. Other than the structure of the algorithm itself, the only explicit prior information that is used in these stages of the algorithm is some knowledge of the *shape* of the convex hull of the object. This information is given by the choice of the prior probability on support vector, and by the constant τ (or $\bar{\tau}$ where appropriate), which is a fundamental parameter of many of our support vector priors.

The next stage of the algorithm consists of the LR algorithm of Chapter 3, which determines an MAP estimate of the full sinogram, subject to the mass and center of mass constraints, and with knowledge of the support information determined in the earlier stages of the hierarchical algorithm. The only explicit prior knowledge required at this stage is the smoothing coefficients γ and β.

The results contained in Section 7.5 indicate that the effect of smoothing the sinogram together with forcing sinogram values to be small outside the estimated region of support can significantly improve reconstructions over conventional CBP applied directly to the data. An additional benefit one obtains when using this approach is that the procedure not only produces an estimate of the object, but also of its mass, center of mass, and convex hull. In Chapters 3 and 7 we did not show an experiment which shows the advantage of constraining the sinogram estimates using the mass and center of mass estimates. We have reserved this demonstration

until Chapter 8, where we discuss constraints in a more general context, and give a specific result that shows quite dramatically the differences between constrained and unconstrained estimates.

Chapter 8

CONSTRAINT-BASED RECONSTRUCTION

8.1 Overview

We have shown in previous chapters that our sinogram MAP method is an effective way to produce high-quality reconstructions from noisy and incomplete projections. As developed in Chapter 3, this method uses a nearest-neighbor MRF prior on sinograms, utilizes support information, and incorporates the mass and center of mass constraints which we presented in Chapter 2. Our most successful optimization algorithm to actually compute the MAP estimate is the local relaxation (LR) method based on the variational formulation of Section 3.5. In this chapter we develop an extension to this method which accounts for *more* of the consistency constraints inherent to the Radon transform (see Section 2.6).

The key motivation for this chapter is the observation that the mass and center of mass constraints are but *two* of an infinite set of constraints given by the Consistency Theorem of (2.1) and (2.2), and that superior results should be obtained if more of these constraints are utilized. One approach that has been used by several investigators is to estimate the higher *free* moments — those analogous to mass and center of mass — from the available data, and to reconstruct the missing projections using these estimates [21,53,76]. We present a brief summary of these methods in Section 7.3. In some ways, our approach in Chapter 3 is similar to these methods

except that we only use two moment estimates instead of several hundred, and the remaining moments are determined (indirectly) by the MAP method using a specific prior probability. One can imagine an extension to the methods of Chapter 3 which uses the MAP formulation with more of these higher moment estimates. However, the approach we present in Section 7.4 focuses not on the higher *free moments*, but on those moments which must be identically *zero* (refer to part (d) in the statement of the Consistency Theorem). Thus, the method presented in this chapter seeks a sinogram MAP estimate while imposing the (in principle infinite set of) constraints which force certain higher moments to be *zero*. The sinogram itself is uniquely selected by the MAP method, and the coefficients of the free moments are never explicitly estimated. The resulting sinogram is, therefore, a *consistent* Radon transform and will produce (using CBP) an artifact-free reconstructed object.

This chapter is organized as follows. In Section 8.2 we review the consistency conditions and motivate and develop the use of the orthonormal Legendre polynomials in the statement of the constraints. In Section 8.3 we review the methods of previous researchers used to estimate the free moments and to estimate the missing projections. Section 8.4 develops the constraint-based methods which constitute the main results of this chapter. Section 8.5 presents some experimental results, and Section 8.6 discusses these results and possible modifications to the proposed algorithm.

8.2 Generalized Fourier Coefficients

As suggested in Section 2.6, it is convenient to use Legendre polynomials in the statement of the consistency conditions for the Radon transform. For convenience, we will assume in this chapter that $T = 1$ and, hence, that the object is contained entirely in the unit circle and the Radon transform $g(t, \theta)$ is (possibly) non-zero only when $-1 \leq t \leq 1$. We denote by $P_k(t)$ the Legendre polynomial of degree k orthonormal over the interval $[-1, 1]$ (cf. [35]). Therefore,

$$\int_{-1}^{1} P_k(t) P_l(t)\, dt = \delta_{kl} \tag{8.1}$$

where δ_{kl} denotes the Kronecker delta function — it has value 1 when $k = l$ and otherwise is zero. The complete set of Legendre polynomials form a complete orthonormal (CON) set of functions on the interval $[-1, 1]$, and the set of Legendre polynomials of degree $\leq k$ form a basis for the space of polynomial functions of degree $\leq k$. Another property of the Legendre polynomials which we shall use is the following symmetry property:

$$P_k(-t) = (-1)^k P_k(t)\,. \tag{8.2}$$

Given the preceding comments it is easily shown that the following condition is equivalent to condition (d) of the Consistency Theorem (Theorem 2.1): If $S_l(\omega)$ is a spherical harmonic of degree l, and if $k < l$ then

$$\int_{|\omega|=1} \int_{-1}^{1} g(t,\omega) P_k(t) S_l(\omega)\, dt d\omega = 0\,. \tag{8.3}$$

Now, as pointed out in Section 2.6, we observe that the product $P_k(t)S_l(\omega)$ for all $k \geq 0$ and $l \geq 0$ form a CON set of functions over $\mathbb{R}^1 \times S^{n-1}$. Therefore, the above condition requires that a certain infinite set of generalized Fourier coefficients — given by the expression on the left-hand side of (8.3) — be identically zero. The remaining coefficients, which correspond to the extension of the mass and center of mass, are free to take on any real value.

It turns out that further constraints on the Fourier coefficients may be found by incorporating the symmetry of the Radon transform given in condition (b) of the Consistency Theorem. For convenience, we denote

$$I_k(\omega) = \int_{-1}^{1} g(t,\omega) P_k(t)\, dt\,. \tag{8.4}$$

From (8.3) we may conclude that $I_k(\omega)$ is a polynomial in ω of degree $\leq k$, and hence may be expanded in orthonormal spherical harmonics as follows

$$I_k(\omega) = \sum_{l=0}^{k} \sum_{m=1}^{N(n,l)} a_{lm}^k S_{lm}(\omega)\,, \tag{8.5}$$

where m indexes the spherical harmonics of degree l, of which there are $N(n,l)$ in $\mathbb{R}^n$ (see Appendix 2.A). We note that in $\mathbb{R}^2$ there are two spherical harmonics of

any degree l — i.e., $N(2,l) = 2$ — and they are given by $S_{l1} = (1/\sqrt{\pi})\cos l\theta$ and $S_{l2} = (1/\sqrt{\pi})\sin l\theta$. It is important to observe that the coefficients a_{lm}^k in (8.5) are precisely the generalized Fourier coefficients from the left-hand side of (8.3). To see this we merely multiply both sides of (8.5) by $S_{ij}(\omega)$ and integrate over ω as follows:

$$\int_{|\omega|=1} I_k(\omega) S_{ij}(\omega)\, d\omega = \int_{|\omega|=1} \sum_{l=0}^{k} \sum_{m=1}^{N(n,l)} a_{lm}^k S_{lm}(\omega) S_{ij}(\omega)\, d\omega\,.$$

Then, when we substitute (8.4) for $I_k(\omega)$ and use the orthogonality of the spherical harmonics, this becomes

$$\int_{|\omega|=1} \int_{-1}^{1} g(t,\omega) P_k(t) S_{ij}(\omega)\, dt d\omega = a_{ij}^k\,, \tag{8.6}$$

which is the desired result.

The additional constraint is now revealed by making the following manipulations

$$\begin{aligned} I_k(\omega) &= \int_{-\infty}^{\infty} g(t,\omega) P_k(t)\, dt \\ &= \int_{-\infty}^{\infty} g(-t,-\omega) P_k(t)\, dt \\ &= (-1)^k \int_{-\infty}^{\infty} g(t,-\omega) P_k(t)\, dt \\ &= (-1)^k I_k(-\omega)\ , \end{aligned} \tag{8.7}$$

from which we may conclude that

$$I_k(\omega) = \sum_{l=0}^{k} \sum_{m=1}^{N(n,l)} a_{lm}^k (-1)^{l+k} S_{lm}(\omega)\,. \tag{8.8}$$

One can now see, by comparing (8.5) with (8.8), that it must be true that $a_{lm}^k = 0$ for $l + k$ odd. Therefore, for $k > l$ — where the Fourier coefficients are *not* constrained by (8.3) to be zero — some coefficients are nevertheless forced to be zero by the above symmetry condition.

8.3 Estimating the Free Fourier Coefficients

Given the discussion of the previous section, we see that one possible approach to the sinogram estimation problem is to estimate from the available data those Fourier

coefficients a_{lm}^k that are not otherwise constrained to be zero — we call these the *free Fourier coefficients.* Then, denoting these estimates as $\hat{a}_{lm}^k$, we may calculate any missing projection — or the estimates of the observed projections themselves if they were noisy — using the expression

$$\hat{g}(t,\omega) = \sum_{k=0}^{K} \sum_{\substack{l=0 \\ k+l \text{ even}}}^{k} \sum_{m=1}^{N(n,l)} \hat{a}_{lm}^k P_k(t) S_{lm}(\omega) \tag{8.9}$$

where K indicates the highest order Legendre polynomial that was computed for each observed projection. This is essentially the method used by Louis [53] and by Ein-Gal [21]. We give a brief summary of their methods in this section.

Let us restrict our attention to the 2-D case, and suppose that we have J observed projections obtained from distinct angles. Then for each observed projection $y(t,\theta_j)$ we calculate

$$\begin{aligned} \tilde{I}_k(\theta_j) &= \int_{-1}^{1} y(t,\theta_j) P_k(t)\, dt \\ &= \int_{-1}^{1} g(t,\theta_j) P_k(t)\, dt + e_k(\theta_j)\,, \end{aligned}$$

for $0 \le k \le K$ where $e_k(\theta_j)$ is an error term which arises from noise in the measurements. From the discussion in the previous section, we know that $I_k(\omega_j)$ can be written as

$$I_k(\omega_j) = \sum_{\substack{l=0 \\ k+l \text{ even}}}^{k} \sum_{m=1}^{N(n,l)} a_{lm}^k S_{lm}(\omega_j)\,, \tag{8.10}$$

which for the 2-D case may be written as

$$I_k(\theta_j) = \sum_{\substack{l=0 \\ k+l \text{ even}}}^{k} \left(a_{l1}^k \frac{\cos l\theta_j}{\sqrt{\pi}} + a_{l2}^k \frac{\sin l\theta_j}{\sqrt{\pi}} \right). \tag{8.11}$$

Now we observe that for a particular k, there are exactly $2\lfloor k/2 \rfloor + 2$ unknown Fourier coefficients — i.e., the coefficients a_{lm}^k — in the summation of (8.11), where $\lfloor x \rfloor$ denotes the integral part of x. Furthermore, we also have J (noisy) observations of $I_k(\theta_j)$, given by the $\tilde{I}_k(\theta_j)$ of (8.10). Therefore, for those values of k that satisfy $2\lfloor k/2 \rfloor + 2 < J$, the system of equations in (8.11) is overdetermined, and the set of coefficients $\{a_{lm}^k \mid m = 1,2;\ l = 1,\ldots,k\}$ may be estimated using least squares.

8.4 Constraint-Based Reconstruction Algorithm

In this section we present a formulation similar to the variational approach of Section 3.5, except that the feasible sinograms are constrained more tightly using the full set of Fourier constraints that were presented in the Section 8.2. The solution is found using steps similar to those given in Appendix 3.A, and the computations are made using the same *generic primal-dual* algorithm of Section 3.5, which alternates between an LR stage and a Lagrange multiplier update stage. This approach, however, has *many* Lagrange multipliers and therefore poses a much greater computational burden than the original method of Chapter 3.

Following very closely the methods of Section 3.5.1, we first state the *Constraint-Based* variational problem, which we will refer to as (C), is to minimize

$$I = \iint_{y_o} \frac{1}{2\sigma^2}(y-g)^2\,dt\,d\theta + \iint_{\mathcal{G}} \kappa g^2\,dt\,d\theta + \iint_{y_T} \left[\beta\left(\frac{\partial g}{\partial t}\right)^2 + \gamma\left(\frac{\partial g}{\partial \theta}\right)^2\right] dt\,d\theta \tag{8.12}$$

where κ, β, and γ are positive constants and $T = 1$, subject to the equality constraints

$$J_{lm}^k = \int_0^{2\pi}\int_{-1}^{1} P_k(t)S_{lm}(\theta)\,dt\,d\theta = 0\,, \tag{8.13}$$

for $m = 1,2$ and for $k,l = 0,1,\ldots$ where $k < l$. The boundary conditions are given by

$$\begin{aligned} g(\mathrm{T},\theta) &= g(-\mathrm{T},\theta) = 0 \\ g(t,0) &= g(-t,\pi) \end{aligned} \tag{8.14}$$

There is a problem with the above statement of (C), however, since the constraints J_{lm}^k contain an integration over the wrong domain. (Recall that the sinogram domain does not contain the angular range $(\pi,2\pi]$.) We may correct this problem, fortunately, using the symmetry relations of $S_{lm}(\theta)$ and $g(t,\theta)$. First, we expand the expression (8.13) as follows:

$$J_{lm}^k = \int_0^{\pi}\int_{-1}^{1} g(t,\theta)P_k(t)S_{lm}(\theta)\,dt d\theta + \int_0^{\pi}\int_{-1}^{1} g(t,\theta+\pi)P_k(t)S_{lm}(\theta+\pi)\,dt d\theta\,.$$

Now we use the fact that $g(t,\theta+\pi) = g(-t,\theta)$ to manipulate the second term, so that we may recombine the two terms as

$$J_{lm}^{k} = \int_0^{\pi}\int_{-1}^{1} g(t,\theta)\left[S_{lm}(\theta)P_k(t) + S_{lm}(\theta+\pi)P_k(-t)\right]\,dtd\theta\,. \tag{8.15}$$

This expression may be simplified further using the symmetry relations $S_{lm}(\theta+\pi) = (-1)^l S_{lm}(\theta)$ and $P_k(-t) = (-1)^k P_k(t)$, from which we obtain the correctly stated constraint on the sinogram domain

$$J_{lm}^{k} = \int_0^{\pi}\int_{-1}^{1} g(t,\theta)S_{lm}(\theta)P_k(t)\left[1+(-1)^{k+l}\right]\,dtd\theta\,. \tag{8.16}$$

It is important to note from (8.16) that when $k+l$ is odd, J_{lm}^{k} is identically zero, and therefore it is not necessary to impose these constraints explicitly. This simply reflects the fact that once we obtain a valid sinogram, the full Radon transform is constructed by periodically replicating the sinogram (in the twisted fashion required by condition (b) of the Consistency Theorem). This causes all the J_{lm}^{k} constraints in which $k+l$ is odd to be trivially met. Therefore, the constraints that we must impose over the sinogram domain are

$$J_{lm}^{k} = \int_0^{\pi}\int_{-1}^{1} g(t,\theta)S_{lm}(\theta)P_k(t)\,dt\,d\theta = 0\,, \tag{8.17}$$

for $m = 1,2$ and $k,l = 0,1,\ldots$ where $k < l$ *and* $k+l$ *is even.*

Now that the variational problem is properly stated we turn our attention to finding a solution. To solve (C) we follow the variational technique used in Appendix 3.A where the only significant difference is that the Lagrange multipliers are scalars here and that there are an infinite number of them. The resulting Euler-Lagrange equation is

$$\left(2\kappa\bar{X}_G + \frac{1}{\sigma^2}\chi_Y\right)g - 2\beta\frac{\partial^2 g}{\partial t^2} - 2\gamma\frac{\partial^2 g}{\partial\theta^2} = \frac{1}{\sigma^2}\chi_Y y - \sum_i \lambda_i\Psi_i\,, \tag{8.18}$$

where the index i denotes the triplet $i = (k,l,m)$ of indices used in the statement of the problem and $\Psi_i = P_k(t)S_{lm}(\theta)$. The solution, as in Chapter 3, must also satisfy the original constraints and boundary conditions and the additional boundary condition

$$\frac{\partial g(t,0)}{\partial t} = \frac{\partial g(-t,\pi)}{\partial t}\,. \tag{8.19}$$

Since the $\{\Psi_i\}$ are orthogonal[1] over $\mathcal{Y}_T$, all that is required to solve for a given Lagrange multiplier λ_j is to multiply both sides of (8.18) by Ψ_j and integrate over $\mathcal{Y}_T$. We get

$$\lambda_j = -2 \int\int_{\mathcal{Y}_T} \left[\left(2\kappa \bar{\chi}_G + \frac{1}{\sigma^2} \chi_Y \right) g - 2\beta \frac{\partial^2 g}{\partial t^2} - 2\gamma \frac{\partial^2 g}{\partial \theta^2} - \frac{1}{\sigma^2} \chi_Y y \right] \Psi_j \, dt d\theta \,, \quad (8.20)$$

and as shown in Appendix 8.A, this may be simplified slightly to yield

$$\begin{aligned} \lambda_j \;=\; & -\int_0^\pi \int_{-T}^T 4\kappa \chi_G g \Psi_j dt \, d\theta - \int_0^\pi \int_{-T}^T \frac{2}{\sigma^2} \chi_Y g \Psi_j dt \, d\theta \\ & + \int_0^\pi 4\beta \frac{\partial g}{\partial t} \Psi_j \Big|_{-T}^T d\theta + \int_0^\pi \int_{-T}^T \frac{2}{\sigma^2} \chi_Y y \Psi_j dt \, d\theta \,. \end{aligned} \quad (8.21)$$

As in Chapter 3, we may derive an approximate expression for λ_j by simplifying the right-hand side of (8.20) and making suitable approximations for each term, where needed. As before, this approximation will allow us to choose starting Lagrange multipliers that are (hopefully) close to the final values, resulting in considerable computational savings. The resulting approximation (detailed in Appendix 8.A) is given by

$$\lambda_j \approx 2 \int\int_{\mathcal{Y}_T} \frac{1}{\sigma^2} \chi_Y y \Psi_j \, dt d\theta \,, \quad (8.22)$$

which is a good approximation when β is small, κ is large, and when χ is 1 on $\mathcal{Y}_T$.

8.5 Computational Methods

In this section we outline the methods used to solve the Constraint-Based algorithm presented in Section 8.4. The methods are very similar to those of the Local Relaxation algorithm of Chapter 3 (see Sections 3.5.2 and 3.5.3, in particular) so that, here, we mainly concentrate on highlighting those aspects which are different.

8.5.1 The Generic Primal-Dual Algorithm

The Constraint-Based algorithm requires finding both $g(t, \theta)$ and p Lagrange multipliers $\lambda_1, \lambda_2, \ldots, \lambda_p$ which satisfy (8.18) so that $g(t, \theta)$ also satisfies the constraints

[1] It can be shown that when $k + l$ is even, $\int\int_{\mathcal{Y}_T} \Psi_i \Psi_j \, dt \, d\theta = \frac{1}{2} \delta_{ij}$.

given by (8.17) and boundary conditions given by (8.14) and (8.19). As in the Local Relaxation solution of Section 3.5, we note that for *fixed* Lagrange multipliers, the PDE of (8.18) may be solved numerically on a discrete lattice system in $\mathcal{Y}_T$ so that $g(t,\theta)$ will satisfy the boundary conditions — this is the *primal* stage. Then if $g(t,\theta)$ also happens to satisfy the constraints, we are done. Otherwise, the Lagrange multipliers must be adjusted — this is the dual stage — so that another primal iteration may be made. We summarize the algorithm below:

Algorithm 8.1 (Constraint-Based Primal-Dual)

1. *Estimate final Lagrange multipliers* $\lambda_1^*, \lambda_2^*, \ldots, \lambda_p^*$ *using (8.22).*
2. *Set* $\lambda_i^0 = \hat{\lambda}_i^*$, *for* $i = 1, \ldots, p$.
3. *Set* $k = 1$ *and* $g^0 = y$.
4. *Solve PDE numerically (using the local relaxation methods described in Section 3.5.2) to yield* g^k.
5. *Does* g^k *satisfy the constraints?*
6. *If not, update Lagrange multipliers* $\lambda_1 \ldots \lambda_p$ *according to*

$$\lambda_i^{k+1} = \lambda_i^k + \alpha\left(0 - \int_0^\pi \int_{-1}^1 g^k(t,\theta)\Psi_i(t,\theta)\, dtd\theta\right) .$$

   Set $k \leftarrow k+1$ *and goto 4.*

7. *Otherwise, we are done and* $\hat{g} = g^k$.

The convergence rate of this algorithm depends on whether the initial Lagrange multiplier estimates are accurate, on the convergence rate of the local relaxation method, and on the choice of α.

8.5.2 Indexing the Basis Functions

The Lagrange multiplier λ_i corresponds to the constraint involving the basis function $\Psi_i(t,\theta) = P_k(t)S_{lm}(\theta)$ where i indexes the triplet (k,l,m). We present here the method by which we order the infinite set of triple indices (k,l,m) so that as

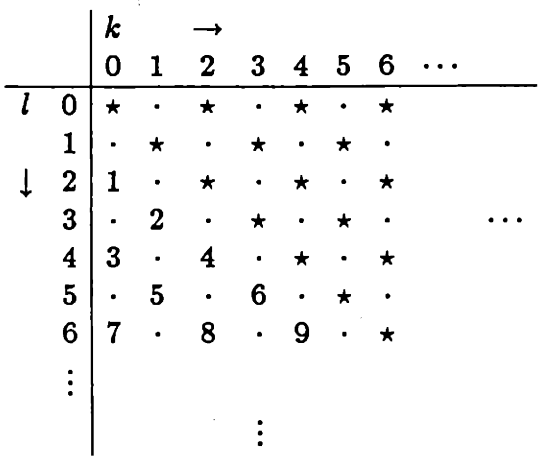


Figure 8.1: The free and constrained Fourier coefficients.

$i \to \infty$ we account for all of the Fourier coefficients and so that we may determine the triplet (k,l,m) which corresponds to any given i.

As developed in previous sections, we know that there are three types of Fourier coefficients: 1) those that are free (unconstrained), 2) those that are constrained to be zero by the polynomial constraint, and 3) those that are trivially zero due to the fact that we enforce the periodicity condition of the Radon transform. This identification is independent of m, so that for either $m = 1$ or $m = 2$ we may consider the classification of the Fourier coefficients by the value of k and l alone. In Fig. 8.1 we show the classification using the symbol $\star$ for the free coefficients, $\cdot$ for the trivially zero coefficients, and an integer for the constraints which must be forced to be zero by our algorithm. The sequence of integers in the figure indexes the infinite sequence of Lagrange multipliers for a given m. Let us denote by j this integer. Then if we let all odd i correspond to $m = 1$ and even i correspond to $m = 2$ we may determine j by

$$j = \begin{cases} (i+1)/2 & \text{for } i \text{ odd} \\ i/2 & \text{for } i \text{ even} \end{cases} . \tag{8.23}$$

Then, we determine k and l from j as follows:

$$k = \begin{cases} 2(i - G^2 + G - 1) & \text{if } S = 0 \\ 2(i - G^2 - 1) + 1 & \text{if } S = 1 \end{cases} \tag{8.24}$$

$$l = 2G + S, \tag{8.25}$$

where

$$G = \lfloor \sqrt{i} + 0.5 \rfloor, \quad \text{and}$$

$$S = \begin{cases} 1 & \text{if } i > G^2 \\ 0 & \text{otherwise} \end{cases}.$$

Given the (k,l,m) coordinates for a particular i, the basis function $\Psi_i(t,\theta)$ is uniquely determined and the integral expressions may be calculated numerically.

8.6 Experimental Results

We present one set of simulations in this section in order to demonstrate the performance of the Constraint-Based algorithm. The object we use is the M I T ellipse described in Section 3.6.1 and we consider the limited-angle case in which we observe the left 40 (out of 60) projections of the same 10.0dB SNR sinogram used in Section 3.6.6. We do not use support information in this simulation; so, the parameters which are variable are the two smoothing parameters, γ and β, and p, the total number of constraints used. The results of these simulations are shown in Figs. 8.2, 8.3, and 8.4. Although the mass and center of mass are never explicitly enforced in the Constraint-Based method, it is interesting to see how well these constraints are met. Therefore, in Fig. 8.4 we show graphs of the mass and center of mass as a function of the view index for the four results.

The sinogram shown in Fig. 8.2a and its reconstruction using CBP shown in Fig. 8.3a are the result of using the Constraint-Based algorithm for $\gamma = 0.0005$, $\beta = 0.01$ and $p = 0$. These values correspond to an amount of vertical smoothing which we found to yield good results in Section 3.6, a small amount of horizontal smoothing, and *no* constraints. The mass and center of mass for the sinogram

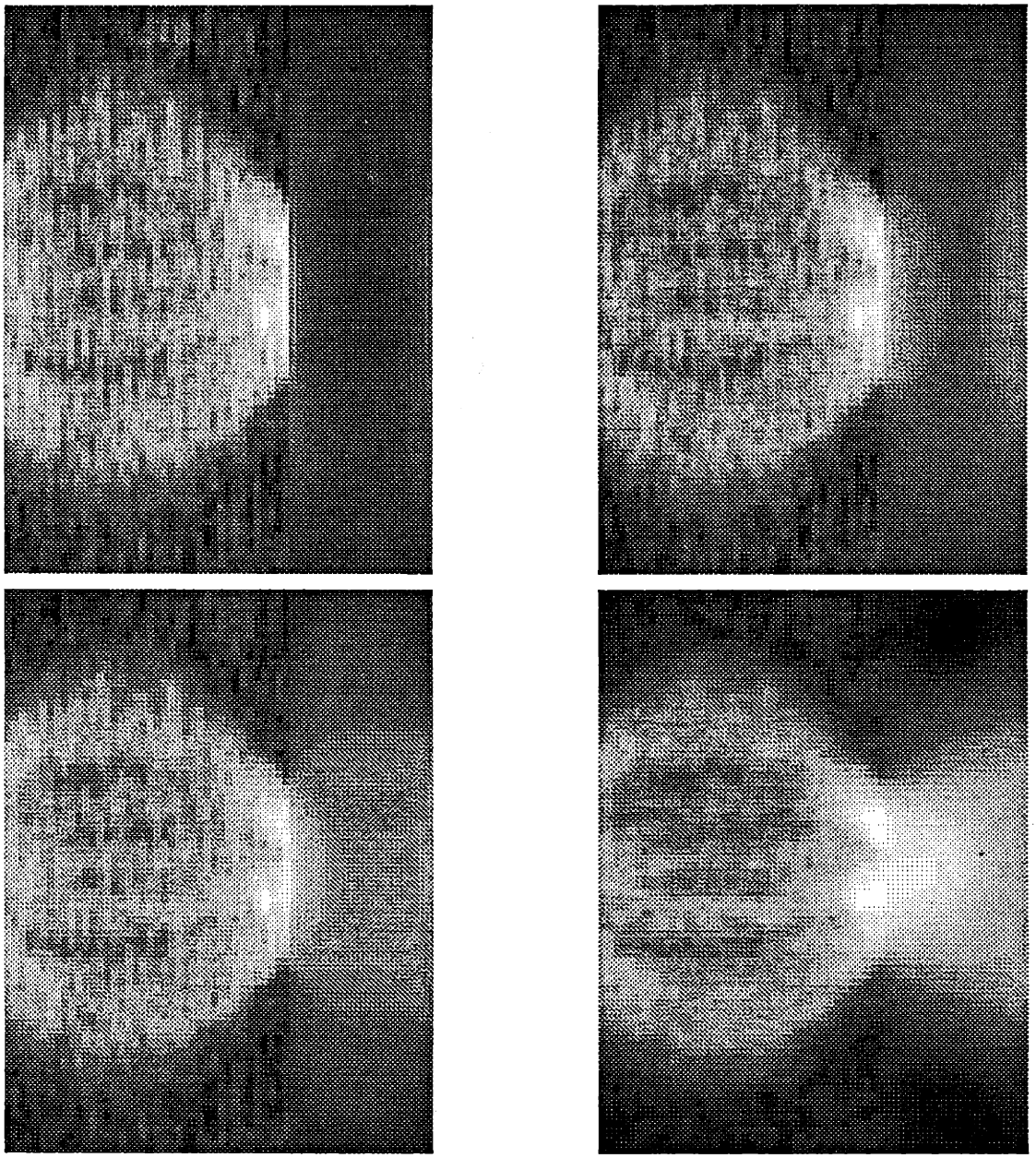


Figure 8.2: Limited-angle studies using the Constraint-Based algorithm: sinogram estimates.

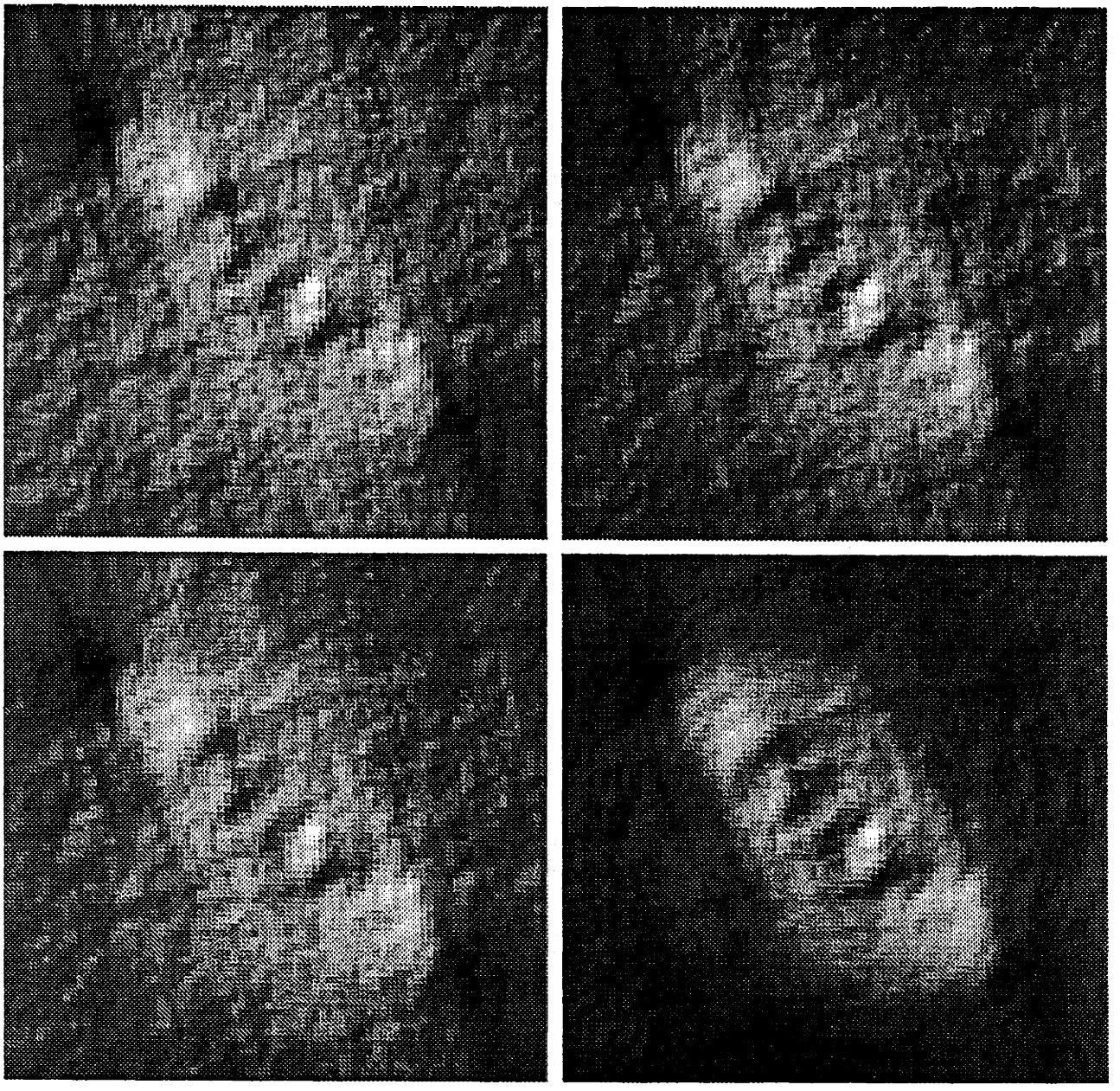


Figure 8.3: Limited-angle studies using the Constraint-Based algorithm: reconstructions from Fig. 8.2.

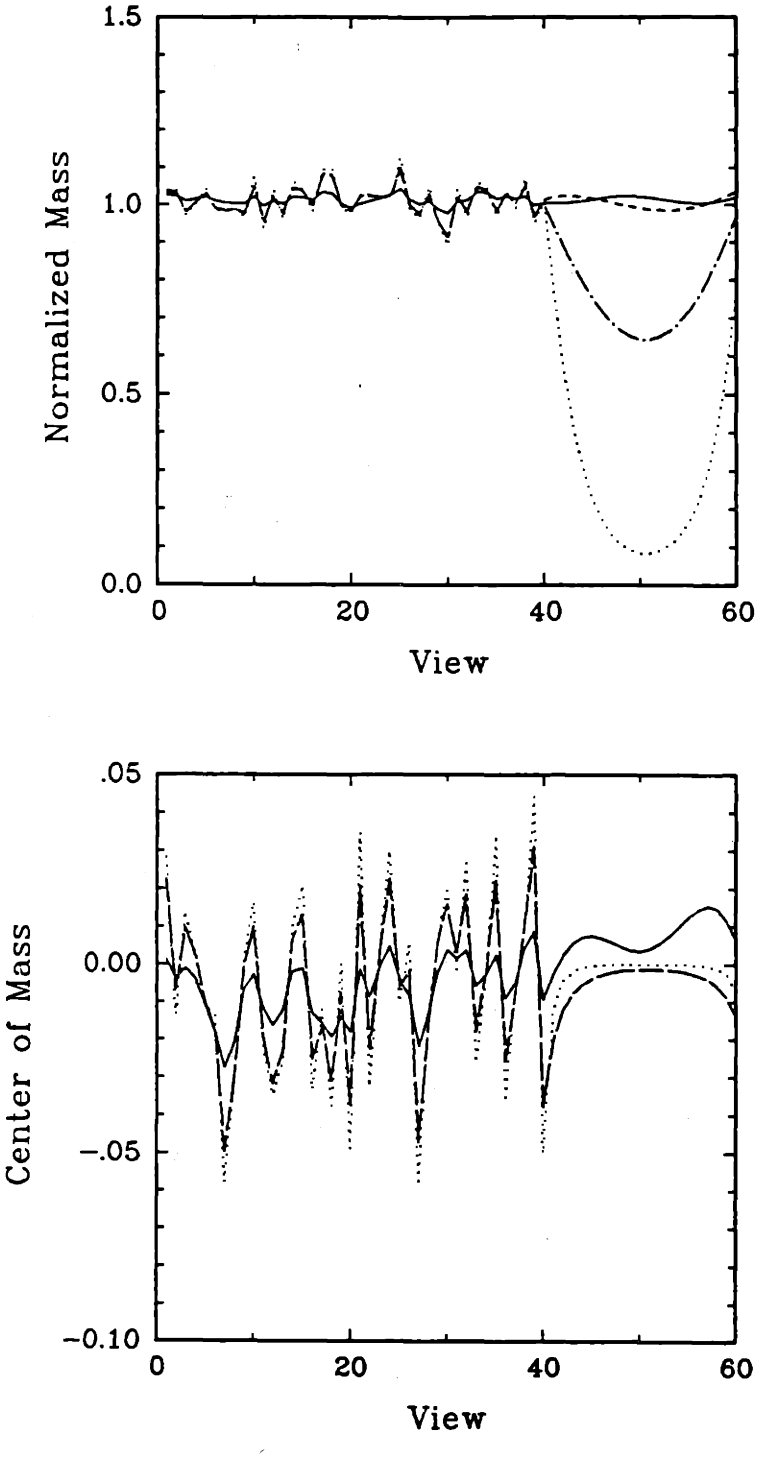


Figure 8.4: Comparison of mass and center of mass constraints for the sinograms shown in Fig. 8.2.

of Fig. 8.2a are shown using the dotted line in Fig. 8.4. The sinogram shown in Fig. 8.2b, its reconstruction in Fig. 8.3b, and the dash/dot curves in Fig. 8.4 correspond to the values $\gamma = 0.005$, $\beta = 0.01$, and $p = 0$. The sinogram shown in Fig. 8.2c, its reconstruction in Fig. 8.3c, and the dashed curves in Fig. 8.4 correspond to the values $\gamma = 0.005$, $\beta = 0.01$, and $p = 2$. Finally, the sinogram shown in Fig. 8.2d, its reconstruction in Fig. 8.3d, and the solid curves in Fig. 8.4 correspond to the values $\gamma = 0.05$, $\beta = 0.01$, and $p = 22$.

There are several important points to be observed from this simulation. First, in examining the differences between the first two experiments, shown in the (a) and (b) figures, we notice that both the mass and center of mass move closer to their correct values (1.0 and 0.0, respectively). But this is entirely due to an increased smoothing effect since no constraints were added. Second, between (b) and (c), the only difference was the addition of two constraints to (c), which causes the mass to become very close to its correct value and, although it is somewhat difficult to see, the center of mass remains completely unchanged. In going from (c) to (d) we make a major change in the input parameters to the algorithm. Here, we increased the horizontal smoothing coefficient γ to the nominal value which we found to produce good results in Chapter 3. At the same time we increased the number of constraints to 22. The result is a sinogram and reconstruction which are noticeably better than the other results in this simulation and somewhat better than the result shown in Fig. 3.13, the comparable result using only the mass and center of mass constraints. Also, the mass and center of mass constraints are more closely met than the other results in this simulation. One point that we observed in other simulations, is that there is no noticeable improvements in the reconstructions for $p > 22$.

8.7 Discussion

The results of this section show that it is possible to improve the reconstructions we obtained in Chapter 3 by enforcing more of the constraints which are inherent to the Radon transform. It is interesting, however, to observe that by strict count, the actual number of *scalar* Lagrange multipliers that were required in the two methods

— that is, the Local Relaxation (LR) method of Chapter 3 and the Constraint-Based (CB) method of this chapter — was greater in the LR method than in the CB method. This is because, in the LR method, the mass and center of mass constraints are independent of θ, and therefore the Lagrange multipliers were actually *functions* of θ. Therefore, when the problem is discretized to solve it on the sinogram lattice, we must maintain $2n_v$ constraints, where n_v is the number of views (projections) in the sinogram. In the examples in this thesis we used $n_v = 60$, so that in the LR method there are actually 120 scalar Lagrange multipliers. In contrast, only 22 scalar Lagrange multipliers were sufficient to produce good results using the CB methods, and using more than 22 did not produce any noticeable differences.

The preceding comments might be construed as a contradiction to our statement that the CB methods use more constraints than the LR methods. Of course, the reason for our statement lies in the fact that we consider the mass and center of mass to be but two constraints. Also, it should be noted that the CB method requires considerably more computation for a single scalar Lagrange multiplier update than does the LR method. This reflects the fact that a scalar Lagrange multiplier in the LR method is associated with the mass or center of mass of a single column (projection) in the sinogram, while a Lagrange multiplier in the CB method is associated with a basis function defined over the entire sinogram. In fact, the dual stage of the CB primal-dual algorithm, is often the most computationally intensive stage, and this is a stage which cannot be parallelized since it involves a global computation. In contrast, each scalar Lagrange multiplier update in the dual stage of the LR algorithm could, in principle, be computed in parallel.

Aside from the difference in the constraints, the CB algorithm has identical structure to the LR algorithm, and therefore many of the comments made in Chapters 3 and 7 concerning, for example, the smoothing coefficients and the support coefficient, apply as well here. In Chapter 9, we summarize some of these general results and make suggestions about possible future research topics.

8.A Approximations for Starting Lagrange Multipliers

The integral expression for λ_j given in (8.20) consists of five separate additive integral terms. In this appendix, we consider each term for possible simplification and approximation.

The first term in (8.20) is approximately zero since g^* (the optimum solution) is nearly zero outside of $\mathcal{G}$. The second term is trivially zero if $\chi = 1$ on $\mathcal{Y}_T$; otherwise we cannot reduce this term any farther. The third term is approximately zero if β is small. To see this we integrate by parts in t twice, and simplify:

$$\begin{aligned}
\int_0^\pi \int_{-1}^1 2\beta g_{tt}\Psi_j \, dt d\theta &= \int_0^\pi \left(2\beta g_t \Psi_j|_{-T}^T - \int_{-1}^1 2\beta g_t \frac{\partial \Psi_j}{\partial t} \, dt \right) \, d\theta \\
&= \int_0^\pi \left(2\beta g_t \Psi_j|_{-T}^T - \left[2\beta g \frac{\partial \Psi_j}{\partial t}\bigg|_{-T}^T - \int_{-1}^1 2\beta g_t \frac{\partial^2 \Psi_j}{\partial t^2} \, dt \right] \right) \, d\theta \\
&= \int_0^\pi \left(2\beta g_t \Psi_j|_{-T}^T + \int_{-1}^1 2\beta g \frac{\partial^2 \Psi_j}{\partial t^2} \right) \, d\theta
\end{aligned}$$

where we have used the fact that $g(T,\theta) = g(-T,\theta) = 0$. The second term in the final expression above is identically zero. To see this we substitute an explicit formula for $\partial^2 \Psi_j / \partial t^2$ into the expression to get for this term only:

$$2\beta \int_0^\pi S_{ml}(\theta) \int_{-T}^T g \frac{\partial^2}{\partial t^2} P_k(t) dt \, d\theta \, .$$

We would like to use the Consistency Theorem to conclude that the above integral is zero, but the range of integration over θ is wrong. To change the range of integration, we note that since $n+m$ is even, we may write

$$2S_{ml}(\theta)P_k(t) = S_{ml}(\theta)P_k(t) + S_{ml}(\theta+\pi)P_k(-t) \, .$$

Therefore, this term may be manipulated as follows:

$$\begin{aligned}
&2\beta \int_0^\pi S_{ml}(\theta) \int_{-T}^T g \frac{\partial^2}{\partial t^2} P_k(t) dt \, d\theta \\
&\quad = \beta \int_0^\pi \int_{-T}^T g \frac{\partial^2}{\partial t^2} \left[S_{ml}(\theta)P_k(t) + S_{ml}(\theta+\pi)P_k(-t) \right] dt \, d\theta
\end{aligned}$$

$$= \beta \int_0^{\pi} \int_{-T}^{T} g \frac{\partial^2}{\partial t^2} S_{ml}(\theta) P_k(t) dt\, d\theta + \beta \int_{\pi}^{\pi} \int_{-T}^{T} g(t, \theta - \pi) \frac{\partial^2}{\partial t^2} S_{ml}(\theta) P_k(-t) dt\, d\theta$$
$$= \beta \int_0^{2\pi} \int_{-T}^{T} g(t, \theta) \frac{\partial^2}{\partial t^2} S_{ml}(\theta) P_k(t) dt\, d\theta$$
$$= \beta \int_0^{2\pi} S_{ml}(\theta) \int_{-T}^{T} g(t, \theta) \frac{\partial^2}{\partial t^2} P_k(t) dt\, d\theta .$$

Now by condition (c) of the Consistency Theorem (Theorem 2.1), we know that the integral over t must result in a polynomial in ω of order $\leq k - 2$ (since the second derivative of $P_k(t)$ is a polynomial of order $k - 2$). Then, since $k < m$, we may conclude that the integral over θ is identically zero.

Finally, the third term in (8.20) may be written as

$$\int_0^{\pi} \int_{-T}^{T} 2\beta \frac{\partial^2 g}{\partial t^2} \Psi_j dt\, d\theta = \int_0^{\pi} 2\beta \frac{\partial g}{\partial t} \Psi_j \Big|_{-T}^{T} d\theta .$$

This term may or may not be nearly zero depending on the size of β and on the size of the support of the observations. For small β and large κ, however, we would expect this term to be nearly zero, because, when κ is large, we expect the optimal $g(t, \theta)$ to be nearly zero from the support values out to the $\pm T$ boundary. Therefore, $\partial g / \partial t$ will be nearly zero at $\pm T$, and provided that β is small, the integral in the above expression will also be nearly zero.

The fourth term in (8.20) is exactly zero. We see this from the following manipulations:

$$\int_0^{\pi} \int_{-T}^{T} 2\gamma \frac{\partial^2 g}{\partial \theta^2} \Psi_j dt\, d\theta = 2\gamma \int_0^{\pi} S_{ml}(\theta) \frac{\partial^2}{\partial \theta^2} \int_{-T}^{T} g(t, \theta) P_k(t) dt\, d\theta$$
$$= \gamma \int_0^{\pi} S_{ml}(\theta) \frac{\partial^2}{\partial \theta^2} \int_{-T}^{T} g(t, \theta) P_k(t) dt\, d\theta$$
$$+ \gamma \int_{\pi}^{2\pi} S_{ml}(\theta - \pi) \frac{\partial^2}{\partial \theta^2} \int_{-T}^{T} g(t, \theta - \pi) P_k(t) dt\, d\theta .$$

The second term of the above sum may be simplified using the symmetries and periodicities of $S_{ml}(\theta)$, $g(t, \theta)$, and $P_k(t)$ to get

$$\gamma \int_{\pi}^{2\pi} S_{ml}(\theta - \pi) \frac{\partial^2}{\partial \theta^2} \int_{-T}^{T} g(t, \theta - \pi) P_k(t) dt\, d\theta$$
$$= (-1)^{k+l} \gamma \int_{\pi}^{2\pi} S_{ml}(\theta) \frac{\partial^2}{\partial \theta^2} \int_{-T}^{T} g(t, \theta) P_k(t) dt\, d\theta .$$

Now since $k+l$ is even, we conclude that the fourth term of (8.20) may be written as

$$\int_0^\pi \int_{-T}^T 2\gamma \frac{\partial^2 g}{\partial \theta^2} \Psi_j dt\, d\theta = \gamma \int_0^{2\pi} S_{ml}(\theta) \frac{\partial^2}{\partial \theta^2} \int_{-T}^T g(t,\theta) P_k(t) dt\, d\theta\,.$$

The consistency theorem tells us that the integral over t is a polynomial in ω of degree $\leq k$, and since the second partial of such a polynomial does not change its degree, the integral over θ must be zero. Finally, we note that the fifth term of (8.20) cannot be simplified.

Taking all the simplifications together we have the following *exact* expression for the λ_j:

$$\begin{aligned} \lambda_j &= -\int_0^\pi \int_{-T}^T 4\kappa \chi_G g \Psi_j dt\, d\theta - \int_0^\pi \int_{-T}^T \frac{2}{\sigma^2} \chi_Y g \Psi_j dt\, d\theta \\ &\quad + \int_0^\pi 4\beta \frac{\partial g}{\partial t} \Psi_j \Big|_{-T}^T d\theta + \int_0^\pi \int_{-T}^T \frac{2}{\sigma^2} \chi_Y y \Psi_j dt\, d\theta \end{aligned} \tag{8.26}$$

Using the approximations from above we have the following approximation for λ_j, which is valid when β is small, κ is large and χ_Y is 1 on y_T:

$$\lambda_j \approx 2 \int_0^\pi \int_{-T}^T \frac{1}{\sigma^2} \chi_Y y \Psi_j dt\, d\theta\,. \tag{8.27}$$

Chapter 9

CONCLUSIONS

In this thesis we have developed a hierarchical method for the reconstruction of images from noisy, and limited- and sparse-angle projections. The method estimates *geometric* features of the object from the available projections, and then uses this information together with a prior probabilistic description of full sinograms to estimate a smoothed sinogram using maximum *a posteriori* estimation techniques. The major areas of investigation required to develop the method can be classified into four areas:

1. Incorporation of prior knowledge.
2. Projection-space algorithms.
3. Computational geometry.
4. Hierarchical reconstruction algorithm.

In this chapter, we highlight the important developments within each of these areas and suggest possible new research topics.

9.1 Incorporation of Prior Knowledge

It is generally acknowledged in the computed tomography literature that in the case of noisy and limited- or sparse-angle data, prior knowledge is essential in order to obtain good reconstructions. We have focused on three types of prior knowledge:

1. Line integrals close in either angle or lateral displacement tend to be similar in value,

2. Radon transform functions are constrained to a certain functional subspace, and

3. Knowledge of the convex hull of the object is equivalent to knowledge of the support of the Radon transform.

We developed in Chapter 3 a Markov random field (MRF) model of sinograms which contains prior information concerning the mass and center of mass of the unknown sinogram, the convex support of the object, and about expected similarity of line integrals close in either angle or lateral displacement. The mass and center of mass are incorporated as constraints on the space of feasible sinograms in the MRF, while the knowledge of convex support and adjacent line similarity are incorporated as penalties. In particular, given an estimate of the convex support of the object, the MRF gives lower probability to sinograms with larger values outside the region of support, given the same values inside — this is a self-potential term in the MRF. Sinograms with adjacent site-values that are similar in value have a higher probability because of the vertical and horizontal pair-potential terms in the MRF. In the Constraint-Based methods of Chapter 8, we modified the domain of feasible sinograms to correspond to those that have a certain set of generalized Fourier coefficients which are zero, rather than constraining the mass and center of mass to be particular values.

Further Research Topics:

- **Binary object constraints.** The objects that we considered in the simulations in this thesis were two-valued, i.e. binary, but we did not use this knowledge to advantage. The reasoning in this case is that we developed a general procedure applicable to all objects with bounded support. But there are many applications in medicine, oceanography, and non-destructive testing, for example, in which the assumption of a two-valued object is appropriate.

Obviously, this additional piece of knowledge strongly constrains the space of feasible solutions, and in particular, we are interested in applying this knowledge in *projection-space*, so that we may use some of the techniques developed in this thesis to make an MAP estimate of the sinogram. It is not difficult to write down a necessary condition for this constraint to be met — just use the exact statement of the inverse Radon transform — but it is not clear how to incorporate it on discretely sampled data and in an efficient manner.

- **Non-local MRF neighborhoods.** The MRF formulation is a very general formulation which permits inclusion of prior knowledge in virtually endless variations. One of the early ideas we had considered is this: if one knows the mass of the object, and the object is nearly or exactly binary, then if one has a very large sinogram value at some (t, θ) coordinate, the support width at $\theta + \pi$ will tend to be large. This phenomenon is particularly noticeable on the ellipse figure that we used in many of our simulations (see Fig. 7.4, for example), where the largest projection values appear in the projections which have the narrowest view of the ellipse, and the projections at 90 degrees away have the widest region of support. This suggests that we might extend the neighborhood system of the MRF to include connections to sites 90 degrees away. One can now imagine including the support values themselves as additional random variables, which are coupled to the sinogram by additional potential terms in the MRF, and which are constrained in their values by the support vector constraint. Then it would theoretically be possible to *jointly* estimate the support together with the sinogram using MAP, instead of using the hierarchical construction given in this thesis.

- **Spatially varying coefficients.** In Chapter 7, we discussed the possibility of using a spatially varying support coefficient κ, in order to better reflect in the MRF our knowledge of support. It is also possible to consider modifying the smoothing coefficients γ and β, in order to better reflect our prior knowledge. For example, since the horizontal smoothing presents itself as an angular smoothing term in the object domain, it may be appropriate to make γ (the

coefficient of horizontal smoothing) *smaller* as $|t|$ increases. In this way, the correlation length of the angular swirling patterns, which appear as unwanted artifacts in many of our reconstructions, may be more constant over the whole image rather than getting larger with $|t|$.

Another reason to design spatially varying smoothing coefficients might be to improve on the estimates of the missing projections. For example, we found that as one moves farther away in angle from the nearest observed projection, the vertical smoothing term tends to dominate the sinogram estimates, and in particular the estimates tend to lose information about the vertical changes in the nearest observed projections. In this case, one might artificially increase the amount of horizontal smoothing and decrease the amount of vertical smoothing for those projections which are not observed. In this way, their estimates will more strongly reflect the values of the nearest observed projections.

9.2 Projection-Space Algorithms.

The algorithms of Chapter 3 and Chapter 8 are projection-space algorithms since they seek to estimate a full sinogram from the available noisy and incomplete data, rather than to seek the object directly. The algorithms solve a maximum *a posteriori* (MAP) estimation problem for the sinogram MRF developed in Chapter 3, with the assumption of a zero-mean white Gaussian noise model. Because of the quadratic form of the energy term in the Gibbs probability (which follows from the MRF), and because of the linear constraints, the solution of the MAP problem is given by a quadratic program. We developed an efficient local relaxation (LR) method to solve this problem (without the positivity constraint) using a primal-dual formulation in which the primal stage is solved by an efficient local-relaxation method, and the dual stage involves a simple Lagrange multiplier update.

Further Research Topics:

- **Incorporation of positivity.** In order to develop the LR method we dropped the positivity constraint which was a part of the original MRF formulation. Therefore, this method does not (necessarily) produce strictly non-negative sinograms. Based on some early experiments, we cannot say that the positivity constraint has a marked effect on the outcome. However, it may be of interest in some applications to enforce this constraint, and in order to do this, one must modify the LR algorithm.

- **Implementation on parallel architecture.** The primal stage of the LR (and Constraint-Based CB) algorithm is readily adaptable to implementation on parallel architectures; however, the dual stages are not since the constraints are global in nature. It may be possible, however, to exchange information on a local basis, or to augment the local neighborhood structure in order to implement the dual stages more efficiently in parallel.

9.3 Computational geometry.

In the area of computational geometry, we investigated one problem in particular: the estimation of convex sets from noisy support line measurements. Our emphasis was on the characterization of support line constraints, and on the utilization of noise statistics and prior shape knowledge. These efforts resulted in a theory of consistent support lines, and included the definition of *support vectors*, *basic objects*, and *the discrete radius of curvature*. With this constraint information and a noise model, we were able to formulate ML and MAP estimation problems which produced reconstructions that were a significant improvement over those of the intersection method. The prior knowledge used in the MAP formulations included information about the expected position, size, and shape of the convex set, and in some cases about the eccentricity and orientation of the set, also.

Further Research Topics:

- **Non-uniform and random angular spacing.** Because the statement of this problem arose from the problem of sinogram estimation, all of the measured support lines have known angles which are evenly spaced over 2π. A more general statement of the problem would allow the support lines to be measured at known angles which are not evenly spaced over 2π, and an even more general problem would have the angles be random variables with some prior distribution. The first extension to the problem — that of known but uneven angles — would require the generation of a new matrix C, and this is not a difficult problem to solve. Also, extensions of the ideas of basic object and discrete radius of curvature should not be difficult to make.

  However, when the angles are random variables, this opens up an entirely new area of investigation, which we feel is a very challenging problem. In particular, since the matrix C depends on the angles $\theta_1, \ldots, \theta_M$, and since these angles and the lateral displacements are random variables, then the constraint $h^T C \leq 0$ is a non-linear function of the unknowns. Although, given the prior distributions of the lateral displacements and the angles, the form of the ML estimate may be simple to write down, it will undoubtedly be a much more difficult problem to solve than the algorithms of Chapters 4 and 5.

- **Further decomposition of support vectors.** In Chapter 5 we decomposed the support vector into the SSS components that represented size, shape, and shift (position). We also developed ideas related to eccentricity and orientation, however, these quantities did not *decompose* a support vector into its constituent coordinates as did the SSS decomposition. There is, however, an elegant structure to the support cone that we have only just begun to explore. For example, one aspect which we did not explore (except to mention in passing) is that the circular rotation of a support vector causes a rotation in the plane (in angular increments of $2\pi/M$) of the basic object. Therefore, one might further decompose a shape vector q into a vector which has a

given orientation; any of the remainder of the shape vectors could be generated by adding an orientation variable which takes on a value in the range of $1,\ldots,M$, and essentially determines the circular rotation of the shape vector. There may also be decompositions related to eccentricity and other shape parameters.

A different type of decomposition may be found as follows: since a shape vector q is an element of a *bounded* polytope, it may be written as convex combination of the *extreme points* of the polytope. It is not hard to see that an extreme point q_e yields a basic object which is either a triangle or a line segment. Therefore, this decomposition takes an arbitrary basic object into a set of coefficients, each associated with a simple triangular or line-segment object. The issues here are: 1) how does one determine the extreme points of this polytope, and how many of them are there (as a function of M), 2) is there a way to specify a unique set of coordinates for any given vector q, and 3) how does one use this decomposition in set reconstruction algorithms.

- **Dual formulations.** Since our observations are lines, we have focused on algorithms to estimate lines, and thereby determine our sets by intersection of the appropriate halfspaces. Another viewpoint on this problem may be obtained by looking at dual formulations in which lines are interpreted as points. For example, one such formulation is to work with the *polar set* [67,29] of each line, which for the line $L(t,\theta)$ is exactly the (dual) point whose polar coordinates are $(1/t,\theta)$. Also, the polar set of an entire basic object is just the convex hull of the dual points of each support line. It would be interesting to examine the fundamental constraint — it should be that all dual points of the given lines lie on the boundary of their own convex hull — and to reformulate the algorithms in terms of their dual variables. Furthermore, there may be some advantages to this viewpoint when random angle problems are considered.

- **Fast algorithms.** Many of the proposed algorithms for estimating support vectors required finding the solution to a quadratic program. In several cases

— e.g., the Closest algorithm and the Joint Ellipse algorithm — the constraint was merely that the estimated support vector must satisfy the basic constraint $h^T C \leq 0$. We did not, however, attempt to modify our QP codes to take advantage of the structure of the support cone (other than what we accomplished with the SSS decomposition) to speed up the computations. It seems likely that, due to the circulant structure of C, there would be some advantages to be obtained by a detailed look at QP methods and special purpose tailoring to this problem.

- **3-D extensions.** There are many applications — e.g, computer vision, robotics, and medical imaging — in which 2-D projections or silhouettes are available, or in which support *plane* measurements are directly available. It is natural to wonder whether there are useful extensions of some of our set reconstruction theory to this 3-D problem. For example, what is the analogous statement of the support vector constraint to sets of measured support lines or planes in 3-D. One complication is that there is no natural ordering for the unit vectors in 3-D, as there is in 2-D. Once the statement of this constraint is worked out, there are many 3-D set reconstruction algorithms which may be designed to parallel some of our algorithms. Also, the ideas analogous to discrete radius of curvature, and the SSS decomposition could be explored.

9.4 Hierarchical Reconstruction Algorithm.

In Chapters 6 and 7 we presented the remaining pieces necessary to combine the the convex support estimation and the sinogram MAP estimation algorithms into one hierarchical algorithm which reconstructs images from noisy, and limited- and sparse-angle projections. Chapter 6 developed two support *value* estimation approaches, the Knot-location and Support-width penalty methods, which serve as inputs to the support *vector* estimation methods of Chapters 4 and 5. This information, together with the estimated mass and center of mass (for which we present methods in Chapter 7) serve as inputs to the sinogram MAP estimation algorithm. The primary contribution in this phase of the research was to demonstrate that the

cascade of these various methods can, in fact, improve on conventional reconstruction which uses CBP applied directly to the data.

Further Research Topics:

- **Multiple objects.** An important feature of the hierarchical algorithm is that there is only one object which is made explicit in both the support estimation phase and the sinogram estimation phase. One possible further research topic might be to explore ways to model more than one object. An example of one way to do this might be to assume that there are several *concentric* objects. In this case, it is conceivable that more than two "support values" per projection could be estimated — using a multiple knot version of the Knot-location method, for example — and the support vector estimation algorithms could be applied in parallel to the various pair of support value estimates. A more difficult problem is one in which there are multiple object but they are isolated; for example, the two disk object in Chapter 7 is such an object. In this case, internal support values might be estimated as above, but they would *cross* themselves within the body of the overall sinogram support, unlike the concentric object case, and this presents an assignment problem if it is desired to use the support vector estimation algorithms.

- **Iteration within the hierarchy.** The hierarchical approach which we outlined in Chapter 7 proceeds forward in a cascade fashion with no feedback or iteration within the hierarchy. One can imagine, however, cases in which some iteration could improve the overall performance. For example, it is reasonable to presume that information about the curvature of the boundary of the object could aid in setting the threshold of the knot-location algorithm. For this reason, an iteration which feeds back from the final support estimation output to the threshold generation algorithm, might yield rewards. In a similar fashion, one can imagine estimating an entire smoothed sinogram, and then using the projections themselves to aid in Knot-location applied to the original data. There are obviously many variations on this theme that might

prove to be interesting avenues of research.

Bibliography

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, 1972.

[2] M. S. Bazarra and C. M. Shetty. *Nonlinear Programming, Theory and Algorithms.* John Wiley and Sons, New York, 1979.

[3] R. Bellman. *Introduction to Matrix Analysis.* McGraw-Hill, New York, 1970.

[4] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods.* Academic Press, New York, 1982.

[5] J. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Royal Statist. Soc. B.*, 36:192–226, 1974.

[6] R. P. Brent. *Algorithms for Minimization without Derivatives.* Prentice-Hall, Englewood Cliffs, N.J., 1973.

[7] Y. Bresler and A. Macovski. 3-d reconstruction from projections based on dynamic object models. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, March 1984.

[8] Y. Bresler and A. Macovski. Estimation of 3-d shape of blood vessels from x-ray images. In *Proc. IEEE Comp. Soc. Int. Symp. on Medical Images and Icons*, July 1984.

[9] Y. Bresler and A. Macovski. A hierarchical Bayesian approach to reconstruction from projections of a multiple object 3-d scene. In *Proc. 7th International Conference on Pattern Recognition*, August 1984.

[10] C. D. Bunks. *Markov random field modeling of geophysical data.* Technical Report, M.I.T., Dept. of Elect. Engr. Comp. Science, 1984. Proposal for Ph.D. research.

[11] C. D. Bunks. *Random field modeling for interpretation and analysis of layered data.* PhD thesis, Massachsetts Institute of Technology, 1987. Dept. Elec. Engr.

[12] M. H. Buonocore. *Fast Minimum Variance Estimators for Limited Angle Computed Tomography Image Reconstruction.* PhD thesis, Stanford University, 1981.

[13] M. H. Buonocore, W. R. Brody, and A. Macovski. A natural pixel decomposition for two-dimensional image reconstruction. *IEEE Trans. Bio. Engr.*, BME-28(2):69–78, 1981.

[14] M. H. Buonocore, W. R. Brody, A. Macovsky, and S. Wood. A polar pixel Kalman filter for limited data CT image reconstruction. In *Proc. SPIE*, pages 109–115, August 1979.

[15] Y. Censor. Finite series-expansion reconstruction methods. *Proc. IEEE*, 71(3):409–419, March 1983.

[16] S. K. Chang. The reconstruction of binary patterns from their projections. *Comm. ACM*, 14(1):21–25, 1971.

[17] S. K. Chang and G. L. Shelton. Two algorithms for multiple-view binary pattern reconstruction. *IEEE Trans. Sys. Man and Cyber.*, 90–94, January 1971.

[18] G. Cross and A. Jain. Markov random field texture models. *IEEE Trans. Pat. Analysis and Mach. Int.*, PAMI-5(1), January 1983.

[19] M. E. Davison and F. A. Grunbaum. Tomographic reconstructions with arbitrary directions. *Comm. Pure Appl. Math.*, 34:77–119, 1979.

[20] S. R. Deans. *The Radon Transform and Some of Its Applications.* John Wiley and Sons, New York, 1983.

[21] M. Ein-Gal. *The Shadow Transformation: An Approach to Cross-Sectional Imaging.* PhD thesis, Stanford University, Dept. of Electr. Engr., 1974.

[22] A. Erdelyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi. *Higher Transcendental Functions.* Volume 2, McGraw-Hill, New York, 1953.

[23] P. C. Fishburn, J. A. Reeds, and L. A. Shepp. Sets uniquely determined by projections. 1986. Preprint.

[24] S. B. Gelfand. *Analysis of simulated annealing type algorithms.* PhD thesis, Massachusetts Institute of Technology, 1987. Dept. of Elec. Engr.

[25] S. Geman. Seminar on MRF's, Gibbs distributions, and Simulated Annealing, given at the Joint Centers for Intelligent Control Computer Vision Workshop, November 21, 1987, Harvard University.

[26] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-6(6):721–741, Nov. 1984.

[27] S. Geman and D. E. McClure. *Bayesian image analysis: and application to single photon emission tomography.* Technical Report, Brown University, 1985. Preprint to appear in 1985 Proc. Amer. Stat. Assoc. Statistical Computing.

[28] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27:1–33, 1983.

[29] J.P. Greshak. *Reconstructing Convex Sets.* PhD thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1985.

[30] K. M. Hanson. Tomographic reconstruction of axially symmetric objects from a single radiograph. 1984. SPIE, vol. 491, High Speed Photography (Strasbourg).

[31] K. M. Hanson and G. W. Wecksung. Bayesian approach to limited-angle reconstruction in computed tomography. *Appl. Optics*, 24:4028–4039, December 1980.

[32] S. Helgason. *The Radon Transform.* Birkhauser, Boston, MA, 1980.

[33] G. T. Herman. *Image Reconstruction from Projections.* Academic Press, New York, 1980.

[34] G. T. Herman, H. Hurwitz, A. Lent, and H.-P. Lung. On the Bayesian approach to image reconstruction. *Inf. Control*, 42:60–71, 1979.

[35] F. B. Hildebrand. *Advanced Calculus for Applications.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[36] F. B. Hildebrand. *Methods of Applied Mathematics.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1965.

[37] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

[38] J. M. Humel. *Resolving Bilinear Data Arrays.* Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1986.

[39] T. Inoye. Image reconstruction with limited angle projection data. *IEEE Trans. Nucl. Sci.*, NS-26(2):2666–2669, 1979.

[40] A. K. Jain and S. Ansari. Radon transform theory for random fields and optimum image reconstruction from noisy projections. In *ICASSP*, pages 12A.7.1–4, March 1984.

[41] F. John. Bestimmung einer Funktion aus ihren Integralen ūber gewisse Mannigfaltigkeiten. *Math. Ann.*, 109:488–520, 1934.

[42] P. J. Kelly and M. Weiss. *Geometry and Convexity – a Study in Mathematical Methods.* John Wiley and Sons, New York, 1979.

[43] J. H. Kim, K. Y. Kwak, S. B. Park, and Z. H. Cho. Projection space iteration reconstruction-reprojection. *IEEE Trans. Med. Imag.*, MI-4(3), September 1985.

[44] R. Kinderman and J. L. Snell. *Markov Random Fields and Their Applications.* American Mathematical Society, Providence, 1980.

[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–611, May 1983.

[46] C.-C. J. Kuo and B. C. Levy. *A two-level four-color SOR method.* Technical Report LIDS-P-1625, MIT Laboratory for Information and Decision Systems, 1986.

[47] C.-C J. Kuo, B. C. Levy, and B. R. Musicus. A local relaxation method for solving elliptic PDE's on mesh-connected arrays. *SIAM J. Sci. Stat. Comput.*, 8(4):550–573, 1987.

[48] A. H. Land and S. Powell. *Fortran Codes for Mathematical Programming.* Wiley-Interscience, London, 1973.

[49] R. M. Leahy. Reconstruction algorithms: transform methods. *Proc. IEEE*, 71(2):390–408, March 1983.

[50] R. M. Lewitt and R. H. T. Bates. Image reconstruction from projections i: general theoretical considerations. *Optik*, 50(1):19–33, 1978.

[51] R. M. Lewitt and R. H. T. Bates. Image reconstruction from projections iii: projection completion methods (theory). *Optik*, 50(3):189–204, 1978.

[52] R. M. Lewitt, R. H. T. Bates, and T. M. Peters. Image reconstruction from projections ii: modified back-projection methods. *Optik*, 50(2):85–109, 1978.

[53] A. K. Louis. Picture reconstruction from projections in restricted range. *Math. Meth. in the Appl. Sci.*, 2:209–220, 1980.

[54] A. K. Louis and F. Natterer. Mathematical problems of computerized tomography. *Proc. IEEE*, 71(3):379–389, March 1983.

[55] D. Ludwig. The Radon transform on Euclidean space. *Comm. Pure Appl. Math.*, 19:49–81, 1966.

[56] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, MA, second edition, 1984.

[57] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley and Sons, Inc., New York, 1969.

[58] B. B. Mandelbrot. *The Fractal Geometry of Nature.* W. H. Freeman and Company, New York, 1983.

[59] J. L. Marroquin. *Probabilistic solution of inverse problems.* PhD thesis, MIT, 1985.

[60] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. N. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1091, 1953.

[61] A. M. Mier-Muth. *Adaptive Knot Location for Spline Approximation.* Master's thesis, Massachusetts Institute of Technology, Dept. Elec. Engr., 1976.

[62] A. M. Mier-Muth and A. S. Willsky. *A Sequential Method for Spline Approximation With Variable Knots.* Technical Report ESL-P-759, M.I.T. Electronic Systems Laboratory, 1977.

[63] B. E. Oppenheim. *Reconstruction tomography from incomplete projections*, pages 155–183. Univ. Park Press, Baltimore, 1977.

[64] J. H. Park, K. Y. Kwak, and S. B. Park. Iteractive reconstruction-reprojection in projection space. *Proc. IEEE*, 73(6):1140–1141, June 1985.

[65] M. J. D. Powell. Corrections and extensions to the Fortran listing of ZQPCVX. February 1984. University of Cambridge.

[66] M. J. D. Powell. *ZQPCVX: a Fortran subroutine for convex quadratic programming.* Technical Report DAMTP/1983/NA17, University of Cambridge, 1983.

[67] F. P. Preparata and M. I. Shamos. *Computational Geometry.* Springer-Verlag, New York, 1985.

[68] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge, 1986.

[69] J. L. Prince. Geometric model-based Bayesian estimation from projections. April 1986. Proposal for Ph.D. research, M.I.T., Dept. of Electrical Engineering.

[70] J. L. Prince and A. S. Willsky. *Estimation algorithms for reconstructing a convex set given noisy measurements of its support lines.* Technical Report LIDS-P-1638, M.I.T. Laboratory for Information and Decision Systems, January 1987.

[71] J. Radon. Ūber die Bestimmung von Funktionen durch ihre Integralwerte lāangs gewisser Mannigfaltigkeiten. *Berichte Sāchsische Akademie der Wissenschaften. Liepzig, Math. — Phys. Kl.*, 69:262–267, 1917.

[72] J. A. Reeds and L. A. Shepp. Limited angle reconstruction in tomography via squashing. *IEEE Trans. on Medical Imaging*, MI-6(2):89–97, June 1987.

[73] A. J. Rockmore and A. Macovski. A maximum likelihood approach to image reconstruction from projections. *IEEE Trans. Nucl. Sci.*, NS-23:1428, 1976.

[74] D. J. Rossi. *Reconstruction from projections based on detection and estimation of objects.* PhD thesis, MIT, 1982.

[75] D. J. Rossi and A. S. Willsky. Reconstruction from projections based on detection and estimation of objects–parts I and II: performance analysis and robustness analysis. *IEEE Trans. ASSP*, ASSP-32(4):886–906, 1984.

[76] T. Saito and H. Kudo. Tomographic image reconstruction from discretely viewed projections. 1987. Preprint.

[77] L. A. Santalo. *Integral Geometry and Geometric Probability.* Volume 1 of *Encyclopedia of mathematics and its applications*, Addison-Wesley, Reading MA, 1976.

[78] J. S. Schneiter. *Automated Tactile Sensing for Object Recognition and Localization.* PhD thesis, Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 1986.

[79] M. I. Sezan and H. Stark. Tomographic image reconstruction from incomplete view data by convex projections and direct Fourier inversion. *IEEE Trans. Med. Imag.*, MI-3(2):91–98, 1984.

[80] M. Spivak. *A Comprehensive Introduction to Differential Geometry.* Volume 2, Publish or Perish, Inc., Berkeley, 1979.

[81] G. Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Wellesley, MA, 1986.

[82] G. Strang. *Linear Algebra and Its Applications.* Academic Press, New York, 1980.

[83] K.-C. Tam, V. Perez-Mendez, and B. Macdonald. 3-D object reconstruction in emission and transmission tomography with limited angular input. *IEEE Trans. Nucl. Sci.*, NS-26(2):2797–2805, 1979.

[84] M. Tasto. Maximum likelihood reconstruction of random objects from noisy objects. In *Proc. 3rd Int'l Joint Conf. Patt. Rec.*, pages 551–555, November 1976.

[85] M. Tasto. A probabilistic object model for computerized transverse axial tomography. In *Second Int'l Joint Conf. Patt. Rec.*, pages 396–400, August 1974.

[86] G. B. Thomas, Jr. *Calculus and Analytic Geometry.* Addison-Wesley, Inc., Reading, MA, alternate edition, 1972.

[87] H. J. Trussell and M. R. Civanlar. The feasible solution in signal restoration. *IEEE Trans. Acoust. Sp. Sig. Proc.*, ASSP-32(2):201–212, Apr. 1984.

[88] J. N. Tsitsiklis. *A survey of large time asymptotics of simulated annealing algorithms.* Technical Report LIDS-P-1623, MIT Laboratory for Information and Decision Systems, 1986.

[89] P. L. Van Hove. *Silhoutte Slice Theorems.* PhD thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1986.

[90] P. L. Van Hove and J. G. Verly. A silhouette-slice theorem for opaque 3-d objects. In *Proc. of the IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc.*, pages 933–936, IEEE ASSP, March 26–29 1985.

[91] H. L. Van Trees. *Detection, Estimation, and Modulation Theory: Part I. Detection, Estimation, and Linear Modulation Theory.* John Wiley and Sons, New York, 1968.

[92] H. L. Van Trees. *Detection, Estimation, and Modulation Theory: Part III. Radar-Sonar Signal Processing and Gaussian Signals in Noise.* John Wiley and Sons, New York, 1968.

[93] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, N.J., 1962.

[94] A. S. Willsky and H. L. Jones. *A generalized likelihood ratio approach to state estimation in linear systems subject to abrupt changes.* Technical Report LIDS-P-538, M.I.T. Laboratory for Information and Decision Systems, 1974.

[95] S. L. Wood. *A system theoretic approach to image reconstruction.* PhD thesis, Stanford University, May 1978.

[96] S. L. Wood, A. Macovski, and M. Morf. Reconstruction with limited data using estimation theory. In Raviv et. al., editor, *Computer Aided Tomography and Ultrasonics in Medicine*, pages 219–233, North-Holland Publishing Co., 1979.

[97] S. L. Wood and M. Morf. A fast implementation of a minimum variance estimator for computerized tomography image reconstruction. *IEEE Trans. on BME*, BME-28(2):56–68, February 1981. Special Issue.

[98] D. C. Youla and H. Webb. Image restoration by the method of convex projections: part 1 — theory. *IEEE Trans. Med. Imaging*, MI-1:81–94, 1982.