Three Dimensional Image Reconstruction from Fourier Magnitude Measurements

by

John Michael Fini

Submitted to the Department of Electrical Engineering and Computer Science

in partial fullfillment of the requirement for the degrees of

Bachelor of Science in Electrical Science and Engineering

and Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

February 1997

Author .....
Department of Electrical Engineering and Computer Science
January 23, 1997

Certified by .....
Lyle G. Shirley
MIT Lincoln Laboratory Research Staff
Thesis Supervisor

Certified by .....
Gregory W. Wornell
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Departmental Commitee on Graduate Theses

Three Dimensional Image Reconstruction from Fourier Magnitude Measurements

by

John Michael Fini

Submitted to the Department of Electrical Engineering and Computer Science
on January 23, 1997, in partial fulfillment of the
requirements for the degrees of
Master of Engineering
and
Bachelor of Science in Electrical Science and Engineering

Abstract

The problem of recovering a three-dimensional image from samples of the Fourier magnitude and *a priori* knowledge of the object structure is addressed. It is shown that this mathematical problem relates to a method for remotely imaging opaque surfaces. Two broad approaches to finding an iterative solution are presented. One manipulates the support of the autocorrelation function, which can be estimated from the Fourier magnitude samples. The other is an FFT-based method which attempts to iteratively improve estimates of the complex scattering function of the object. The two approaches are shown to combine with some success. Simulations demonstrate the potential of the method, while the algorithms performance on actual measured data have limited success.

Thesis Supervisor: Lyle G. Shirley
Title: MIT Lincoln Laboratory Research Staff

Thesis Supervisor: Gregory W. Wornell
Title: Associate Professorof Electrical Engineering

Acknowledgments

I would like to thank all of the people in the Speckle Lab. I thank Dr. Shirley for keeping the lab going and driving all of us, Greg Hallerman for his patience and help, and Harold Payson for useful advise. I thank Michael Mermelstein who has not only taught me a lot about the details, but has made sure I was always growing and having fun. Thanks to Kavitha, whose support through this thesis and all of my endeavors has been invaluable. Finally, this research has been supported by Lincoln Laboratory, MIT's Department of Electrical Engineering, and The National Science Foundation; I thank all of them for the opportunity they have given me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of the Speckle Lab

This thesis represents work done in the Laser Speckle Lab of Lincoln Laboratory's Group 35. The Laser Speckle Lab develops novel measurement techniques for a range of applications using interference patterns of coherent light. A family of techniques has been developed to quickly and remotely form 3D height maps of opaque surfaces, and these are now the main thrust of the Speckle Lab's work [1, 2]. The problem of recovering Fourier phase information from magnitude measurements alone arises in one particular technique used for measuring heights. This thesis develops "phase retrieval," or Fourier phase recovery, algorithms in order to enable this technique.

The Laser Speckle Lab's earliest work was directed towards target identification in missile defense [3, 4, 5]. More recently, it has become clear that the same principles used for generating signatures of target objects in defense applications could also be used to generate 3D images of many opaque objects. Attempts to find off-the-shelf components to replace extremely expensive equipment were successful, and communication with industry revealed a true need for novel imaging techniques that overcome the difficulties of current methods.

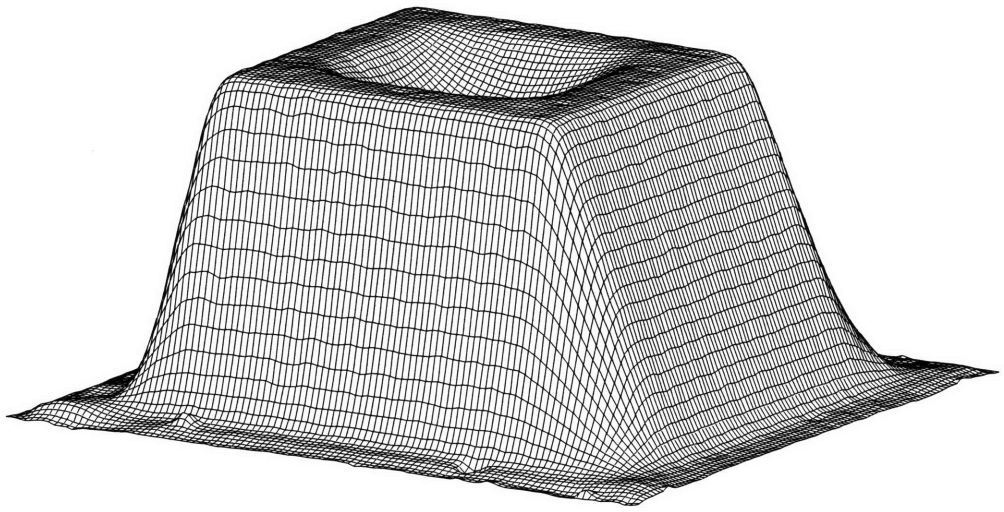

Figure 1-1: Image of a stamped aluminum object using speckle pattern sampling with a phase reference

1.2 Motivation for Phase-Retrieval Processing

Currently, the most successful of the Speckle Lab's methods do not rely on phase-retrieval. Some of these operate on simple principles and have been shown to readily produce detailed images; for example, the height map in Figure 1-1 was obtained with the speckle-pattern-sampling method with a reference point, which will be described in the next chapter. While little work has been done on the phase retrieval problem applied to complex-valued scattering functions in three dimensions, it is apparent that the problem becomes very difficult for objects of considerable detail.

As compared with measurements already done using other techniques at the Speckle Lab, phase retrieval may be extremely computationally intensive and far less robust. Despite the decreasing cost of computation, in many situations there will simply be no payoff for using this method in favor of others. However, situations arise where other methods cannot be used due to physical limitations, such as the need to put a reference mirror near the object or the failure of some methods as the angle between source, object, and detector approaches zero. The method I will discuss is monostatic (meaning that it does not require an angle between the illuminating

source and detector[1]), has range resolution which is independent of the total distance to the object, and does not require that the object be physically accessible. For applications such as imaging a distant satellite, all of these properties are desirable if not essential. Other applications, such as forensic imaging of the back of a gun barrel (or any long tube or deep crevice), call for monostatic imaging even if the object is small and accessible.

Development of effective algorithms for reconstructing objects from their Fourier magnitude thus extends the applicability of the Speckle Lab's family of methods. Furthermore, there is great promise that further algorithm development will yield improved robustness and images with more interesting detail.

1.3 Previous Work

Shirley, [1], and Paxman, [6], have each presented work on the problem of extracting Fourier phase related to Speckle Pattern Sampling. The work discussed in this thesis is a continuation of work done at the Speckle Lab by Shirley, Rahn, and Hallerman. Shirley's work presents an entirely support based approach, which allows a great deal of data reduction to be done before iterations begin. Paxman uses a very different method, much more akin to iterative transform techniques which will be discussed in Chapter 4. He performs a minimization of errors in the observed intensities over a parameterized space of allowable objects.

In both cases, results are presented that are encouraging, but fall short of enabling a general surface-imaging technique. Shirley presents a reconstruction from actual measurements of a 25- point array, obtained using somewhat specialized software to utilize knowledge of the 2D structure of the object. Paxman presents only simulations of a 6-point and an 18-point discrete object on a regular grid. It is desirable to build on these results so that interestingly fine samples of continuous scattering objects can

[1] triangulation methods, for example, do require such an angle

be obtained. Such improved performance would make the imaging concept described in the next chapter a very flexible technique for obtaining remote measurements.

1.4 Currently Implemented System

In this thesis, I begin to bridge the gap between previous results obtained and a realistic imaging system. My work consists of

1. algorithmic extensions of previous phase retrieval work
2. further application of convenient notation and theory which will allow support-based methods to be more carefully improved
3. the first simulations at the Speckle Lab with non-ideal Fourier magnitude estimates for both support and Transform methods
4. engineering of code which is capable of producing images and offers ease of continued development and experimentation.

Several features not contained in older programs have been included in a C implementation. Strategies for support algorithms have been explored further with some success but no definitive optimal strategy. I implemented a non-parametric Iterative Transform algorithm using an opaqueness constraint, which has not been done before to my knowledge. The implementation was successful: within a limited but interesting class of simulated objects, reconstructions had essentially no error. Most importantly, the two types of methods have been shown to have an interaction: their combination is shown to give better simulated reconstructions than either one alone. The real data reconstructions presented here are not of high quality, but provide a proof of concept. This work may also provide a basis for deciding whether future work is worth while; especially given the success of the Speckle Lab's other techniques, resources should only be allocated towards further development if the method shows sufficient potential.

Chapter 2

Background and Models

2.1 Speckle-Pattern Sampling

In the speckle-pattern-sampling (SPS) technique, developed by Shirley and also presented in [7], a coherently illuminated object surface can be reconstructed from the intensity pattern of light scattered as the illuminating laser frequency is tuned. Using the simplest possible models, we illustrate the mathematical results behind this imaging concept. The first result to derive is that scattered light in the far-field has a 3D Fourier transform relationship with the complex scattering function. Knowing the complex scattering function of an object surface gives us the shape and brightness of the surface—everything which we would like to measure. The second result is that we can easily obtain a grid of samples of this Fourier Transform magnitude which allow for rapid inverse-transformation using the FFT. For a much more complete treatment of Speckle-Pattern Sampling, including analysis of various sources of error, see [1].

Figure 2-1 schematically shows the essential geometry of the system: light comes from a source point $\boldsymbol{r}_s$ and illuminates a collection of object points with complex scattering function $g(\boldsymbol{r}_o)$. The scattered light is detected at point $\boldsymbol{r}_d$. If we have an approximately ideal point source of light, we can use a simple complex-exponential wave representation of the illumination at frequency ν, and can easily express the optical field at an object point. The contribution to the field at $\boldsymbol{r}_d$ made by light scattered from $\boldsymbol{r}_o$ then follows directly from knowledge of the scattering function at

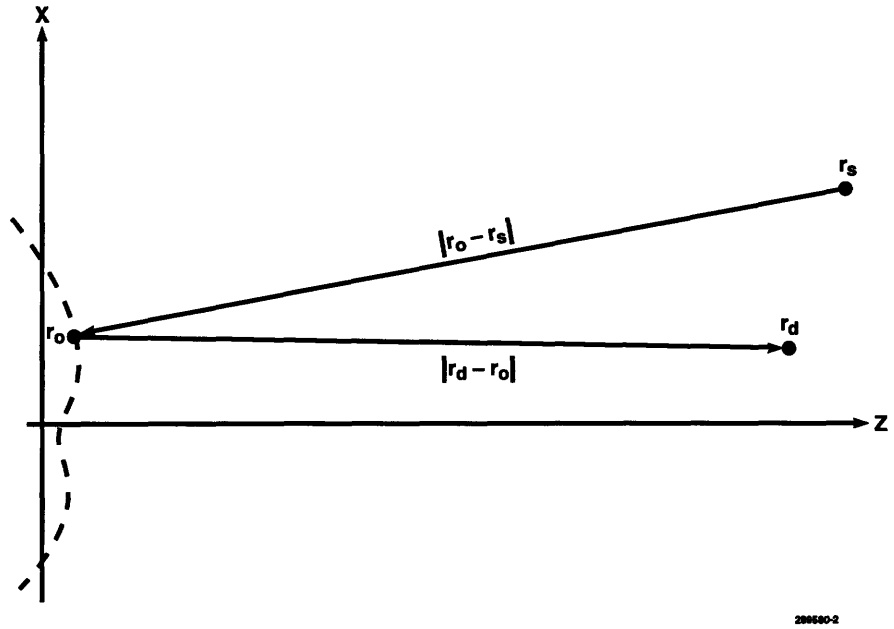

Figure 2-1: Geometry of the Speckle-Pattern Sampling technique for 3D imaging

that point. I will simplify things from the beginning by assuming that $|\boldsymbol{r}_s - \boldsymbol{r}_o|$ is approximately the same for all object points in the surface. This means that the magnitude of the optical field is uniform across the object, and we need to keep track only of the dependence of phase on position $\boldsymbol{r}_o$. This assumption reflects a simple far-field design, that $|\boldsymbol{r}_s| \gg |\boldsymbol{r}_o|$, so that $|\boldsymbol{r}_s - \boldsymbol{r}_o| \approx |\boldsymbol{r}_s|$ for all object positions of interest. Assuming a uniform magnitude, $|V_o|$, the field at the object is

$$V(\boldsymbol{r}_o) = |V_o| e^{\frac{i2\pi\nu}{c}|\boldsymbol{r}_o - \boldsymbol{r}_s|}, \tag{2.1}$$

which makes the contribution from $\boldsymbol{r}_o$ at $\boldsymbol{r}_d$

$$V(\boldsymbol{r}_d; \boldsymbol{r}_o) = |V_o| e^{\frac{i2\pi\nu}{c}|\boldsymbol{r}_o - \boldsymbol{r}_s|} g(\boldsymbol{r}_o) e^{\frac{i2\pi\nu}{c}|\boldsymbol{r}_o - \boldsymbol{r}_d|}. \tag{2.2}$$

The sum of contributions from all $\boldsymbol{r}_o$ is

$$V(\boldsymbol{r}_d) = |V_o| \int g(\boldsymbol{r}_o) e^{\frac{i2\pi\nu}{c}(|\boldsymbol{r}_o - \boldsymbol{r}_s| + |\boldsymbol{r}_o - \boldsymbol{r}_d|)} d\boldsymbol{r}_o. \tag{2.3}$$

Expressing $\boldsymbol{r}_s$ as $R_s\hat{\boldsymbol{r}}_s$, and $\boldsymbol{r}_d$ as $R_d\hat{\boldsymbol{r}}_d$ with $\hat{\boldsymbol{r}}_s$ and $\hat{\boldsymbol{r}}_d$ unit vectors, we design the system so that $R_s, R_d \gg |\boldsymbol{r}_o|$, and $\hat{\boldsymbol{r}}_s$ and $\hat{\boldsymbol{r}}_d$ are almost entirely in the z direction. Since R_s and R_d are large, we can approximate $|\boldsymbol{r}_o - \boldsymbol{r}_s|$ as $R_s + \hat{\boldsymbol{r}}_s \cdot \boldsymbol{r}_o$, and $|\boldsymbol{r}_o - \boldsymbol{r}_d|$ as $R_d + \hat{\boldsymbol{r}}_d \cdot \boldsymbol{r}_o$. This allows us to write Eq. (2.3) in a form which extends the familiar 2D Fourier optics result. That is,

$$V(\boldsymbol{r}_d) = |V_o| e^{\frac{i2\pi\nu}{c}(R_s+R_d)} \int g(\boldsymbol{r}_o) e^{\frac{i2\pi\nu}{c}(\hat{\boldsymbol{r}}_s+\hat{\boldsymbol{r}}_d)\cdot\boldsymbol{r}_o} d\boldsymbol{r}_o. \tag{2.4}$$

We have now obtained the first result: the integral has the form of a Fourier transform, and is multiplied by an overall scaling factor and a phase (the phase will not be detected at all, since we take only intensity measurements). It remains to find a convenient sampling of the Fourier domain. First rewrite Eq. (2.4) with $\tilde{g}(\boldsymbol{f})$ defined as $\mathcal{F}\{g(\boldsymbol{r}_o)\} \equiv \int g(\boldsymbol{r}_o) e^{-i2\pi \boldsymbol{f}\cdot\boldsymbol{r}_o} d\boldsymbol{r}_o$:

$$V(\boldsymbol{r}_d) = |V_o| e^{\frac{i2\pi\nu}{c}(R_s+R_d)} \tilde{g}(\boldsymbol{f}). \tag{2.5}$$

And so the frequency vector is,

$$\boldsymbol{f} = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = -\frac{\nu}{c} \begin{pmatrix} \frac{x_s}{R_s} + \frac{x_d}{R_d} \\ \frac{y_s}{R_s} + \frac{y_d}{R_d} \\ \frac{z_s}{R_s} + \frac{z_d}{R_d} \end{pmatrix}. \tag{2.6}$$

For a fixed optical frequency and source position, we can place an array of detectors to sample a surface of values $\boldsymbol{f}$. For example, placing a CCD array perpendicular to the z axis allows us to sample at a grid of x_d and y_d values with approximately the same R_d. Now remember that $\boldsymbol{f}$ has a dominant z component—to first order, $f_z = -2\nu/c$. The fact that $\frac{z_s}{R_s} + \frac{z_d}{R_d}$ is so much larger than the corresponding x and y components allows us to sample a volume of $\boldsymbol{f}$-points on a rectangular grid with a single, fixed 2D detector array using the following trick: tune ν by an incremental amount $\Delta\nu$ such that the change in f_x and f_y is negligible. The change in f_z is then $2\Delta\nu/c$, which is much larger—it can be large enough to sample a reasonable band of frequency space,

assuming the system is designed with appropriate ratios $\Delta\nu/\nu$, f_y/f_z, and f_x/f_z. In this way, we have approximately independent variation of f_x, f_y, and f_z through x_d, y_d, and $\Delta\nu$, respectively. Capturing a rectangular volume of samples appropriate to the FFT is very natural. Further analysis of the above assumptions, conditions for their validity, and compensation can be found in [1].

2.2 Optically Rough Objects

Most objects of interest are rough on an optical scale; optical wavelengths are less than one micron, and so many object surfaces have fine-grained variations in height on the order of a wavelength or greater. This roughness gives rise to scattered light which seems to form a random pattern. If we have no interest in resolving the grain of the object, but only of imaging the macroscopic structure, a natural approach is that found in in Dainty [8], where local variations in height are modeled as a random phase function. Shirley adapts this type of model to write

$$g(\boldsymbol{r}) = \alpha(x, y)\delta[z - h(x, y)], \tag{2.7}$$

where α is a delta-correlated, circular, complex random process, representing the limit of infinitely fine grain of surface texture. So although the scattering function is a random process, the height function, h, represents only the macroscopic or deterministic part of the surface height. The observed Fourier Transform will naturally be a realization of a random process. While tiny changes in the surface texture of an object may cause the intensity patterns to change considerably, the second-order statistics, or the "average" shape of speckles, will be the same. In practice, we find that even the periodogram, a simple FFT-based estimate of second-order statistics with no averaging of data, gives high-quality height maps of objects.

2.3 Magnitude Measurements

It is well known that only optical *intensity* can be directly measured electronically. For our technique, this means that samples of $\tilde{g}(\boldsymbol{f})$ are not obtained by our detector, but rather samples of its squared-magnitude. When we take the FFT, we get an estimate not of $g(\boldsymbol{r}_o)$, but of it's autocorrelation.

This issue can be dealt with by designing a more clever geometry which allows phase to be measured indirectly through intensity. Often a phase reference of some kind is used. Several schemes have been used at the Speckle Lab and elsewhere. One method which has produced successful measurements fits very easily into the framework developed above. This is to use exactly the same Speckle-Pattern Sampling method described, but place a curved mirror near the object during measurement. The mirror focuses the incoming illumination down to a point, which is now a small, bright point in $g(\boldsymbol{r}_o)$. We can now write the effective scattering function by adding an impulse to the original function: $g(\boldsymbol{r}) = g_1(\boldsymbol{r}) + \delta(\boldsymbol{r} - \boldsymbol{r}_{\text{ref}})$. It is easy to show that the autocorrelation function of g now contains four parts: a bright point at the origin, the autocorrelation of g_1 (also centered on the origin), and two copies of g_1: $g_1(\boldsymbol{r} + \boldsymbol{r}_{\text{ref}})$ and $g_1^*(\boldsymbol{r}_{\text{ref}} - \boldsymbol{r})$. Since copies are offset from the origin by $\pm\boldsymbol{r}_{\text{ref}}$, we can place this reference point so that the copies do not overlap with the other parts of the autocorrelation, assuming the object is of finite extent. An estimate of g_1 is trivially extracted from the autocorrelation, since copies are physically offset from other supported points.

SPS with a reference point is a good solution to imaging problems where it is physically reasonable to put a reference mirror near the object. If we cannot, one of many other ways to obtain a reference phase involves building another leg of beam path and interfering with a reference mirror, thus creating a virtual point near the object. Here too, there is the question of whether it is physically desirable to build a second leg and reference point. For applications where a phase reference is not desirable, the question is how do we obtain g from samples of $|\tilde{g}|$. Equivalently, how do we obtain g from an estimate of its autocorrelation?

The general problem of uniquely obtaining g from $|\tilde{g}|$ is of course impossible: any phase function, $\phi(\boldsymbol{f})$, can be combined with $|\tilde{g}|$ to generate a function $g' \equiv \mathcal{F}^{-1}\{|\tilde{g}(\boldsymbol{f})|e^{i\phi(\boldsymbol{f})}\}$. If we have some *a priori* knowledge of the structure of g, then certain of the phase functions, ϕ, can be deemed less valid, and we can hope to reconstruct the correct scattering function uniquely. Accomplishing the stated reconstruction is the subject of this thesis.

The question of uniqueness is discussed in some of the basic literature, for example [9], including the important difference between problems of one dimension and those of higher dimensions. In a number of 2D applications, phase-retrieval has been successfully used to uniquely determine g. Note that, for any $g(\boldsymbol{r})$, all shifted versions of both $g(\boldsymbol{r})$ and $g^*(-\boldsymbol{r})$ give the same Fourier magnitudes. We will consider these all to be in an equivalence class, and regard uniqueness accordingly. Previous applications typically assume that a sufficiently tight support of g is known *a priori*. The support constraints available in our problem are somewhat non-standard, but seem tight enough in comparison with common 2D supports used, for example, in [9]. Uniqueness is actually thought to be more easily achieved for higher dimensions [10].

Chapter 3

Support Methods

We have seen that our problem can be posed as extracting the support of a scattering function from an estimate of its autocorrelation. If we argue that much of the useful information in the autocorrelation function is contained in its support alone, we can take an entirely support-based approach towards reconstruction, with this support function as our starting point. Work done by Shirley along these lines is presented in [1, 11]. The basic observation is most simply explained assuming a scattering function, $g(\boldsymbol{r})$, composed of a finite number of scatterers. If

$$g(\boldsymbol{r}) = \sum_i g_i \delta(\boldsymbol{r} - \boldsymbol{r}_i), \tag{3.1}$$

the autocorrelation takes the form,

$$a(\boldsymbol{r}) \equiv (g \star g)(\boldsymbol{r}) = \sum_i g_i^* (g(\boldsymbol{r}) * \delta(\boldsymbol{r} + \boldsymbol{r}_i)), \tag{3.2}$$

where $*$ denotes convolution (or complex conjugation, when it appears as a superscript), and $\star$ denotes correlation: $v \star w$ is defined as convolution of $w(\boldsymbol{r})$ with $v^*(-\boldsymbol{r})$. We read (3.2) to say that a is a weighted sum of shifted copies of g. Then the autocorrelation function is nonzero wherever any of the copies in (3.2) are nonzero, except at points where the contributions from different copies cancel. The issue of cancellation is important, and will be discussed in Section 3.3. For the moment, we note that by

taking a larger record of speckle pattern samples, we can estimate the statistics of $|\tilde{g}|$ more accurately, and reduce the problem of cancellations. We have the (approximate) result that the support of the autocorrelation is determined in a straightforward way from the support of g. When this relationship can be inverted, then the assumption made at the beginning of this chapter, that much of the useful information of a is contained in its support, is justified.

This problem has been investigated in [12], where a convenient mathematical framework is developed. Fienup uses set notation and manipulations; for example, S and A are the sets of points in the supports of g and a respectively. His results are interesting but of limited use, as they treat the problem of finding exact, guaranteed *locator sets*, or sets including an acceptable solution S. Not much can be done in the general case, and those special cases which he treats do not apply to our problem.

Shirley, in [1], takes this approach further, exploiting the specific structure of S which results from assuming an opaque object. At least for perfect measurements, he has shown that acceptable reconstructions can be obtained. He has also accomplished reconstruction from real data for an object with known 2D structure.

3.1 The Shift-and-Intersect Method

3.1.1 Basic Algorithm

In this section, I describe an iterative method for improving estimates of S developed by Shirley, [1]. I start with some notation. The following intuitive definitions of set addition, subtraction, and negation are taken from [12]: For arbitrary sets B and C,

$$B + C \equiv \{d | d = b + c, b \in B, c \in C\}, \tag{3.3}$$

$$-B \equiv \{-b | b \in B\}, \tag{3.4}$$

$$C - B \equiv \{d | d = c - b, c \in C, b \in B\}, \tag{3.5}$$

$$C - b \equiv \{d | d = c - b, c \in C\}, \tag{3.6}$$

and

$$c - B \equiv -(B - c). \tag{3.7}$$

Note that these definitions are very different from the algebra of sets given, for example, in [13]. Some further explanation of set operations is given in Appendix A. We will see that many of the manipulations of support functions can be elegantly expressed using these definitions. As a first example, we recognize from Eq. (3.2), and by neglecting cancellations, that the support of the autocorrelation function, A, is the set of all differences of points in S, that is $S - S$.

We proceed by rewriting Eq. (3.2) as

$$A = S - S = \bigcup_{s \in S} (S - s) = \bigcup_{s \in S} (s - S). \tag{3.8}$$

So we already have a set which contains many shifted copies of S, some of which are flipped about the origin. Keep in mind that any shifted or inverted copies of the actual S are acceptable solutions to our problem.[1] We think of A as our first estimate of S, and call it S_1. Without making any special assumptions about S, we can also obtain a smaller locator set by taking $A \cap (A - a)$ for any $a \in A$. As it is an improved guess at S (containing fewer unwanted points), call this S_2. To see that it does contain a solution, define $s_1 \in S$ and $s_2 \in S$ such that $a = s_1 - s_2$, and

$$A \cap (A - a) = \left(\bigcup_{s \in S} (S - s) \right) \cap \left(\bigcup_{s \in S} (S - s - s_1 + s_2) \right). \tag{3.9}$$

Each union has a term with $s = s_1$ and one with $s = s_2$, so $S - s_1$ lies within $A \cap (A - a)$. By the same argument, using the second form in Eq. (3.8), $s_2 - S$ lies in $A \cap (A - a)$ as well. Each of these is a shifted (possibly flipped) copy of S, and is an acceptable solution to our problem. So this locator set actually contains two solutions. In general, there is no simple way of separating one of these copies from unwanted points in S_2. If additional shift and intersect operations can be made, they

[1] Furthermore, the ambiguity in the solution is both unavoidable and of little consequence to the utility of the method

will likely eliminate unwanted points rapidly, but how do we choose those shifts which will not also eliminate points of the desired copy?

We know that many shifts do exist, since any $A - a_k$, $a_k = s_1 - s_k$ contains $S - s_1$ by the same argument given for $A - a$. We can make an entirely parallel argument for $s_2 - S$, but this becomes repetitive, and will be omitted from the remainder of this thesis. If we could determine some of these a_k's, or equivalently, if we could find any $b \in S - s_1$ (and set $b = -a_k$), we could proceed with more shifts. That is, if even one additional point of $S_k \supset S - s_1$ can be determined to lie in $S - s_1$, we can obtain S_{k+1} through,

$$S_{k+1} = S_k \cap (A - a_{k+1}), \tag{3.10}$$

or more explicitly,

$$S_k = \bigcap_{i=1}^{k} (A - a_k). \tag{3.11}$$

These expressions are consistent with the above definitions of S_1, S_2, and a_k.[2] The key, then, is to use some *a priori* knowledge of the structure of S to identify desired points within S_k. In particular, if we can assume that certain vertical columns of S_k must contain *some* a_k, then we can hope to find one such a column which only has one point left. If so, this point is a guaranteed "true" point. For general objects, we would hope to no avail—there is no reason to think that singly-populated columns would arise. The specific objects we are imaging, on the other hand, are opaque, so each copy of S has at most one point per column, and it is plausible if not likely that singly-populated columns would arise at each iteration of the reconstruction. While I have done no analysis of the probability of these arising, my own and previous simulations have shown that shifting using singly-populated columns is very effective. If no guaranteed points can be found, it is still possible that these or any other *a priori* constraints might provide a list of likely shifts and some criteria for determining whether incorrect shifts have been made, allowing us to traverse a tree of candidate

[2]We can think of the base case in the recursion as $S_1 \equiv A$, or define S_0 to be the set of all permissible points in our space.

shifts. Other strategies for making shifts which seem promising will be discussed in the following sections.

3.1.2 Choosing Shifts and Converging

The shift-and-intersect method described above raises two questions: how can we choose the "best" shifts, and how do we decide that we've converged? Clearly the best shifts are those that are very likely to lie in $S - s_1$, but keep in mind that correct shifts which do not eliminate extraneous points are of no use. We could regard the algorithm "converged" when enough points have been eliminated so that we can find the surface heights, but on the other hand, as soon as there are few enough points that we can systematically check the consistency of remaining points, no more shifts are needed. I will discuss the final stage of the reconstruction before strategies for choosing shifts.

Convergence Through Consistency Check

Shirley has previously developed a reasonable algorithm for testing and sorting a set of points to find a large subset which is consistent. Consistency is defined here to mean that any pair of points, p_1, p_2, in the set correspond to a vector difference $\boldsymbol{p}_2 - \boldsymbol{p}_1$ which is in the support, A. This may be the most that the support can give us: if there are still spurious points, there is no straightforward way to remove them using the autocorrelation support alone. In practice, there are few if any spurious points in this solution, although for imperfect A estimates it is common for points to be missing from the reconstruction.

Ideally, we would like an algorithm which extracts the largest possible set of consistent points from S_k. One very impractical way to do this is to enumerate all possible subsets of S_k, and choose the largest one which does not contain an inconsistent pair. This method is useful only in pointing out that it is possible to find an optimal, if not unique, solution in finite time. S_k with M points has a number of subsets which increases exponentially in M, and so the impracticality of the method.

Shirley's consistency matrix reduction algorithm (CMR), on the other hand, can be implemented to run in N^2 time, but does not entirely guarantee that the consistent set is the largest which can be extracted. This may only be a small drawback, since we can take the set of consistent points to be a series of shifts to be executed, which will eliminate the bulk of the unwanted points and leave points missed by the CMR algorithm. I have not studied the behavior of this modified version in detail, but an example is presented below.

The CMR algorithm conceptually starts by forming a consistency matrix, or a 2D table of ones and zeros representing the pairwise consistency of remaining points p_i in the locator set. The matrix is $M \times M$ and the i, j^{th} entry is 1 exactly when $p_j - p_i$ is in the autocorrelation support. The matrix is then reduced in (at most) M iterations. At each iteration, we find the point which is inconsistent with the fewest other points, and eliminate this point from the set, crossing out the corresponding row and column of the matrix. If all points are consistent, we return the set without further reduction. Versions of the algorithm existing before were implemented in Mathematica, and were not written with efficiency in mind. I have so far found and tested the version presented as C code in Appendix B, which is equivalent and achieves the claimed M^2 performance in time while avoiding the M^2 memory overhead of the original Mathematica implementation. Further improvements may be possible.

Table 3.1 shows a consistency matrix which I have found which will not necessarily be reduced to give the maximum consistent set. A table of the algorithm's progression is shown in Table 3.2. Note that the columns are already ranked in order of highest consistency at all iterations, although several "ties" in consistency occur during the reduction. The algorithm gives points p_1 and p_2 as output, although p_5 is consistent with both, and so is "overlooked" by the algorithm. This is a case where shifts for all consistent points (p_1 and p_2) and retrying the consistency matrix trivially gives the correct answer. There may be cases in which this method breaks down as well, and future study can be given to this issue, to the breaking ties, and to whether alternative algorithms obtain the optimum set in polynomial time.

Choosing Shifts

Prior to this thesis, shifts have been chosen randomly among those points in singly populated columns. If each of these is likely to be a good shift, this method may be good enough; simulation results obtained prior to this thesis gave perfect reconstructions for ideal autocorrelation estimates. In fact, even the results of just a few shifts were already recognizable.

As false singly-populated columns come up (due to thresholding difficulties, noise, etc.) we may want to choose shifts more carefully. To motivate this, we look at a simulation of how the random-choice strategy does starting with imperfect A. An estimate which is a noiseless $|a|$ thresholded at one thousandth its peak value was used. The original 2D object support of a triconic target is shown in Figure 3-1, and Figures 3-2 and 3-3 show the results after 2 and 6 shifts in a series, respectively. The shadings in the latter two figures represent the status of columns: black indicates an empty column, dark grey a singly-populated column, and light grey a column with multiple points. The resulting sets are small enough to perform CMR, which leaves us with Figures 3-4 and 3-5, respectively. If we consider these results unacceptable, they demonstrate that, even without noise, thresholding can cause the original algorithm to fail. If we start over and make different random choices, we can obtain acceptable, although imperfect reconstructions. One is represented in Figure 3-6 after only two shifts, and in Figure 3-7 after CMR. The true object is represented analogously to Figure 3-7 in Figure 3-8. After trying with several random seeds, I concluded that neither successes nor failures were rare for this object and threshold.

	p_1	p_2	p_3	p_4	p_5
p_1	1	1	1	0	1
p_2	1	1	0	1	1
p_3	1	0	1	1	0
p_4	0	1	1	1	0
p_5	1	1	0	0	1

Table 3.1: Consistency matrix reduced unsuccessfully.

Iteration	Points Consistent With					Result
	p_1	p_2	p_3	p_4	p_5	
1	4	4	3	3	3	Drop p_5
2	3	3	3	3	-	Drop p_4
3	3	2	2	-	-	Drop p_3
4	2	2	-	-	-	Converged: p_1, p_2

Table 3.2: Progression of unsuccessful CMR iterations.

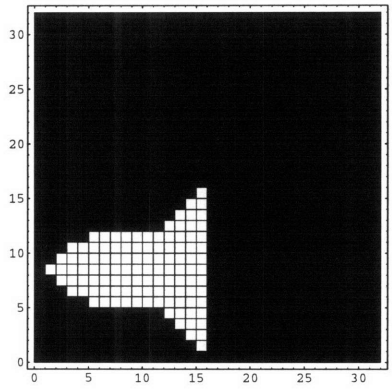

Figure 3-1: 2D support of true object.

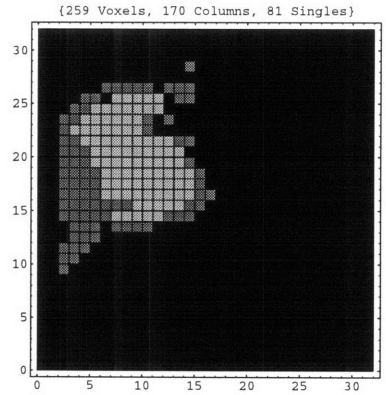

Figure 3-2: Result of two unlucky shifts, randomly chosen in singly populated columns. Object Shape is generally intact, but not yet separated.

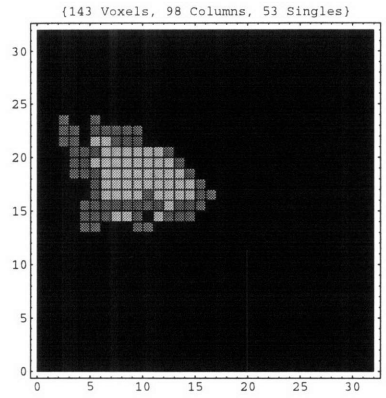

Figure 3-3: Result of six unlucky shifts. Significant loss of information has resulted from the chosen shifts.

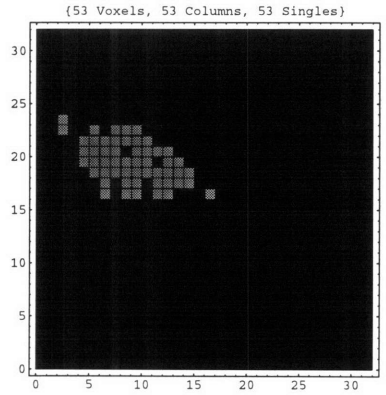

Figure 3-4: Result of CMR after two unlucky shifts. The shape of the object is not intact.

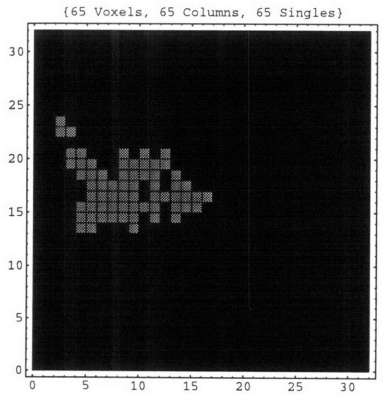

Figure 3-5: Result of CMR after six unlucky shifts. Again, the shape is not intact.

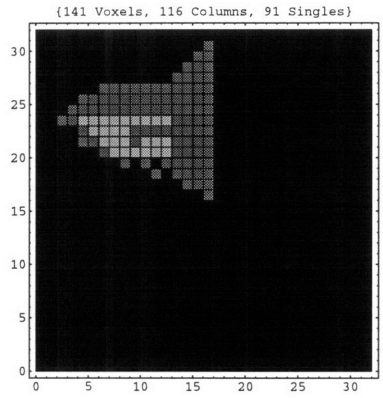

Figure 3-6: Result of two lucky shifts. Most columns seem already to have converge to a singly-populated state.

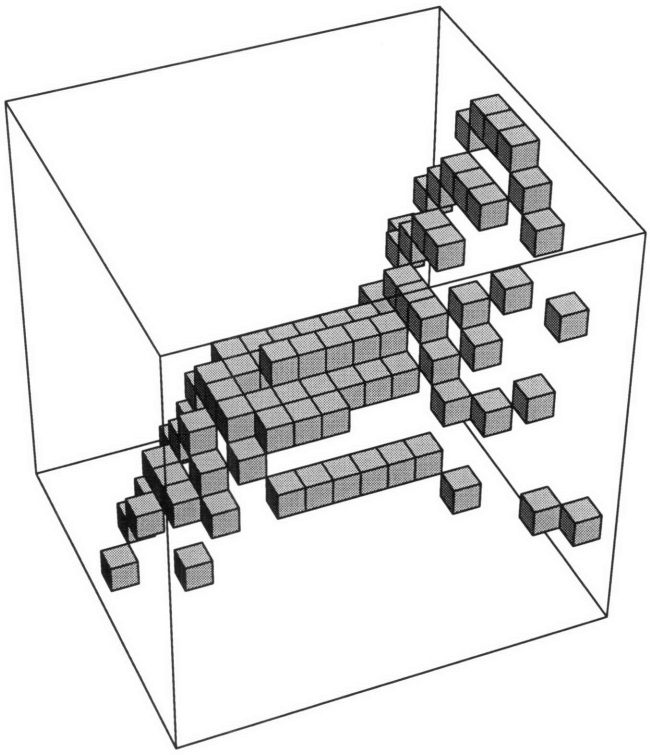

Figure 3-7: Representation of an acceptably converged object obtained from CMR after two lucky shifts. While some points are missing, image quality is fairly good.

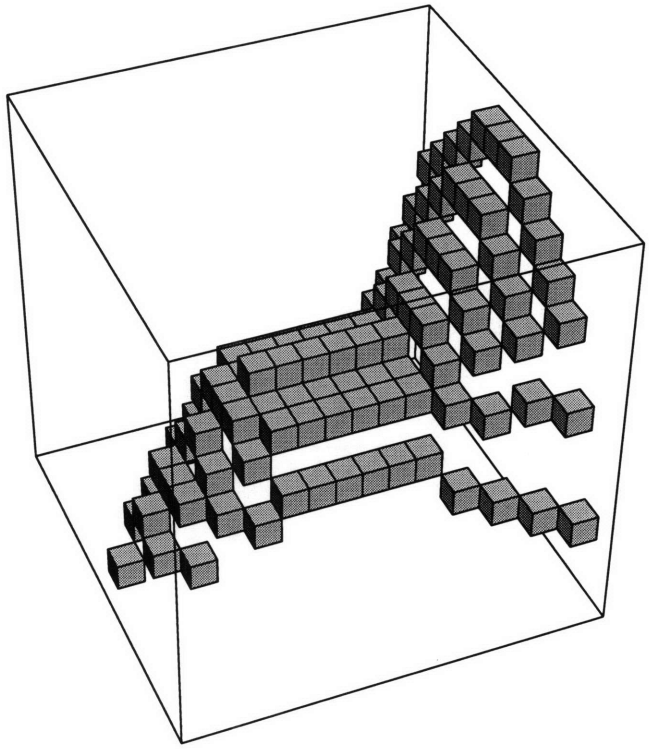

Figure 3-8: Representation of true object.

The CMR algorithm actually has relevance to choosing early shifts as well as for putting the finishing touches on the reconstruction. Although the original CMR was intended to be done on the entire set S_k, I observed that there is nothing preventing us from reducing a much smaller set of points on the basis of consistency. In particular we can use CMR on the set of all points in singly populated columns. If these are in short supply, we can throw in doubly or triply-populated columns, or any other points that we think are likely to be in $S - s_1$. The advantage of columns with few points is that they should guarantee that even a reasonably sized consistency matrix contains several points in the true support. Even with noisy estimates, then, many of these points should be determined as consistent by CMR, and thus good candidate shifts. Among these, we can choose all of them as shifts, or any which seem better than others. For example, if we don't mind deviating slightly from our exclusively support-based approach, we can pick those of the consistent points $\boldsymbol{p}_i$ with greatest $|a(\boldsymbol{p}_i)|$. This is unattractive because of the asymmetrical importance of $|a(\boldsymbol{r} - \boldsymbol{a}_1)| = |a(\boldsymbol{r})|$ over all of the other $|a(\boldsymbol{r} - \boldsymbol{a}_k)|$. (Since $\boldsymbol{a}_k$ are defined to correspond to a_k, the support of $|a(\boldsymbol{r} - \boldsymbol{a}_k)|$ is $A - a_k$. While juggling set and functional notation is a bit clumsy, the correspondence is just the obvious one). The point with the highest "combined" $|a(\boldsymbol{p} - \boldsymbol{a}_k)|$'s seems the best bet, but this begs the question of how to combine them. Proceeding in this direction seems to call for a more thorough generalization of the CMR idea.

Finally, we may want a method for choosing first shifts so that a bright "copy" of g in the sum (3.2) is chosen among the many possible copies. Clearly, there will be brighter and darker copies which correspond directly to the larger and smaller values of g_i in that sum. By setting the threshold high enough that even most true points in A are undetected, we obtain an estimate A much smaller than the true support which tends to contain points corresponding to pairs of bright points. If we then do a complete consistency reduction on this small set, we may obtain a handful of points in a bright "copy" of S. We can then remember these points, start over again with a lower threshold and new A and S_1. The points which we remembered still lie in S_1, but this time we have regained points in that bright copy which fell below the earlier

threshold. If we choose these consistent points from the previous stage as our first few shifts now, we should have an excellent start in this stage. In future sections, we will refer to this strategy as the variable-threshold approach.

3.1.3 Two-Dimensional Support Considerations

The singly-populated column strategy is simple and has been effective. It does assume that certain columns must contain supported points, and so some care should be taken in making this assumption and in deciding for which columns it is valid. I will put forward some guidelines which may improve existing methods of choosing shifts.

Before proceeding, it is important to note that those algorithms which simply assumed that all populated columns in S_k would be supported columns worked fairly well. In simulations done prior to this thesis, when random choices were made of the singly-populated columns, correct shifts were often chosen, and entire reconstructions could be made with no shortage of singles. However, simulations confirm that after one or two iterations, it is possible to identify singly-populated columns that actually lie outside of the 2D projection of the desired copy of S. That is, these columns do not fall within the 2D support of the object, and so we cannot conclude that the point in each is a guaranteed true point. In fact, shifts corresponding to these points are bad shifts, and cause loss of desired points from the reconstruction. This differs from results previously mentioned because it involves a perfect A estimate. Bad shifts are obtained for unsupported columns lying both on the edge and in the center of S_k, for an object derived from that of Figure 3-8, but with several points removed. Points were removed so that the object had a "hole" in its 2D support.

One way of obtaining a guideline of which columns are guaranteed assumes that we already know the 2D support of S. Call the 2D support P. In practice this may be obtained from a conventional 2D imaging system, or estimated from $|\tilde{g}|$. Given a locator set, S_k, we can take its 2D projection. Call this projection P_k. The thing to notice now is that for any S_k tighter than A, there are columns which fall within all of the shifted copies of P that are contained within P_k. This must be true since even $P_k = P_1$, equal to the 2D projection of A, has one such column—the origin. So by

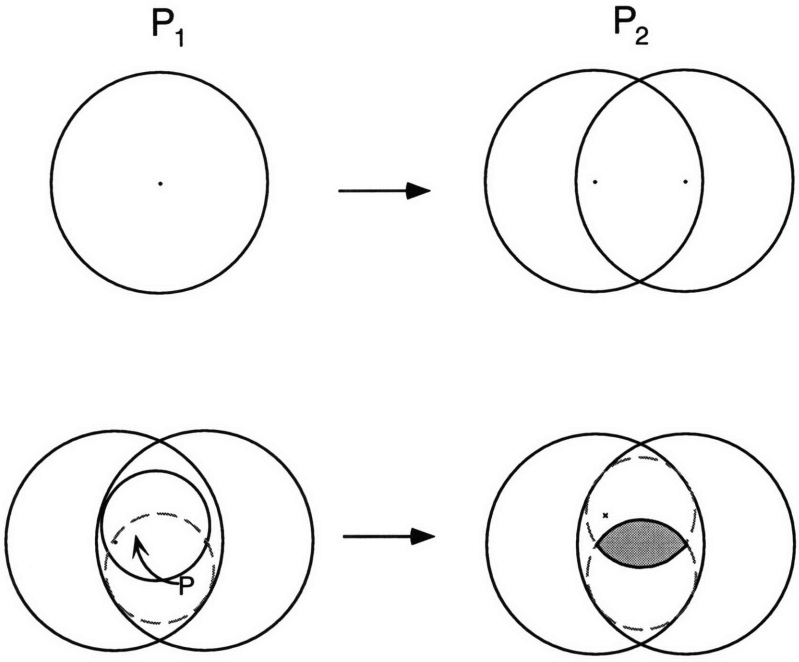

Figure 3-9: Conceptual sketch of a method for finding safe columns.

taking the intersection visualized in Figure 3-9 of all of the copies of P which do not conflict with P_k, we can obtain a set of "safe" columns. A looser estimate of P will mean a smaller intersection, but not necessarily no intersection, and so the method may even be effective when P is known imprecisely.

We would prefer not to need a measurement of P. In some sense, we can define safer columns as those which are not inconsistent with any other columns. Interpreting consistency of columns as the presence of their vector difference in the 2D support (analogously to our definition for points), we need only the projection of A to judge consistency. We might then restrict ourselves to shifts in the set of columns consistent with all other columns in P_k. As depicted in Figure 3-10, this is the set of columns v such that $P_1 - v$ contains P_k. If the correct, partially reconstructed copy of P is called P_r, then the above v are guaranteed to be within the intersection,

$$\bigcap_{p \in P_r} P_1 - p.$$

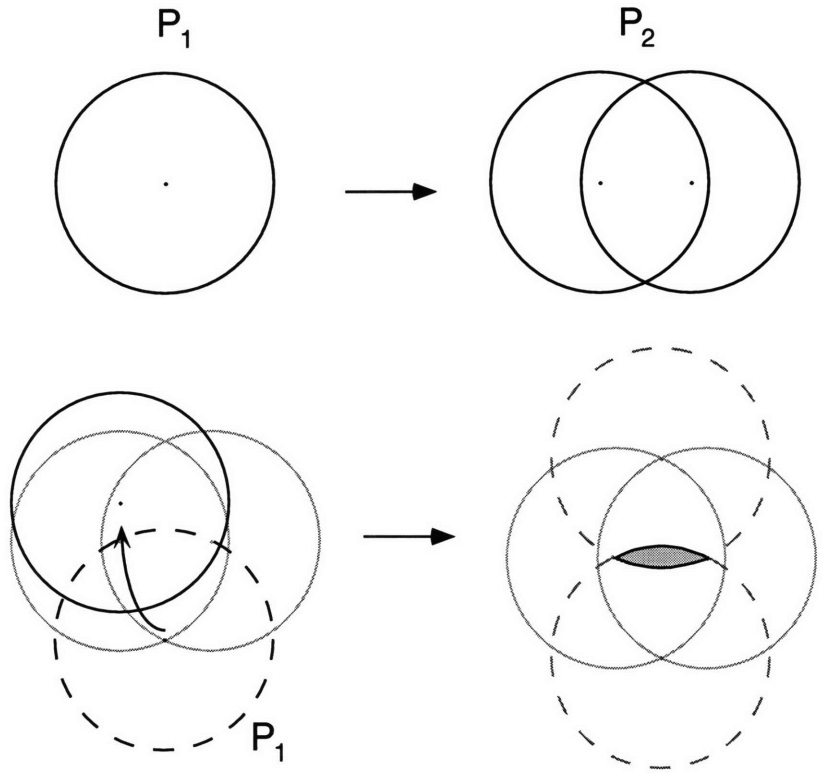

Figure 3-10: Conceptual sketch of another method for finding safe columns.

This gives us some assurance that the set is tight enough to provide some safety. It may be worth exploring whether any reasonable assumptions about S imply that the resulting columns, p are guaranteed to be in P_r.

We should keep in mind that safety is not our only criterion for good shifts; there may be some trade-off between safe shifts and ones which rapidly generate tight locator sets. While the above methods may be helpful, the partial consistency matrix method which has been discussed seems more powerful and as easy to implement. They can be used together, if desired.

3.2 A New Interpretation—Deconvolution

In this section, I would like to show an interpretation of, or alternate way of arriving at, the shift-and-intersect method. The interpretation comes with natural potential generalizations. Here we assume that at the k^{th} iteration, we have obtained an imperfect guess, V_k, and would like to find an improved guess, V_{k+1}. So far this is a very common and general approach. Of course there are many types of imperfect guesses that one might have. Assume that V_k is a subset of a correct solution S: $V_k \subset S$, $S - S = A$. This implies that

$$(V_k - S) \subset (S - S) = A. \tag{3.12}$$

It is then reasonable to use Eq. (3.12) to obtain an approximation to S associated with V_k. In particular, it makes sense to find the largest set W_k such that

$$V_k - W_k \subset A. \tag{3.13}$$

To interpret the meaning of W_k, notice that Eqs. (3.12) and (3.13) imply $S \subset W_k$ as long as $V_k \subset S$. Then W_k is a locator set in which we can search for additional points to include in V_{k+1}. We have chosen the smallest set guaranteed to contain a solution for S to Eq. (3.12) and so in some sense have fully utilized V_k. What we will see next is that the set W_k can be generated easily from V_k, and furthermore is

equivalent to the intersection of shifted copies of A already described. This provides some additional assurance that intersecting copies of A is a reasonable way to proceed at each iteration. It is *not* intended as a proof that we have used all of the information available in A and V_k. In fact, we tend to assume that all of the information needed to generate S or $-S$ uniquely (to within a shift and flip) is contained in A and *a priori* assumptions alone, so if all of the information were utilized, we would converge to $V_1 = S$ on the first iteration.

We can refer to this method of obtaining W_k from V_k as deconvolution since, in functional notation, our definition (3.13) becomes: $w_k(\boldsymbol{r})$ is the largest support function such that

$$(v_k \star w_k)(\boldsymbol{r}) \leq s_a(\boldsymbol{r}), \tag{3.14}$$

where $\star$ is again used for correlation and s_a is the support of a. Here is the demonstration that W_k, the "deconvolution" of A and V_k is equivalent to the result of the shift and intersect method with shifts obtained directly from V_k.

Without loss of generality, assume that the origin is one of the points of V_k (otherwise just translate all points so that one lands on the origin). Then, we also have, from (3.13), that $0 - W_k = -W_k \subset A$. That is, we only have to check points within A to see whether they satisfy (3.13). But this already suggests the whole shift-and-overlap method. By the same reasoning, for each point x_{kj} in V_k,

$$x_{kj} - W_k \subset A \Longrightarrow W_k \subset -A + x_{kj} = A + x_{kj}, \tag{3.15}$$

and furthermore, all of these equations taken simultaneously imply (3.13). But the largest set satisfying $W_k \subset A + x_{kj}$ for all j is of course

$$W_k = \bigcup_j (A + x_{kj}). \tag{3.16}$$

This is precisely the shift-and-intersect formula. The question remains: how do we obtain further shifts, or an improved $V_{k+1} \supset V_k$, from W_k? Exactly as in the shift-

and-intersect method, *a priori* object constraints can be used to choose guaranteed, or at least the most likely shifts.

Note also that the deconvolution can be interpreted as a standard matched filter with a threshold at the output. Eq. (3.14) dictates that $w_k(\boldsymbol{r}_i) = 1$ only if the support function $v(\boldsymbol{r}+\boldsymbol{r}_i)$ lies entirely inside of the support s_a. This means that with threshold, t, equal to the number of points in V_k, w can be defined as

$$w(\boldsymbol{r}) = \left\{ \begin{array}{lll} 1 & , & (v * s_a)(\boldsymbol{r}) > t \\ 0 & , & (v * s_a)(\boldsymbol{r}) < t \end{array} \right\}. \tag{3.17}$$

It may be useful to generalize the above matched filter in the case of noisy autocorrelation measurements. If the threshold is set some number below the size of V_k, then the shift-and-intersect procedure will "forgive" each point of that number of "bad" shifts which may disagree with it. While this will allow more unwanted points to persist, the true points will have a statistical advantage, and so highly forgiving algorithms may have good asymptotic performance. Going further along the lines of a matched-filter, we can think of a shift and add method as a matched filter searching for high return, or "brightness," at particular points, which we have determined are supported. We define the real valued W_k as,

$$W_k(\boldsymbol{r}) = (A \star V_k)(\boldsymbol{r}). \tag{3.18}$$

A is the "brightness," not just the support, of the autocorrelation, and V_k can be taken as the fragment of the desired support obtained so far. If we do restrict V_k to be a support function, then we can still expect fairly speedy computation.

3.2.1 Other Generalizations

Part of the power of the shift-and-intersect and the consistency matrix methods is that it is fairly easy to generalize their forms. Looking at a broader class of methods, we can decide if improvements on the original method are possible and at what expense. We have just seen one path for generalization. Now, we can attempt a more inclusive

description of reasonable solutions to our problem: at each iteration, we have a set of points, $\{a_i\}$, likely to be in the support we are trying to reconstruct, and a function of $\boldsymbol{p}$ related to the likelihood that there exist supported $\boldsymbol{r}_i, \boldsymbol{r}_j$ with $\boldsymbol{p} = \boldsymbol{r}_j - \boldsymbol{r}_i$. We ultimately make a binary decision at each point—whether that point is to be considered supported based on its consistency with the trusted points. In accordance with this description, we can think of $b(\boldsymbol{p}) = B(|a(\boldsymbol{p})|)$ as the brightness measure of a point in the autocorrelation; any monotonic function of $|a|$ is reasonable. A binary decision at $\boldsymbol{r}$ based on the consistency of points with the assumed set is then

$$F(b(\boldsymbol{r} - \boldsymbol{a}_1), b(\boldsymbol{r} - \boldsymbol{a}_2), ...)$$

For the shift-and-intersect, we can think of B as a simple thresholding operation and of F as a logical AND. One simple variant of this is to threshold with B, and require that at most l of the thresholds failed. This reduces to shift-and-intersect for $l = 0$, and is the same as the "forgiving" filter described in the previous section. With the same threshold, higher l it will reduce the number of unwanted points in S_k more slowly than a less forgiving F, but it may still be useful if noisy measurements make it likely that even true points need forgiveness.

One may want to go further and retain the information normally thrown away with a thresholding B. In other words, should a thresholding operation be performed on some combined measure of a point's consistency with *all* assumed a_i? This suggests we choose $B(|a|)$ to be some non-decreasing function of $|a|$ such as $|a|$ or $\log(|a|)$, and F to be some accumulating operation which is symmetric and non-decreasing in its arguments like summing, maximum, minimum, etc., and then thresholding. Shift-and-intersect then gives rise to variations like shift-and-add, or shift-and-multiply. In the next section, we discuss generalizations in the context of performance metrics for support methods.

We could proceed along the same lines in generalizing the consistency matrix—using real numbers instead of ones an zeros would parallel the generalization of B. But without further understanding the failures in performance of the CMR algorithm,

and outlining performance metrics for generalizations, they would be very *ad hoc* and arbitrarily chosen.

3.2.2 Performance Metrics

Although some intuition can be built by simply playing with variations on the shift-and-intersect method, it is appropriate to carefully define performance metrics. These serve as a basis for precise empirical comparison and a starting point in a search for theoretically better or optimal methods. Ultimately of interest is whether we can get an acceptable reconstruction using a particular method or strategy, but this does not help us decide how well we are doing at each iteration. If we think of finding all supported points at a given iteration as a set of stochastic detection problems, there is no need to reinvent any terminology. At each point in autocorrelation space we either detect a point or not. Natural performance metrics are then the probability of detection (given there is a true point to detect), P_D, and probability of false alarm, or probability that a false detection is made given there is no true supported point, P_{FA}.

Extensive modeling of object surface characteristics which will effect the above probabilities is beyond the scope of this thesis. Clearly, the detection problems at different points are interrelated, but we can obtain understanding without exploring these dependencies. Assume only that α is random, h is deterministic but unknown, and that there is an unknown set of true supported points of a determined by h. We imagine that there are functions representing the conditional probability distributions of $b(\boldsymbol{r})$ given that $\boldsymbol{r}$ does represent some true pair in the autocorrelation. Then a familiar and fairly general graphic of a single point's detection problem is shown in Figure 3-11, and the hypothesis, T, is that $\boldsymbol{r}$ is a true point in the autocorrelation support. Although specific density functions can give rise to arbitrary decision regions, we often end up comparing b to some threshold c. The shaded regions in the graph correspond to the conditional probabilities of error, $1-P_D$ and P_{FA}. There is a trade-off involved since reducing one tends to increase the other. Unless we derive the forms of $p_{b(\boldsymbol{r})|T}$ and $p_{b(\boldsymbol{r})|\bar{T}}$, we do not actually use use this graph to choose an optimal

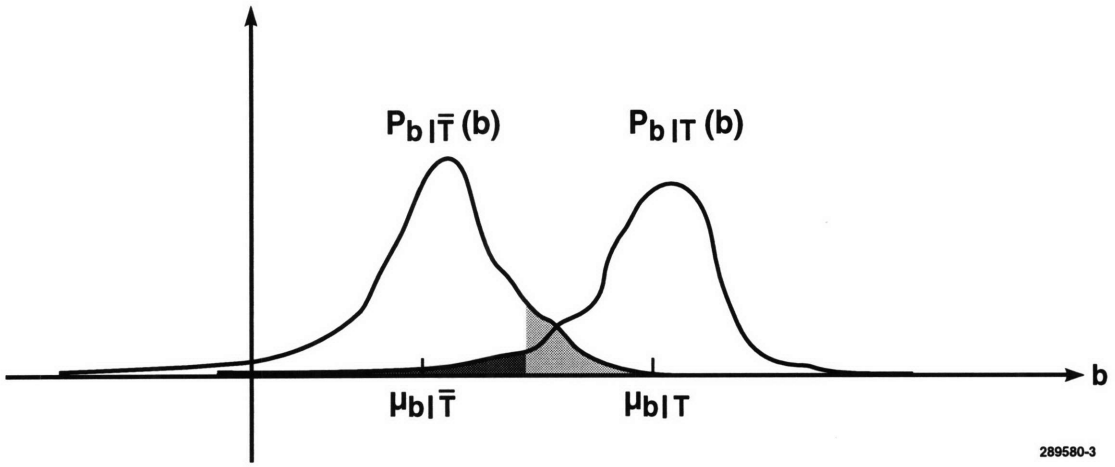

Figure 3-11: Decision based on a single value of the autocorrelation.

threshold, but it does give understanding which is useful as we extend to multiple measurements. In practice me may use empirical receiver operating characteristics, which simply plot the P_D,P_{FA} relation.

After several shifts, we would like to simultaneously decide whether all of the points $\boldsymbol{r} - \boldsymbol{a}_i$ corresponding to shifts, $\boldsymbol{a}_i$, are supported. For only two $\boldsymbol{a}_i$, it is easy to extend the graphic of Figure 3-11 to the contour plots shown in Figure 3-12. The decision rule will now be based on 2D regions of this space, not 1D intervals, and our task is to pick a boundary which discriminates the TT case from the other three cases. Again, we can assume a form for the four density functions and solve the problem exactly. Even less specific information such as correlations between points in b could help, but I have avoided this type of solution because it is sensitive to the noise model and lacks simplicity and generality.[3]

It is most obvious to think of the shift-and-intersect as thresholding based on the marginal, as opposed to the joint, probabilities. Another way of looking at it is to draw the curves $\min(b(\boldsymbol{r}), b(\boldsymbol{r} - \boldsymbol{a}_2)) = c$ onto the contour plot as in Fig (3-13). Asking if both brightnesses are above the threshold is equivalent to asking whether their minimum is, so each contour represents the boundary between the two decision

[3]In the future, it may be worth it, for example, to at least correct for pronounced systematic errors such as sidelobes coming from a bright point.

regions corresponding to the shift-and-intersect method for a particular threshold, c. We can compare to shift-and-add (which actually becomes shift-and-multiply if the function b is chosen as the log-magnitude instead of just the magnitude); now each diagonal $b_{\text{sum}} \equiv b(\boldsymbol{r}) + b(\boldsymbol{r} - \boldsymbol{a}_2) = c$ will bound the decision regions arising from threshold c. If we imagine taking the conditional derived distributions of b_{sum}, that is the diagonal marginal of the joint distribution, we see that discrimination of the $T\bar{T}$ and $\bar{T}T$ cases from the TT case will generally be expected to get worse than for the the standard marginals: the probability "blobs" will be equally broad in either case assuming there is no correlation between points, but the distance between the centers of the blobs drops by a factor $\sqrt{2}$. On the other hand, the $\bar{T}\bar{T}$ is expected for the same reason to be better discriminated by b_{sum}. We might expect the ideal boundary to have a softer corner in its boundary curve than the shift-and-intersect, but in general the original method holds up very well considering it has the advantage of greater simplicity and efficiency over almost any variant. A set of contours corresponding to the optimum Bayesian decision for a particular fabricated set of jointly gaussian distributions is given in Figure 3-14. Note that we can get quite different looking "ideal" decision boundaries by changing the assumed parameters, such as *a priori* probabilities of the four cases.

Some preliminary simulations have been done using shift-and-multiply, but this is not a promising generalization since it sacrifices the ability to discriminate the more difficult cases ($T\bar{T}$ and $\bar{T}T$ in the above example, and generally the cases with mostly T's and only a few $\bar{T}$'s) in favor of better performance on the easy cases.

Observing the performance metrics described above for simulated or otherwise accurately known objects is straightforward. Once we have obtained an ideal support of a to compare against, the probabilities are just estimated as ratios of points correctly and incorrectly detected. We can look at a particular true point $\boldsymbol{r}$ over a number of different pseudo-random $\alpha(x, y)$ to observe its P_D, or use a less careful estimate by taking as our ratio, P_D, the fraction of all true points in A which are detected for a single α. In the latter case, we are assuming that different points are independent even though they are not. This method judges our thresholding scheme relative to the

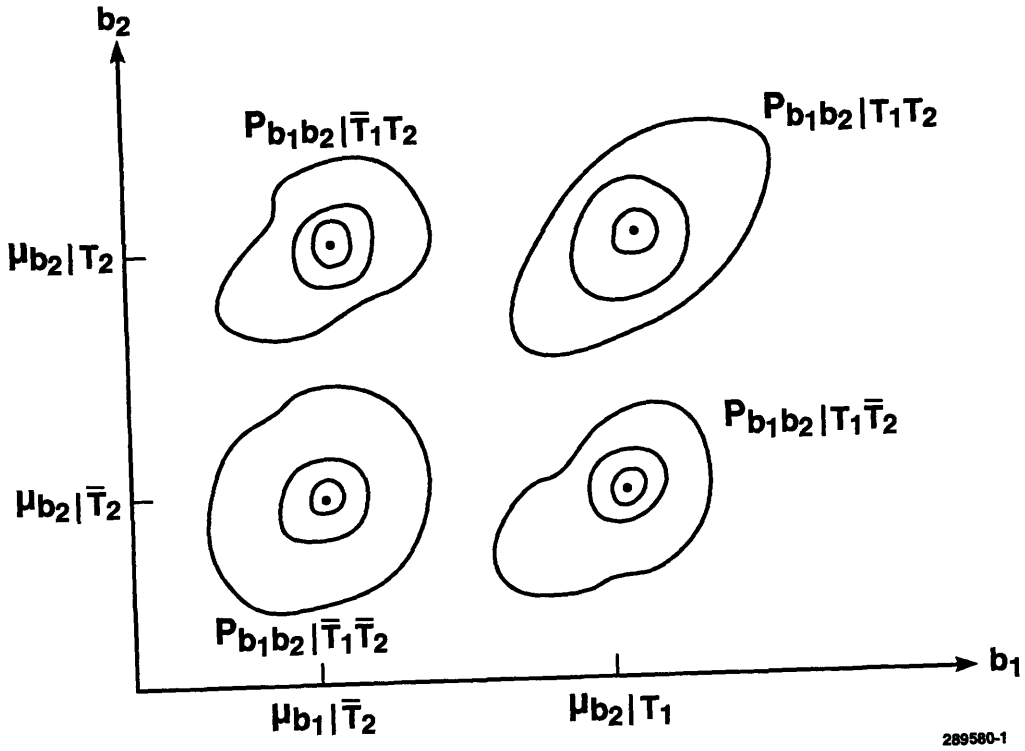

Figure 3-12: Decision based on two values of autocorrelation.

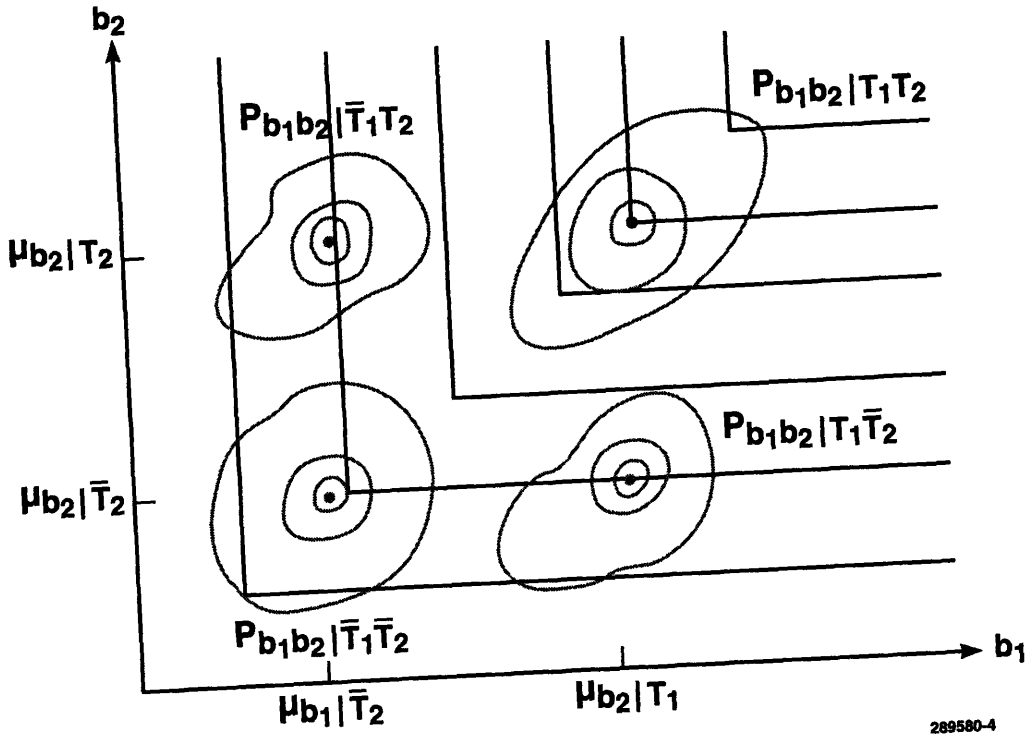

Figure 3-13: Decision boundaries corresponding to shift-and-intersect method.

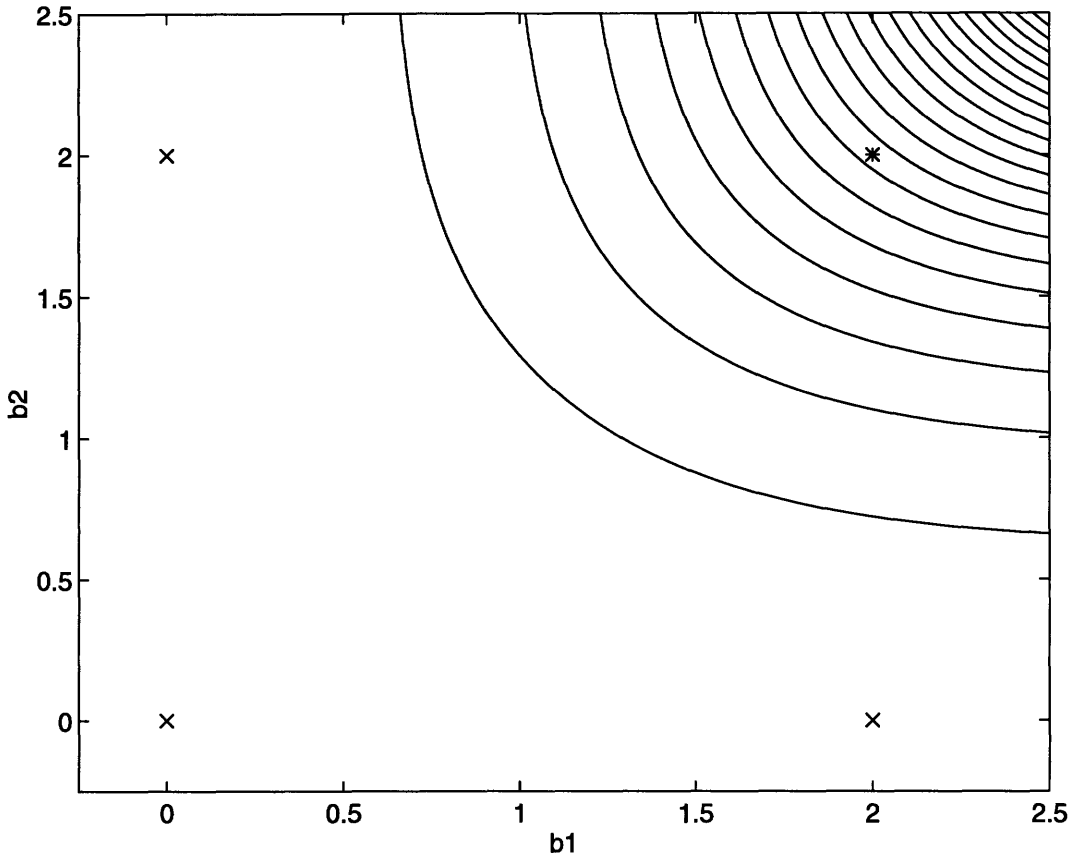

Figure 3-14: Some Bayesian decision boundaries for a simple distribution.

same shift-and-intersect iterations performed on ideal A, and says nothing about how good our strategy for choosing shifts is. We can define analogous ratios which are performance metrics for the shifts, although it is somewhat less natural to think of these ratios as probabilities. The fraction of points in the copy of S to be reconstructed which are correctly detected by the shifts $\{a_i\}$ is analogous to P_D, and the fraction of points not in this copy falsely detected is analogous to P_{FA}. To interpret this as a probability, we think of the the macroscopic height h is one of many possibilities each with a certain likelihood, or probability. To get a good idea of performance of shift strategies, we should simulate many different height functions and average the above ratios on this sample space.

Other ratios of practical interest are the fraction of points in the best remaining copy of S which are detected at an iteration, and the fraction of unwanted points not yet eliminated, both resulting from the combined shift strategy and threshold.

3.3 Implementation of Autocorrelation Support

We have until now assumed that an estimate of A is available. Obtaining this estimate from measured speckle intensity is in fact a non-trivial part of engineering a useful technique. From noisy measurements of the sampled Fourier intensity, we would like to decide which voxels in the autocorrelation function represent true pairs of scatterers in the sum (3.2). The sum can be expanded further:

$$a(\boldsymbol{r}) = \sum_i \sum_j g_i^* g_j \delta(\boldsymbol{r} + \boldsymbol{r}_i - \boldsymbol{r}_j), \tag{3.19}$$

and so the autocorrelation brightness at point $\boldsymbol{p}$ is

$$\sum_{(i,j)\to\boldsymbol{p}} g_i^* g_j, \tag{3.20}$$

where the sum is taken over all pairs (i, j) such that $\boldsymbol{p} = \boldsymbol{r}_j - \boldsymbol{r}_i$. Typically, estimating the support consists of two parts: estimating the "brightness" of a at each voxel,

and then finding a suitable threshold to determine whether this brightness is approximately zero. The first of these turns out to be a spectral estimation problem: given measurements of the stationary process, $s(\boldsymbol{r}) \equiv |\tilde{g}(\boldsymbol{r})|^2$, we would like to estimate $|\mathcal{F}\{s(\boldsymbol{r})\}|^2$. Cancellations occur whenever the above sum falls below the threshold used. As our threshold approaches zero the number of cancellations will fall to zero except in very special cases. However, noise prevents us from lowering our threshold too much, since any noise causes A to become hopelessly cluttered as the threshold drops to zero.

3.3.1 Autocorrelation Brightness Estimate

Spectral estimation is a very well-studied problem, and many of the common 1D results have straightforward extensions to our 3D problem. The periodogram or averaged periodogram are natural choices for speed and ease of implementation. The latter allows us to measure and utilize more data than for the simple periodogram and thus reduce some types of noise.[4] If noise in the support estimate becomes a significant obstacle to the imaging technique, it is worth considering estimates which show lower bias and variance. The minimum variance method has proven useful in some two dimensional scenarios as reported in [14], where we also find extensions of AR estimates and the Levinson algorithm to 2D. I expect no trouble finding or deriving 3D results, and expect that performance of these estimates will be as good in 3D as they have been in 1D and 2D.

Interesting 3D objects typically lie in large spaces, and estimating the correlation matrix, R_{ss}, of our data will be somewhat expensive in time. Using the simple estimate $R_{ss}(\boldsymbol{r}) = \sum_{\boldsymbol{r}'} s(\boldsymbol{r}')s(\boldsymbol{r}' - \boldsymbol{r})$ it is clear that the computation takes time on the order of the number of resolution cells in the autocorrelation space squared. Objects of the very modest size $16 \times 16 \times 16$ will have an autocorrelation space 8 times this size and so require around one billion multiplications, while each additional

[4]Systematic sources of noise may persist despite averaging. For example, stray light scattered off of unwanted background points will create virtual scatterers and alias them into our object space. Periodogram averaging will not remove this source of error.

doubling in all dimensions represents a 64-fold increase in computation. I have found this to be prohibitive for the quick reconstructions which I did, although performing billions of operations in a reasonable amount of time is well within the capabilities of today's workstations. I feel correlation-based methods may have a place in the future development of 3D Phase-Retrieval problems.

3.3.2 Thresholding

Ideally, in choosing a threshold, we would have some knowledge of the brightnesses of supported points, the brightnesses of unsupported points, the *a priori* probabilities of each outcome, and the consequences for making each of the two incorrect decisions. In practice, accurate knowledge of these quantities is difficult to obtain. Unsupported points will have brightnesses corresponding to any noise present. Noise due to the finite precision of the detector can be thought of as uncorrelated from pixel to pixel, and so will lead to white noise in the autocorrelation domain. On the other hand, noise due to the discrete grid and bias of the spectral estimates, such as sidelobes of a brightly lit point, will have systematic structure linked to the true signal. Another source of error is the failure of the *sampled* object to meet the opacity constraint. That is, even a perfectly opaque continuous surface will pass through multiple voxels in the same column often, except in very special cases.[5]

Similarly, depending on the shape of the object, there may be more or fewer terms in the sum (3.20) at a typical point in the autocorrelation support.[6] This in turn effects the range of brightnesses seen among supported points. My approach for real data has been similar to Shirley's in not trying to choose a thresholding scheme which corresponds to any particular model of the noise sources or of the object shape. Instead, I chose thresholds which visually seemed to leave the 2D support neither full of holes nor flared well beyond the dimensions of the object.

[5]Arguably, multiple voxels in a column should not even be considered an error at the thresholding stage, since each voxel represents true scattering pairs. A flexible algorithm for processing the autocorrelation support with some tolerance for non-opaqueness is called for.

[6]Think, for example of a planar object, where a typical supported autocorrelation point $\boldsymbol{r}_i - \boldsymbol{r}_j$ actually corresponds to *many* different (i, j) pairs.

Intuitively, points falsely placed in the support should be less harmful than missed points in the support: Any point missing from the copy which we are trying to reconstruct will not be recovered, and missing points are a cause of incorrect shifts when searching within singly populated columns. (Spurious points only cause false shifts in combination with an error in the 2D support estimate or with a missing correct point). False points will tend to slow convergence of the algorithm, which is generally less damaging to the overall result than incorrect shifts. Of course, too many false points will almost surely to cripple the algorithm. I feel these reasons confirm Shirley's intuition that overly high thresholds should be avoided.

One theoretical result that does merit attention here is the scaling of the brightness distributions for supported points with the number of scatterers, or resolved scattering cells, in the object. As we move from an object with a single scatterer to one with some large number, N, of scatterers, the expected "histogram" of the brightnesses of the supported points becomes more and more spread. Some points will represent a single pair of weak scatterers and some will represent many pairs summing through Eq. (3.20). Intuitively, any broadening of the range of brightnesses means that it is harder to choose a threshold which enough of the points are above.

The following simulations support this concern. In two simulations, $|a|$ is derived from the simulated observable of $|\tilde{g}|^2$ which suffers from quantization error. I converted the $|\tilde{g}|^2$ samples (generated by performing the FFT on randomly generated opaque objects) into integers with peak value normalized to 50, so that each measurement represents about 6 bits of data. In one simulation, the discrete space was kept a fixed size while N increased, so that scatterers filled up the space. In the other, the number of scatterers and total number of x, y pairs in the space were scaled up together, so that scatterers occupied one point in every column. Histograms for the first and second simulation are given in Figures 3-15 and 3-16, respectively, each for four different values of N.

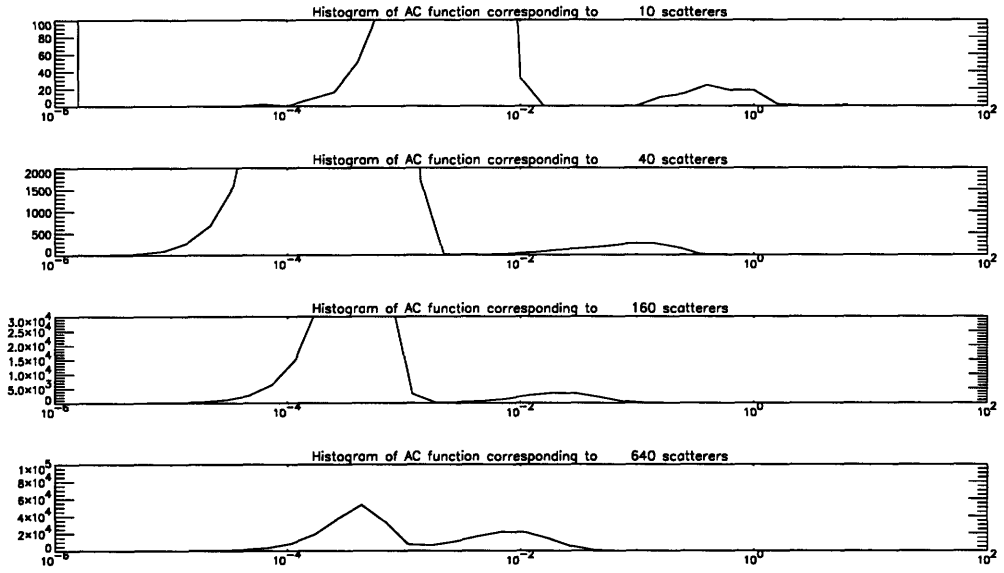

Figure 3-15: Histograms of the brightnesses of points in the autocorrelation function for four values of N, with space of fixed size.

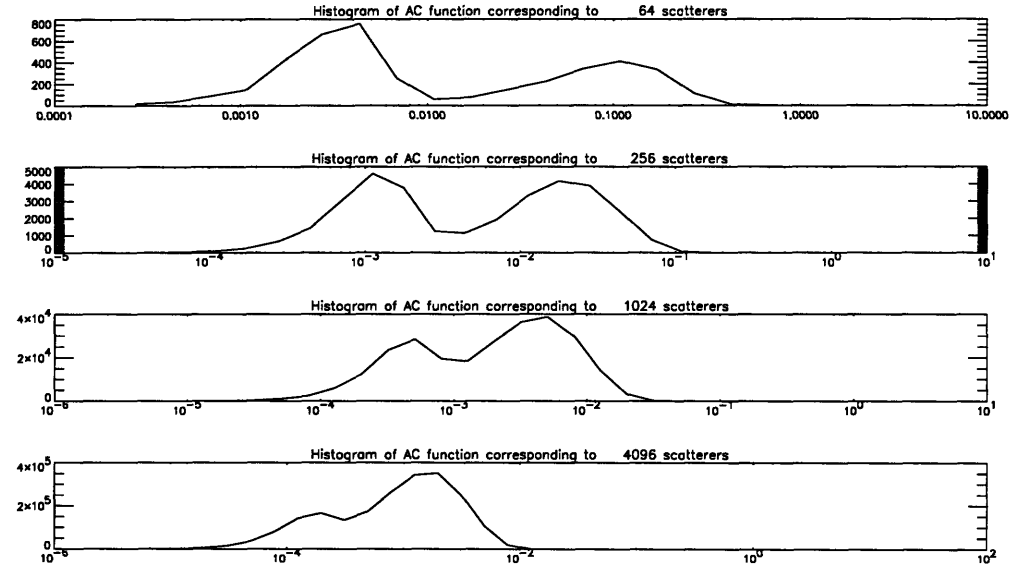

Figure 3-16: Histograms of the brightnesses of points in the autocorrelation function for four values of N, with space scaled in size along with N.

The points which are supported only because of quantization error are clearly discernible for small N, but as N becomes moderately large, it becomes impossible to cleanly separate supported points from errors on the basis of $|a|$ alone. For many applications, I feel the desirable resolutions will cause us to push for larger N, making thresholding an important issue.[7] Measurements using other techniques often involve tens or hundreds of thousands of x, y pixels. This is one reason to think that thresholding a may involve too great a loss of information. Approaches to phase retrieval which are not support based may be appropriate. Some have been mentioned already and others will be discussed in the next chapter.

3.4 Scaled Approach

When imaging a continuous object using some arbitrary sampling pitch in the autocorrelation domain, it is natural to consider how the solutions at different resolutions relate to one another. That is, if the shift-and-intersect method gives a reconstruction at one resolution, how does this predict the results which would be obtained at a lower resolution, starting with a more coarsely sampled A? A more interesting question is, can we gain anything by attempting to reconstruct a surface at some resolution while already having estimated the surface at some coarser resolution? This is the type of thinking which has been so successful in the FFT and many other algorithms. They generally improve speed, which is not an issue right now for the support methods (shifts can be computed essentially instantaneously within a program that involves printing and interacting with the user). In our case, a more important advantage would be that the lower resolution image's autocorrelation estimates may provide further noise immunity. There will be a trade-off of speed for noise suppression, since even performing autocorrelation estimates at one resolution can be time consuming.

This type of staged algorithm represents an important direction for future research. Looking strictly within a low-resolution solution after rescaling seems likely

[7] Although, I have not proven here that increasing the number of resolution cells in a continuous object will give behavior following the simulation.

to run into problems for objects that are not sufficiently smooth. In general, images taken at a resolution coarser than the features of the imaged object will tend to violate opacity, and I would expect the complexity of the program will increase substantially. Some of the reasons for trying this out are shared by the varying-threshold staged shift method already described, and comparisons of the two should prove interesting.

3.5 Simulation Results

Simulations using the object in Figure 3-17 were used to test the methods put forward in this chapter. The object consists of 256 points in a $32 \times 32 \times 128$-point autocorrelation space, with points lying in two continuous sheets.[8] The "observable" $|\tilde{g}|^2$ were obtained by taking the magnitude squared of the result of the n-dimensional FFT function found in [15]. The autocorrelation magnitude squared was obtained using the inverse transform, and the resulting data was processed exactly as estimates of $|a|^2$ derived from real measurements would be. Several threshold and shift-strategy combinations failed to produce results. Notably, the random choice of singly populated columns failed for some thresholds because no singly populated columns were found after the first shift. The variable-threshold approach was tried, finding fully consistent sets before moving on to a lower threshold. These preliminary attempts were not successful, although with no noise in the $|\tilde{g}|^2$ estimate, some of advantage of setting the threshold high is lost; the strategy should not be discarded on the basis of these simulations.

The results of a particular strategy which could very easily be automated is in Figure 3-18.[9] The strategy was basically the same at each iteration: form a consistency matrix of all points in columns with at most three points (up to a maximum of 200 points, in my implementation), perform CMR, and then perform a shift-and-intersect using that point of those returned by the CMR which has the highest $|a(\boldsymbol{r})|$. The final reconstruction had 203 of 256 points, and involved at least one bad shift. Of

[8]The autocorrelation space is double the extent of the object in each dimension. It determines the size of the FFTs which we will perform.

[9]Some points plotted with height 0 actually represent empty columns, and thus lost points of S.

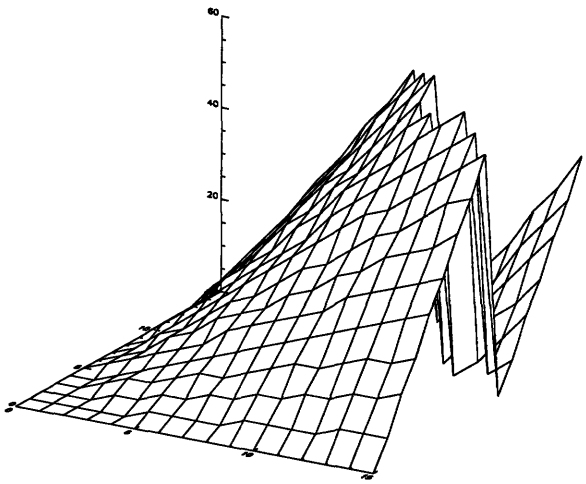

Figure 3-17: True simulated object to be reconstructed, and also the reconstruction of Section 4.5.

the remaining points, only 185 are in a single copy of S—the others could perhaps be eliminated with further shifts. After starting over only a handful of times, I was able to converge to a solution with 232 of 256 points, all in the same copy, with no bad shifts performed (missing points resulted from thresholding). No consistent strategy was used in this reconstruction, so the method may not be easily and effectively automated.

Some quick results on performance metrics may give an idea of their utility. In all cases, P_{FA} relative to the ideal-A shifts was zero; this would not be the case if noise were added to our simulated speckle measurements.

We can get a rough idea of our P_D, and of the difficulty in thresholding an object of this size by looking at Figure 3-19. The fraction of points detected in A before shifts are performed is plotted as a function of threshold, t, defined by the comparison $|a|^2 > t$ with the peak value of $|a|^2$ normalized to 1. For the above successful reconstructions, $t = 10^{-6}$ was used. This corresponds to the same peak-to-threshold ratio used in the simulations of section 3.1.2. Of several attempts made with $t = 10^{-5}$, all failed. Future work should relate the threshold needed for reconstruction to the required noise limit on speckle measurements. Curves such as in Figure 3-20 can tell us the

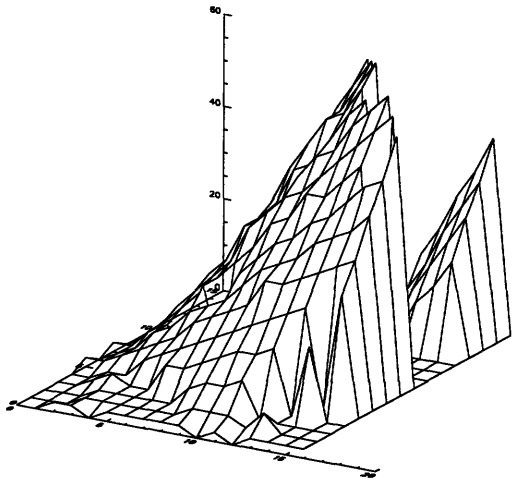

Figure 3-18: Reconstructed object resulting from shift-and-intersect operations.

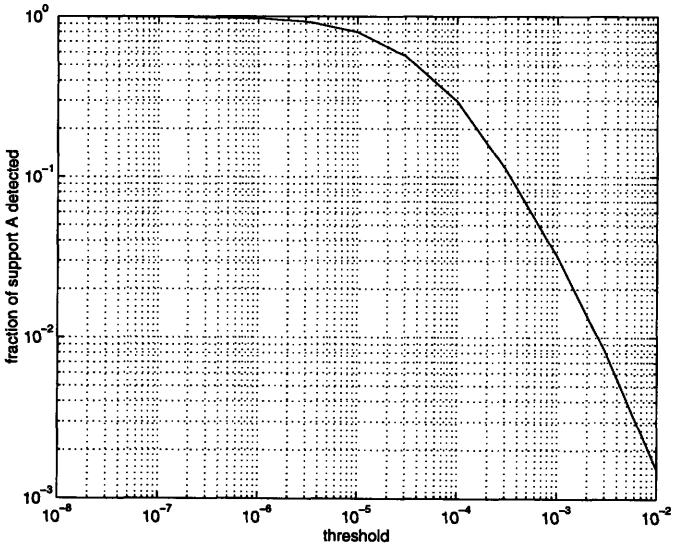

Figure 3-19: Rate of detection of supported points as a function of threshold.

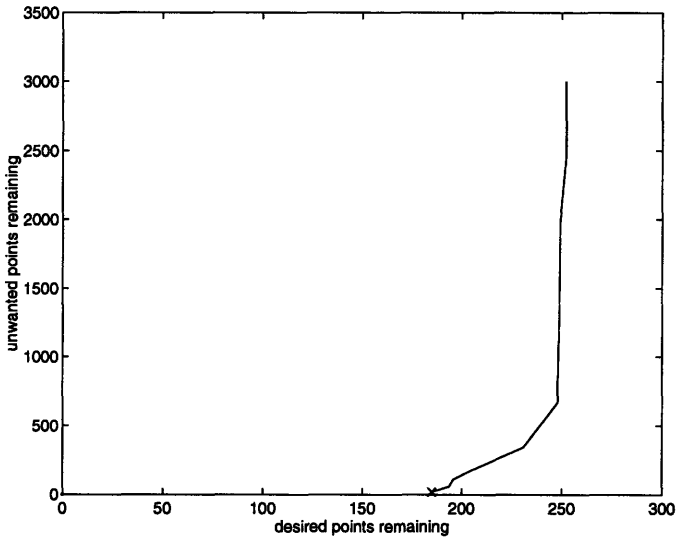

Figure 3-20: Progress of reconstruction for an automated shift strategy.

overall performance of our shifts and thresholding as the algorithm progresses. It is simply the number of desired points plotted against the number of unwanted points left in our estimate of S. Clearly, we would like to move towards the bottom without slipping too far to the left.

While more simulations can be done, the general result is that objects with only 256 scatterers are already non-trivial to reconstruct. Objects several times larger than this cause all of my first-cut extensions to fail. Although there are many avenues for improvement within support methods, the difficulties motivate looking into other alternatives as well.

Chapter 4

Transform Methods

So far, we have approached the extraction of height information by discarding most of the measured Fourier intensity data and keeping only the autocorrelation support. An iterative extraction scheme then depends on the ability to choose good shifts, and may involve processing and sorting lists of next likely candidates for a shift. If enough good shifts cannot be found to reconstruct a surface without losing many of the desired points, or if strategies for choosing shifts become sufficiently time consuming, we would like to explore other avenues of processing. In addition, if the values of g, and not only its support, are desired, we will need non-support based methods—hopefully ones which utilize results of the (partial) support reconstruction.

Since the support approach discards a good deal of the data—and perhaps the information—present in the autocorrelation function, we can be hopeful that non-support methods can show improvement over or compliment these methods. One alternative is to take small steps away from this drastic data reduction—through more elaborate weighting and thresholding schemes, for example. But to utilize the data fully, we would at least like to consider directly estimating the complex scattering function which gives rise to the autocorrelation through Eq. (3.2). There may also be advantages in having the flexibility to use both kinds of methods, for example to impose approximate or imperfect constraints. Further, the methods I discuss here involve typical DSP operations like convolution and FFT, thereby utilizing the efficiency of available hardware for performing these tasks.

It turns out that a framework exists for approaching this problem numerically. The problem of 2D image reconstruction with fixed support has been successfully posed and well-studied in the context of successive approximations and global minimization. There are several approaches that can be generalized to suit our problem. Further, there is one example of a 3D complex object reconstruction using the same opacity constraint that concerns us here [6]. This paper used a conjugate-gradients minimization of the deviation in frequency magnitude to solve the problem; it encourages speculation that many of the methods can be successfully extended, but does not use real data or objects of sufficiently complex shape to be of use in our applications. We would like, then, to demonstrate extensions of their results to realistically complex and corrupt data. There is, of course, a huge amount of literature on successive approximations, minimization, and operator equations in general. It may be possible to draw ideas from this literature which can be applied to our particular problem, although the time-frame of this thesis did not permit any attempt to generalize existing methods. We will see that although extensions of the iterative-transform methods are successful for simple objects, achieving rapid convergence becomes very difficult when the number of scatterers becomes large.

4.1 Building on Two-Dimensional Results

The existing literature on the phase retrieval problem for two dimensions is fairly extensive. It addresses the question of obtaining g from $|\tilde{g}|$ using *a priori* knowledge of the object-domain structure, as with our problem. The previous knowledge is also, as in our case, knowledge of the support of the finite-extent function, g. Unlike our case, the supports most commonly assumed have fixed shape, not our "opaque" support, which means any support function with at most one cell per vertical column. The basic literature, for example [9], is very useful in developing intuition for how our specialized case will work. Some results which I have already observed are analogous to results reported for 2D reconstruction problems. Experimenting with any of the methods of speeding up convergence is potentially worthwhile.

On the other hand, much of the literature consists of detailed studies of special cases and of methods which speed convergence sometimes and slow it down others. We must then be cautious not to expect clear and rapid understanding of the many variations of the algorithm, or of the dependence upon parameters. Instead, I have tried to develop some understanding of the most common variations of the iterative-transform method, and make some reasonable guesses at how to write simple but effective versions which apply to our problem.

4.2 Fixed Point and Optimization Forms

It is helpful to restate the problem: estimate a scattering function $g(\boldsymbol{r})$ which agrees with the observed speckle pattern samples, $|\tilde{g}_{obs}(\boldsymbol{f}_i)|^2$ through,

$$|\tilde{g}(\boldsymbol{f}_i)|^2 = |\tilde{g}_{obs}(\boldsymbol{f}_i)|^2, \tag{4.1}$$

and is consistent with the opacity constraint,

$$g(\boldsymbol{r}) = \alpha(x,y)\delta(z - h(x,y)), \tag{4.2}$$

and any other *a priori* support assumptions. Here the $\boldsymbol{f}_i$ represent the discrete sampled frequencies corresponding to measured intensities. In some cases, we may reasonably impose additional object-domain constraints. In particular, I will use a three-part constraint:

1. The scattering object is at least approximately opaque; Eq. (4.2) holds at least approximately.

2. $\alpha(x,y)$ has support known within a shift and an inversion. We can imagine that this information can be approximately obtained with a conventional 2D image, but at the least we have an upper bound on the total horizontal extent of the object.

3. The support of the autocorrelation function measured will be regarded as approximately the autocorrelation of the support of the scattering function to be reconstructed. This has been discussed in the previous chapter. If we arrive at a locator set using methods of the previous chapter, we can require that all nonzero values of g occur within this locator set.

We can trivially satisfy either the object or frequency-domain constraint; then a sensible approach is to regard one equation as a hard constraint and minimize some measure of the the error in the other equation. This approach is taken in [6], where the hard constraint is (4.2) and the error measure is the sum of squared errors at each frequency sample. We could also minimize any measure of the combined errors of the object and frequency-domain constraints. A third approach is to combine the two equations into a fixed-point condition, and then apply successive approximation methods. The error-reduction algorithm is the simplest example of this approach. To obtain the fixed point form, $x = Tx$, with operator T and unknown x, we combine (4.1) and (4.2) to give,

$$g(\boldsymbol{r}) = Tg(\boldsymbol{r}) = R_o \mathcal{F}^{-1} R_F \mathcal{F}\{g(\boldsymbol{r})\}, \tag{4.3}$$

where the R_o and R_F are operators which in some sense "impose" the object-domain and frequency-domain restrictions respectively. There is some freedom in choosing these operators, and the choice has an impact on the success of the method. The choice corresponding to the error-reduction algorithm is $R_o\{g\} = \hat{g}$ minimizing $|g - \hat{g}|$ among those $\hat{g}$ which obey the support (and any other) constraint(s), and similarly $R_F\tilde{g} = \check{g}$ minimizing $|\tilde{g} - \check{g}|$ among all $\check{g}$ obeying (4.1). It is easily shown that for this choice of T, iterating according to

$$g_{k+1}(\boldsymbol{r}) = Tg_k(\boldsymbol{r}), \tag{4.4}$$

guarantees that neither error will grow from one iteration to the next [9]. While this does not quite prove that the transformation is a contraction (the error could

simply stay the same), it gives us some assurance that the method is sound. In a preliminary look at the literature, I found no similar proofs for other T, but did find that particular T empirically show faster convergence. The input-output family of algorithms result from choosing R_o differently, but in all cases R_o and R_F leave unchanged any solution to both constraints, thus maintaining the fixed-point form.

The optimization and fixed-point approaches are closely related, and lend themselves to almost identical implementations (see [9]). I have coded the error-reduction algorithm as a C program, and built output-output, hybrid-input-output, and the conjugate-gradients method on top of the original structure. Using randomly generated heights for points on a regular grid, I simulated Fourier samples in the absence of noise and observed the performance of the algorithms. For "small" objects, that is sixteen points within a $4\times4\times4$ unit cube, the programs could often compute solutions in seconds, although times varied greatly according to (randomly-generated) initial guesses, the particular method used, *a priori* support assumptions, and parameters of the methods. All methods were prone to either stagnation or instability (conjugate-gradients); a truly robust implementation would have a large degree of adaptation or perhaps a more creative choice of operator T.

4.3 Customizing the Algorithms

The 3D support, or locator set, is crucial to rapid reconstruction, and care should be taken in finding and using a locator set. I described a three-part object-domain constraint earlier. The second condition is very familiar from other phase retrieval problems, but the first and third have not been fully explored. Opacity does not present us with a rigid support to impose (within a shift) like typical phase-retrieval problems; one cannot ask on a point-by-point basis whether or not the condition is met. Instead one point is chosen in a vertical column in favor of the others. Making this choice is the first major modification of the algorithm which I made. Choosing the brightest point seems like a very natural choice—more importantly, it corresponds to the error-reduction choice. If this is not obvious, recall the error-reduction condition:

search among all "opaque" $\hat{g}$ for that with smallest least-square distance from g. If we choose $\hat{g}$ column-by-column, we are restricted to placing any nonzero value in at most one point in each column. Whichever point we place a value at, it is best to make our $\hat{g}$ agree with g exactly there. The total error of that column is then the sum of squares of all values of g except our chosen point, and so we pick the point with largest magnitude. An important observation is that, if any other assumption further restricts the support of g within this column, the above reasoning still holds almost unchanged. Given that we can choose only among certain points, we still choose the allowed point with largest magnitude. This will allow support methods of the previous chapter to plug in to this transform method very easily.

Incorporation of support methods is well motivated since even an incomplete support-based solution, for example performing only a few shift and intersect operations, should improve convergence of the iterative transforms. It is well known that tighter supports, or locator sets, lead to rapid reduction of errors. The points eliminated in even the first few shifts may make a difference between rapid convergence and stagnation. Further, a constraint that is too tight in some places and too loose in others can still steer the iterations near a solution. Both the literature and my own results show that once near a solution, much looser constraints will bring us quickly to convergence. A thorough study of the interactions between the algorithms is not intended here, but I have obtained sufficiently good results in simulation to demonstrate that a combination the two methods can perform better than either method alone. This leaves much to be done, but is encouraging in itself.

4.3.1 Parametric and Sampled Objects

The approach of [6], of optimizing only within the space of opaque objects is an interesting alternative to the direction I have taken here. In some senses, a framework of samples may be more general or flexible, allowing different constraints to be used without any structural changes of the data or the program, although this is not yet clear. Another difference is the possibility of super-resolution achieved in the parametric case, although again, one can devise schemes for using high-resolution transforms in

the later stages of the program I am presenting as well. Further, talking about high resolution before robustness issues have been resolved may be premature. A clearer difference which I see is that the use of locator-sets in a parametric optimization will be much less natural. Whether it will be less effective is not clear.

4.3.2 Other Possibilities—Magnitude Constraint, etc.

The modifications which I have made were those which seemed most appropriate to the specifics of our problem: implementing an opacity constraint for example. Naturally, once you look at the iterative-transform method in the context of fixed-point problems, there are many modifications which one can dream up. For example, I have seen no mention at all of using an alternative operator R_F, of Eq. (4.3). It is easy to show that the error-reduction choice previously defined amounts to $R_F\tilde{g} \equiv |\tilde{g}_{obs}|e^{i\angle(\tilde{g})}$, where $\angle$ simple returns the phase angle of its complex argument. One can think up any number of generalizations, perhaps

$$T_F\tilde{g} \equiv \tilde{g} + \gamma(|\tilde{g}_{obs}|e^{i\angle(\tilde{g})} - \tilde{g}),$$

or

$$T_F\tilde{g} \equiv \tilde{g}\left(\frac{|\tilde{g}_{obs}|}{|\tilde{g}|}\right)^{\gamma}$$

While these possibilities are somewhat interesting, it is difficult to justify spending time to experiment with them. If future work is to be done here, it may help to back up and develop a broader theory to direct our numerical experiments.

4.4 Multi-Staged Algorithm

While building a truly robust algorithm is not within the scope of this thesis, I did experiment with several types of flexibility or adaptation so that convergence could reasonably be expected for simple problems. A major motivation for adaptation was the unavoidable result that any tight support was bound to be missing some of the desired points. If this support is rigidly enforced, the iterations will never

be able to converge completely. I have already mentioned that a tight support is more important at earlier iterations. The best situation then might be to loosen the support as the iterations progress. This is both straightforward to implement and effective in simulations. I have done this both by lowering the threshold for $|a|$ and by undoing some of the shifts in the shift and intersect method.

Further motivation to adapt comes from various results in the literature[1], and also from the simple fact that failure of the algorithm can result simply from starting with a bad initial guess for g. The latter result suggests that we may want to give up on helplessly stagnated guesses and restart with a new random g. This too was easily and effectively implemented once an arbitrary criterion for stagnation was decided upon.

Proceeding in the direction of adaptation clearly can complicate a program and leave us with many parameters to set arbitrarily or explore empirically. I leave the question of which adaptations are most desirable open-ended, but also comment that the need for many variations and parameters is a serious drawback of the method in itself. For many industrial applications, keeping tight control of the measuring environment, even if costly, may seem much more attractive than a method tied together by elaborate software patches.

In the short time I have developed my programs, I have built in a fair amount of flexibility without allowing the number of parameters to explode. Both loosening the support and restarting have made the difference between stagnation and convergence for many cases. While I could generate statistics on the increased probability of convergence given these parameters, it is more interesting to know that there are cases in which adapting is necessary.

4.4.1 Rescaling Between Iterations

A particularly interesting adaptation in a 2D iterative-transform problem is described in [16]. The sampling pitch in the object domain is initially very coarse and is made

[1]For example, mixing cycles of hybrid-input-output with error-reduction improves performance.

finer at later iterations, so that (we hope) high frequency information is only utilized once coarser structure has already been resolved. While I have not implemented anything similar, I feel this method is worth looking into, and in fact that the reported speedup of up to three times is overly conservative. Though I have not studied the method careful, I see no evidence that the algorithm could not dramatically improve computation time for large problems; 3D problems do tend to be large.

4.5 Simulation Results

Early simulations were done using height functions of independent, pseudorandom heights at every point—that is, at the limit of jagged, discontinuous objects. Several variants of the iterative-transform method were used; after some *ad hoc* controls were put in to prevent instability, the conjugate-gradients method as described in [9] would converge completely and in a matter of minutes for objects of interesting detail. This was the most successful variant. The problem became difficult for autocorrelation space sizes approaching 64 samples in each dimension. Since these jagged objects may not be representative of targets one would actually image, I moved to smoother simulated objects, which actually hurt performance considerably. Even with the few variations I implemented, it was impossible to test all versions over an interesting range of object surface types, space sizes, threshold values, etc., and I will present only the most relevant results here.

The motivation presented at the beginning of this chapter was that iterative-transform methods may recover information lost in support-based reconstructions. Recall the reconstruction of 203 points in Figure 3-18 reconstructed from the true object of Figure 3-17. An error-reduction algorithm was done using the result of the first 5 shifts only and recovered all 256 of the 256 points correctly, with no spurious points. The algorithm was implemented in two stages. Only the first stage, with 45 iterations, used the shift and intersect's output support. The second stage automatically began once the frequency-domain "error," that is the deviation from the frequency domain constraint (4.1), drops to some predefined level. Convergence,

similarly defined in terms of object-domain error levels, is achieved after twelve additional iterations. The algorithm is thus very crudely adaptive, although none of the parameters were optimized for this object. Adjusting the adaption parameters and switching over to conjugate gradients would likely improve speed considerably. Since even this version took only several minutes on a multitasking workstation with the gcc compiler, there may be real-time potential for this method, although getting the algorithm to work for less perfect data is a higher priority.

We have seen that the iterative-transform algorithm has given us something that the support methods alone did not. The natural question is what the support methods add which iterative transforms alone do not have. The result which we might expect is that, without a tight enough support constraint, the error-reduction algorithm will stagnate. Once stagnated, iterations of 2D transform algorithms have been observed to barely progress for tens or hundreds of thousands of iterations [9]. This is sufficiently slow to be considered a failure of the method (remember that each iteration includes two 3D-FFTs). To compare with the previous simulation, I started the iterative-transform method using a conventional support constraint. Specifically I did only a single shift and intersect operation using the first of the five shifts used in the previous simulation.[2] The program is designed to restart with a new initial guess whenever stagnation is detected, which has been helpful in previous simulations. In this case, after restarting four times and a total of 400 iterations, the algorithm fails. To the extent that frequency-domain errors reflect how well the current guess g_k fits, the iterations hardly show any progress at all.

[2]The locator set resulting from a single shift and intersect operation has been used in the past, and is one of the favored supports, for example, in [12]. It makes a fair "control"—it tells us how well transform methods do without the specific locator sets obtained by assuming opacity.

Chapter 5

Measurements

Previous chapters have helped bridge the gap between prior work on this problem and an effective measurement system. Before making any conclusions about the effectiveness of the techniques developed or the challenges to overcome in future work, we should look at how the techniques perform on real data. I describe a system used for acquiring raw data and present the first reconstructions which have been obtained using general-purpose imaging software.

5.1 Basic Measurement Setup

A schematic for the optical path in a setup used for measurements prior to this thesis is shown in Figure 5-1. If this looks much more complicated than Figure 2-1, that is because it folds a long path into a space which fits on an optical table, and allows the flexibility to magnify the speckle patterns. Magnifying the speckle allows us to achieve spacings Δx_{d} which do not correspond to the spacing of pixels on our CCD array. Simulating a long path is appropriate since long-distance imaging, for example of a satellite, is one of our motivating applications. The setup I used was essentially the same, and so minimal design was necessary. I selected a new lens to accomplish the desired magnification, brought in a beam from a NuFocus external cavity tunable diode laser at wavelength around 780 nm, and set up a Burleigh Wavemeter and UDT power meter. The laser was tuned using a Newport linear actuator and PM500

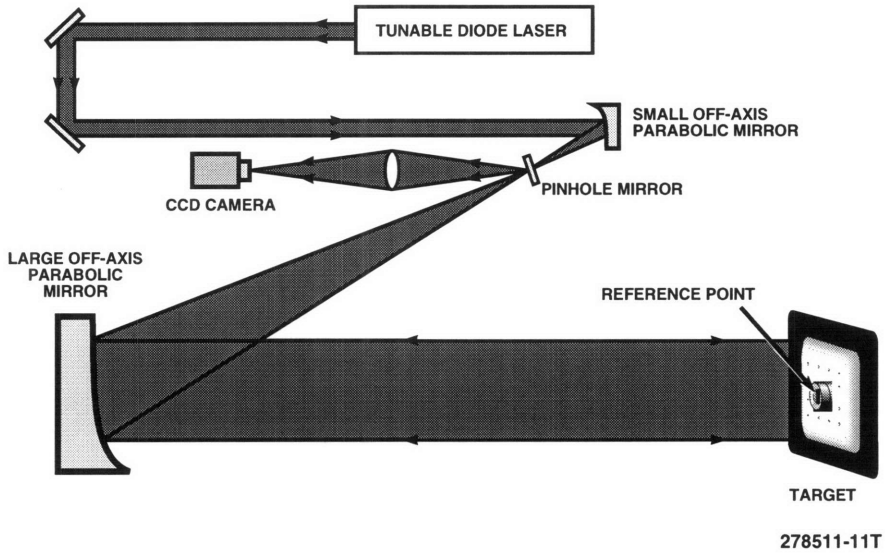

Figure 5-1: Schematic of optical system

controller with a LabView interface. The LabView software was previously developed by Greg Hallerman for similar measurements, and provided semi-automated stepping of the frequency, with manual fine adjustments made at each step. Intensity data was recorded by a Photometrics camera in 512×512 frames (the full field of the camera), which, along with power meter readings, were saved for processing. For all objects, 256 frames of data were taken and used to estimate the autocorrelations. In accordance with the method used in the past, intensity frames were scaled down by the power, which does vary appreciably as the laser is tuned. I find this a very reasonable way to compensate for these variations, since intensity should vary directly with input laser power.

There was some distortion in the optics, which I believe had only a minor effect on final results. Taking the 2D transform of a single frame revealed a pincushion effect on the expected 2D autocorrelation function. Compare Figure 5-2 with an analogous, but lower resolution, measurement taken in a previous optical setup, Figure 5-3, to see this effect. I used data sets previously acquired for the first target as well as new measurements in testing my algorithm.

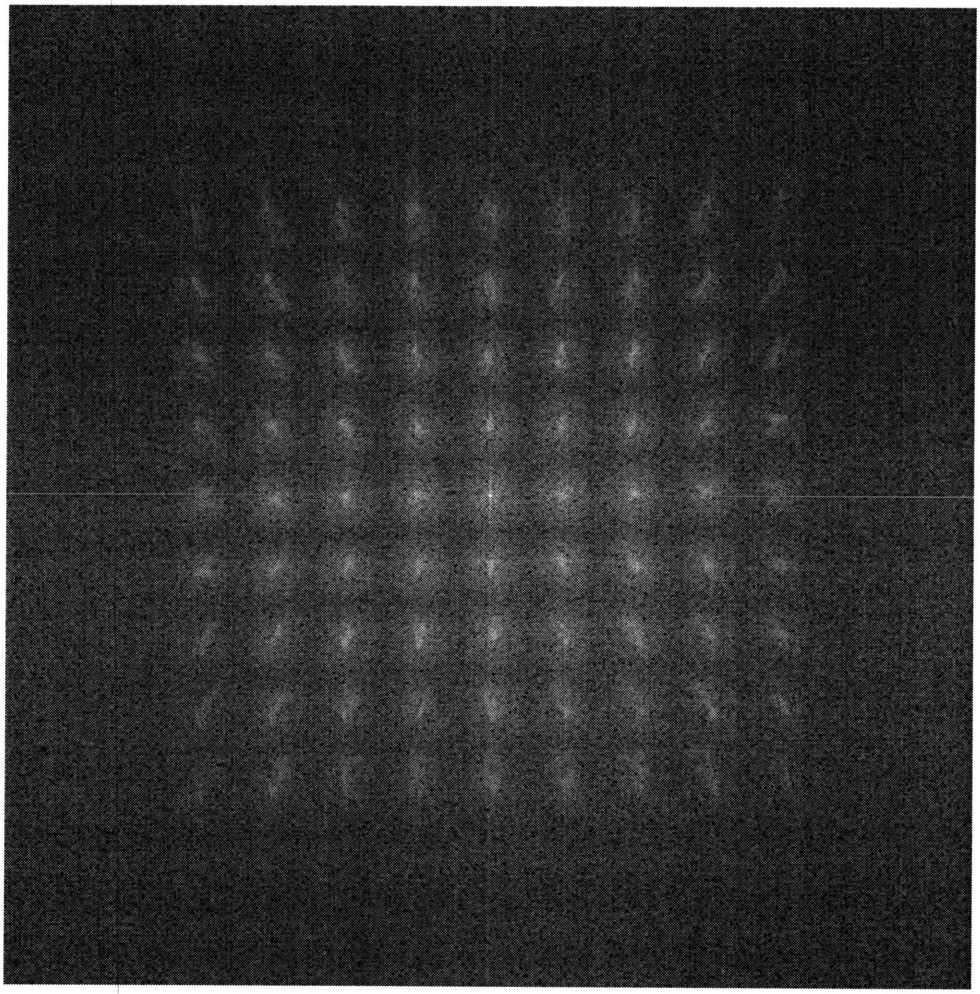

Figure 5-2: 2D autocorrelation with visible distortion

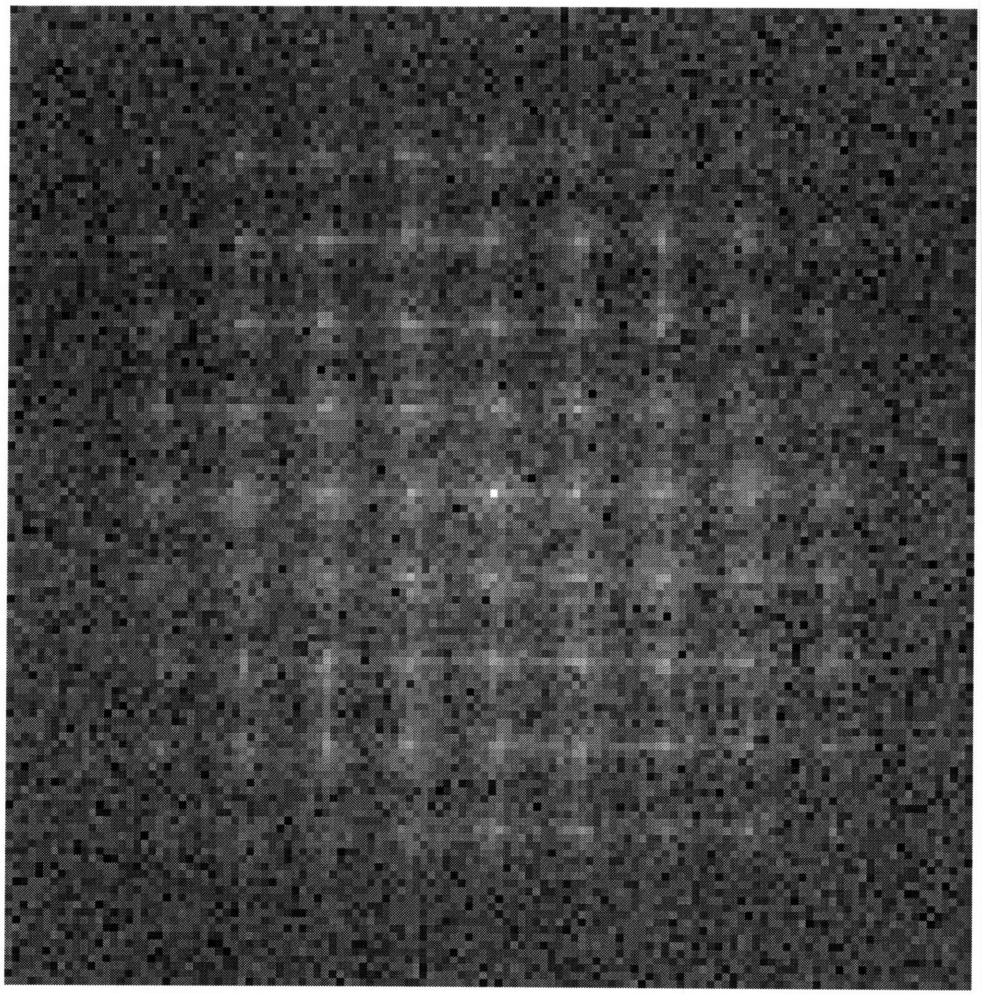

Figure 5-3: 2D autocorrelation with no distortion

Targets were chosen to test the system with different levels of difficulty. The first target, referred to as the pin array, is a a 5×5 array of curved, polished tips which create virtual points at different heights. The space of the autocorrelation estimate I used was $32 \times 32 \times 128$ samples. The object was designed for testing the imaging system of Shirley, Hallerman, and Rahn [11]. A photograph is shown in Figure 5-4. I chose this as an easy target, since it only has twenty five virtual points scattering light. There was no attempt to align these points with the natural grid of the autocorrelation function (in fact, the points are not spaced by an integer number of samples), so that this easy target is not trivially easy. I also did not take advantage of known 2D structure. The discrete points are either blurred out into more than one voxel, in which case they may as well be multiple scatterers, or will be imaged as discrete points. In the latter case the algorithm still has the task of deciding that some columns are dark, or unsupported; it does not assume this. Naturally, the bottom line is that fewer points present fewer sidelobes and generally simpler thresholding problems, but the measurement is not fundamentally different from that of general object surfaces.

The second target chosen was a model satellite with very simple structure. A photograph is shown in Figure 5-5. This object shape has no special advantage for imaging except that its coarse shape is visually recognizable at low resolution. I used an autocorrelation space of $32 \times 32 \times 128$ samples. It is painted with a special reflective paint which boosts the signal returned with respect to stray light. This is a fair advantage which could be engineered some other way in real applications (by using a stronger laser, for example). Naturally, its opaqueness makes the object desirable, but many real objects also have this quality. The satellite target is then not a special case.

5.2 Averaging

Even a quick look at the autocorrelation magnitude estimated using the straight FFT showed an obvious need to do better. Plots of single z-slices of the log-magnitude

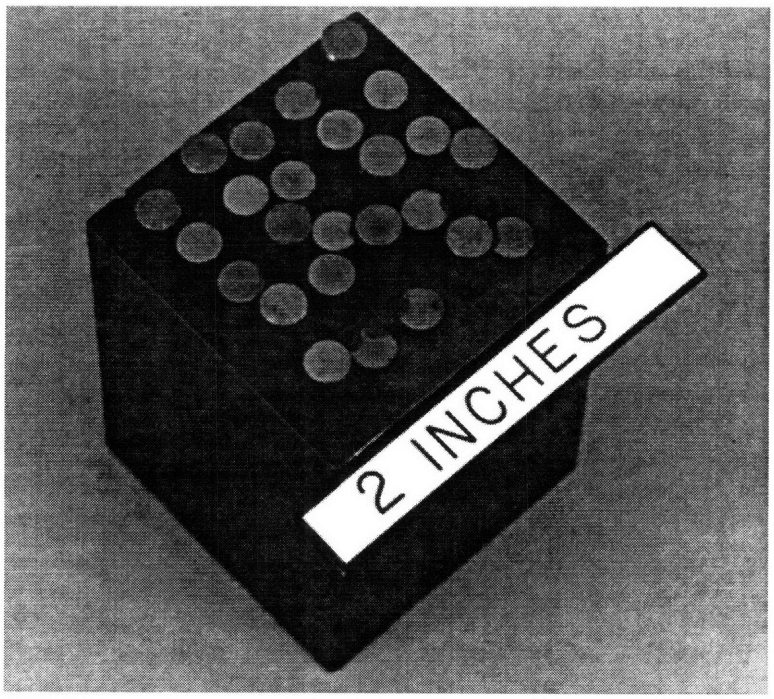

Figure 5-4: Photograph of pin array

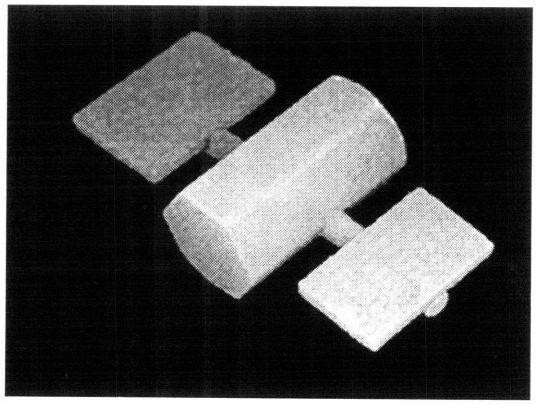

Figure 5-5: Photograph of model satellite

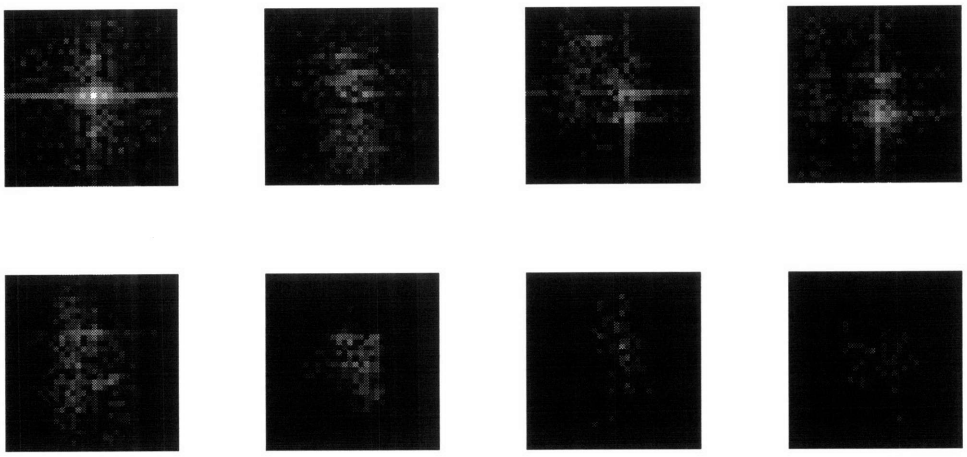

Figure 5-6: Several z-slices of the autocorrelation log-magnitude without averaging

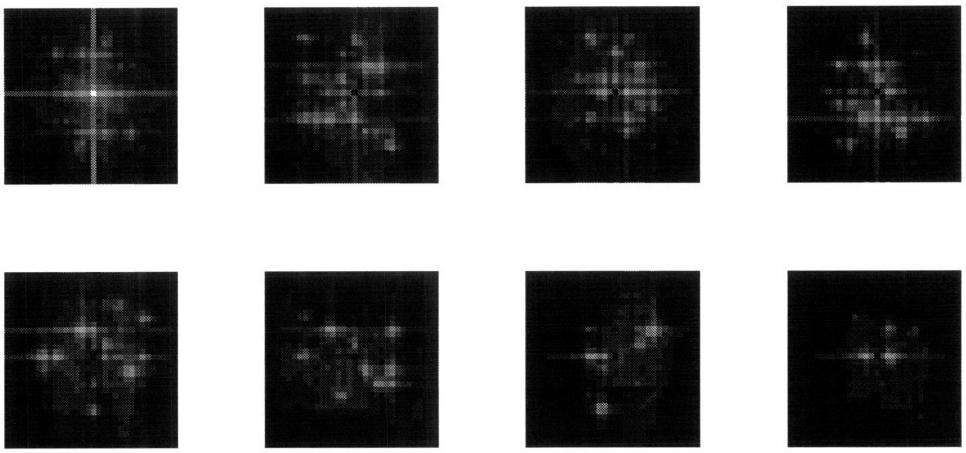

Figure 5-7: Several z-slices of the autocorrelation log-magnitude with averaging

of the autocorrelation function such as for the pin array shown in Figure 5-6 show that sidelobes and noise are a significant problem. We can look at these pictures and see that the thresholding problem—of deciding which points are true solely on the basis of brightness—is difficult. For this reason, better spectral estimates are key in making successful real measurements, though they were unimportant in simulations.

I have already mentioned that periodogram averaging was my preferred method, although slower correlation-based estimates may have more promise in the long-term. A series of frames analogous to Figure 5-6 is shown in Figure 5-7. To obtain the latter estimate, periodograms of 32 non-overlapping, windowed blocks of speckle data

have been averaged. Here it is much easier to discern the regular grid arising from the 2D structure of the target, and to discriminate bright true points from generally dim artifacts. Sidelobes are still an issue, and probably more of a problem than other noise sources. For this reason, later estimates used in the final reconstructions used larger block sizes with (spherical kaiser window) windowing. Since the windowing reduces the effective resolution, the larger blocks were then subsampled down to the desired size. The net result is to reduce the bias of the estimates due to sidelobes at the cost of having fewer blocks to average (and thus higher variance estimates). I coded the program into PVWave with enough generality that block sizes and kaiser window β-parameters can be changed at will to trade-off bias and variance. For the final measurements of the satellite, sixteen blocks were averaged, each were double the autocorrelation size in each dimension, and β-parameters were in the range of 5 to 7. For a discussion of spectral estimates and the relevant trade-offs, see [17, 18, 14].

The sidelobes of the large central peak in the autocorrelation function are particularly problematic. It is important while we are doing shifts that points in the same column not be considered consistent with each other. After all, for an opaque S, A will always have exactly one point in the central column, at the origin. For only approximately opaque objects, the center column of a might have a small cluster of supported points, all near the origin. If our estimate of A does not even reflect approximate opacity, multiple points within a column can be deemed consistent. We may end up performing shifts corresponding to multiple points in the same column. These are guaranteed to include bad shifts, and will damage performance. Bright sidelobes can cause this situation—they tend to make a large number of points in the center column appear in A. While there are more careful ways of dealing with this problem, I simply set all values in the center column of $|a|$ to 0 except the peak at the origin. This pre-processing does not represent any new assumption—we have already been using an opacity assumption throughout this thesis. Performance was immediately improved by this for the pin array. Moving to a more forgiving opacity constraint (with fewer points in the center column suppressed) may be desirable.

10.5	18.5	20.5	38.0	25.5
28.0	27.5	39.0	31.5	–
7.5	8.5	0.0	41.5	–
10.5	20.5	33.5	21.5	–
31.5	–	18.5	43.5	–

Table 5.1: Pin heights measured after two stages of shifts

5.3 Results

I have mentioned that looking at real data immediately necessitated improved spectral estimates. The situation was similar with shift strategies and other aspects of the reconstruction. The basic ideas presented in Section 3.5 were experimented with in a number of ways to try to overcome the many potential causes of failure associated with real data. Results focus on support-based methods; an assessment of the effectiveness of transform methods for the measured objects is left for future work. However, moving to real data did motivate some of the adaptation and flexibility which has been built into the transform part of the algorithm.

5.3.1 Quantitative Comparison of Pin Array

The discrete points in the pin array have been adjusted to known heights. This allows us to easily make quantitative comparisons. The reconstructions which I present have not squeezed every bit of information out of the data, but they do use a method of choosing shifts which would be easy to automate and is quite similar to that used to obtain the 203-point reconstruction of Figure 3-18. The heights in Table 5.1 were obtained in two stages of shifts, the stages having different thresholds. The horizontal spacing of pins was between 2 and 3 samples. A fractional height $h + \frac{1}{2}$ was assigned whenever a cluster of points with heights h and $h + 1$ appeared at the expected location of the pin. The 2D structure of the array could be easily discerned in the image. Since several points were still missing, I lowered the threshold and recovered a few more, which gives heights in Table 5.2. (Lowering the threshold causes additional

10.5	18.5	20.5	38.0	25.5
28.0	27.5	39.0	31.5	25.5
7.5	8.5	0.0	41.5	4.5
10.5	20.5	33.5	21.5	–
31.5	24.0	18.5	43.5	1.5

Table 5.2: Pin heights measured after a special stage

-1.5	-0.9	-0.4	-0.4	-0.4
-1.4	-0.9	-0.8	0.2	1.8
-1.0	-0.7	0.0	0.0	-36.
-1.3	-0.9	-0.4	0.5	–
-2.2	-0.8	-0.5	0.0	-5.1

Table 5.3: Errors in pin heights

points to arise in previously converged columns. These points were ignored.) This left only one point absent, and I proceeded to compare this measurement with the known height settings.

Rather than do additional measurements to calibrate the system, I took the heights relative to the center height, and scaled the maximum of the true heights to 43.5, so that the units are resolution cells. Table 5.3 shows the errors in the measured heights. Although the measurement includes bad points, the validity of the method has been clearly confirmed. Of the four points added between Tables 5.1 and 5.2, only one would appear to have been a correct point.

Similar measurements were done for different resolution sizes—if we accept poorer range resolution, we can get better convergence through windowing and averaging.

5.3.2 Qualitative Satellite Results

I will give only a very qualitative idea of the performance of the satellite reconstruction. Several shift strategies were tried. In all cases the image quality was very poor, but some did contain the coarsest structure of the satellite: a large body with two smaller, parallel panels. One such image is shown in Figure 5-8. Keeping in mind

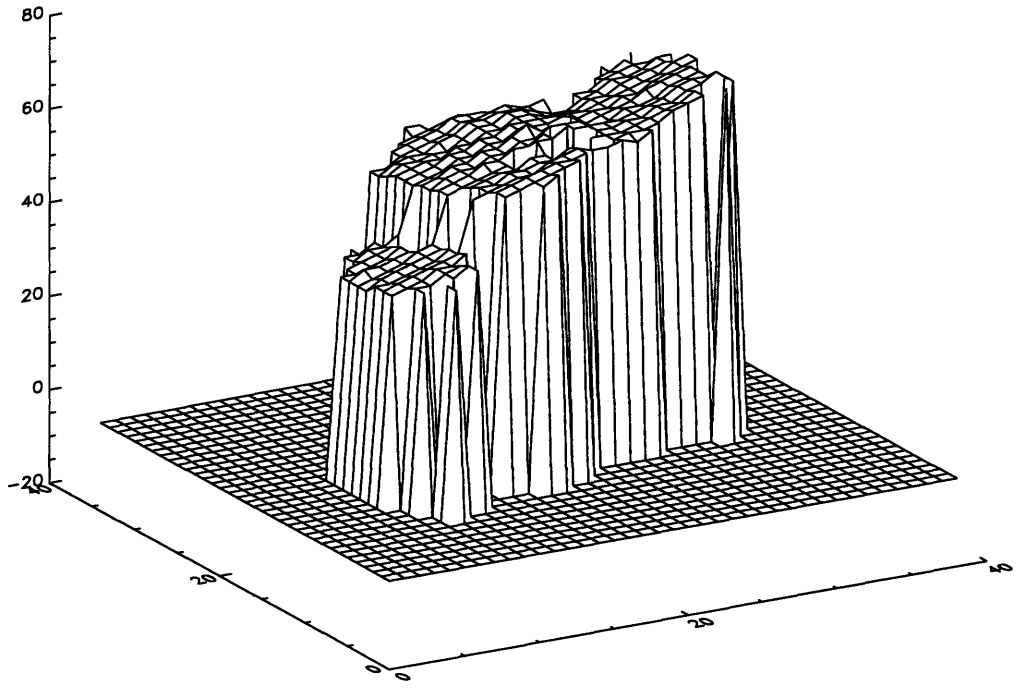

Figure 5-8: Reconstruction of the satellite model

that one of the simulations showed very persistent unwanted points, we can say that the satellite image is very likely to have spurious points as well as the obvious missing points.

Chapter 6

Conclusions

The motivation for exploring the phase retrieval problem in this thesis has been to enable an imaging technique, and help determine its practicality. The very positive results of simulations and general success of the pin-array measurement show that the method has potential. On the other hand, the very rough satellite image could be taken as either encouragement or discouragement. Clearly, some first attempts at improving and combining algorithms have not been sufficient to make an image which will resolve detail on continuous objects. This is not to say that there could not be an application for which the imaging system presented here presents useful information, but the results prented in the previous chapter do not yet compels us to find applications for this technology.

Both simulated and real results suggest that achieving the desired performance from phase retrieval methods will involve a good deal more algorithm development. While many directions have been proposed in previous chapters, it is difficult to say which will lead to significant improvements. Choosing shifts and converging through CMR-like algorithms are possible areas for improvement. Using information throughout the dynamic range of $|a|$ is difficult, and may involve major structural changes. Transform methods present many possible variations. In all cases, there is no clear way to proceed and no guarantee that improvements will be forthcoming.

One result not yet presented is that as modifications of the algorithm were made, noticeable increases in performance were observed. So while we cannot predict which

modifications will give the greatest improvements, we can have some confidence that we have not yet reached the limits of what can be done, and that some understanding of where the method fails is being developed. Using the satellite as a benchmark, I expect a clean image at the resolution shown in the previous chapter could be obtained within several weeks of experimenting with the various algorithmic modifications already mentioned. The fact remains that simple and elegant versions tried so far have not been sufficient. Moving towards a version which is empirically optimized or based on poorly justified models is something we do reluctantly, and only where simpler methods cannot be used. On the other hand, there are many examples in technology where elaborate modeling and programming are done to produce results which, whether or not we consider them "elegant", are very impressive.

Part of the answer to whether we have arrived at an acceptable solution to the proposed imaging problem is that it depends on the application. The methods presented, of course, will not constitute a cheap alternative to more traditional surface imaging techniques. If applications truly out of the regimes of other methods provide sufficient motivation to solve the problem, however, we have reason to think it can be solved. In addition to cumbersome algorithmic details, there are ways in which we can obtain assured improvements. For example, a data set of $512 \times 512 \times 256$ speckle pattern samples seemed quite large for the purpose of this thesis, but we can certainly imagine using thousands of frames with millions of pixels each. Dedicated hardware or carefully streamlined software for performing correlation and spectral estimates would yield very clean autocorrelation magnitude functions. As another example, proper development of a thresholding scheme which varies from one point to the next would not have been appropriate for this thesis, but we can easily imagine that special-purpose software could be developed to deal with specific artifacts in the spectral estimates while taking the support. Again, there is little question that this would improve performance, only of how much effort would be required. The algorithms have not at all been pushed to their limits.

Appendix A

Sets Operations

We assume that capital letters are used for sets and lowercase letters denote points. Generally, in this paper, point p_i in set notation corresponds directly to vector $\mathbf{p}_i$ in the space on which our scattering function and autocorrelation function are defined.

It is natural then to refer the point corresponding to $\mathbf{p}_i + \mathbf{p}_j$ as $p_i + p_j$, and refer to the point corresponding to $-\mathbf{p}_i$ as $-p_i$. Once addition has been defined on points, the definition which we have given for sets,

$$A + B \equiv \{c | c = a + b, a \in A, b \in B\}, \tag{A.1}$$

is very natural. For example, if $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2\}$, the set $A + B$ contains the six points:

$$\begin{aligned} &a_1 + b_1, \quad a_2 + b_1, \quad a_3 + b_1, \\ &a_1 + b_2, \quad a_2 + b_2, \quad a_3 + b_2 \ . \end{aligned}$$

Note that the definitions given for $A - b$ and $a - B$ are consistent with thinking of a as a set containing only one point.

The binary union and intersection operators are denoted by $\cup$ and $\cap$, respectively. That is, $A \cup B$ denotes the set of all elements contained in either A or B or both, while $A \cap B$ is the set of elements contained in both A and B. These symbols are

used also in a way analogous to Σ for addition or Π for multiplication, that is:

$$\bigcup_i A_i = A_1 \cup A_2 \cup ..., \tag{A.2}$$

and,

$$\bigcap_i A_i = A_1 \cap A_2 \cap ..., \tag{A.3}$$

The expression $A \subset B$ signifies that every element of A is also an element of B, and $B \supset A$ is defined as being equivalent to $A \subset B$.

Appendix B

Modified CMR Algorithm

```
Form_Consistency_New(int **points, int points_length, float *autocor){
  int this_point, other_point;

  cm_new_length = points_length;
  /* the structure cm_new_consistence is a list of consistency values
     for each of the points.  The ith entry contains an integer which
     is the number of points the ith point is consistent with.
     The trick is that as points are removed from the matrix, we can
     update this list without ever having formed the matrix itself as
     a data structure*/                                              10

  for(this_point=0;this_point<points_length;this_point++){
    cm_new_consistence[this_point] = 0;     /* zero the entry */
    cm_lookup_new[this_point] = this_point;
    for(other_point=0;other_point<points_length;other_point++){
      if(PairConsistence(points[this_point], points[other_point], autocor))
          cm_new_consistence[this_point]++;   /*count the consistent points*/
    }
  }
}                                                                   20

Reduce_Consistency_Matrix_New(int **points, float *autocor){
  int swap_slave;
```

```
  int worst_col=0, last_col, col;
  int fewest;

  /*here is the reduction algorithm.  Note that swapping two points
    simply means swapping two integers in the lookup table, not
    reordering any lists*/
                                                                    30
  for(last_col=cm_new_length-1;last_col>=0;last_col--){
    /*each iteration finds the worst column up through last_col and makes it the last*/
    fewest = cm_new_length+2;  /*bigger than max possible*/
    for(col=0;col<=last_col;col++){
      if(cm_new_consistence[cm_lookup_new[col]] <= fewest){
          fewest = cm_new_consistence[cm_lookup_new[col]];
          worst_col = col;
      }
    }
    /*swap worst with last, unless perfectly consistent*/             40
    if(fewest == last_col+1){
      printf("\nNew method got %d consistent\n", last_col+1);
      Compare_CM_Methods();
      return(last_col+1);
    }
    swap_slave = cm_lookup_new[worst_col];
    cm_lookup_new[worst_col] = cm_lookup_new[last_col];
    cm_lookup_new[last_col]= swap_slave;
    for(col=0;col<cm_new_length;col++){
      cm_new_consistence[col] -=                                    50
          PairConsistence(points[col],
                          points[cm_lookup_new[last_col]],
                          autocor);
    }
    /*continue with next-to-last column*/
  }
  return(0);
}
```

Bibliography

[1] L. G. Shirley and G. R. Hallerman. Technical report 1025, applications of tunable lasers to laser radar and 3d imaging. *Lincoln Laboratory Journal*, 1996.

[2] Michael S. Mermelstein. A large-scale three-dimensional imaging system based on laser speckle. Master's thesis, Massachussets Institute of Technology, Department of Electrical Engineering and Computer Science, 1995.

[3] Lyle G. Shirley, Emory D. Ariel, Gregory R. Hallerman, Harold C. Payson, and John R. Vivilecchia. Advanced techniques for target discrimination using laser speckle. *Lincoln Lab Journal*, 1992.

[4] Lyle G. Shirley and Peter A. Lo. Bispectral analysis of the wavelength dependence of speckle: Remote sensing of object shape. *Journal of the Optical Society of America*, 1994.

[5] Peter Andrew Lo. Implementation of a bispectrum algorithm for laser radar range-resolved cross-section estimation. Master's thesis, Massachussets Institute of Technology, Department of Electrical Engineering and Computer Science, 1993.

[6] Richard G. Paxman, John H. Seldin, James R. Fienup, and Joseph C. Marron. Use of an opacity constraint in three-dimensional imaging. *SPIE v.2241*, 1994.

[7] J. C. Marron and K. S. Schroeder. Three-dimensional lensless imaging using laser frequency diversity. *Applied Optics*, 31, 1992.

[8] J. C. Dainty. *Laser Speckle and Related Phenomena.* Springer-Verlag, New York, 1984.

[9] Thomas S. Huang. *Image Reconstruction From Incomplete Observations.* Jai Press Inc., Greenwich, Connecticut, 1984.

[10] R. P. Millane. Redundancy in multidimensional deconvolution and phase retrieval. *Proceedings of SPIE v 1351*, 1990.

[11] Lyle G. Shirley and Benjamin Rahn. Remote 3-d imaging of a point-scattering array using laser speckle. 1994.

[12] James R. Fienup. Reconstruction of an object from the support of its autocorrelation. *Journal of the Optical Society of America*, 1982.

[13] K. Kuratowski and A. Mostowski. *Set Theory with an Introduction to Descriptive Set Theory.* North Holland Publishing Company, New York, 1976.

[14] S. Lawrence Marple. *Digital Spectral Analysis, With Applications.* Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[15] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes.* Cambridge University Press, New York, 1992.

[16] Norman R. Guivens. Reduced computation algorithm for phase retrieval. *Proceedings of the SPIE*, 1351, 1990.

[17] Stephen M. Kay and Stanley L. Marple Jr. Spectrum analysis—a modern perspective. *Proceedings of the IEEE*, 69(11), 1981.

[18] Alan V. Oppenhem and Jae S. Lim. *Advanced Topics in Signal Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1988.