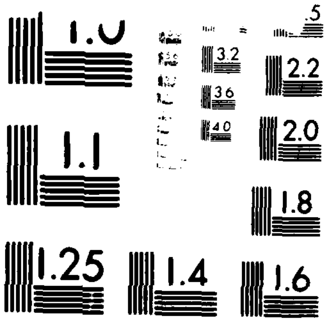
AD-A168 792 RECONSTRUCTION OF TWO-DIMENSIONAL SIGNALS FROM THE FOURIER TRANSFORM MAGNITUDE(U) MASSACHUSETTS INST OF TECH CAMBRIDGE RESEARCH LAB OF ELECTRON. 1/2

UNCLASSIFIED D IZRAELEVITZ JUN 86 TR-517 F/G 9/3 NL


1.0
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

[illegible]

12

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
Research Laboratory of Electronics
Room 36-615
Cambridge, MA 02139

Reconstruction of Two-Dimensional Signals from the Fourier Transform Magnitude

David Izraelevitz

Technical Report No. 517

This work has been supported in part by the Advanced Research Projects Agency monitored by ONR under Contract No. N00014-81-K-0742 and in part by the National Science Foundation under Grant ECS-8407285.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A168792

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Research Laboratory of Electronics
Massachusetts Institute of Technology

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
Office of Naval Research
Mathematical and Information Scien. Div.

6c. ADDRESS (City, State and ZIP Code)
77 Massachusetts Avenue
Cambridge, MA 02139

7b. ADDRESS (City, State and ZIP Code)
800 North Quincy Street
Arlington, Virginia 22217

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Advanced Research Projects Agency

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-81-K-0742

8c. ADDRESS (City, State and ZIP Code)
1400 Wilson Boulevard
Arlington, Virginia 22217

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
		NR 049-506	

11. TITLE (Include Security Classification) Reconstruction of Two-Dimensional Signals from Fourier Trans. Magn.

12. PERSONAL AUTHOR(S)
David Izraelevitz

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT (Yr., Mo., Day)
June 1986

15. PAGE COUNT
160

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis is concerned with the problem of reconstructing a discrete two-dimensional signal of known support from the Fourier transform magnitude only. This problem arises in many fields where imaging is desired, such as astronomy and wavefront sensing.

Since the autocorrelation function is easily calculated from the Fourier transform magnitude, we attack the equivalent problem of signal reconstruction from a known autocorrelation function. The main result of the thesis is a new algorithm for realizing this reconstruction. This algorithm is guaranteed to yield the correct solution given accurate measurements and is much more computationally attractive than previous reconstruction algorithms. The result is based on the detailed analysis of the zeros

(cont.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Kyra M. Hall
RLE Contract Reports

22b. TELEPHONE NUMBER (Include Area Code)
(617) 253-2569

22c. OFFICE SYMBOL

EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

19. Abstract continued

of a polynomial which is essentially the two-dimensional z-transform of the known autocorrelation signal. From this analysis, a large number of zeros of the z-transform of the unknown discrete signal are extracted. This set of zeros is then used to extract the signal values via the solution of a set of linear equations.

Examples of the application of this algorithm to several families of images is presented, along with a discussion of the accuracy and computational requirements of the new algorithm. We conclude with a discussion of the application of the ideas of this thesis to the area of two-dimensional filter design and stability testing.

Reconstruction of Two-Dimensional Signals from the Fourier Transform Magnitude

by

David Izraelevitz

Submitted in partial fulfillment of the requirements for the degree of Doctor of Science at the Massachusetts Institute of Technology.
May 8, 1986

Abstract

This thesis is concerned with the problem of reconstructing a discrete two-dimensional signal of known support from the Fourier transform magnitude only. This problem arises in many fields where imaging is desired, such as astronomy and wavefront sensing.

Since the autocorrelation function is easily calculated from the Fourier transform magnitude, we attack the equivalent problem of signal reconstruction from a known autocorrelation function. The main result of the thesis is a new algorithm for realizing this reconstruction. This algorithm is guaranteed to yield the correct solution given accurate measurements and is much more computationally attractive than previous reconstruction algorithms. The result is based on the detailed analysis of the zeros of a polynomial which is essentially the two-dimensional z-transform of the known autocorrelation signal. From this analysis, a large number of zeros of the z-transform of the unknown discrete signal are extracted. This set of zeros is then used to extract the signal values via the solution of a set of linear equations.

Examples of the application of this algorithm to several families of images is presented, along with a discussion of the accuracy and computational requirements of the new algorithm. We conclude with a discussion of the application of the ideas of this thesis to the area of two-dimensional filter design and stability testing.

Thesis Supervisor: Jae S. Lim
Title: Associate Professor of Electrical Engineering.

With Terry

Acknowledgements

Acknowledgements are always fun. I have heard of only one person who wrote the acknowledgements before writing the rest of the thesis and I believe he ended up never finishing the thesis. How could he, when the best part of the whole dissertation has been done and there is still left at least three years of hard work?

The rest of us, I suspect, wisely wait until the very end to slip in this page. By this time all the hard work is done; all the writing is complete, the equations and figures checked and the advisor and readers have approved of the contents. All the anxiety is over with and one can think back with fondness of all the help received on the way here.

My sincere thanks go to Professor Jae S. Lim for the advice and focus offered throughout the doctoral program. I have worked with Jae for the last five years, ever since I came to MIT, and during this time have learned from him a tremendous amount on the fine art of research. I also thank Professor Alan Oppenheim for his continuing interest and insight into the problem of signal reconstruction in general and phase retrieval in particular. Professors Michael Artin of the Mathematics Dept. at MIT and Eric Kaltofen of the Computer Science Dept. at RPI provided valuable help in the initial stages of my research. I also want to acknowledge the value of several rewarding conversations with Avideh Zakhor and Dr. Susan Curtis, who are involved in research related to the work presented here. In addition, the results of one of the appendices of this thesis is due to a collaboration between Ms. Zakhor and myself. Many, if not most of the members of the DSPG group engaged in the sometimes painful task of listening and discussing my half-baked ideas. Dr. Michael Wengrovitz, Cory Myers and Webster Dove were specially abused in this respect.

Of course, one does not complete a thesis by signal processing alone. Ever since I left home for college, I have always relied on the love, support and Sunday phone calls of my parents. I thank them now as I will thank them forever for this. Finally, at about the same time as I started my doctoral thesis, I fell in love with a cute girl from St. Louis. The deepest thanks go to my wife, Terry, for being with me. It made it all easy.

Contents

Chapter 1

Introduction

The magnitude and phase of the Fourier transform of an arbitrary multidimensional signal are independent functions of frequency. In many applications, however, there is additional information regarding the signal which provides a very strong connection between its Fourier transform magnitude and phase. One example of such additional information is the common condition that the signal is non-zero only over a specified region. In this case, it has been shown that almost all multidimensional signals which are non-zero only over a specified region are uniquely specified, in a sense, by knowledge of only its Fourier transform magnitude [1,2]. Hence, once the Fourier transform magnitude is known, the Fourier transform phase is determined as well. For this reason, the problem of reconstruction from Fourier transform magnitude is also called the phase retrieval problem.

The reconstruction of a two-dimensional signal from its Fourier transform magnitude has been the object of much study. This interest is guided by the wide range of applications of results in this area. One such application is in astronomy [3]. The effect limiting the resolution capabilities of the largest optical telescopes is not the diffraction limit of the lens, but rather the turbulence of the earth's atmosphere. During a short time period, the atmosphere can be considered to introduce on the incoming optical wave a spatially varying, time-invariant random phase delay due to inhomogeneities induced by thermal gradients. These inhomogeneities are slowly varying with respect to short exposure times. Thus, over such a short time period, the atmosphere can be modeled as a glass plate of spatially varying thickness over the telescope aperture.

This phase aberration, although it blurs each individual exposed image, does not affect the spatial autocorrelation function. A way of circumventing this blurring effect is to first measure an accurate estimate of the spatial autocorrelation function. This can be done via Labeyrie interferometry [4]. In this procedure an interferometer is used to image the spatial autocorrelation function over a small time period. This estimate will be very noisy because of the short exposure time. The signal-to-noise ratio however can be increased by averaging several short exposures. Thus a diffraction limited autocorrelation function can be measured which is not affected by atmospheric blurring. It is clear that a reliable method for extracting the image of the astronomical object from such interferometer data would in effect greatly increase the resolution capabilities of earth-based telescopes.

A possible application of phase retrieval to electron microscopy lies in the possibility of indirect phase measurement from magnitude measurement [5]. Photographic film can only record the intensity of the field impinging on it. However, the phase of the field provides important information on the object being viewed. For example, thin objects may be considered as modulating the phase of the electron wave while not affecting the magnitude. The actual phase delay introduced depends on the thickness and composition of the specimen. Retrieving the phase delays from the recorded field intensity would yield an indirect way of measuring the specimen properties.

X-ray crystallography is a third realm where reconstruction from Fourier transform magnitude may prove useful [6]. Physical arguments show that the angles at which the x-rays are diffracted from a crystal specimen and the intensity of the diffracted wave at each angle are related to the Fourier transform magnitude of the electron density of the crystal under study. An important part of crystallography is the task of deducing the arrangement of atoms in the crystal from knowledge of such diffraction data.

The importance of the phase retrieval problem has led several researchers to propose algorithms for reconstruction from Fourier transform magnitude. However, previously presented algorithms fall into either of two categories; they are heuristic algorithms which often do not converge to the true reconstruction, or they are computationally too expensive for even moderate size signals. The purpose of this thesis is to present a new algorithm for reconstruction of multidimensional discrete signals from Fourier transform magnitude which is a closed form solution to the problem and which has been used with success in reconstructing signals of moderate size.

The thesis is divided into eight chapters; Chapter 2 develops an appropriate notation, reviews basic properties of signals and introduces some mathematical concepts. Chapter 3 is concerned with the review of previous results in reconstruction of both one- and two-dimensional signals from the Fourier transform magnitude. The formulation of the phase retrieval problem as a bivariate polynomial factorization problem is given in Chapter 4. Previous algorithms for factoring polynomials in two variables and their connection and possible application to the phase retrieval problem are also discussed. Based on this framework, a new algorithm for factoring large polynomials in two variables is developed in Chapter 5. This leads to a new closed form algorithm for solving the phase retrieval problem. Examples of the application of the new phase retrieval algorithm is the subject of Chapter 6. Chapter 7 discusses the application of the ideas developed in this thesis to the areas of general bivariate polynomial factorization and filter stability testing. A summary of the work presented and suggestions for future research is the subject of Chapter 8.

Chapter 2

Notation and Basic Properties

In this chapter we review some concepts from digital signal processing and polynomials in one or several variables. The purpose of this review is primarily to develop a consistent notation and provide a base for our later development. The interested reader may consult [7] for a more detailed discussion of the signal processing topics described here. A development of the properties of polynomials reviewed here is contained in [8].

2.1 One- and Two-Dimensional Signals

A one-dimensional discrete signal $x[n]$ or a two-dimensional discrete signal $x[m, n]$ is a real or complex function of a single integer index n or two integer indices m and n respectively. The support of $x[n]$ is the set (or sometimes a superset) of all indices n such that $x[n]$ is non-zero. The support of $x[m, n]$ is also defined as the set of all index pairs such that $x[m, n]$ is non-zero.

In our discussion we will find special cases of support to be especially useful. A signal, either one- or two-dimensional, is said to be of finite extent if its support is bounded; in one dimension, this means that there is an index pair, (n_{min}, n_{max}) such that $x[n] = 0$ for $n > n_{max}$ and $n < n_{min}$. Similarly in two dimensions a signal is of finite extent if there is an index quadruple $(m_{max}, m_{min}, n_{max}; n_{min})$ such that $x[m,n] = 0$ whenever any of the four conditions below hold:

$$m > m_{max},\ m < m_{min},\ n > n_{max},\ n < n_{min} \tag{2.1}$$

Pictorially, this means that the non-zero values of $x[m,n]$ can be enclosed in a box, Figure 2.1.

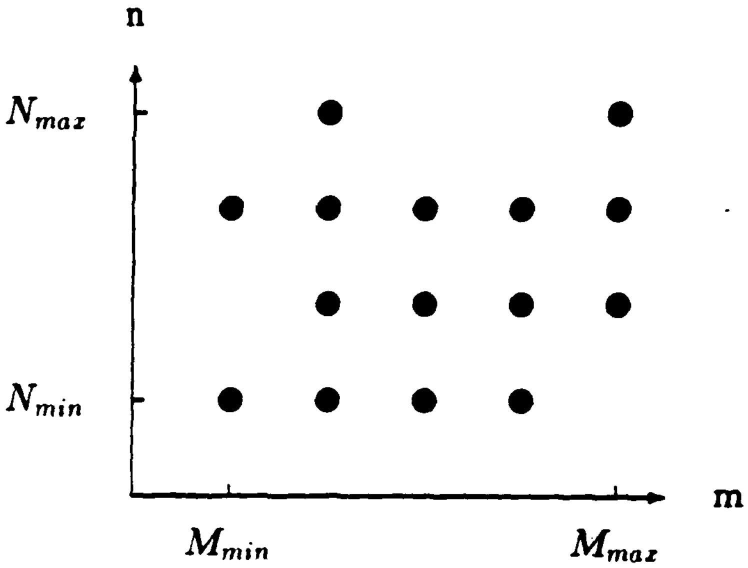


Figure 2.1: A two-dimensional signal with support $[M_{min}, M_{max}] \times [N_{min}, N_{max}]$.

We will denote a region of support consisting of all indices $a \leq n \leq b$ as $[a,b]$. In two dimensions a support comprising all index pairs (m,n) satisfying $a \leq n \leq b$, $c \leq m \leq d$ will be denoted by $[a,b] \times [c,d]$. We will abbreviate $[a,b] \times [a,b]$ by $[a,b]^2$.

Associated with any signal $x[n]$ is a Laurent series called its z-transform,

$$X(z) = \sum_n x[n]z^{-n} \tag{2.2}$$

where z is a complex number. The set of all z where the summation converges is called the region of convergence (ROC) of $X(z)$. In this thesis, we will be dealing primarily with signals of finite extent, in which case the ROC includes all of the complex z-plane with the possible exclusion of the origin or infinity. Evaluating the z-transform in (2.2) on the unit circle $|z| = 1$ yields the Fourier transform,

$$X(e^{jv}) = \sum_n x[n]e^{-jnv} \tag{2.3}$$

Even if $x[n]$ is real, its Fourier transform may be complex-valued, and can thus be represented in terms of its real and imaginary components,

$$X(e^{jv}) = X_r(e^{jv}) + jX_i(e^{jv}) \tag{2.4}$$

or in polar form

$$X(e^{jv}) = |X(e^{jv})|e^{j\Theta_x(v)} \tag{2.5}$$

Two-dimensional discrete signals also have a z-transform which is a complex-valued function of two complex numbers w and z,

$$X(w,z) = \sum_m \sum_n x[m,n]w^{-m}z^{-n} \tag{2.6}$$

The definition of an ROC also applies to two-dimensional z-transforms. Whenever the ROC includes the bicircle $|w| = 1$, $|z| = 1$, the Fourier transform of $x[m,n]$ is defined:

$$X(e^{ju}, e^{jv}) = \sum_m \sum_n x[m,n]e^{-jum}e^{-jvn} \tag{2.7}$$

An important signal which is generated from $x[n]$ is its autocorrelation function, defined via the summation below,

$$r[n] = \sum_{l} x[l]x^*[l+n] \tag{2.8}$$

It is straightforward to show that if $x[n]$ is a finite extent signal of support $[a, b]$ then $r[n]$ has support $[-(b-a), (b-a)]$. Note that the autocorrelation function is invariant to multiplication of $x[n]$ by a constant of unit magnitude or to a translation of $x[n]$.

From the convolution theorem [7], the following relationship is derived between $X(z)$ and the z-transform of $r[n]$, $R(z)$,

$$R(z) = X(z)X^*(\frac{1}{z^*}) \tag{2.9}$$

When evaluated on the unit circle, the above equation collapses to a relationship between the Fourier transforms of $x[n]$ and $r[n]$,

$$R(e^{j\nu}) = |X(e^{j\nu})|^2 \tag{2.10}$$

From (2.10) one can make an important observation that if two signals have the same autocorrelation function, they must have the same Fourier transform magnitude, and vice versa.

The autocorrelation function of $x[m, n]$ is similarly defined by

$$r[m,n] = \sum_{k}\sum_{l} x[k,l]x^*[k+m, l+n] \tag{2.11}$$

The support of $r[m, n]$ is given by $[-(b-a), (b-a)] \times [-(d-c), (d-c)]$ for $x[m, n]$ with support $[a, b] \times [c, d]$.

The convolution theorem also applies to the two-dimensional case; the z-transform of $r[m,n]$, $R(w,z)$, is given by

$$R(w,z) = X(w,z)X^*(\frac{1}{w^*}, \frac{1}{z^*}) \qquad (2.12)$$

On the unit bicircle, (2.12) becomes

$$R(e^{ju}, e^{jv}) = |X(e^{ju}, e^{jv})|^2 \qquad (2.13)$$

Thus we see that as in the one-dimensional case, if the Fourier transform magnitude of a signal is known then its autocorrelation function is known also and vice versa.

2.2 Polynomials in One and Two Variables

In the subsequent discussion, we will be dealing primarily with signals of finite extent. In this case, the corresponding z-transforms are essentially polynomials. Therefore, it is important to understand some properties of polynomials which will be used later on.

A polynomial in one variable z is a function of the form

$$p(z) = \sum_{n=0}^{N} p_n z^n \qquad (2.14)$$

where the p_n are complex numbers and N is finite. The degree of a polynomial in one variable (or one-dimensional polynomials) is the largest power to which the indeterminate variable is raised. The degree of a polynomial $p(z)$ will be denoted by $deg(p)$. Thus the polynomial $p(z)$ in (2.14) above has degree $deg(p) = N$.

As a result of the Fundamental Theorem of Algebra [8], all one dimensional polynomials of degree N can always be expressed as a product of N polynomials of degree

1 and a constant,

$$p(z) = p_N \prod_{n=1}^{N} (z - z_n) \tag{2.15}$$

This product representation is unique up to a permutation of the product terms. Since $p(z_n) = 0$ for all n, z_n is called a zero of $p(z)$. We note that z_n will generally be complex, even if the p_n set is real.

A polynomial $p(z)$ with $deg(p) > 0$ is called reducible if it can be expressed as the product of two polynomials $p_1(z)$, $p_2(z)$ with $deg(p_1) > 0$ and $deg(p_2) > 0$, i.e.,

$$p(z) = p_1(z)p_2(z) \tag{2.16}$$

If no such decomposition is possible, then $p(z)$ is called irreducible. From the decomposition presented in (2.15), we see that the only irreducible polynomials in one variable are polynomials of degree 1 [1]. We will see, however, that the situation is quite different for polynomials in two (or more) variables.

Associated with any polynomial is a "mirror" polynomial consisting of coefficients in reversed order and conjugated. For example, for the polynomial $p(z)$ in (2.14), the mirror polynomial $\tilde{p}(z)$ is defined by

$$\tilde{p}(z) = \sum_{n=0}^{N} p^*_{N-n} z^n \tag{2.17}$$

There is a very simple relationship between the zeros of $p(z)$ and $\tilde{p}(z)$; namely, if z_0 is a zero of $p(z)$, then z_0^{*-1} is a zero of $\tilde{p}(z)$.

[1] Strictly speaking, irreducibility is defined with respect to a specific field. Whether a polynomial is irreducible or not may depend on the field of interest. However, in our discussion we will only be dealing with polynomials over the field of complex numbers.

One can also study polynomials in two variables, also called two-dimensional polynomials or bivariate polynomials. In this case there are two variables w, z,

$$p(w,z) = \sum_{m=0}^{M} \sum_{n=0}^{N} p_{m,n} w^m z^n \tag{2.18}$$

where again $p_{m,n}$ are allowed to be complex numbers. The degree in w, $deg_w(p)$, of $p(w,z)$ is the highest order to which the indeterminate w is raised. In the example above, $deg_w(p) = M$. Similarly, the degree in z of $p(w,z)$, $deg_z(p)$, is N. The degree of the polynomial $p(w,z)$ $deg(p)$, is defined by the pair of integers $(deg_w(p), deg_z(p))$; in this case, $deg(p) = (M,N)$. We will also define the total degree of $p(w,z)$, $totdeg(p)$, as the degree of the univariate polynomial $p(w,w)$. We note that in some areas of mathematics, e.g., algebraic geometry, the total degree is considered ***the*** degree of $p(w,z)$.

A decomposition of an arbitrary bivariate polynomial into a product of polynomials of a lower degree is not always possible. This is because, unlike the case for one-dimensional polynomials, there are irreducible polynomials in two variables for any degree, except of course polynomials which are of degree 0 in one of the variables.

Example: The polynomial of degree (M,N) below is easily checked to be irreducible for any $N > 0$ and $M > 0$ [9].

$$p(w,z) = p_0 + p_1 w + \cdots + p_1 w^M + z^N \tag{2.19}$$

If a product decomposition does exist, then it is essentially unique as expressed in the following theorem [10],

Theorem 2.1 *Every polynomial $f(w,z)$ is expressible as the product*

$$f = P_1 P_2 \cdots P_k \tag{2.20}$$

of a finite number of irreducible polynomial factors P_i, and every other such expression for f has factors which are the same except possibly for constant multipliers.

Note that this theorem is equivalent to the Fundamental Theorem of Algebra except that in this case there is no specification of the degrees of each irreducible factor P_i.

An important representation of a bivariate polynomial $p(w,z)$ is to consider it as a polynomial in w which has coefficients consisting of polynomials in z,

$$p(w,z) = p_0(z) + p_1(z)w + \cdots p_M(z)w^M \tag{2.21}$$

If $deg(p) = (M,N)$ then the degree of each $p_n(z)$ must be less than or equal to N. Of course, one can similarly consider the same polynomial $p(w,z)$ as a one-dimensional polynomial in z with coefficients consisting of polynomials in w.

Similarly, $\tilde{p}(w,z)$, the "mirror" image of $p(w,z)$, is defined by

$$\tilde{p}(w,z) = \sum_{m=0}^{M}\sum_{n=0}^{N} p^*_{M-m,N-n} w^m z^n \tag{2.22}$$

The zeros of $p(w,z)$ and $\tilde{p}(w,z)$ are related via a relationship analogous to the univariate case; if (w_0, z_0) is a zero of $p(w,z)$, then (w_0^{*-1}, z_0^{*-1}) is a zero of $\tilde{p}(w,z)$.

2.3 Z-Transforms of Finite Extent Signals

Recall that a one-dimensional discrete signal $x[n]$ with finite extent support $[a,b]$ has z-transform,

$$X(z) = \sum_{n=a}^{b} x[n]z^{-n} \tag{2.23}$$

The expression above can be written equivalently as

$$X(z) = z^{-b}\sum_{n=0}^{b-a} x[b-n]z^{n} \tag{2.24}$$

The second term can be identified as a polynomial in the variable z with coefficients $p_n = x[b-n]$. We will call the expression

$$p_x(z) = \sum_{n=0}^{b-a} x[b-n]z^n \tag{2.25}$$

the polynomial *associated* with $x[n]$. Thus, the two signals $x[n]$ and $x[n+k]$ for any fixed k have the same associated polynomial. We will assume that the degree of $p_x(z)$ is as small as possible; thus, any associated polynomial of degree n has $a_0 \neq 0$ and $a_n \neq 0$. Formally, we define $p_x(z)$ as the polynomial of least degree such that $X(z) = z^k p_x(z)$ for some integer k.

There is a relationship concerning the associated polynomials of $x[n]$ and $r[n]$, which is very similar to the convolution theorem relationship of (2.9),

$$p_r(z) = p_x(z)\tilde{p}_x(z) \tag{2.26}$$

where we recall that $\tilde{p}_x(z)$ is the mirror polynomial of $p_x(z)$.

We can also define associated polynomials for two-dimensional signals. Consider the z-transform of a two-dimensional signal $x[m,n]$ of support $[a,b] \times [c,d]$,

$$X(w,z) = w^{-d}z^{-b} \sum_{m=0}^{d-c} \sum_{n=0}^{b-a} x[d-m, b-n]w^m z^n \tag{2.27}$$

Again, we define the polynomial

$$p_x(w,z) = \sum_{m=0}^{d-c} \sum_{n=0}^{b-a} x[d-m, b-n]w^m z^n \tag{2.28}$$

as the polynomial associated with $x[m,n]$. The corresponding relationship between the polynomial associated with $x[m,n]$ and $r[m,n]$ is given by

$$p_r(w,z) = p_x(w,z)\tilde{p}_x(w,z) \tag{2.29}$$

2.4 The Phase Retrieval Problem

Using the notation presented in this chapter, we can state succinctly what is the specific problem we are trying to solve. Consider an unknown two-dimensional discrete signal $x[m,n]$. The problem is to reconstruct $x[m,n]$ given,

- The support of $x[m,n]$.
- The Fourier transform magnitude, $|X(e^{ju}, e^{jv})|$, of $x[m,n]$, at all frequencies [2].

This problem can be reformulated in several different ways. From the relationship of (2.13), we know that knowledge of the Fourier transform magnitude is equivalent to knowledge of the autocorrelation function of $x[m,n]$. Thus the phase retrieval problem can be stated as, reconstruct $x[m,n]$ given,

- The support of $x[m,n]$.
- The autocorrelation function, $r[m,n]$, of $x[m,n]$.

The polynomial, $p_x(w,z)$ associated with $x[m,n]$ is an almost invertible representation of a signal, since from the coefficients of $p_x(w,z)$ we can restore, to a linear shift, the original signal. We will find that this shift ambiguity is inherent in phase retrieval. Using associated polynomials, the phase retrieval problem can be considered as reconstructing $p_x(w,z)$ given,

- the degree of $p_x(w,z)$.

[2] It can be shown that for finite extent signals it is only required that the Fourier transform magnitude be known on a sufficiently dense sampling grid which depends on the known support constraint. We will not consider this aspect of the problem. For a discussion of this question, see [9]. In the case where the autocorrelation function is known directly, then the support of $x[m,n]$ can be estimated.

- the polynomial associated with the autocorrelation function, $p_r(w, z)$.

In considering the phase retrieval problem we will make use of all three formulations.

Chapter 3

Previous Results in the Phase Retrieval Problem

The earliest results on reconstruction from Fourier transform magnitude for signals with given support relate to the one-dimensional case. These results are reviewed in this chapter, both because they provide some insight into the general problem of phase retrieval and also because solving one-dimensional problems turns out to be an integral part of several proposed algorithms for two-dimensional phase retrieval.

The second part of this chapter discusses the two-dimensional problem. A review of the known results on the uniqueness and characterization of the solution is presented followed by a review and critique of several algorithms which have been proposed for reconstruction of images from the Fourier transform magnitude.

3.1 Reconstruction of One-Dimensional Signals from Known Support and Magnitude

The earliest reference to the one-dimensional case seems to be a paper by Akutowicz [11] which provided a complete characterization of the problem of one-dimensional continuous reconstruction from Fourier transform magnitude. This relationship was rediscovered simultaneously by Walther [12] and Hofstetter [13].

While Akutowicz, Walther and Hofstetter considered the reconstruction of a continuous one-dimensional signal from its Fourier transform magnitude, more recently Hayes [9] has considered the discrete one-dimensional case. The situation of a discrete one-dimensional signal can be derived in a manner similar to the argument followed by Akutowicz, Walther and Hofstetter. His development is given here while stressing some points which will become important in our later discussion.

In the section below, we discuss virtually simultaneously the problems of characterizing the number of solutions to the one-dimensional phase retrieval problem and an algorithm for producing such a reconstruction. This is in contrast with our discussion of the two-dimensional phase retrieval problem later in this chapter, where reconstruction algorithms are developed at a distance from uniqueness considerations.

The object of the algorithm is to find all signals of extent $[0, N]$ which have the Fourier transform magnitude $|X(e^{j\nu})|$. From (2.10) we know that we can immediately calculate the autocorrelation function of any such signal via,

$$F^{-1}\{|X(e^{j\nu})|^2\} = r[n] \tag{3.1}$$

Since the autocorrelation function of an $N+1$ point signal has support $[-N, N]$ the z-transform of $r[n]$ becomes

$$R(z) = \sum_{n=-N}^{N} r[n]z^{-n} \tag{3.2}$$

The polynomial associated with $r[n]$ is

$$p_r(z) = \sum_{n=0}^{2N} r[N-n]z^n \tag{3.3}$$

The next step is to develop the product decomposition of $p_r(z)$. Since $r[n] = r^*[-n]$, $p_r(z)$ is its own mirror polynomial. Thus if $p_r(z_0) = 0$, then $p_r(z_0^{*-1}) = 0$ also. Furthermore, since the Fourier transform of $r[n]$ is non-negative, the zeros z_0 and z_0^{*-1} will be distinct, or occur in even multiplicities. Therefore, $p_r(z)$ can be expressed as

$$p_r(z) = A \prod_{i=1}^{N} (z - z_i)(1 - z_i^* z) \tag{3.4}$$

where A is positive. Now suppose that $x[n]$ with associated polynomial $p_x(z)$ is one solution, i.e., $x[n]$ has $r[n]$ as an autocorrelation function. Recall that

$$p_r(z) = p_x(z)\tilde{p}_x(z) \tag{3.5}$$

The object of the discussion below is to generate a $p_x(z)$ that satisfies (3.5). From the set of zeros in (3.4), one zero is picked from each pair (z_i, z_i^{*-1}) and a polynomial with such zeros is generated but with an unknown scale constant α,

$$p_x(z) = \alpha \prod_{i \in I} (z - z_i) \prod_{j \notin I} (1 - z z_j^*) \tag{3.6}$$

where I is a subset of $\{1, N\}$. The mirror polynomial of $p_x(z)$ is $\tilde{p}_x(z)$ and has product decomposition given by

$$\tilde{p}_x(z) = \alpha^* \prod_{i \in I} (1 - z z_i^*) \prod_{j \notin I} (z - z_j) \tag{3.7}$$

Now the product of $p_x(z)$ and $\tilde{p}_x(z)$ will equal $p_r(z)$ if $|\alpha|^2 = A$. Thus, α can be picked to be any complex number such that $|\alpha| = \sqrt{A}$.

From the procedure above, a polynomial associated with a possible solution signal has been constructed. Consider two different polynomials which have been constructed using the algorithm above. They will have some zeros in common and some zeros which are "flipped" pairs. Thus, one can generate alternate polynomials from a single construction by "zero-flipping".

At this moment we should review the process of generating the associated polynomial of a solution to the phase retrieval problem. The first step is to extract the autocorrelation function from the Fourier transform magnitude information. From this signal, its associated polynomial is extracted. The zeros of this polynomial are calculated and these zeros occur in N pairs. From each of these N pairs one member is chosen and thus determine the zeros of the solution associated polynomial. Since one can pick either member from each pair, a total of 2^N possible zero sets, and corresponding polynomials can be constructed. The final step is to find an appropriate scale constant for the solution associated polynomial. This scale constant has a prescribed magnitude but arbitrary phase.

If the desired solution must be real then zeros chosen for a solution associated polynomial must include complex conjugate pairs, thereby decreasing the number of solutions from 2^N possibly down to $2^{N/2}$ if all zeros are complex. The scale constant is also restricted to being real, so only a sign ambiguity instead of a phase ambiguity remains.

Once the appropriate associated polynomial has been specified we can label as a solution any signal which is associated with that polynomial. For example, if the extracted polynomial is given by

$$p_x(z) = \sum_{n=0}^{N} a_n z^n \tag{3.8}$$

then a possible solution is

$$\begin{aligned} x[n] &= a_{N-n} \quad \text{for } 0 \leq n \leq N \\ &= 0 \qquad\quad \text{elsewhere} \end{aligned} \tag{3.9}$$

In fact any signal of the form

$$\begin{aligned} x[n+k] &= a_{N-n} \quad \text{for } 0 \leq n \leq N \\ &= 0 \qquad\quad \text{elsewhere} \end{aligned} \tag{3.10}$$

for any integer k is a valid solution.

Although the algorithm presented in the previous discussion is rather simple and at first sight does not seem to deserve the painstaking attention which has been directed to it, we will find that the two-dimensional problem can be attacked in much the same way; the only difference, and unfortunately it is a big difference, is that extracting the appropriate zeros of the polynomial associated with a solution will be much more involved.

3.2 Reconstruction of Two-Dimensional Signals from Known Support and Magnitude

3.2.1 Theoretical Results

Although the one-dimensional reconstruction from Fourier transform magnitude given a finite support has been completely solved, the two-dimensional problem has proven to be both more challenging and more interesting. The key to characterizing the general reconstruction from Fourier transform magnitude problem is to note how the multiplicity of solutions in the one-dimensional problem is introduced. This multiplicity depends on the ability to decompose the polynomial associated with a solution into a product of lower order polynomials, in fact, polynomials of degree 1. It is this property that allows for the "zero-flipping" described above, resulting in several solutions. We will discuss the fact that most two-dimensional signals do not allow such zero-flipping; hence, the solution to the phase retrieval problem becomes essentially unique.

The original insight on the relationship between factorability of the z-transform and uniqueness of the Fourier transform magnitude reconstruction was presented by Bruck and Sodin [1], and later formalized by Hayes [2].

Consider a known Fourier transform magnitude $|X(e^{ju}, e^{jv})|$. We want to find a signal $x[m, n]$ of support $[0, M] \times [0, N]$ with the given Fourier transform magnitude. From the Fourier transform magnitude we can calculate the autocorrelation function $r[m, n]$ which must have support $[-M, M] \times [-N, N]$. The polynomial associated with $r[m, n]$ must be of the form,

$$p_r(w, z) = p_x(w, z)\tilde{p}_x(w, z) \tag{3.11}$$

Thus, $p_r(w, z)$ must satisfy two conditions: first, it must be a polynomial factorable into two smaller polynomials of degree (M, N) each. Second, each of the factors must be mirror polynomials of each other. Any factorization of $p_r(w, z)$ which satisfies these two conditions corresponds to a valid signal $x[m, n]$ which satisfies the original information.

An important question is whether there is only one factorization. To answer this question it is necessary to look at the factorability of $p_x(w, z)$ itself. Following Hayes [9] define the following equivalence class,

$$y[m, n] \sim x[m, n] \text{ if } y[m, n] = e^{j\theta} x[k_1 \pm m, k_2 \pm n] \tag{3.12}$$

for some integer pair (k_1, k_2) and some θ. Thus, $y[m, n]$ and $x[m, n]$ are equivalent if they are related by a linear shift, a phase change, and/or a rotation by 180 degrees. From the properties of associated polynomials, we see that $x[m,n] \sim y[m,n]$ is the same as stating that $p_y(w,z) = \exp(j\theta)p_x(w,z)$ or $p_y(w,z) = \exp(j\theta)\tilde{p}_x(w,z)$. Using this definition, we can state a uniqueness theorem for reconstruction from Fourier transform magnitude,

Theorem 3.1 *If $x[m,n]$ has an irreducible associated polynomial, then all other $y[m,n]$ which have the same Fourier transform magnitude must be equivalent to $x[m,n]$.*

The complete proof of this statement is given in [9] for the case of real $x[m,n]$. The case for complex $x[m,n]$ can be shown in a similar manner.

It has been shown that factorable polynomials form a zero-measure set in the space of two-dimensional signals [14]. A subsequent contribution by Sanz et al. [15] showed that not only do signals with a factorable z-transform form a zero-measure set, but they also form a non-dense set in the space of two-dimensional signals. From these results we can conclude that most two-dimensional signals have an associated polynomial $p_x(w,z)$ which is irreducible. Moreover, the polynomial remains irreducible if the signal is perturbed by a small amount.

Several comments are in order regarding Theorem 3.1. We want to note first of all that a finite extent constraint must be placed on $y[m,n]$, i.e., in the Hayes (and Bruck and Sodin) proof, it is implicitly assumed that $y[m,n]$ is a finite extent signal. Otherwise, "infinite order polynomials" do not satisfy a unique factorization property of Theorem 2.1. This observation implies that some knowledge of the support of the signal is needed in order to assure a well-behaved uniqueness result. For example, if the support of $x[m,n]$ is $[0,3]^2$, there may be only essentially one signal of finite extent with the same Fourier transform magnitude as $x[m,n]$ but there may be many signals of extent $[0,100]^2$ with *almost* the same Fourier transform magnitude as $x[m,n]$.

Under some conditions, it may be possible to specify the $x[m,n]$ with a given Fourier transform magnitude to a greater extent than just the fact that it is a member of the equivalence class defined above. For example, if the signal is composed of image intensities, then the sign ambiguity disappears. Also, if the support of the signal is known to be such that it is not invariant to a 180 degree rotation, for example a triangular support, then the rotation ambiguity is removed. However, if the support is symmetric, for example, square, or only bounded by a symmetric region, then we cannot distinguish between the original image and the image reversed. This ambiguity is all that is left of the multiplicity of solutions which was observed in the one-dimensional case that was due to the "zero-flipping"; either no zeros are flipped, or they all are. The linear shift ambiguity itself is seldom of importance.

The continuous two-dimensional case has only been studied in depth recently. An early discussion is due to Huiser and Van Toorn [16] who first studied the question. More recently, Sanz et al. [17] showed that the non-uniqueness of the case studied by Huiser was due strictly to the fact that the Laplace transform Huiser considers is factorable, and in fact showed that multidimensional continuous signals with Laplace transforms which are non-factorable entire functions can be uniquely reconstructed from the magnitude of the Fourier transform.

3.2.2 Study of Previous Algorithms for Phase Retrieval

Fienup Algorithm

The development of algorithms for reconstruction of two-dimensional signals from Fourier transform magnitude preceded the estabiishment of conditions for uniqueness. The earliest family of algorithms were of an iterative nature, pioneered by Fienup [18,19]. These algorithms are based on work by Gerchberg and Saxton [20], and since the introduction by Fienup have been studied by Hayes [2], Sanz and Huang [21], Levi and Stark [22], and Won et al. [23], among others. In this section we describe and study three versions introduced by Fienup. The simplest algorithm is the Gerchberg-Saxton-Fienup (GSF) algorithm which iteratively imposes the known support in the spatial domain and the known Fourier transform magnitude in the frequency domain. The second algorithm considered is the Gerchberg-Saxton-Fienup algorithm with positivity constraint (GSF-P) which forces each estimate to be non-negative as well as of the given support. Although positivity is not a requirement for uniqueness of reconstruction, it is hoped that using such additional information will aid in speed of convergence. The final algorithm considered is the hybrid input-output (HIO) algorithm developed by Fienup which is described below. The object of this section is to describe these algorithms and evaluate their performance.

The algorithms begin with an initial estimate which can be derived either from a priori knowledge of the signal, i.e., a low resolution image, or in the case of no other a priori knowledge, from an image with either a random phase component which satisfies the Fourier transform magnitude constraint, or from an image with random coefficients which satisfies the support constraint. In our discussion, we take the last approach.

Given an estimate $x_i[m,n]$ of $x[m,n]$ which satisfies the support constraints, an auxiliary signal $\hat{x}_i[m,n]$ is calculated by the three algorithms, where $\hat{x}_i[m,n]$ has the correct Fourier transform magnitude and the same phase as $x_i[m,n]$,

$$\hat{x}_i[m,n] = F^{-1}\{|X(e^{ju}, e^{jv})|e^{j\arg(X_i(e^{ju},e^{jv}))}\} \qquad \forall\ (m,n) \tag{3.13}$$

where $X_i(e^{ju}, e^{jv})$ is the Fourier transform of $x_i[m,n]$. However, in general, $\hat{x}_i[m,n]$ will not satisfy the known support and/or positivity constraints. The difference between the three algorithms is how the next estimate $x_{i+1}[m,n]$ is calculated from $\hat{x}_i[m,n]$ and $x_i[m,n]$.

For GSF, $x_{i+1}[m,n]$ is calculated as,

$$x_{i+1}[m,n] = \begin{cases} \hat{x}_i[m,n] & (m,n) \in S \\ 0 & \text{elsewhere} \end{cases} \tag{3.14}$$

where the S denotes all index pairs where $x[m,n]$ is allowed to be non-zero. In the GSF-P algorithm, $x_{i+1}[m,n]$ is derived from,

$$x_{i+1}[m,n] = \begin{cases} \hat{x}_i[m,n] & (m,n) \in S \text{ and } \hat{x}_i[m,n] \geq 0 \\ 0 & \text{elsewhere} \end{cases} \tag{3.15}$$

Finally, HIO uses both the previous estimate $x_i[m,n]$ and $\hat{x}_i[m,n]$,

$$x_{i+1}[m,n] = \begin{cases} \hat{x}_i[m,n] & (m,n) \in S \text{ and } \hat{x}_i[m,n] \geq 0 \\ x_i[m,n] - \hat{x}_i[m,n] & \text{elsewhere} \end{cases} \tag{3.16}$$

From this estimate, $x_{i+1}[m,n]$, the iteration is repeated. We note that implementation requirements dictate that the algorithm use a sampled Fourier transform. Moreover in order to be able to impose a support constraint, we need to oversample the Fourier transform magnitude. In our examples, if the support is enclosed in an N by N "box" we use a $2N$ by $2N$ discrete Fourier transform of the appropriately zero-padded image.

There is considerable disagreement between researchers in the field considering the conditions under which these iterative algorithms converge to the correct reconstruction. It can be shown [19] that the GSF algorithm decreases at each iteration the mean squared error between the estimate and true Fourier transform magnitude. Thus, the algorithm must converge to a fixed point of the iteration. This convergence does not imply, however, that the error will decrease to zero.

Fienup [18,19] has reported on the wide applicability of the iterative algorithms, especially the HIO algorithm. On the other hand, Hayes [2] has found that the GSF algorithm was not successful in reconstructing several original images. In a response, Fienup [24] has noted that the positivity constraint, which is used in the GSF-P and HIO algorithms, is important in ensuring convergence to the original image. However, Sanz et. al [21] has produced an example where the GSF-P algorithm did not converge to the original image even though the algorithm makes use of the positivity constraint.

In Appendix A we present a study of the convergence properties of the algorithms described above. A Monte Carlo study was performed on a large number of randomly generated 4 by 4 and 8 by 8 images to assess the convergence rate of each algorithm presented. The results of this study are summarized in Tables 3.1 and 3.2. These tables present the percent of trials which converged to a solution which had a normalized mean squared error (defined in Appendix A) of either less than 10^{-2} or less than 10^{-4}.

Success rate of convergence (percent)				
support type	rectangular		triangular	
cutoff	10^{-2}	10^{-4}	10^{-2}	10^{-4}
GSF	21	9	59	53
GSF-P	22	12	60	55
HIO	22	12	60	52

Table 3.1: Summary of algorithm convergence as a function of support and criterion for 4 by 4 image.

Success rate of convergence (percent)				
support type	rectangular		triangular	
cutoff	10^{-2}	10^{-4}	10^{-2}	10^{-4}
GSF	0	0	44	40
GSF-P	0	0	54	50
HIO	0	0	54	48

Table 3.2: Summary of algorithm convergence as a function of support and criterion for 8 by 8 image.

From Tables 3.1 and 3.2 we see that the iterative algorithms perform substantially different for the case of symmetric and non-symmetric supports. For the case of a non-symmetric support, the success rate was on the order of 40 to 60 percent while for symmetric supports, the algorithms succeeded in at most 20 percent of the trials. The reason for this difference may be that for non-symmetric supports, there is no rotation ambiguity in the reconstruction, i.e., one cannot rotate the image by 180 degrees and be able to "fit" the image in the same support. Moreover, while the convergence rate for the non-symmetric support remained about the same as the image size increased, the iterative algorithms degraded dramatically for 8 by 8 square images compared to 4 by 4 square images.

We must make the final observation that although there was a difference in the convergence rate among the three algorithms, the success rate among the three algorithms, as defined in Tables 3.1 and 3.2 is about the same; that is, the frequency with which the application of any of the algorithms considered resulted in a good reconstruction of the original image was about the same.

There are several conclusions we can derive from the study presented above. For the family of symmetric signals considered, we can surmise that the algorithms considered converged to the correct reconstruction in at most 20 percent, depending on the reconstructed image fidelity cutoff used and the image size. As the image size increased the success rate decreased. In the case where a nonsymmetric region of support was known, the iterative algorithm performed substantially better, achieving a final estimate close to the true signal in about 50 to 60 percent of the trials, the success rate staying constant as the image size was increased. The use of positivity information did not aid the convergence rate substantially.

Bates Algorithm

Bates [25,26,27] has developed an algorithm for the two-dimensional reconstruction which tries to estimate the phase of the Fourier transform directly from samples of the Fourier transform magnitude. For simplicity, consider a two-dimensional signal $x[m,n]$ which has a region of support $[0,N]^2$. Consider the samples of the Fourier transform at frequencies $u = 2\pi k/(N+1)$, $v = 2\pi l/(N+1)$,

$$X_{k,l} = X(e^{j\frac{2\pi k}{N+1}}, e^{j\frac{2\pi l}{N+1}}) \tag{3.17}$$

Since $X(e^{ju}, e^{jv})$ is the Fourier transform of an $[0,N]^2$ signal, it must satisfy the following "interpolation" formula,

$$X(e^{ju}, e^{jv}) = e^{jN\frac{u+v}{2}} \sum_{k=0}^{N} \sum_{l=0}^{N} X_{k,l}\, sinc(\frac{2\pi k}{N+1} - u)\, sinc(\frac{2\pi l}{N+1} - v) \tag{3.18}$$

where

$$sinc(x) = \frac{\sin\left(\frac{(N+1)x}{2}\right)}{\sin\left(\frac{x}{2}\right)} \tag{3.19}$$

Note that for $u = 2\pi k/(N+1)$, or $v = 2\pi l/(N+1)$, (3.18) can be simplified to

$$\begin{aligned} X(e^{ju}, e^{j2\pi l/(N+1)}) &= e^{jN\frac{u+2j\pi l/(N+1)}{2}} \textstyle\sum_{k=0}^{N} X_{k,l}\, sinc(\frac{2\pi k}{N+1} - u) \\ X(e^{j2\pi k/(N+1)}, e^{jv}) &= e^{jN\frac{2j\pi l/(N+1)+v}{2}} \textstyle\sum_{l=0}^{N} X_{k,l}\, sinc(\frac{2\pi l}{N+1} - v) \end{aligned} \tag{3.20}$$

The basic assumption in the Bates algorithm is that the sidelobes of the $sinc()$ in (3.19) function can be ignored. Using this assumption, (3.20) is approximated for "in-between" samples $X_{k+\frac{1}{2},l}$ or $X_{k,l+\frac{1}{2}}$ by

$$\begin{aligned} X_{k+\frac{1}{2},l} &\approx e^{j2\pi N\frac{k+l+1/2}{2(N+1)}}(\delta X_{m,n} + \delta X_{n+1,m}) \\ X_{k,l+\frac{1}{2}} &\approx e^{j2\pi N\frac{k+l+1/2}{2(N+1)}}(\delta X_{m,n} + \delta X_{m,n+1}) \end{aligned} \tag{3.21}$$

where δ is a constant which is "tuned" to the specific signal.

The algorithm proceeds in a sequential manner. Let us suppose that we already know the phase of $X_{k,l}$ and want to find the phase of $X_{k+1,l}$. By assumption, we know the magnitude of all the terms in (3.21), therefore, we can use the "law of cosines" to find the cosine of the difference in phase between $X_{k,l}$ and $X_{k+1,l}$,

$$\cos(arg\{X_{k+1,l}\} - arg\{X_{k,l}\}) \approx \frac{|X_{k+\frac{1}{2},l}|^2 - \delta^2|X_{k+1,l}|^2 + \delta^2|X_{k,l}|^2}{2|X_{k,l}||X_{k+1,l}|} \tag{3.22}$$

Similarly we can find a formula for $X_{k,l+1}$.

$$\cos(arg\{X_{k,l+1}\} - arg\{X_{k,l}\}) \approx \frac{|X_{k,l+\frac{1}{2}}|^2 - \delta^2|X_{k,l+1}|^2 + \delta^2|X_{k,l}|^2}{2|X_{k,l}||X_{k,l+1}|} \tag{3.23}$$

Note that (3.22) does not specify the phase of $X_{k+1,l}$ uniquely; there are actually two phase differences (negative of each other) which satisfy (3.22).

The algorithm begins by setting the phase of $X_{0,0}$ to zero. As Bates points out, this is not a restriction to the algorithm since all signals which differ by a constant phase have the same Fourier transform magnitude. The algorithm then proceeds to calculate the two possible approximations to the phase of $X_{1,0}$ via (3.22). One of these phase approximations is picked arbitrarily. Again this is not really a restriction since the only effect is to possibly rotate the resulting reconstruction by 180 degrees. The two possible values for the phase of $X_{0,1}$ are then calculated via (3.23). At this point both estimates for the phase of $X_{0,1}$ must be kept since all the degrees of freedom have been used in setting the phase of $X_{0,0}$ and $X_{1,0}$. The algorithm then proceeds by using both possible values for the phase of $X_{0,1}$ and (3.22) to calculate *four* possible values for the phase of $X_{1,1}$.

At this point the phase of $X_{0,0}$ and $X_{0,1}$ is known approximately, there are two possible phase values for $X_{1,0}$ and four possible phase values for $X_{1,1}$. The next step is to remove the ambiguity in the phase of $X_{1,1}$ by re-calculating the phase of $X_{1,1}$ in an independent manner, namely via the knowledge of the phase of $X_{1,0}$. Since the phase of $X_{1,0}$ has been uniquely specified, we can use (3.22) to calculate only two possible values of the phase of $X_{1,1}$. Bates maintains that the two sets of possible values for the phase of $X_{1,1}$ derived via the two different paths

$$X_{0,0} \rightarrow X_{1,0} \rightarrow X_{1,1} \tag{3.24}$$

and

$$X_{0,0} \rightarrow X_{0,1} \rightarrow X_{1,1} \tag{3.25}$$

will, in general, have only one value in common, or more precisely only one pair which almost matches. Using this value, the phase of $X_{1,1}$ is thus uniquely specified. Tracing our steps back, we can then get a unique phase estimate for $X_{0,1}$. The algorithm then proceeds to calculate other values of phase by working sequentially away from $X_{0,0}$.

Obviously, this algorithm makes many assumptions which make it a questionable solution to the phase retrieval problem although the authors have had some success in implementation [26,27]. First of all, the whole algorithm is predicated on the linear assumption made in (3.20); this is rarely valid. Second, as the authors suggest, this assumption will cause the two sets of phase values for $X_{1,1}$ not to have *any* points in common, necessitating the use of a "closest" criterion. As soon as one error is made in choosing a phase value, all later phase estimates will be incorrect. Even if no error is made in choosing the "closest" match, higher frequency phase estimates will be increasingly poor.

Because the phase corresponding to higher frequency terms get progressively poorer, we expect that the algorithm would be able to extract with some accuracy the lower frequency content of the image. In fact, this is the observed phenomenon; the reconstruction seems to be a severely blurred version of the original [23]. More recently, some effort has been made in using the Bates algorithm to generate an initial estimate for the Fienup algorithm. For the case where the image has mostly low frequency content, the Bates algorithm gives an adequate initial estimate for the Fienup algorithm [28].

Canterakis Algorithm

Recently, a third algorithm has been proposed by Canterakis [29]. This algorithm differs from the algorithms presented so far in that it is guaranteed to yield an exact solution to the phase retrieval problem. The aigorithm can be described as a conversion of the two-dimensional reconstruction problem to a one-dimensional reconstruction problem. Thus, it is important to keep in mind the discussion of Section 3.1 in evaluating this algorithm. For simplicity, suppose that the support of the unknown signal $x[m,n]$ is $[0,N]^2$. Consider now the one-dimensional signal $\hat{x}[n]$ given by

$$\hat{x}[m+(2N+1)n] = x[m,n] \quad \text{for } 0 \le m,n \le N \tag{3.26}$$

$$= 0 \quad \text{elsewhere} \tag{3.27}$$

This can be viewed as attaching N zeros to each row of $x[m,n]$ and concatenating the rows. It is easily shown that $\hat{r}[n]$, the autocorrelation function of $\hat{x}[n]$, satisfies

$$\hat{r}[m+(2N+1)n] = r[m,n] \tag{3.28}$$

where $r[m,n]$ is the autocorrelation function of $x[m,n]$. Note moreover that (3.28) is a reversible transformation; $\hat{r}[n]$ is generated from $r[m,n]$ by concatenating the rows of $r[m,n]$. The Canterakis algorithm consists of first calculating $r[m,n]$ from the known Fourier transform magnitude via the relationship of (2.13). Then $\hat{r}[n]$ is formed and *all* solutions to the one-dimensional reconstruction problem are calculated. Since $\hat{x}[n]$ is an $2N^2+2N+1$ point signal, there are about 2^{N^2} solutions to the one-dimensional problem posed, taking into account the fact that $\hat{x}[n]$ is assumed to be real. Each candidate $\hat{x}[n]$ is then checked to see if it can correspond to a $x[m,n]$; namely, whether it consists of the alternation of non-zero and zero samples consistent with a "transformed" two-dimensional signal. For each successful $\hat{x}[n]$, the corresponding $x[m,n]$ is formed, resulting in a solution to the two-dimensional reconstruction problem.

It is clear, as the author pointed out, that this algorithm requires exponentially increasing computations as the signal size increases. Because there are on the order of 2^{N^2} solutions to the associated one-dimensional problem, the computational load increases rapidly as N increases. Clearly this algorithm cannot begin to be applied to signals of reasonable size. We should note, however, that the Canterakis algorithm does provide an interesting perspective on reconstruction from magnitude. One point is that the phase retrieval problem is posed as that of extracting $x[m,n]$ from its autocorrelation function instead of extracting $x[m,n]$ from the Fourier transform magnitude directly. The second interesting point is that the Canterakis method, as we will see later, is an implicit attempt at factoring the polynomial associated with the autocorrelation function of $x[m,n]$. This is, in fact, the specific approach we will pursue in Chapters 4 and 5.

Deighton, Scivier and Fiddy Algorithm

Recently Deighton, Scivier and Fiddy [30] have developed an algorithm reminiscent of the Bates algorithm. This algorithm however does not make any approximations, and therefore, like the Canterakis algorithm, yields an exact reconstruction.

Suppose that, as in the discussion of the Bates algorithm, one seeks to calculate the phase difference between the Fourier transform samples $X_{k,l}$ and $X_{k+1,l}$. Let $x_l[n]$ be defined by

$$x_l[n] = \sum_{m=0}^{N} x[m,n] e^{-j\frac{2\pi l m}{N+1}} \tag{3.29}$$

where again, $x[m,n]$ has support $[0,N]^2$, and let the Fourier transform of $x_l[n]$ be denoted by $X_l(e^{jv})$. Using $x_l[n]$ above, we can express $X_{k,l}$ and $X_{k+1,l}$ as

$$X_{k,l} = X_l(e^{\frac{2\pi k}{N+1}}) \tag{3.30}$$

$$X_{k+1,l} = X_l(e^{\frac{2\pi(k+1)}{N+1}}) \tag{3.31}$$

Thus the Fourier transform of $x_l[n]$ is the strip in the (u, v) plane along $v = (2\pi l)/(N+1)$. Since $|X(e^{ju}, e^{jv})|$ is known for all (u, v), the Fourier transform magnitude of $x_l[n]$ is also known for all frequencies. The signal $x_l[n]$ has support $[0, N]$ and the results on solving the one-dimensional phase retrieval problem can be used to generate a set of at most 2^N signals of which one *must* be $x_l[n]$. By calculating the Fourier transform phase of each of these possible signals, the possible phase difference between $X_{k,l}$ and $X_{k+1,l}$ is thus restricted to be one of at most 2^N different possibilities. From the same process, 2^N different possible values for the phase difference between $X_{k,l+1}$ and $X_{k+1,l+1}$ can be generated. Combining the two results above, we have 2^{2N} possible phase differences between $X_{k,l}$ and $X_{k+1,l+1}$. By using the $X_{k,l} \rightarrow X_{k,l+1} \rightarrow X_{k,l}$ path, we get another 2^{2N} possible phase differences. Comparing the two lists, we expect that only two phases will be in both lists, the true phase and the phase of the reversed image. Although no proof has been presented, experimentally it has been found that only those two phase values will be in common in both lists.

The Deighton algorithm proceeds as follows: first, the phase of $X_{0,0}$ is arbitrarily set to zero. Via the phase matching process just described the phase of $X_{0,1}$, $X_{1,0}$ and $X_{1,1}$ are calculated. Moreover, the phase of $X_{0,k}$, $X_{1,k}$ and $X_{k,1}$ can be calculated for all k. Suppose that the phase of $X_{k,2}$ is now desired for $k \geq 2$. Using the same procedure, all 2^N solutions for $x_2[n]$ are calculated. The one solution which agrees with the previously

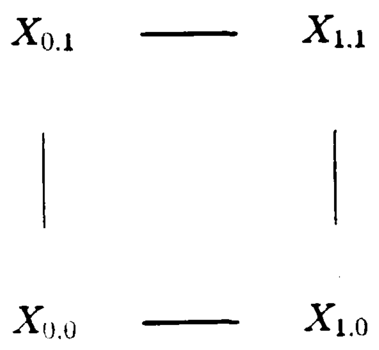


Figure 3.1: Phase differences to be used in Deighton algorithm.

calculated phase difference between $X_{0,2}$ and $X_{1,2}$ is then used to calculate the phase of $X_{k,2}$ for all other values of k. This way, the rest of the phase values of $X_{k,l}$ can be calculated.

The most computationally demanding part of the Deighton algorithm is the initial matching of phases. Figure 3.1 displays the number of possible phase differences among the four Fourier components $X_{0,0}$, $X_{0,1}$, $X_{1,0}$ and $X_{1,1}$. The characterization of the number of phase calculations needed turns out to involve a tradeoff between computation and storage. Consider first trying to find a match for the phase difference between $X_{0,0}$ and $X_{1,1}$ in the manner just discussed. Then, the algorithm needs to compare two lists with $2^{N-1+\lfloor N/2 \rfloor}$ elements each. Thus the lists grow at a rate of about $2^{3N/2}$. Such a list becomes too large to be accommodated in main memory (6 Mbytes of available memory) for images larger than 12 by 12. Since we will be dealing with larger images, it is imperative that the memory requirements of the Deighton algorithm be reduced for comparison purposes. This can be done by finding a match for the phase difference between $X_{1,0}$ and $X_{0,1}$ instead of $X_{0,0}$ and $X_{1,1}$. This allows us to use a list of size approximately $2^{2\lfloor N/2 \rfloor}$ so we could consider images of size up to about 18×18. This storage reduction comes at a price however, since now the number of comparisons required is about 2^{2N} instead of $2^{3N/2}$.

To the author's knowledge, of all the algorithms so far published for phase retrieval which is guaranteed to yield the correct solution, the Deighton algorithm is the most computationally efficient. In order to be able to compare the performance of our algorithm with the Deighton algorithm, the Deighton algorithm was implemented using the second method described here. In Figure 3.2 we show the number of CPU seconds on a Vax-750 required to reconstruct an image with support $[0, N-1]^2$ as a function of N.

The severe storage or computational requirements of Deighton algorithm, especially its quick growth with image size, is directly related to the combinational approach used of trying all possible solutions. Since the resources required grow exponentially, one quickly runs into a storage or computational barrier.

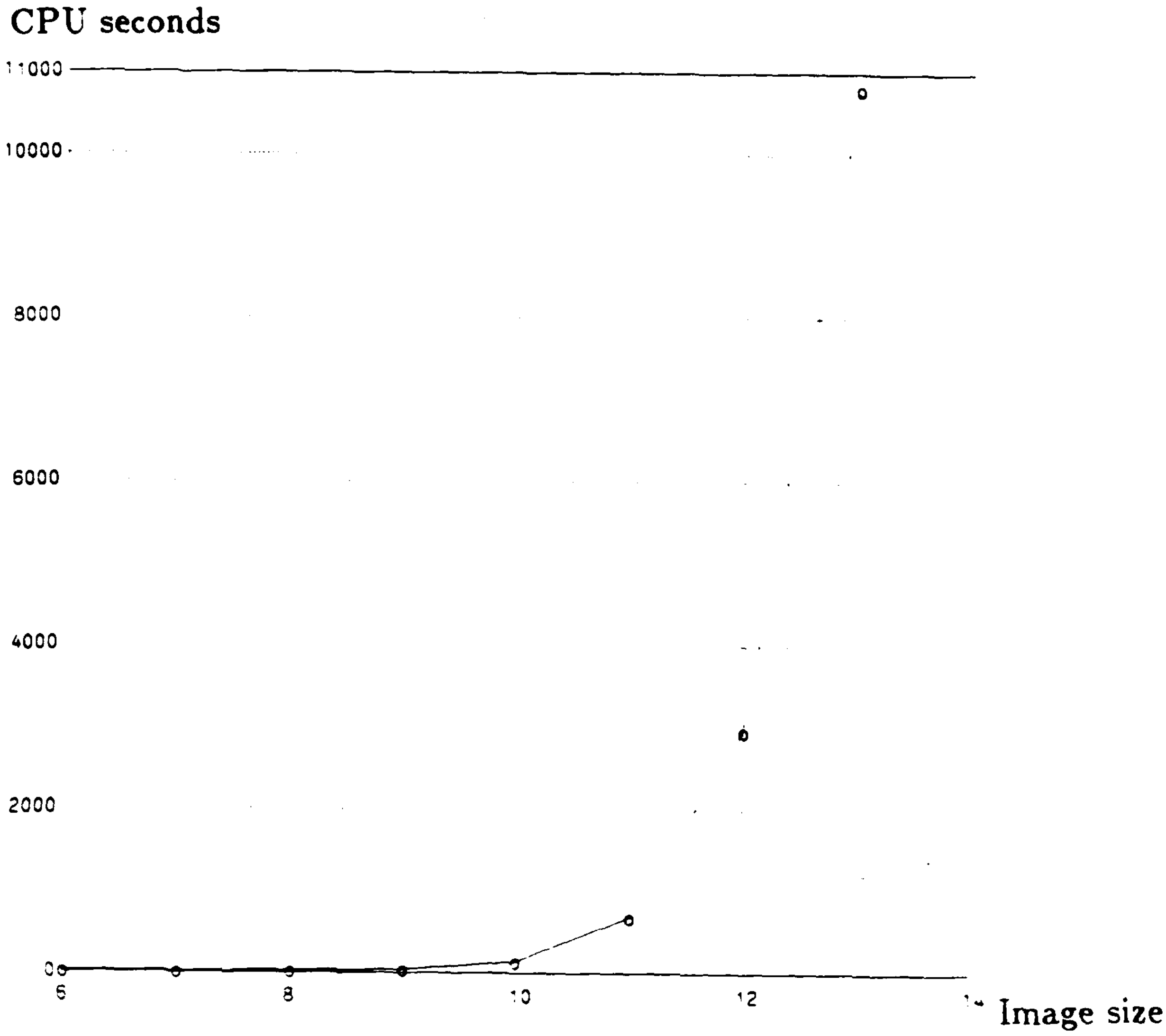


Figure 3.2: Time required for reconstruction of an N by N image as a function of N for Deighton algorithm.

Chapter 4

Phase Retrieval via Bivariate Polynomial Factorization

We have so far discussed several approaches to reconstruction from Fourier transform magnitude. The algorithms of Fienup and Bates are computationally efficient and easily applied. However, there is no assurance that using these algorithms will yield a solution to the phase retrieval problem. We have also considered two algorithms which provide a closed form solution to the reconstruction problem. The first algorithm studied, developed by Canterakis, rapidly becomes prohibitively expensive as the image size is increased, or equivalently, as the image resolution is increased. The second closed form algorithm, studied by Deighton et al., is more attractive computationally. However, the required computational and storage load also increase very rapidly as a function of image resolution.

As described in the last chapter, the results of Bruck and Sodin, Hayes, and Sanz point to the intimate relationship between conditions for uniqueness of reconstruction from Fourier transform magnitude and the factorability of polynomials. Moreover, we found that a general procedure for solving the one-dimensional phase retrieval problem consists of factoring the polynomial associated with the autocorrelation function. The approach we introduce in this chapter in order to generate a new algorithm for phase retrieval is to view the relationship between phase retrieval and polynomial factorization as also pointing to a method for solving the two-dimensional phase retrieval problem. We will also find that this approach, although explicitly expressed here, is the implicit basis for the Canterakis algorithm described earlier.

If the coefficients of the unknown signal are known to be rational numbers, then the theory of integer polynomial factorization becomes applicable. This is the approach considered by Berenyi, Deighton, and Fiddy [31]. There are several difficulties with the rational coefficient assumption. First of all, an infinitesimal amount of noise will render the algorithm useless. Second of all, algorithms for factoring polynomials are very computationally intensive. In Berenyi's paper, the largest image considered was a 6 by 6 image.

One novel idea of this thesis is to consider algorithms for factoring polynomials over the reals or complex numbers. This makes the problem more robust to noise and also applicable to the case where the signal to be reconstructed has complex values. The final result, described in the next chapter, is a new factorization algorithm which leads to a closed form solution to the phase retrieval problem which is much more computationally efficient than the Canterakis or Deighton algorithms. As a result, the reconstruction of images of sizes up to 25 by 25 becomes feasible.

In this chapter we first pose the two-dimensional phase retrieval problem in analogy to our previous one-dimensional presentation of Section 3.1 and emphasize how factorization of the resulting bivariate polynomial leads to a solution of the phase retrieval problem. This leads to the topic of this chapter, which is the discussion of several known algorithms for factoring polynomials in two variables over the complex numbers. In order to do this we first introduce some results from algebraic function theory. We follow this by a discussion of the algorithms of Kronecker and Kaltofen for factoring bivariate polynomials. In the next chapter we introduce the main result of this thesis, which is a new algorithm for factoring polynomials in two variables over the reals or complex numbers and its application to the phase retrieval probiem.

4.1 Phase Retrieval as Polynomial Factorization

We have already found that the polynomial associated with the autocorrelation function is always factorable into the product of $p_x(w, z)$, the polynomial associated with the unknown image $x[m, n]$ and the mirror polynomial $\tilde{p}_x(w, z)$,

$$p_r(w, z) = p_x(w, z)\tilde{p}_x(w, z) \tag{4.1}$$

Moreover, if $p_x(w, z)$ is not factorable, then $\tilde{p}_x(w, z)$ is not factorable either, and thus (4.1) is the only non-trivial product decomposition of $p_r(w, z)$. Thus, if $p_x(w, z)$ is not factorable, we can generate $x[m, n]$ from $r[m, n]$ by factoring $p_r(w, z)$ into its irreducible components. This observation is the basis for the result by Bruck and Sodin, and Hayes. From (4.1) we see the possible utility of algorithms for factoring bivariate polynomials; namely, if we can factor $p_r(w, z)$ then we will retrieve $p_x(w, z)$ or $\tilde{p}_x(w, z)$, and from $p_x(w, z)$ or $\tilde{p}_x(w, z)$ we can extract $x[m, n]$ or $x[-m, -n]$.

4.2 Zeros of Bivariate Polynomials

4.2.1 Motivation

In the search for factorization mechanisms for polynomials over the complex numbers, zeros of such polynomials play a crucial role. The importance of zeros is essentially due to the property that the set of zeros of a polynomial is equal to the union of the set of zeros of its factors. For example, the set of zeros z_i of a univariate polynomial is equal to the union of the zeros of each of its irreducible factors, namely each factor $(z - z_i)$. The same situation holds for polynomials in two or more variables. Unlike univariate polynomials, however, the zeros of polynomials in two variables cannot be broken up into independent pieces consisting of an isolated zero each. If the polynomial is reducible, its set of zeros can be broken up into a union of zeros corresponding to its irreducible factors. The main difficulty is that performing such a dissection is a much more intricate matter in the bivariate case than the univariate case. For this reason the discussion below is concerned with developing some of the properties of zeros of polynomials in two variables.

4.2.2 Algebraic Functions

We already discussed that there are complex numbers z_i associated with a polynomial $p(z)$, called the zeros of $p(z)$, such that $p(z_i) = 0$. We also noted that the number of such zeros is finite. In fact, there are at most $deg(p)$ zeros of $p(z)$.

Bivariate polynomials have a much more interesting set of zeros. In fact a large branch of mathematics is concerned with the study of the zeros of polynomials in two variables. A very readable discussion of this topic is contained in [10]. Most of the development below is based on this reference. In order not to stray too far from our main subject, some of the relevant results are rederived here in a restricted yet adequate form, but without the necessity of introducing certain peripheral subjects.

Recall that a zero of the polynomial of degree (M, N)

$$p(w,z) = \sum_{m=0}^{M} \sum_{n=0}^{N} a_{m,n} w^m z^n \tag{4.2}$$

is a *pair* of complex numbers (w_i, z_i) such that $p(w_i, z_i) = 0$. However, unlike the one-dimensional case, the number of such zeros is not finite. In fact, as shown below, for any complex number z we can always find a w such that (w, z) is a zero of $p(w, z)$.

Consider $p(w, z)$ written as a polynomial in w with coefficients which are polynomials in z,

$$p(w,z) = p_0(z) + p_1(z)w + \cdots + p_M(z)w^M \tag{4.3}$$

where

$$p_j(z) = \sum_{k=0}^{N} p_{j,k} z^k \tag{4.4}$$

For each value of z, the equation $p(w, z) = 0$ defines a function $w(z)$ implicitly by

$$p(w(z), z) = 0 \tag{4.5}$$

This function is called an *algebraic function* because for all z, $w(z)$ satisfies the algebraic or polynomial equation (4.5). This function $w(z)$ will be multi-valued; specifically, it will normally consist of the M distinct finite roots of (4.3)

$$w_1(z), w_2(z), \cdots, w_M(z) \tag{4.6}$$

These are called the *branches* of the algebraic function.

It is straightforward to generate these roots; namely, each $p_j(z)$ in (4.3) is evaluated at z, reducing (4.3) to a polynomial in w with numerical coefficients. The roots of this one-dimensional polynomial is exactly what we mean by (4.6).

A point $z = a$ in the complex z-plane where there are M distinct finite zeros of $p(w, a)$, i.e., where all the branches of $w(z)$ are finite and do not intersect, is called an *ordinary point* of the algebraic function $w(z)$. A point where one of the branches becomes infinite or two branches coalesce into a single root is called a *singular point*. Therefore, each point in the z-plane is either an ordinary or singular point of $w(z)$. The case where two branches coalesce into a single root is also called a *branch point*. Thus, a branch point is a special case of a singular point. Singular points which occur for finite values of z will be called *finite singular points*.

Example: Consider the polynomial,

$$p(w,z) = 1 - z + z^2 + (-1 + z - z^2)w + (1 - z)w^2 \qquad (4.7)$$

A plot of $w(z)$ as a function of z between .5 and 1.5 is shown in Figure 4.1. In this case there are two branches since the degree of $p(w,z)$ in w is 2. Thus for almost all z, there are two values of w such that $p(w,z) = 0$. One singular point of $w(z)$ occurs at $z = 1$, where it can be seen from the figure that one branch becomes infinite at this value of z. A branch point of $w(z)$ occurs at approximately $z = 7.913$, $w = 2.01647$ where the two branches join. For real z less than the location of the branch point, the branches of $w(z)$ are complex conjugate pairs. Figure 4.1 in this case shows, in dotted lines, the real part of w plus or minus the imaginary part of w.

The set of singular points can be characterized by the following two theorems,

Theorem 4.1 *Let $p(w,z)$ be a polynomial of degree (M,N), expressed as in (4.3), then all $w_i(z)$ evaluated at z_0 must be finite unless $p_M(z_0) = 0$, where $p_M(z)$ is defined in (4.3). The number of such z_0 is at most N.*

Proof: We first need to invoke a result by Cauchy [32, p.123],

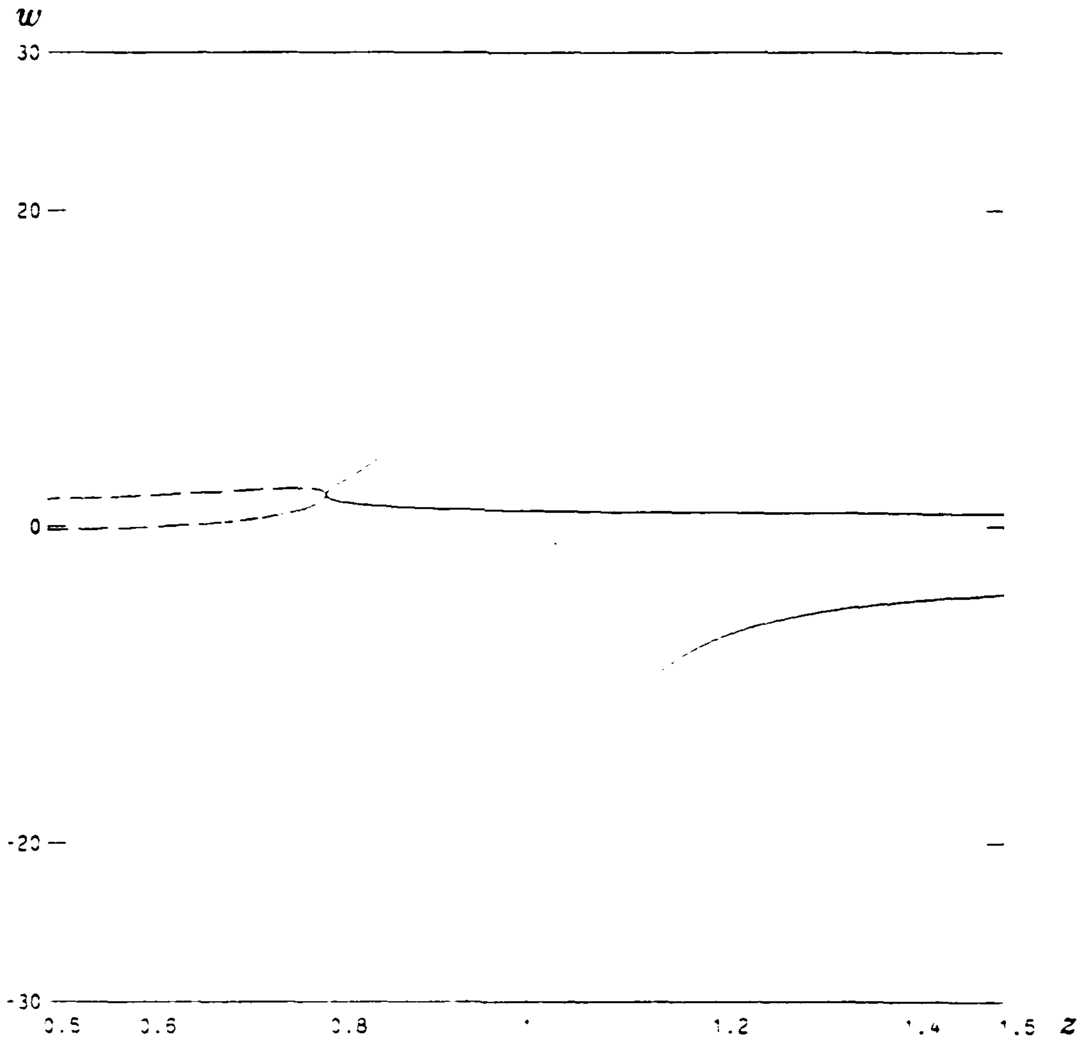


Figure 4.1: Zeros of polynomial in example for real z. For z less than .79, the zeros become complex.

Lemma 4.1 *All the zeros of*

$$f(z) = a_0 + a_1 z + a_2 z^2 + \ldots + a_n z^n \tag{4.8}$$

where $a_n \neq 0$ *lie in the circle*

$$|z| < 1 + \max |a_k / a_n| \textit{ for } k = 0, 1, \cdots, n-1 \tag{4.9}$$

Applying the lemma to (4.3), we find that if (w, z) is a zero of $p(w, z)$, the following inequality must hold,

$$|w| < 1 + max|p_k(z)/p_M(z)| \text{ for } k = 0, 1, \cdots, M-1 \tag{4.10}$$

If $w(z)$ becomes unbounded then we need that $p_M(z)$ tend to zero. Since $p_M(z)$ has at most N zeros, the theorem is proven. □

The second theorem concerns itself with branch points, the second type of singularity considered.

Theorem 4.2 *The complex number* z_0 *is a branch point, i.e., two roots of* $p(w, z)$, $w_i(z)$ *and* $w_j(z)$, *are equal at* z_0 *where* $w_i(z_0) = w_j(z_0) = w_0$, *if and only if the partial derivative of* $p(w, z)$ *with respect to* w *is zero when evaluated at* (w_0, z_0).

Proof: We will denote the partial derivative of $p(w, z)$ with respect to w as $p_w(w, z)$. A well known result from the study of polynomial functions of a single variable states that a polynomial $p(z)$ has a multiple zero at z_0, if and only if z_0 is also a zero of the derivative of $p(z)$ [32]. This result can be applied trivially to the case of a bivariate polynomial. If (4.3) when evaluated at z_0 has a multiple root at w_0, then from the discussion above, the polynomial

$$p_1(z_0) + 2p_2(z_0)w_0 + \cdots + Mp_n(z_0)w_0^{M-1} \tag{4.11}$$

must also be zero. However, this polynomial is just $p_{ir}(w_0, z_0)$. □

Example: Consider again the polynomial of the previous example. In this case we can express the branches of $w(z)$ in explicit form.

$$w_1(z) = \frac{-1 + z - z^2 + \sqrt{-3 + 6z - 5z^2 + 2z^3 + z^4}}{2 - 2z} \tag{4.12}$$

$$w_2(z) = \frac{-1 + z - z^2 - \sqrt{-3 + 6z - 5z^2 + 2z^3 + z^4}}{2 - 2z} \tag{4.13}$$

According to Theorem 4.1, at each zero of $p_2(z) = 1 - z$, at least one branch of $w(z)$ must go to infinity. Here the zero occurs at $z = 1$, and we see from (4.13) and Figure 4.1 that $w_2(z)$ becomes infinite as expected.

The polynomial $p_{ir}(w, z)$ for our example is given by

$$p_{ir}(w, z) = (-1 + z - z^2) + 2(1 - z)w \tag{4.14}$$

Calculating $p_{ir}(w, z)$ at the branch point $z = .7913$, $w = 2.01647$, we find that $p_{ir}(w, z)$ is approximately zero, as predicted by Theorem 4.2.

From (4.12), we see that all branch points must satisfy the polynomial equation

$$0 = -3 + 6z - 5z^2 + 2z^3 + z^4 \tag{4.15}$$

Thus for this example, at least, $p(w, z)$ can only have at most 4 branch points. We will later characterize more generally, the number of branch points a given polynomial may have.

Near an ordinary point, the functions $w_i(z)$ can be associated with analytic functions, i.e., functions possessing a Taylor series expansion.

Theorem 4.3 *Near an ordinary point $z = a$, the M values of the algebraic function $w(z)$ are defined by M convergent series*

$$w = w_{i0} + w_{i1}(z - a) + w_{i2}(z - a)^2 + \cdots \quad (i = 1, \cdots, M) \tag{4.16}$$

where the numbers w_{i0} are the M distinct roots of $p(w, a) = 0$.

Proof: This theorem is a straightforward application of the Implicit Function Theorem [33, p.109],

Lemma 4.2 *Let $F(w, z)$ be a function of two complex variables which is analytic in a neighborhood $|z - z_0| < r$, $|w - w_0| < \rho$ of the point (w_0, z_0), and suppose that*

$$F(w_0, z_0) = 0, \quad \frac{\partial F(w_0, z_0)}{\partial w_0} \neq 0 \tag{4.17}$$

Then there are neighborhoods $\mathcal{N}(z_0)$, $\mathcal{N}(w_0)$ such that the equation $F(w,z) = 0$ has a unique root $w = w(z)$ in $\mathcal{N}(w_0)$ for any given $z \in \mathcal{N}(z_0)$. Moreover, the function $w = w(z)$ is single-valued and analytic on $\mathcal{N}(z_0)$ and satisfies the condition $w(z_0) = w_0$.

To use this theorem we first note that $p(w,z)$ is an analytic function of w and z for all finite w, z. Furthermore, from Theorem 4.2, if (w_{i0}, z_0) is an ordinary point, then $p_w(w_0, z_0) \neq 0$; thus, the condition of (4.17) is satisfied. We conclude then that a locally analytic *unique* function $w(z)$ exists such that $p(w_i(z), z) = 0$ in a neighborhood of z_0 and such that $w_i(z_0) = w_{i0}$. Since there are M solutions to the equation $p(w, z_0) = 0$, there are M such functions. □

A corollary of this theorem is that each $w_i(z)$ is differentiable. It is straightforward to calculate the derivative of each $w_i(z)$. From the relationship,

$$p(w_i(z), z) = 0 \tag{4.18}$$

we get that

$$\frac{dp(w_i(z), z)}{dz} = 0 \tag{4.19}$$

Using the chain rule,

$$p_z(w_i(z), z) + p_w(w_i(z), z)\frac{dw_i(z)}{dz} = 0 \tag{4.20}$$

or

$$\frac{dw_i(z)}{dz} = -\frac{p_z(w_i(z), z)}{p_w(w_i(z), z)} \tag{4.21}$$

The crucial importance of the study of $w(z)$ is that each branch of $w(z)$ can be associated with an irreducible factor of $p(w,z)$. This relationship is stated in the following theorem,

Theorem 4.4 *For each branch $w_i(z)$ of $w(z)$ given by (4.16) and each ordinary point z_0 there is an irreducible factor of $p(w,z)$, $p_i(w,z)$ such that*

$$p_i(w_i(z),z) = 0 \textit{ for all } z \textit{ in a neighborhood of } z_0 \tag{4.22}$$

Moreover, this is the only irreducible polynomial which satisfies the equation above.

Proof: Recall that any polynomial can be expressed as an essentially unique product of irreducible polynomials,

$$p(w,z) = p_1(w,z)p_2(w,z)\cdots p_k(w,z) \tag{4.23}$$

Let $w_{i0} = w_i(z_0)$. Since $p(w_{i0},z_0) = 0$, we have according to (4.23),

$$p(w_{i0},z_0) = p_1(w_{i0},z_0)p_2(w_{i0},z_0)\cdots p_k(w_{i0},z_0) = 0 \tag{4.24}$$

Thus at least one of the terms $p_i(w_{i0},z_0)$ must be zero. However, since z_0 is an ordinary point, only one of these terms can be zero. Without loss of generality, let $p_1(w_{i0},z_0) = 0$. Since z_0 is an ordinary point of the algebraic function of $p(w,z)$, it must also be an ordinary point of the algebraic function corresponding to $p_1(w,z)$. Thus from Theorem 4.3 we know that there is a locally analytic function $g(z)$ such that $p_1(g(z),z) = 0$ everywhere in a neighborhood of z_0 and $g(z_0) = w_{i0}$. However, if $p_1(g(z),z) = 0$ then $p(g(z),z) = 0$. Since according to Lemma 4.2, the function $w_i(z)$ is unique in a neighborhood of z_0, $g(z)$ and $w_i(z)$ must be the same in this neighborhood. Therefore, $p_1(w_i(z),z) = 0$ for all z in a neighborhood of z_0. That this is the only such irreducible polynomial follows from the fact that two irreducible polynomials can have only a finite number of common zeros. A theorem to this effect is introduced later in the discussion of Bezout's theorem. □

From the theorem above, we see that we can associate with each branch of $w(z)$ an irreducible factor of $p(w,z)$. The analogous procedure for univariate polynomials is the fact that we can associate with each zero z_i of $p(z)$, an irreducible factor of $p(z)$, namely $(z - z_i)$. The crucial difference is that we are not able to break up the zeros of $p(w,z)$ into isolated zeros; the best one can do is to break up the zeros of $p(w,z)$ into different continuous sets of zeros. The reader should be aware that each branch of $w(z)$ is *not* associated with a distinct factor of $p(w,z)$, that is, the fact that $w(z)$ has M branches does not imply that $p(w,z)$ has M irreducible factors. For example, we could have that both $w_1(z)$ and $w_2(z)$ correspond to $p_1(z)$.

Although we will have the opportunity to discuss algebraic functions further, the properties described above are sufficient to understand the rest of this chapter. In Chapter 5, we will take up the topic again, in order to develop some more results necessary for the establishment of our new phase retrieval algorithm.

4.3 Bivariate Polynomial Factorization

4.3.1 Kronecker's Algorithm

Although univariate polynomial factorization has a long and productive history, the factorization of bivariate polynomials has enjoyed relatively little progress. The first algorithm for factoring a polynomial in several variables is due to Kronecker in 1882 [34]. Of course, Kronecker's interest was in the mathematical problem of factoring polynomials in several problems, not phase retrieval. The basic idea is to convert the bivariate polynomial into a univariate polynomial, and then factors of the resulting univariate polynomial are associated with factors of the original bivariate polynomial.

Consider a general polynomial $p(w, z)$ of degree (M, N). We seek to find a polynomial $g(w, z)$ which is a factor of $p(w, z)$. The algorithm consists of four steps as delineated below:

1. **Compute degree bound** Obtain an integer $d > \max(M, N)$.

2. **Reduction of bivariate to univariate polynomial** Generate $\hat{p}(w)$ which is given by

   $$\hat{p}(w) = p(w, w^d) \tag{4.25}$$

   Note that because of the way we have picked d, this transformation between $\hat{p}(w)$ and $p(w, z)$ is reversible.

3. **Factorization of univariate polynomial** Factor $\hat{p}(w)$ into irreducible, i.e., linear factors,

   $$\hat{p}(w) = \hat{g}_1(w)\hat{g}_2(w)\cdots\hat{g}_s(w) \tag{4.26}$$

4. **Inverse reduction and trial division** From the factorization in step 3, generate all polynomials which divide $\hat{p}(w)$. Call a typical factor $\hat{g}(w)$. Convert $\hat{g}(w)$ into a bivariate polynomial $g(x, y)$ via the relation

   $$g(w, w^d) = \hat{g}(w) \tag{4.27}$$

   and check if $g(w, z)$ divides $p(w, z)$. If so, stop; if not, try a different factor If there is no appropriate $\hat{g}(w)$, then $p(w, z)$ is irreducible.

The relationship between the Canterakis algorithm and Kronecker's algorithm becomes apparent if we note that the polynomials associated with $\hat{r}[n]$ and $r[m, n]$ in the

Canterakis algorithm are related by

$$\hat{p}_r(w) = p_r(w, w^{2N+1}) \tag{4.28}$$

Thus, the Canterakis algorithm picks $d = 2N + 1$ as the parameter in Kronecker's algorithm. Since the degree of $p_r(w, z)$ is $(2N, 2N)$ for an N by N image, this is an appropriate value for d. Then, the algorithm proceeds to factor $p_r(w, z)$ via Kronecker's algorithm.

A severe problem with Kronecker's algorithm which precludes its general use is the tremendous work required in step 4 of the algorithm. The amount of computation increases exponentially as the size of the polynomial increases. Since the Canterakis algorithm is basically Kronecker's algorithm, it also suffers from this exponential growth.

4.3.2 Kaltofen's Algorithm

In 1982, Kaltofen [35], presented an algorithm for reducing a bivariate polynomial factorization problem to a univariate factorization problem which avoids the exponential growth experienced by Kronecker's algorithm. Again, this algorithm was developed solely in the context of the mathematical problem of polynomial factorization.

The first step in Kaltofen's algorithm is to find an ordinary point of the algebraic function $w(z)$. It can be supposed without loss of generality that $z_0 = 0$; otherwise, we just need to make a linear change of variables in z. Thus, it can be assumed that a w_0 such that $p(w_0, 0) = 0$ has been calculated and that $(w_0, 0)$ is an ordinary point.

The second step of the algorithm is based on developing a power series expansion of a branch of the algebraic function $w(z)$ around zero. That such an expansion exists is guaranteed by Theorem 4.3,

$$p(w_0 + w_1 z + w_2 z^2 + \cdots, z) = 0 \tag{4.29}$$

There are many ways of calculating the $\{w_i\}$ set. The simplest way conceptually is by matching powers of z, i.e. expanding (4.29) to a power series in z and then setting each coefficient in the power series to zero. This yields a manner of evaluating $\{w_i\}$ successively. There are much faster algorithms for finding the $\{w_i\}$ set which are based on an algebraic version of Newton's method [36,37]; however, it is really not pertinent to our discussion that we describe these methods here.

The fourth and last step of Kaltofen's algorithm is based on the fact that each branch of $w(z)$ is characteristic of an irreducible factor of $p(w,z)$ by virtue of Theorem 4.4, i.e., there is an irreducible polynomial $g(w,z)$ that divides $p(w,z)$ and, again for all z in a neighborhood of $z = 0$,

$$g(w_0 + w_1 z + w_2 z^2 + \cdots, z) = 0 \tag{4.30}$$

From the discussion concerning Theorem 4.4, we see that which of the irreducible factors of $p(w,z)$ is specified is determined by which irreducible factor of $p(w,z)$ satisfies $g(w_0, 0) = 0$. Picking a different solution w_0 may result in a different irreducible factor being identified.

Now that Taylor series defined by the $\{w_i\}$ set has been extracted and associated this series with one of the irreducible factors of $g(w,z)$, the question becomes how to extract the coefficients of $g(w,z)$ given the coefficient set $\{w_i\}$. The answer is to use (4.30) in "reverse", i.e., calculate the coefficients of $g(w,z)$ from the known $\{w_i\}$ by matching coefficients in z and setting them to zero. This yields a set of homogeneous linear equations in the unknown *coefficients* of $g(w, z)$. The algorithm tries all possible total degrees for $g(w, z)$ up to $totdeg(p)$ until it finds that the homogeneous linear equations have a non-trivial solution. In the reconstruction from Fourier transform magnitude problem of a signal with an irreducible associated polynomial, we would not have to try all possible sizes, since the support of the signal, and therefore the degree of the polynomial factor, is known.

Note that the Kaltofen algorithm does not have to go through the "combinatorial explosion" required by the Kronecker algorithm. Thus, it is a promising vehicle for factoring polynomials of reasonable size. From this discussion the Kaltofen algorithm seems like a good candidate to effect the factorization of the polynomial associated with the autocorrelation function as was done in the Canterakis algorithm, but without the enormous computational requirements.

Indeed, we have tried using this algorithm and have found that it successfully reconstructs signals with support up to $[0, 5]^2$. However, as is well known, Taylor series coefficients have to be calculated exceedingly accurately; otherwise, small errors in the coefficients lead to great changes in the function as the series is evaluated away from the origin. Similarly, the Taylor series coefficients in Kaltofen's algorithm have to be calculated very precisely, or else the equations to be solved do not match the actual solution. We have found that these round-off effects preclude the use of the algorithm for signals with support larger than $[0, 5]^2$. Another way of viewing the source of instability in the Kaltofen algorithm is that it essentially uses local information to extract the factors of a polynomial; namely, the value and derivatives of the locus of roots of the irreducible polynomial to be extracted.

An interesting aspect of this algorithm, however, is that it uses the zeros of $p(w,z)$ to isolate an irreducible factor of $p(w,z)$. Unfortunately, it uses this strategy in a numerically sensitive manner. An improvement of this method would be to find a way of specifying the global characteristics of a single branch of $p(w,z)$. It is hoped that by using such global information, a more attractive way of isolating one irreducible factor of $p(w,z)$ can be found. The development and analysis of one such strategy is the subject of the next chapter.

Chapter 5

New Closed Form Algorithm for Reconstruction from Fourier Transform Magnitude

We have hinted in the introduction that the approach we consider in this thesis is to explicitly view phase retrieval as a polynomial factorization problem and to search for an algorithm to factor polynomials efficiently and accurately. We have already noted that the Canterakis approach for reconstruction from Fourier transform magnitude is implicitly the use of Kronecker's algorithm to factor the polynomial associated with the signal autocorrelation function. However, we found that this algorithm is extremely time consuming for even small signals. An alternative is Kaltofen's algorithm for factoring polynomials. This algorithm, although not as computationally expensive as Kronecker's algorithm, suffers from severe numerical instability due to the use of information which is very sensitive to round-off effects.

In this chapter a new algorithm for factoring bivariate polynomials is described. Although the algorithm is applicable to the factorization of any bivariate polynomial, it is specifically developed as a vehicle for the phase retrieval problem. This algorithm does not increase in complexity as the region of support increases as quickly as the

Canterakis or Deighton algorithms do and is more numerically stable than applying the Kaltofen algorithm, allowing us to factor larger polynomials and thus reconstruct larger images from the Fourier transform magnitude.

The algorithm is based on analyzing the set of complex pairs (w, z) where the polynomial associated with the autocorrelation function is zero. Although the use of this set of zeros to factor $p_r(w, z)$ is original, the importance of this set of zeros to the problem of phase retrieval has long been recognized, beginning with the early work of Napier and Bates [38].

5.1 Background and Overview

Before developing the algorithm formally, it is helpful to sketch the general idea behind the method. Consider a reducible polynomial $p(w, z)$ with two irreducible factors $r(w, z)$ and $s(w, z)$; thus

$$p(w, z) = r(w, z)s(w, z) \tag{5.1}$$

If (w, z) is a zero of $s(w, z)$ then it must also be a zero of $p(w, z)$. However, if (w, z) is a zero of $p(w, z)$ then a priori one cannot tell whether this zero corresponds to $r(w, z)$ or $s(w, z)$. Of course, (w, z) *must* be a zero of at least one of them. Generating zeros of $p(w, z)$ is easy since $p(w, z)$ is known; the difficulty resides in assigning each calculated zero of $p(w, z)$ to the appropriate irreducible factor of $p(w, z)$. The main objective of the development discussed below is to generate such an assignment strategy, so that eventually we have a large number of zeros which are all guaranteed to correspond exclusively to either $r(w, z)$ or $s(w, z)$.

Suppose, then, that we have at our disposal a large number of zeros of, for example,

$s(w,z)$. Then, for each zero (w_k, z_k), the following equation must hold,

$$s(w_k, z_k) = 0 = s_{00} + s_{10}w_k + s_{01}z_k + s_{11}w_k z_k + \cdots \tag{5.2}$$

where s_{ij} are the coefficients of $s(w,z)$. This equation is *linear* in the unknown s_{ij}. By examining the null vectors of this set of equations we will be able to reconstruct the coefficients of $s(w,z)$ and thus $s(w,z)$ itself.

5.1.1 New Factorization Algorithm

An interpretation of $w(z)$, the algebraic function corresponding to $p(w,z)$, which will prove to be crucial in our later development, is to consider it as a multi-valued mapping of a path in the z-plane to several paths in the w-plane. A path is a complex-valued function of a real parameter t, where t varies from t_0 to t_1

$$z(t) = x(t) + jy(t) \tag{5.3}$$

such that $x(t)$ and $y(t)$ are continuous. Since we will be free to choose $z(t)$, it will be convenient to assume also that $x(t)$ and $y(t)$ are also piece-wise differentiable functions of t. The mechanism by which this mapping occurs is easy to visualize. Consider an ordinary point $z_0 = z(0)$ and all M solutions to the (univariate) polynomial equation $p(w, z_0) = 0$. We denote each of these solutions by $w_i(0)$ for $i = 1, \cdot, M$, Figure 5.1. As z_0 is changed in a continuous manner, thus generating $z(t)$, the corresponding roots of the equation $p(w, z(t))$ will also change in a continuous manner, resulting in the paths $w_i(t)$, Figure 5.2. This view of $w(z)$ mapping paths in the z-plane to paths in the w-plane is supported by the following theorem [10, p.25],

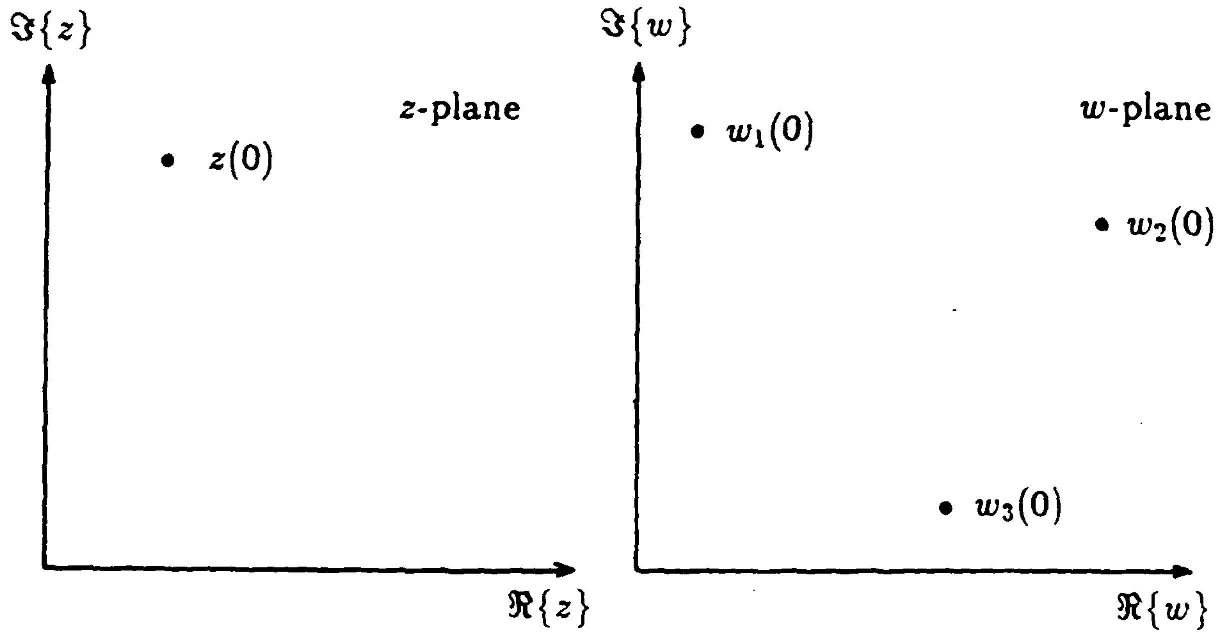


Figure 5.1: Points in the w-plane corresponding to a value of z.

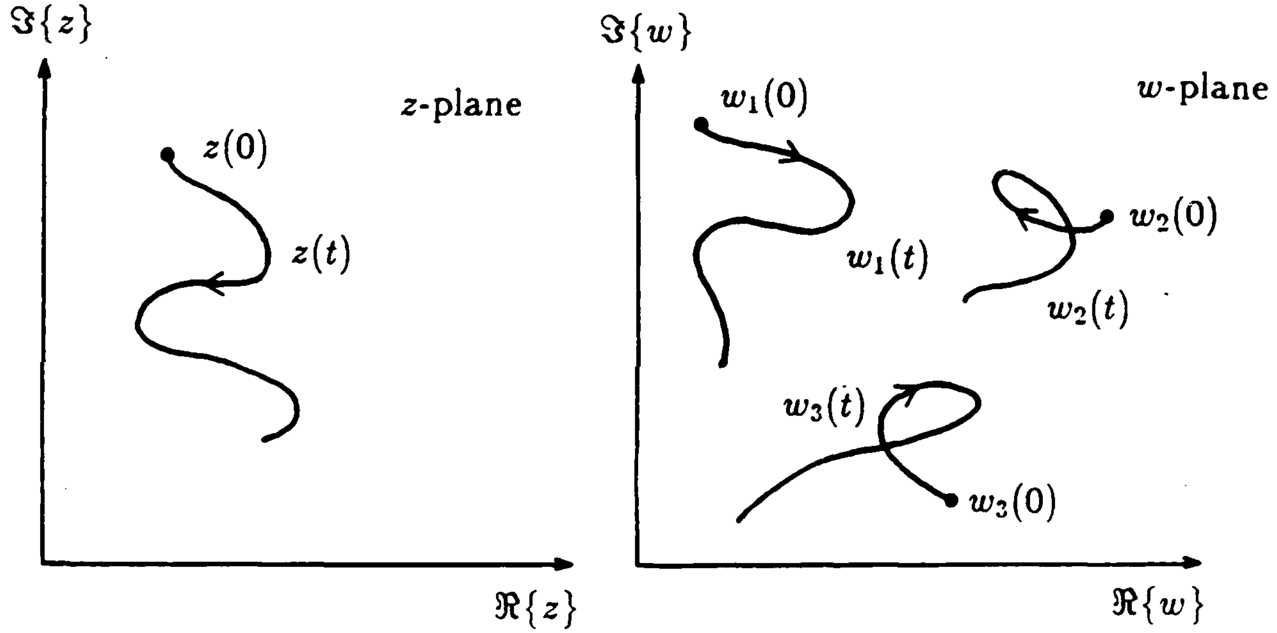


Figure 5.2: Paths in the w-plane which result from a path in the z-plane.

Theorem 5.1 *If $z = z(t)$ $(t_1 \leq t \leq t_2)$ is a path consisting entirely of ordinary points of an algebraic function $w(z)$, then the values of $w(z)$ along $z(t)$ form a set $w_k(t)$ $(t_1 \leq t \leq t_2;\ k = 1, \cdots, M)$ of M paths.*

In terms of the M branches of $w(z)$, the paths described in this theorem are given by,

$$w_k(t) = w_k(z(t)) \tag{5.4}$$

Thus, the trajectory of each of the paths is specified by each of the M branches of $w(z)$.

Example: As an illustration, consider a path in the z-plane given by $z(t) = e^{j2\pi t}, (0 \leq t \leq 1)$. Figure 5.3. The polynomial under study is $p(w, z) = w^2 - z$. We now evaluate the two paths $w_1(t)$ and $w_2(t)$ which form the image of $z(t)$ under $w(z)$. At each value of t, we have, $p(w_i(t), z(t)) = 0$, or $w_1(t) = e^{j\pi t}$, $w_2(t) = e^{-j\pi t}$. The resulting paths are displayed in Figure 5.4.

Theorem 5.1 extends our notion of $w(z)$ from the local characterization given by Theorem 4.3 to a global one. Note that this global characterization comes at the expense of no longer being able to consider each $w_i(z)$ individually.

Example: Consider the two paths $z_a(t)$ and $z_b(t)$ $(0 \leq t \leq 1)$ given by

$$z_a(t) = e^{j2\pi t} \tag{5.5}$$

$$z_b(t) = 2 - e^{j2\pi t} \tag{5.6}$$

They are depicted in Figure 5.5. We consider again, the polynomial $p(w, z) = w^2 - z$. Since $z_a(0) = z_b(0)$, we can associate with $z_a(t)$ and $z_b(t)$ two paths which begin at the same value of $w = 1$, $w_a(t)$ and $w_b(t)$. The corresponding paths are drawn in Figure 5.6. Although $z_a(1) = z_b(1)$, we find that $w_a(1) \neq w_b(1)$, i.e., we have "moved" from one branch of $w(z)$ to another.

The reader may at this point realize the similarity between the Figures 5.3 to 5.6 and the technique of root locus analysis which forms an important part of classical control analysis. The observation is well-justified since root locus analysis seeks to find the zeros of the function

$$P(s, K) = H(s) + KG(s) \tag{5.7}$$

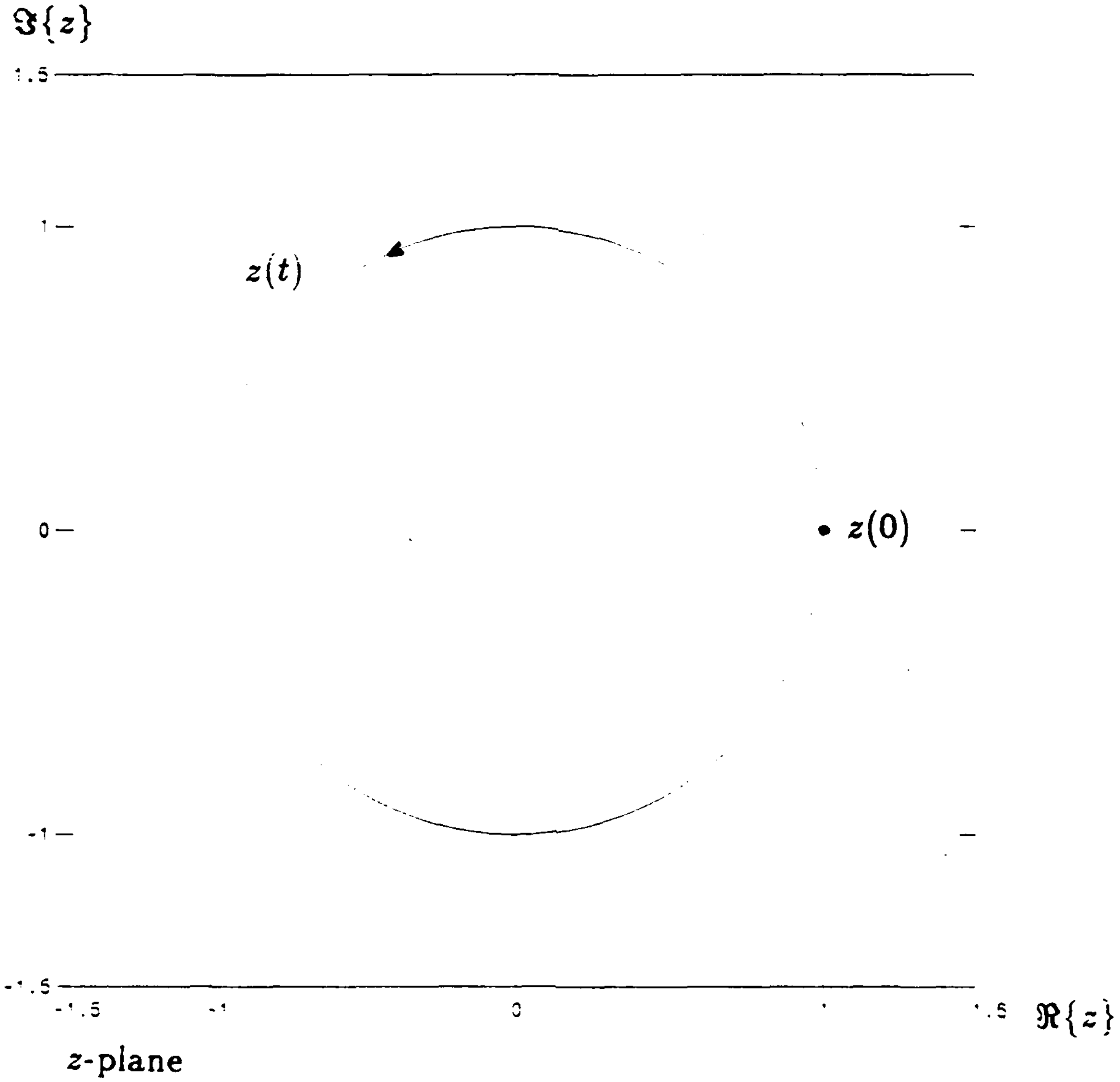


Figure 5.3: Path in the z-plane given by $z(t) = e^{j2\pi t}$.

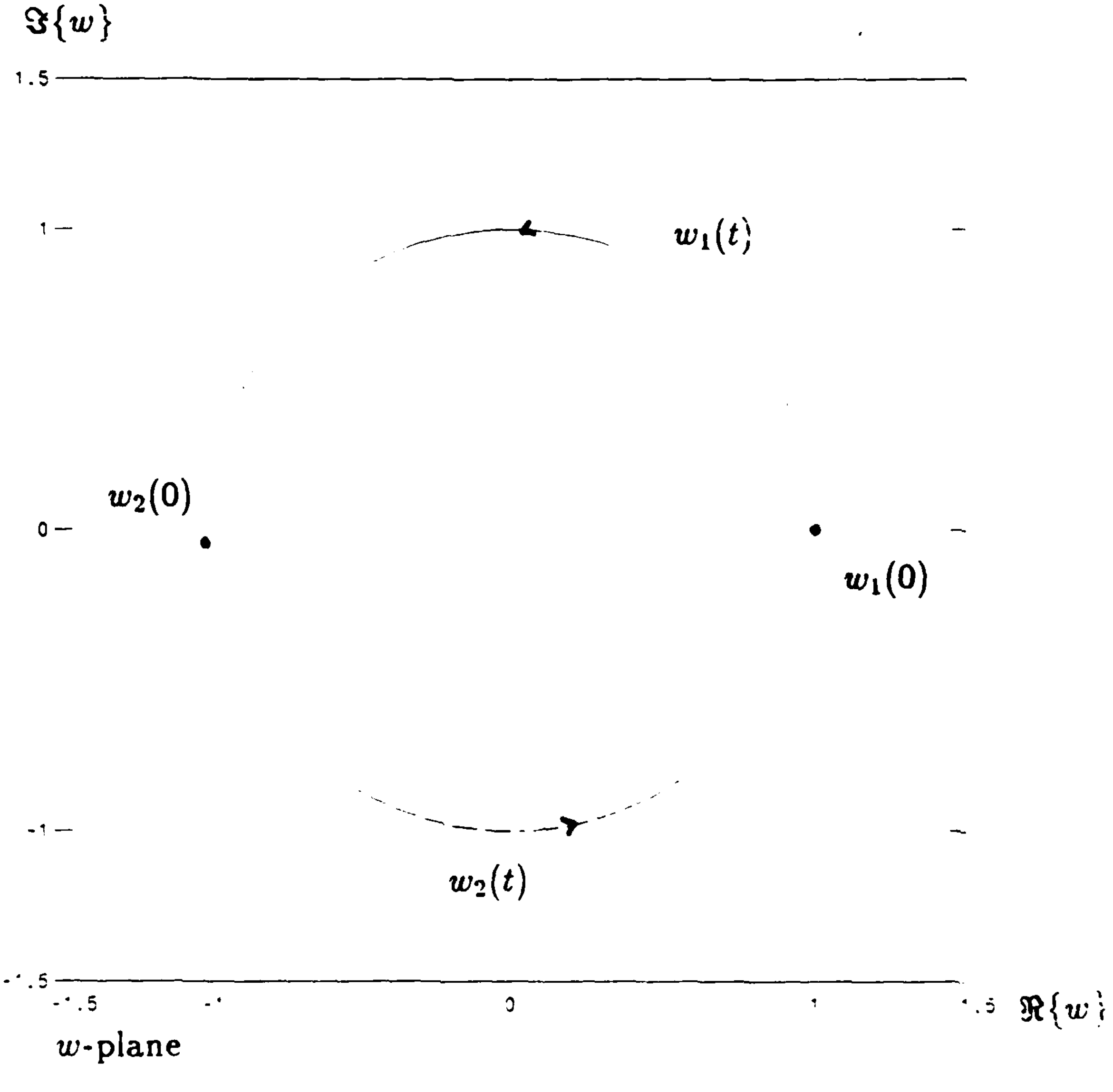


Figure 5.4: Image of $z(t) = e^{j2\pi t}$ under the algebraic function of $w^2 - z$.

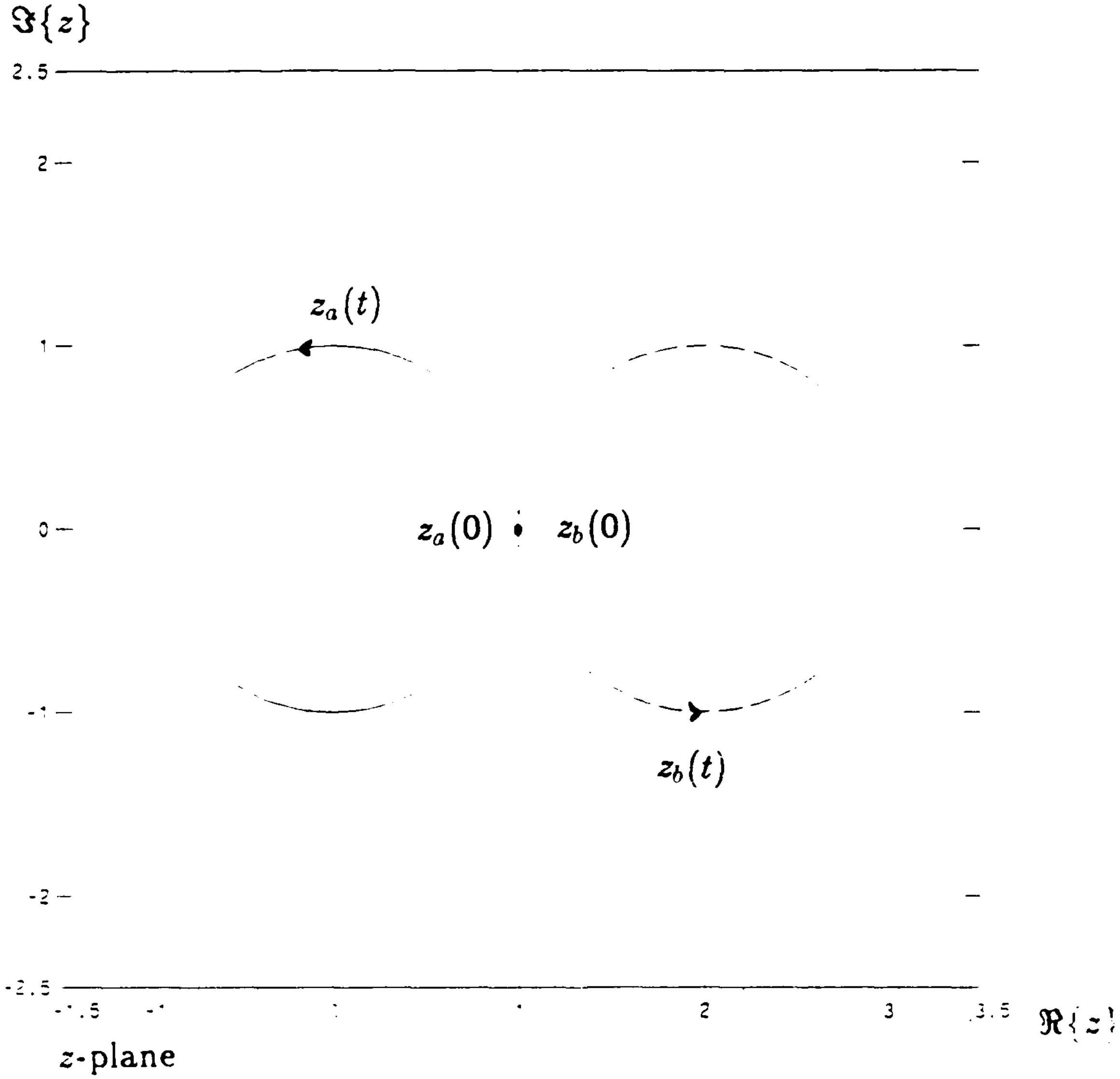


Figure 5.5: Two paths which begin and end at the same point in the z-plane.

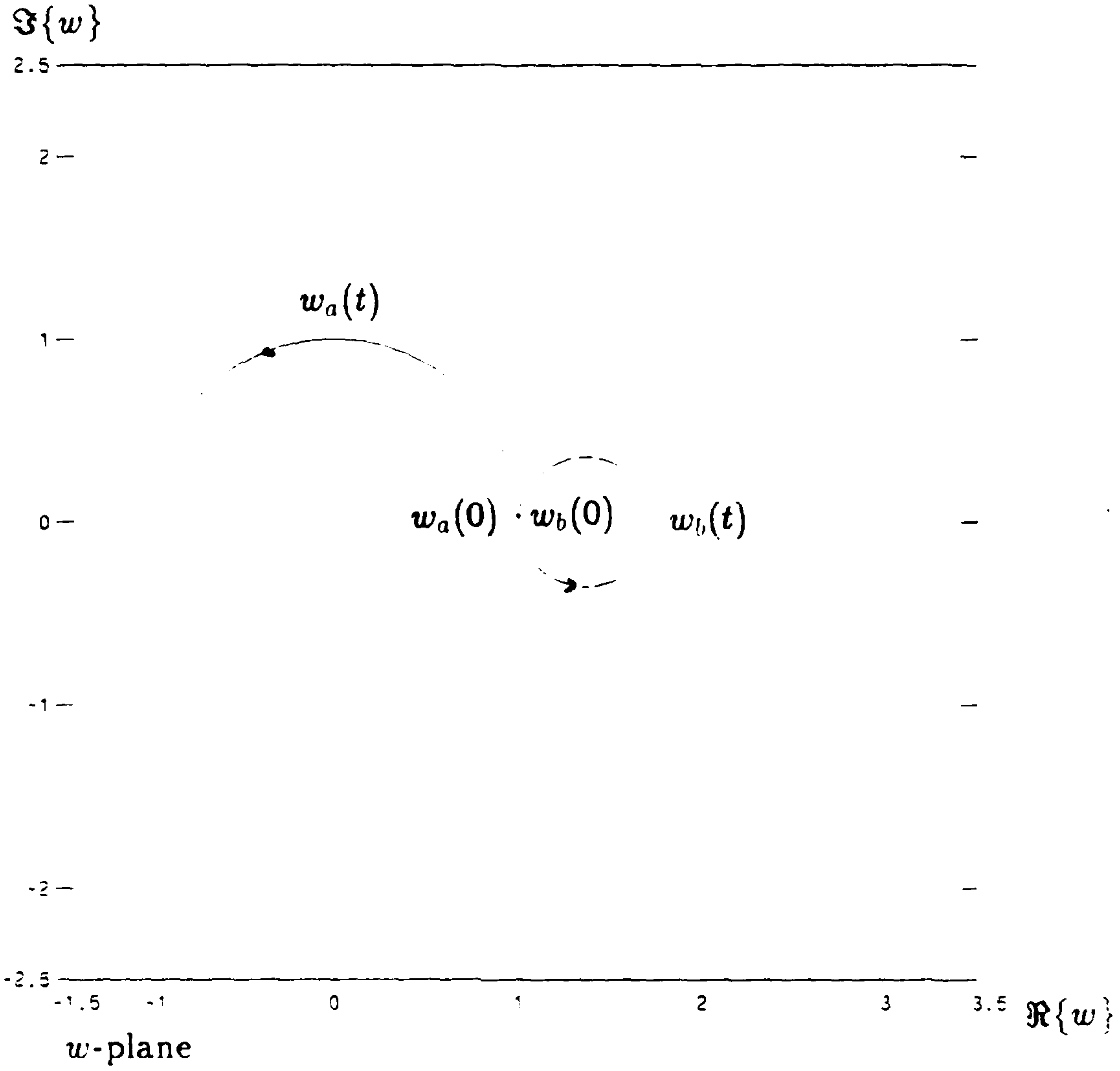


Figure 5.6: Mapping of paths to the w-plane.

where $H(s)$ is the feed-forward transfer function numerator, $G(s)$ is the feed-forward transfer function denominator and K is the feedback gain. The similarity becomes obvious once we consider (5.7) as a polynomial in the two variables s and K. Moreover, it is easy to show that $P(s, K)$ is an irreducible polynomial as long as $H(s)$ and $G(s)$ do not have a common factor. Thus, many of the observations we have made so far and will be making in the future also pertain to root-locus analysis. Where our discussion and root-locus analysis depart is that although in root-locus analysis, K is restricted to be real, we will allow both w and z to take on complex values.

We need one more result regarding the path mapping property of algebraic functions. This next result is analogous to Theorem 4.4 which was used by Kaltofen to develop his bivariate polynomial factorization algorithm. Recall that Theorem 4.4 stated simply that each branch $w_i(z)$ of the algebraic function $w(z)$ corresponding to a polynomial $p(w, z)$ can be locally associated with an irreducible factor of $p(w, z)$, i.e., there is an irreducible polynomial $p_i(w, z)$ which divides $p(w, z)$ and for which $p_i(w_i(z), z) = 0$ is true everywhere in a neighborhood of an ordinary point. This next theorem states a similar result for each path $w_i(t)$ which is an image of $z(t)$ under $w(z)$.

Theorem 5.2 *Let $z(t)$ be a path in the z-plane consisting exclusively of ordinary points of an algebraic function $w(z)$ corresponding to a polynomial $p(w, z)$. If the path $w_i(t)$ is an image of $z(t)$ via $w(z)$ in a given interval, then there is an irreducible factor $p_i(w, z)$ which divides $p(w, z)$ such that $p_i(w_i(t), z(t)) = 0$ for all t in the given interval.*

Proof: This follows rather directly from the fact that $z(t)$ only includes ordinary points and that $z(t)$ and all the polynomials involved are continuous functions. Suppose that for $t = t_0$, $(w_i(t_0), z(t_0))$ is a zero of $p_i(w,z)$ where $p_i(w,z)$ is an irreducible factor of $p(w,z)$. Express $p(w,z)$ as the product $p_i(w,z)q(w,z)$. We want to show that for all subsequent values of t, $(w_i(t), z(t))$ is a zero of $p_i(w,z)$. We will have to make use of Theorem 5.1, i.e., if $z(t)$ consists of only ordinary points, then $w(t)$ is also a continuous path.

Suppose that at some point, the hypothesis is not true, i.e., $p_i(w_i(t), z(t)) \neq 0$, say at t_1. Since $(w_i(t), z(t))$ is a zero of $p(w,z)$ for all, we must have that $q(w(t_1), z(t_1)) = 0$. Because of continuity, there must be a transition t^* between t_0 and t_1 such that $p_i(w(t^*), z(t^*)) = q(w(t^*), z(t^*)) = 0$. However, this would mean that $p_w(w(t^*), z(t^*)) = 0$ or that $z(t^*)$ is a singular point of $w(z)$. This contradicts the assumption that $z(t)$ consists of only ordinary points. The fact that this irreducible polynomial $p_i(w,z)$ is the only one which contains $(w_i(t), z(t))$ as a zero for all t will become obvious once we introduce Bezout's theorem later on. □

The theorem above is exactly what is needed. From Theorem 5.2 we see that the zeros of $p(w,z)$ which lie on a path pair $(w(t), z(t))$ must all be from one irreducible factor of $p(w,z)$. An algorithm for isolating an irreducible factor of $p(w,z)$ can be described in the following three steps:

- Pick a path $z(t)$ consisting entirely of ordinary points of $p(w,z)$.
- Find a path $w(t)$ such that $p(w(t), z(t)) = 0$ for all t of interest.
- Find the irreducible polynomial $d(w,z)$ such that $d(w(t), z(t)) = 0$ for all t of interest. According to Theorem 5.2, the resulting polynomial $d(w,z)$ will divide $p(w,z)$.

5.1.2 Picking a Path Consisting of Ordinary Points

In order to study which paths in the z-plane consist of only ordinary points, we need to discuss in more detail the behavior and propensity of points in the z-plane which are not ordinary, i.e., the singular points. Recall that there are two types of singular points: the first type is composed of all z such that one branch $w_i(z)$ becomes infinite. We have already shown in Theorem 4.1 that there are $deg_w(p)$ such z. The second type of singular point consists of all z such that two branches coalesce at z, i.e., $w_i(z) = w_j(z)$ at z. Theorem 4.2 stated that if (w_0, z_0) is such that $w_0 = w_i(z_0) = w_j(z_0)$, then not only $p(w_0, z_0) = 0$ but also $p_w(w_0, z_0) = 0$ as well. Since $p_w(w, z)$ is also a bivariate polynomial and since (w_0, z_0) resides in the common zeros of the two polynomials $p(w, z)$ and $p_w(w, z)$, we need to discuss briefly the topic of the intersection of the zeros of bivariate polynomials.

In general, two bivariate polynomials will only have a finite number of common zeros. This can be supported intuitively by the fact that a possible common zero (w_0, z_0) of $p(w, z)$ and $q(w, z)$ can be represented by the four dimensional real vector $[Re(w_0), Im(w_0), Re(z_0), Im(z_0)]^T$. On the other hand, (w_0, z_0) must satisfy simultaneously the four equations below,

$$Re(p(w_0, z_0)) = Im(p(w_0, z_0)) = Re(q(w_0, z_0)) = Im(q(w_0, z_0)) = 0 \qquad (5.8)$$

The maximum number of such common zeros is the subject of Bezout's theorem below [8],

Theorem 5.3 *If $p(w,z)$ and $q(w,z)$ are bivariate polynomials of total degree r and s respectively with no common factors, then there are at most rs distinct pairs (w,z) where*

$$p(w,z) = q(w,z) = 0 \tag{5.9}$$

Example: An example where the maximum number of common zeros is attained with finite w and z is the pair of polynomials

$$p(w.z) = w^2 + z^2 - 4 \tag{5.10}$$
$$q(w.z) = w^2 - z^2 - 1 \tag{5.11}$$

The set of zeros of these polynomials for real $(w.z)$ is plotted in Figure 5.7. The total degree for each polynomial is 2. The common zeros occur at $(\pm\sqrt{3}.\pm\sqrt{5})$. Since the total degree of each polynomial is 2. Bezout's theorem predicts a maximum number of common zeros of 4. which is attained exactly in this case.

One application of Bezout's theorem which we will have need to make use of later on, is that it specifies the minimum number of zeros needed to specify an irreducible polynomial to a scale factor. We will discuss this point in more detail in a later part of this chapter.

Bezout's theorem applies to any two polynomials with the specified total degree. In much of our discussion we will know the degree in each variable as well as the total degree of the polynomial in question. An obvious question then is whether we can somehow tighten the bound on the number of common zeros if the degree of the polynomials involved is known. In a sense we cannot do this because there is a strong form of Bezout's theorem where it is shown that the bound described above is actually achieved. If the degree of $p(w,z)$ is (M_p, N_p) with the $w^{M_p}z^{N_p}$ term non-zero and the degree of $q(w,z)$ is (M_q, N_q) with the $w^{M_q}z^{N_q}$ term non-zero then the strong form of Bezout's theorem states that there are $(M_p + N_p)(M_q + N_q)$ common zeros. However, many of these zeros are "phantom" zeros where w or z or both are infinite. In our study we will be concerned with finite zeros. Therefore, it is important to be able to

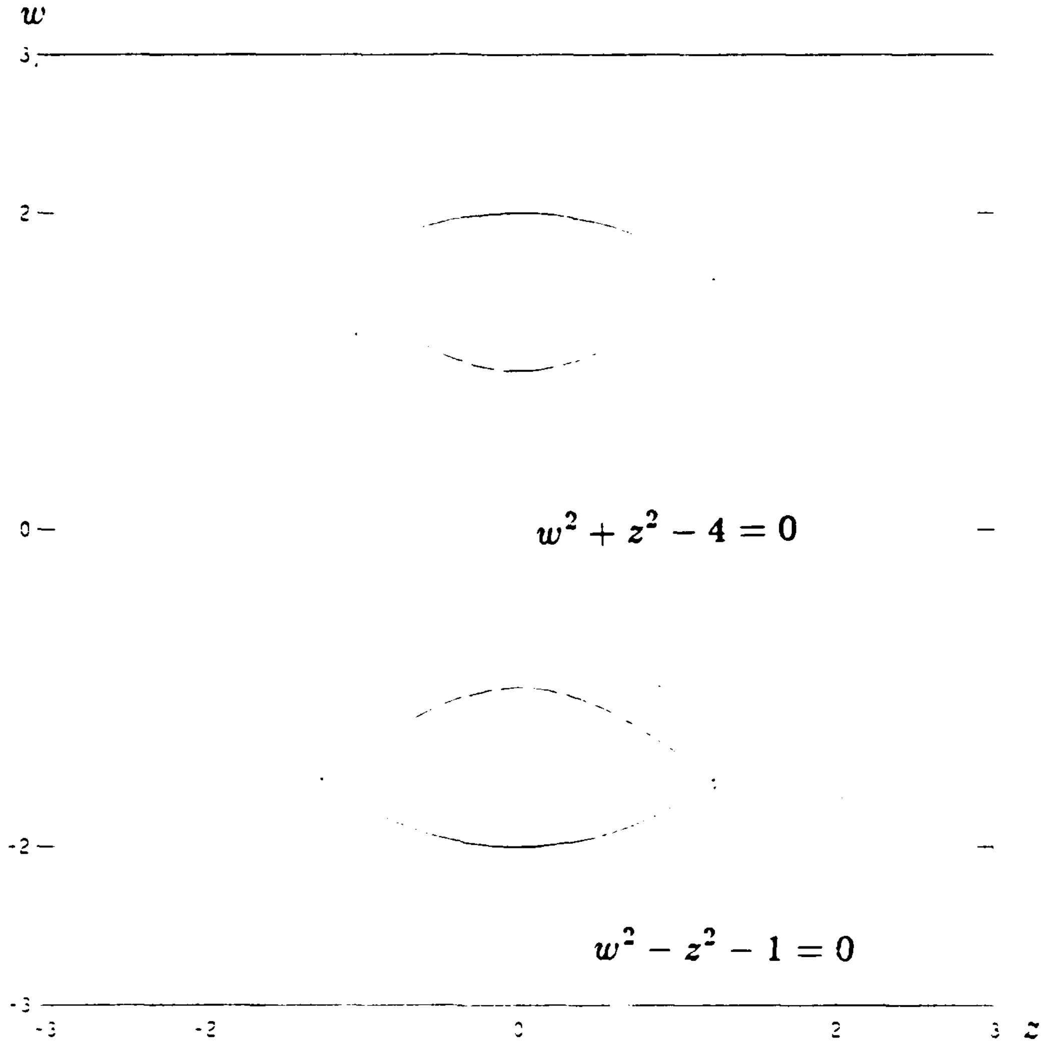


Figure 5.7: Real zeros of $w^2 + z^2 - 4$ and $w^2 - z^2 - 1$

study the number of finite zeros which two polynomials with specified degree can have in common. This is the topic of Appendix B where the following theorem is proven,

Theorem 5.4 *Let $p(w,z)$ and $q(w,z)$ be two polynomials of degree (M_p, N_p) and (M_q, N_q) with no common factors, then there are at most $N_pM_q + N_qM_p$ pairs of finite complex numbers (w,z) such that*

$$p(w,z) = q(w,z) = 0 \tag{5.12}$$

That the bound above can be attained exactly for some $p(w,z)$ and $q(w,z)$ is demonstrated by the following example:

Example: Consider the two polynomials of degree $(1,1)$

$$p(w,z) = wz - 2 \tag{5.13}$$

$$q(w,z) = wz + w - z - 3 \tag{5.14}$$

The zero sets for w, z real are drawn in Figure 5.8. Since these two polynomials have 2 common finite zeros, the bound of Theorem 5.4 is achieved.

Recall that branch points of $p(w,z)$ reside in the intersection of the zeros of $p(w,z)$ and $p_w(w,z)$. The theorems just discussed specify the number of zeros in common between two polynomials which are relatively prime. As soon as we show that in most cases $p(w,z)$ and $p_w(,z)$ are relatively prime, then we can use these theorems to specify the maximum number of branch points $p(w,z)$ may have, which in turn bounds the number of singular points in the z-plane.

Theorem 5.5 *Let $p(w,z)$ be expressed as a product of irreducible primitive polynomials and a constant*

$$p(w,z) = \alpha \prod_{i=1}^{P} p_i(w,z) \tag{5.15}$$

If $p_i(w,z) \neq p_k(w,z)$ for $i \neq k$, then $p(w,z)$ and $p_w(w,z)$ are relatively prime.

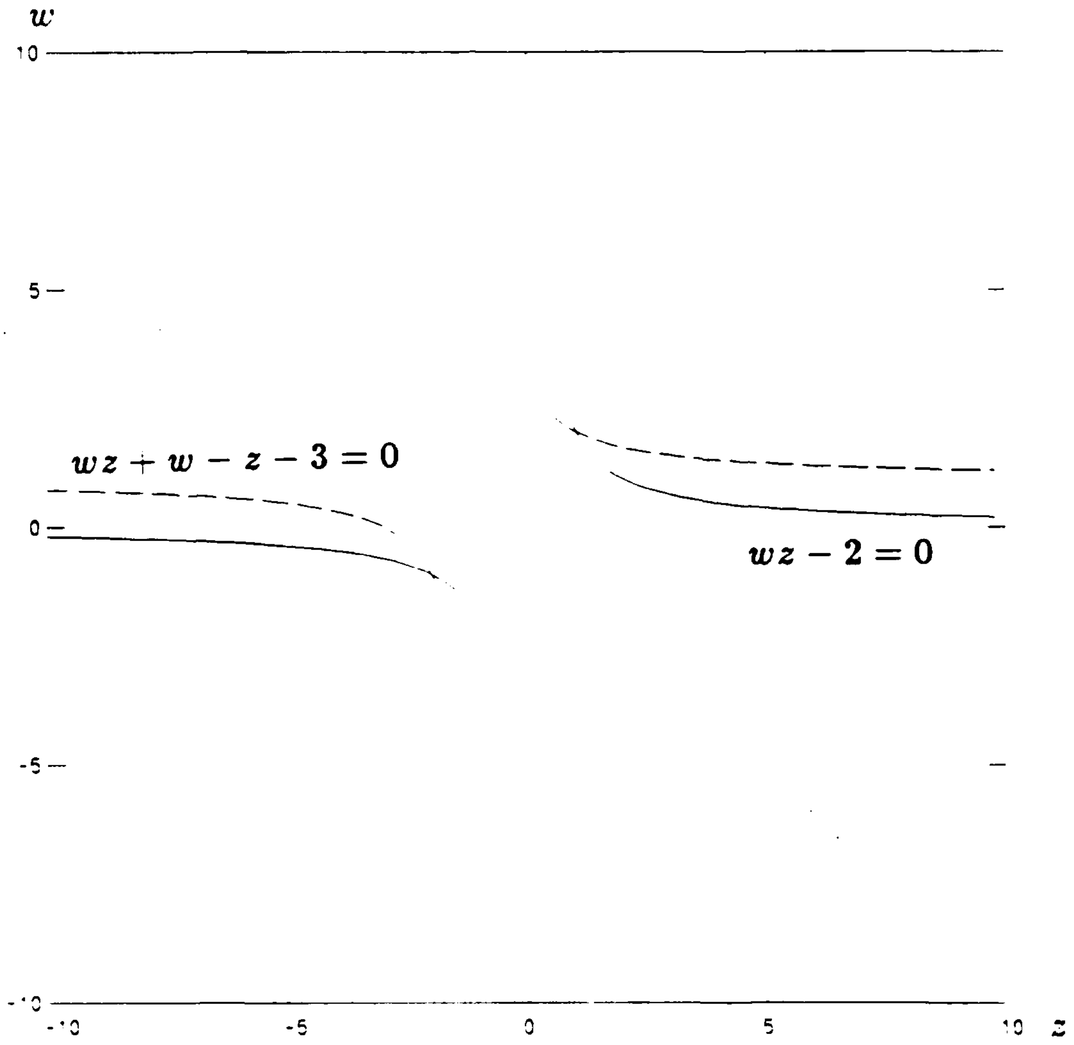


Figure 5.8: Intersection of zeros of $wz - 2$ and $wz + w - z - 3$.

Proof: A polynomial such that its irreducible factors do not repeat is called a *square-free* polynomial. Thus, the condition above is that $p(w,z)$ be square-free. Assume that $p(w,z)$ and $p_w(w,z)$ have a common non-trivial irreducible factor $p_1(w,z)$. Therefore $p_1(w,z)$ divides $p_w(w,z)$, say $p_w(w,z) = p_1(w,z)q(w,z)$. However $p_w(w,z)$ can be expressed as,

$$\begin{aligned} p_w(w,z) &= \alpha \textstyle\sum_{i=1}^{P} p_{i.w}(w,z) \prod_{j\neq i} p_j(w,z) \\ &= p_{1.w}(w,z)A(w,z) + p_1(w,z)B(w,z) \end{aligned} \tag{5.16}$$

where $p_{i.w}(w,z)$ is the partial derivative of $p_i(w,z)$ with respect to w,

$$A(w,z) = \prod_{i=2}^{P} p_i(w,z) \tag{5.17}$$

and $B(w,z)$ is a polynomial in w and z. Since we also have that $p_w(w,z) = p_1(w,z)q(w,z)$,

$$p_{1.w}(w,z)A(w,z) = [q(w,z) - B(w,z)]\, p_1(w,z) \tag{5.18}$$

Therefore $p_1(w,z)$ divides $A(w,z)$ or $p_{1.w}(w,z)$. However, since $p(w,z)$ has no repeated factors, $A(w,z)$ and $p_1(w,z)$ are relatively prime. Furthermore, $p_{1.w}(w,z)$ is of lower total degree than $p_1(w,z)$ and therefore cannot contain $p_1(w,z)$ as a factor since $p_1(w,z)$ is irreducible. Therefore $p_{1.w}(w,z)A(w,z)$ and $p_1(w,z)$ are relatively prime. Thus the assumption that $p(w,z)$ and $p_w(w,z)$ have a common factor leads to a contradiction.

□

Using Theorem 5.5, we can deduce that for bivariate polynomials which are square-free, the number of singular points in the z-plane is finite, and in fact the maximum number of finite singular points can be bounded by the minimum of the bounds of Theorems 5.3 and 5.4.

Theorem 5.6 *Let $p(w,z)$ be a square-free polynomial of degree (M,N) and total degree n_p, then the number of finite singular points is bounded by $min(n_p(n_p-1), 2(M-1)N) + N$.*

Proof: Since $p(w,z)$ is square-free, $p(w,z)$ and $p_w(w,z)$ are relatively prime. Since the total degree of $p_w(w,z)$ is $n_p - 1$ and the degree of $p_w(w,z)$ is $(M-1,N)$, the maximum number of finite common zeros is $min(n_p(n_p-1), 2(M-1)N)$. The first term is due to Bezout's theorem; the second term is due to our intersection Theorem 5.4. From Theorem 4.2, this means that the number of branch points of $p(w,z)$ is bounded by $min(n_p(n_p-1), 2(M-1)N)$. From Theorem 4.1, the number of singular points where a branch of $w(z)$ becomes infinite is at most N. Combining the two sets of singular points we conclude with the total bound of $min(n_p(n_p-1), 2MN-N)+N$.

□

The importance of the theorem above at this point is not the actual bound on the number of singular points, but rather that the number is finite. This means that if we pick a path $z(t)$ at random, it will almost surely consist exclusively of ordinary points.

The previous discussion showed that almost all paths in the z-plane consist of only ordinary points, since singular points form a very sparse set, consisting of only a finite number of points. In the following section we address the problem of specifying a path $w(t)$ for a given path $z(t)$ such that $p(w(t), z(t))$ is zero for all t of interest.

5.1.3 Tracking Zero Paths

Given a path $z(t)$ consisting of ordinary points, and an initial zero of $p(w,z)$, say $p(c, z(0))$, it is straightforward to pose a differential equation that $w(t)$ must satisfy.

Recall from (4.21) we calculated the derivative of each branch of $w(z)$ as,

$$\frac{dw_i(z)}{dz} = -\frac{p_z(w_i(z), z)}{p_w(w_i(z), z)} \tag{5.19}$$

This in turn implies that the derivative of $w(t)$ with respect to t must be

$$\frac{dw(t)}{dt} = -\frac{p_z(w(t), z(t))}{p_w(w(t), z(t))} \frac{dz(t)}{dt} \tag{5.20}$$

We are left with the following first order initial value problem:

Find the function $w(t)$ which satisfies, $w(0) = c$ and

$$\frac{dw(t)}{dt} = -\frac{p_z(w(t), z(t))}{p_z(w(t), z(t))} \frac{dz(t)}{dt} \tag{5.21}$$

By solving the initial value problem above, we can calculate a $w(t)$ for any $z(t)$ such that $p(w(t), z(t)) = 0$ for all t of interest.

5.1.4 Extracting Polynomial Coefficients from Zeros

At this point we have, at least conceptually, an arbitrarily large number of zeros of one irreducible factor of $p(w, z)$, say $d(w, z)$; namely all the complex pairs $(w(t), z(t))$. The next part of the procedure extracts the coefficients of $d(w, z)$ from its zeros. This problem was first considered and solved by Curtis [39,40]. Basically, we sample $(w(t), z(t))$ at t_k, $t = 1, \cdots, K$ to yield K simultaneous homogeneous linear equations for the coefficients of $d(w, z)$,

$$\sum_{m=0}^{deg_w(d)} \sum_{n=0}^{deg_z(d)} d_{m,n} w_k^m z_k^n = 0 \text{ for } k = 1, \cdots, K \tag{5.22}$$

where $d_{m,n}$ is the set of coefficients of $d(w, z)$.

There are two points that need to be addressed at this point. First is the question of how many samples of the zeros of $d(w,z)$ are needed to guarantee that there will be essentially only one solution to (5.22). The second question is how to specify $deg_w(d)$ and $deg_z(d)$ without knowing $d(w,z)$.

The question of determining the degree of $d(w,z)$ is addressed in Appendix C. The reason why we do not discuss the problem here is that in the case of the phase retrieval problem, the issue does not arise as we will see later. Let us assume then that the degree of the irreducible factor of $p(w,z)$ to be extracted is known. We now want to specify K such that the coefficients of $d(w,z)$ will be retrievable from (5.22).

Consider a non-zero solution to (5.22), say the coefficients $a_{m,n}$. We can associate with this set of coefficients a polynomial $a(w,z)$ given by,

$$a(w,z) = \sum_{m=0}^{deg_w(d)} \sum_{n=0}^{deg_z(d)} a_{m,n} w^m z^n \tag{5.23}$$

Since $d(w,z)$ is irreducible, and $a(w,z)$ and $d(w,z)$ have the same degree, $d(w,z)$ and $a(w,z)$ must be relatively prime or related by a constant factor. From Theorem 5.4, then $d(w,z)$ and $a(w,z)$ can have at most $2deg_w(d)deg_z(d)$ zeros in common. Since $a_{m,n}$ is a solution to (5.22), the polynomial $a(w,z)$ must satisfy,

$$a(w_k, z_k) = 0 \tag{5.24}$$

Thus, $a(w,z)$ and $d(w,z)$ have K zeros in common. If we pick $K > 2deg_z(d)deg_w(d)$, then $a(w,z)$ must be related to $d(w,z)$ by a constant factor which implies in turn that $a_{m,n}$ and $d_{m,n}$ are related by a constant factor. Therefore, by finding a null vector of the matrix displayed in (5.22) we have recovered an irreducible factor of $p(w,z)$. Note that using Theorem 5.3, the minimum number of equations required to guarantee the correct reconstruction is twice the number specified by Theorem 5.4.

Let us review the steps necessary to find an irreducible factor of a given square-free bivariate polynomial $p(w, z)$.

1. Find an ordinary point z_0 of the algebraic function corresponding to $p(w, z)$.

2. Using z_0, find a w_0 such that (w_0, z_0) is a zero of $p(w, z)$. Such a w_0 can be calculated via the method described in Section 4.2.2. This complex pair will also be a zero of an irreducible factor of $p(w, z)$, $d(w, z)$.

3. Generate a path $z(t)$ which begins at z_0 and which consists only of ordinary points. Since a square-free polynomial has a finite number of singular points, almost any path will consist of only ordinary points.

4. Use the differential equation (5.21) and the initial condition $w_0 = w(0)$ to calculate a path $w(t)$ which satisfies $p(w(t), z(t)) = 0$ for all t of interest.

5. Sample the path pair $(w(t), z(t))$ to yield (w_k, z_k) and solve (5.22) to get the coefficients of an irreducible factor of $p(w, z)$, $d(w, z)$.

5.2 Considerations for Phase Retrieval

In the discussion above we considered the polynomial factorization problem in general. Here we would like to consider the phase retrieval problem specifically. We first note that since in almost all practical cases, the polynomial $p_x(w, z)$ is irreducible, the autocorrelation polynomial $p_r(w, z)$ will only have two factors, namely $p_x(w, z)$ and

$\tilde{p}_x(w, z)$. Therefore, unlike the general polynomial factorization problem, the degree of the factors is known; if $deg(p_r) = (2M, 2N)$ then $deg(p_x) = deg(\tilde{p}_x) = (M, N)$. One final point to consider is under what situations $p_r(w, z)$ is a square-free polynomial which implies in turn that $p_r(w, z)$ has a finite number of singular points. Since in practical cases, $p_x(w, z)$ is irreducible, we restrict our attention to this case only. In this situation, $p_r(w, z)$ has only two factors, $p_x(w, z)$ and $\tilde{p}_x(w, z)$. Thus, if $p_x(w, z) \neq \tilde{p}_x(w, z)$ then $p_r(w, z)$ is square-free. Pictorially, this means that $x[m, n]$ when rotated by 180 degrees does not look identical to the original image. Thus, we have the following theorem,

Theorem 5.7 *If the polynomial associated with $x[m, n]$, $p_x(w, z)$ is irreducible, and $p_x(w, z)$ is not equal to its mirror polynomial, then the polynomial associated with $r[m, n]$ will be square-free.*

Since most images do not have this symmetry property we can be assured that $p_r(w, z)$ will be square-free and therefore it will have a finite number of singular points. This implies in turn that most paths in the z-plane consists of only ordinary points.

An M by N pixel image will have an associated polynomial of degree $(M - 1, N - 1)$. Thus, if this polynomial is irreducible, it is specified, according to Theorem 5.4, by a number of zeros greater than $2(M - 1)(N - 1)$. The minimum number of equations in (5.22) necessary to assure a unique solution becomes $2(M - 1)(N - 1) + 1$.

The total algorithm for reconstruction from Fourier transform magnitude of a signal with an irreducible associated polynomial can be decomposed into the following steps.

1. Calculate $r[m, n]$ from $|X(e^{ju}, e^{jv})|$ via (2.13).

2. Form $p_r(w,z)$ from $r[m,n]$.

3. Pick z_0 such that it is an ordinary point of $p_r(w,z)$. Almost all values of z_0 will be adequate.

4. Find a w_0 such that $p_r(w_0,z_0) = 0$.

5. Pick a path in the z-plane consisting of ordinary points such that $z(0) = z_0$ and solve the differential equation using w_0 as the initial condition and (5.21).

6. Pick enough (w,z) pairs such that (5.22) has a unique solution. By our version of Bezout's theorem the minimum number is $2(M-1)(N-1)+1$ for an M by N image.

7. Since $p_x(w,z)$ is irreducible, then the solution to the set of equations above yields coefficients proportional to the image values $x[m,n]$ or $\hat{x}[-m,-n]$.

8. Normalize the coefficients found such that

$$\sum_{m,n} x[m,n]^2 = r[0,0] \tag{5.25}$$

A conceptual flowchart of the steps involved in the algorithm is depicted in Figure 5.9.

Example: Before considering some practical images, it is instructive to illustrate the algorithm using a small example. Consider the situation where we are to reconstruct the following image from the Fourier transform magnitude.

$$x[m,n] = \begin{matrix} 4 & 2 & 1 \\ 0 & 3 & 3 \\ 3 & 0 & 1 \end{matrix} \tag{5.26}$$

This image has associated polynomial given by.

$$p_x(w,z) = 4 + 2z + z^2 + 3wz + 3wz^2 + 3w^2 + z^2w^2 \tag{5.27}$$

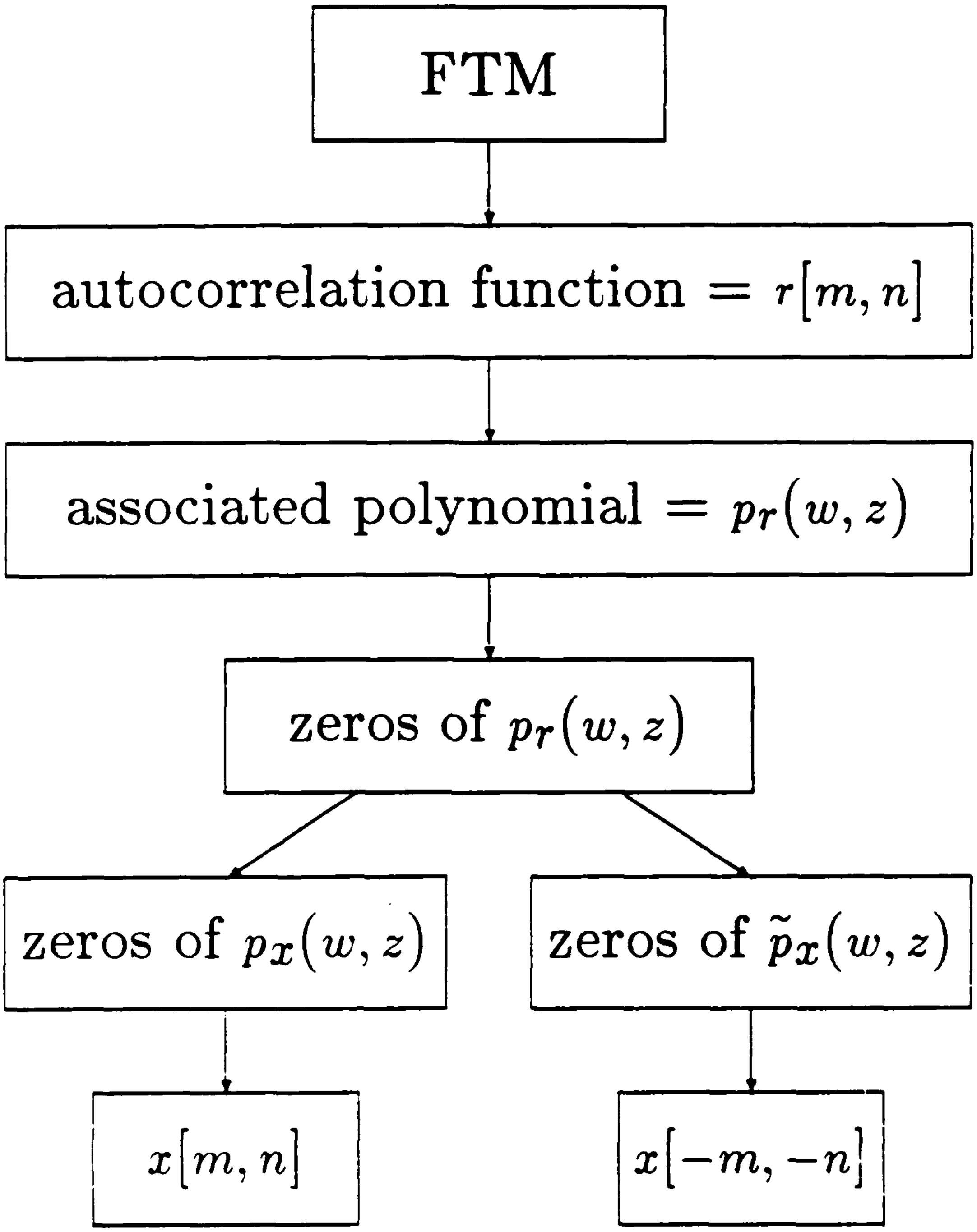

FTM
autocorrelation function = $r[m,n]$
associated polynomial = $p_r(w,z)$
zeros of $p_r(w,z)$
zeros of $p_x(w,z)$
zeros of $\tilde{p}_x(w,z)$
$x[m,n]$
$x[-m,-n]$

Figure 5.9: Algorithm steps in new phase retrieval algorithm.

As a path in the z-plane, we will use $z(t) = .8e^{j2\pi t}$, for t between 0 and 1, i.e., a circle of radius .8 centered at the origin, Figure 5.10. The reason for using this path is discussed in the next section. For this path, we can calculate the paths $w_i(t)$ in the w-plane such that $p_z(w_i(t), z(t)) = 0$ for all t of interest. These paths are drawn in Figure 5.11. We can similarly calculate the paths in the w-plane which correspond to zeros of the mirror polynomial of $p_z(w, z)$, i.e., the polynomial associated with the signal.

$$\begin{matrix} 1 & 0 & 3 \\ 3 & 3 & 0 \\ 1 & 2 & 4 \end{matrix} \tag{5.28}$$

These paths are shown in Figure 5.12. Comparing Figures 5.11 and 5.12, we see that the paths

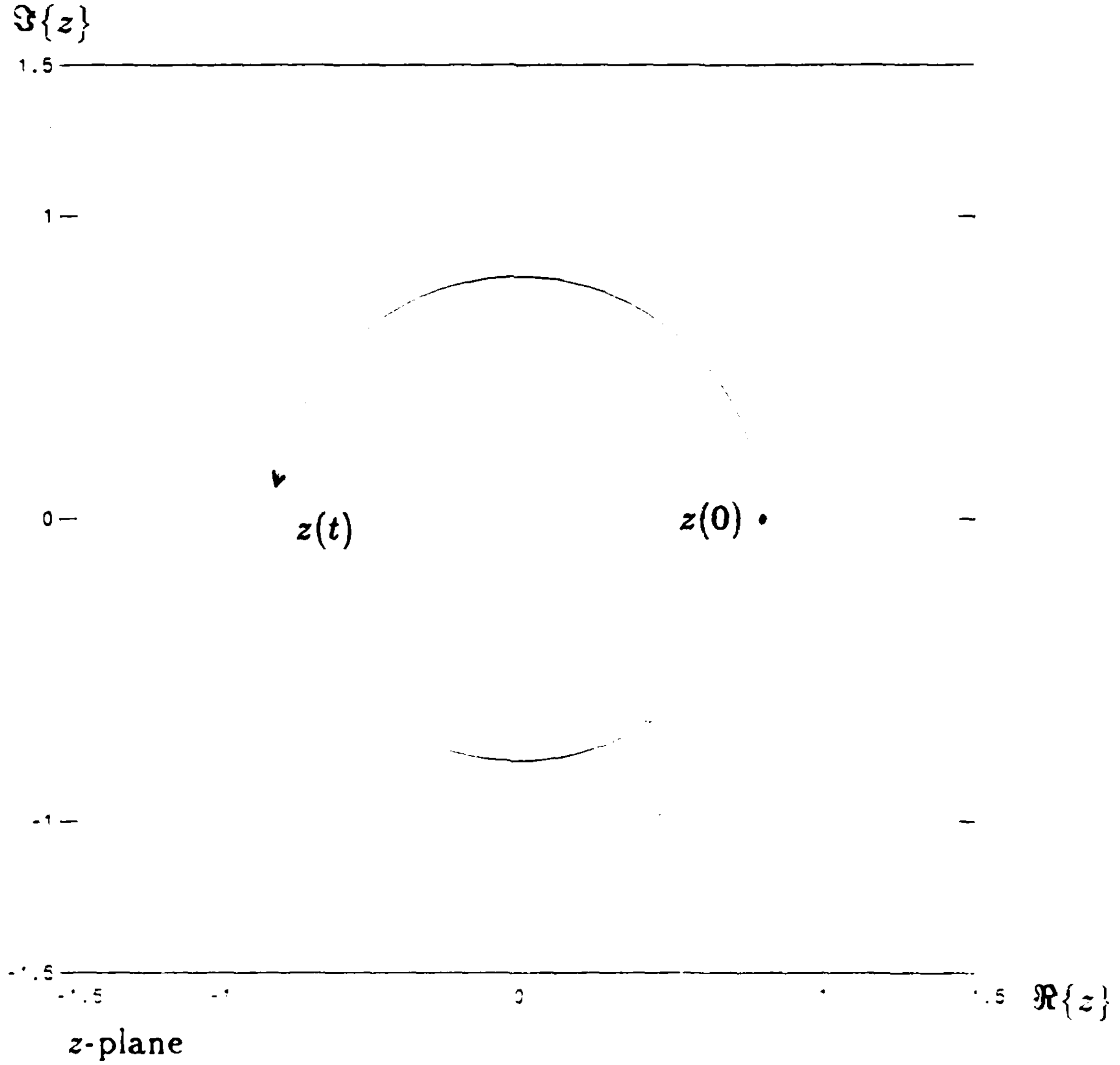


Figure 5.10: Path chosen in the z-plane for reconstruction example. The unit circle is drawn as a dotted line.

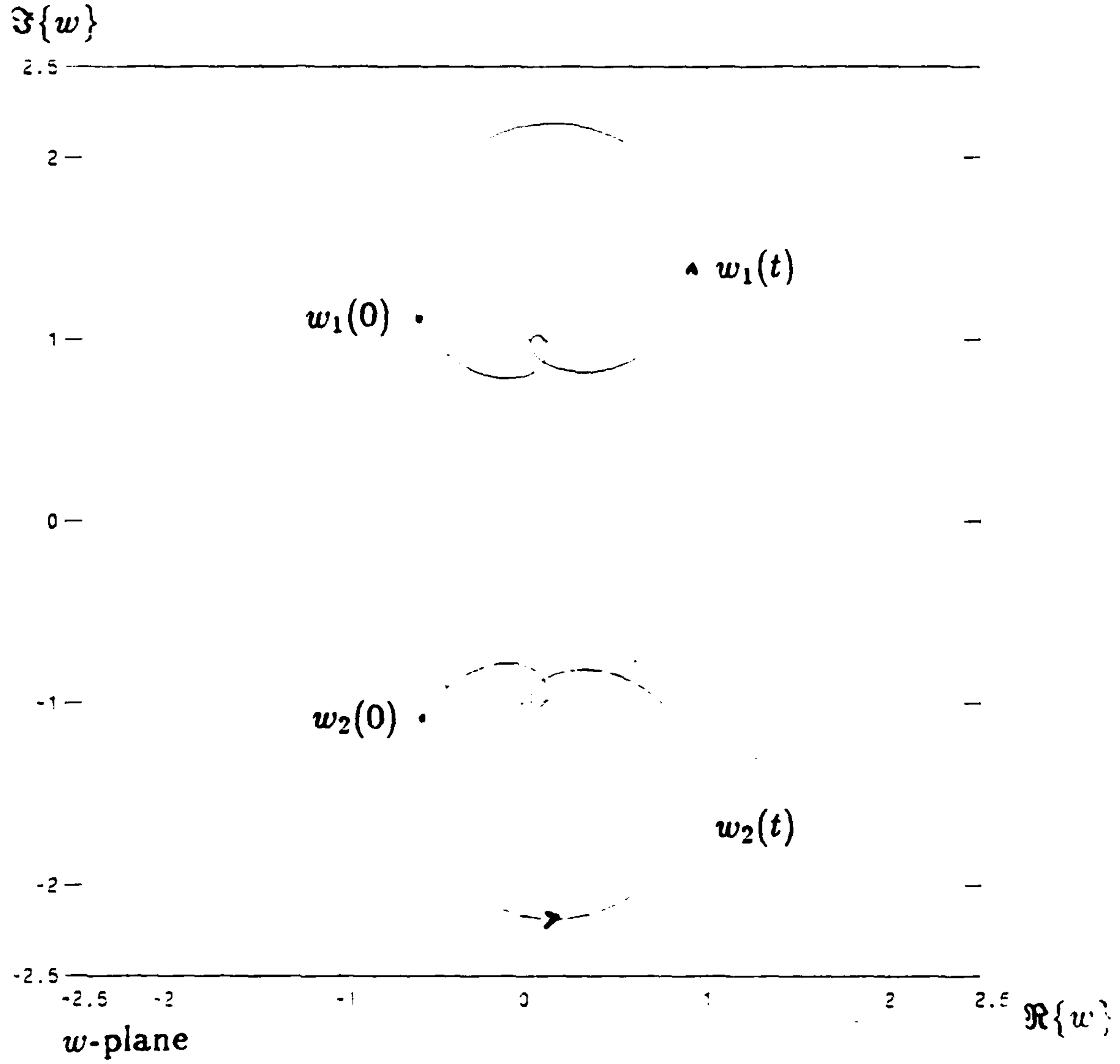


Figure 5.11: Resulting paths in the w-plane for given $z(t)$. The path $w_1(t)$ is drawn as a solid line, while $w_2(t)$ is drawn as a dashed line.

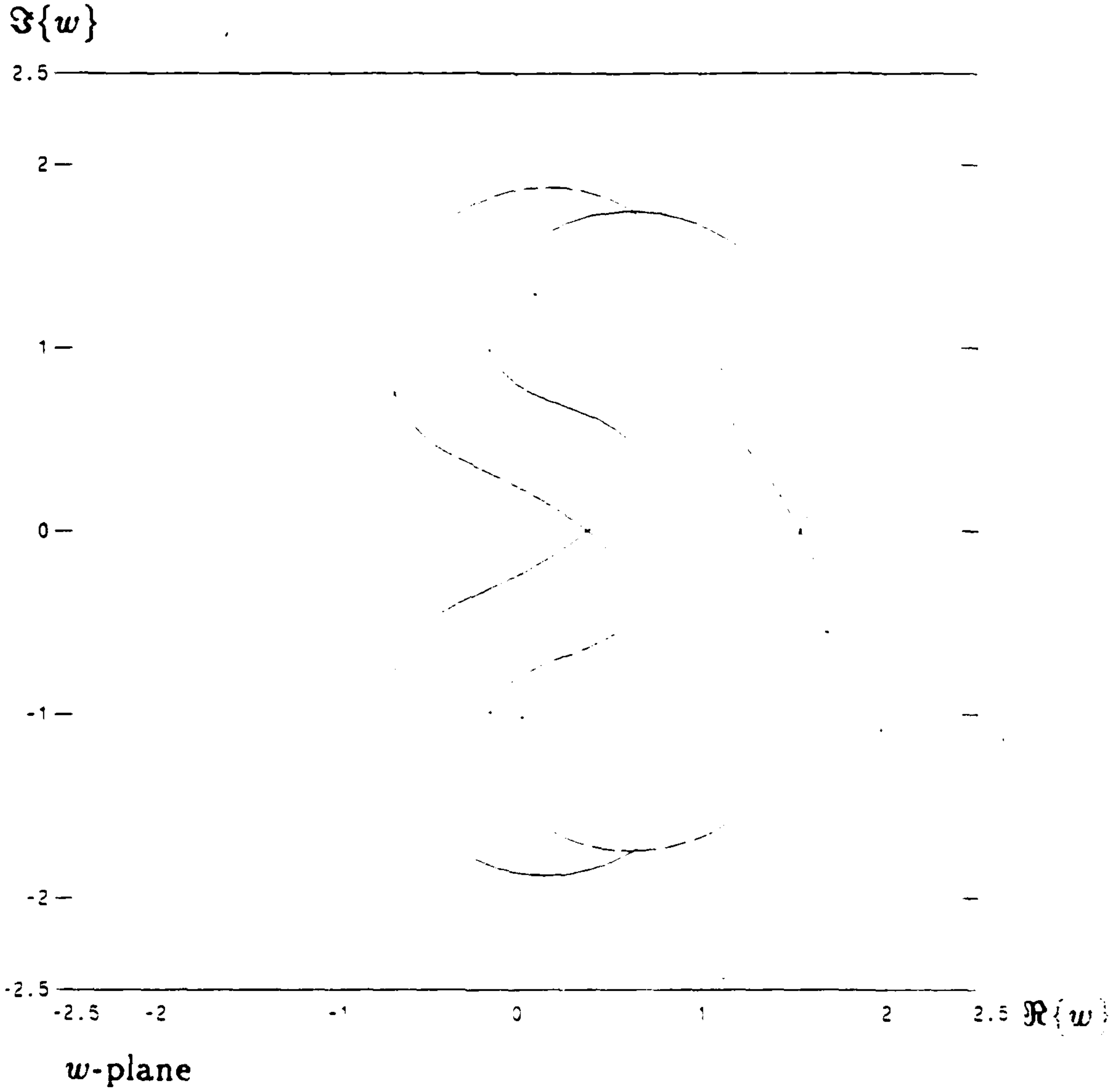


Figure 5.12: Resulting paths in the w-plane for given $z(t)$ for mirror polynomial of example.

are not reciprocal images as might be expected. This would only be the case if the path chosen in the z-plane were the unit circle exactly.

Of course, we assume that this image is actually unknown but that the autocorrelation function is given or readily calculated.

$$r[m,n] = \begin{array}{ccccc} 4 & 2 & 13 & 6 & 3 \\ 12 & 21 & 12 & 12 & 9 \\ 7 & 19 & 49 & 19 & 7 \\ 9 & 12 & 12 & 21 & 12 \\ 3 & 6 & 13 & 2 & 4 \end{array} \qquad (5.29)$$

We can similarly calculate the polynomial associated with $r[m,n]$, $p_r(w,z)$. The paths $w_i(t)$ in the w-plane such that $p_r(w_i(t),z(t)) = 0$ are shown in Figure 5.13. Note that the paths shown in Figure 5.13 consist of those corresponding to original image, drawn as a solid line, and those corresponding to the image rotated, drawn as a dashed line. The present algorithm does not calculate all these paths in the w-plane; rather, it begins with a one zero of $p_r(w,z(0))$ and generates the *single* path which begins at this zero, Figure 5.14. One may at first glance believe that there might be some ambiguity at the location in the w-plane where two paths cross. However, the reader should note that an ambiguity will only occur if the paths cross in the w-plane at the same value of z and w. In other words, one may visualize these paths as trajectories of infinitesimal particles which travel over the w-plane. An ambiguity which would preclude the possibility of isolating a single path would only occur if two particles collide, i.e., are on the same position at the same time. Using samples of the path pair $(w(t),z(t))$, the linear equation (5.22) is solved, and after normalization, results in the image,

$$\begin{array}{ccc} 1 & 0 & 3 \\ 3 & 3 & 0 \\ 1 & 2 & 4 \end{array} \qquad (5.30)$$

Comparing Figures 5.11, 5.12 and 5.14, we see that we followed a path which corresponds to the mirror polynomial of $p_x(w,z)$. Thus, the resulting reconstruction is rotated by 180 degrees.

The next section is concerned with some issues involved in implementing the algorithm just presented for images much larger than that of this simple example.

5.3 Implementation Issues

The theory developed in the previous chapter has been applied to write a computer program to extract an image from its Fourier transform magnitude. In developing this program several practical considerations need to be addressed; they all revolve, as would be expected, around the effects of finite precision representation of numbers in

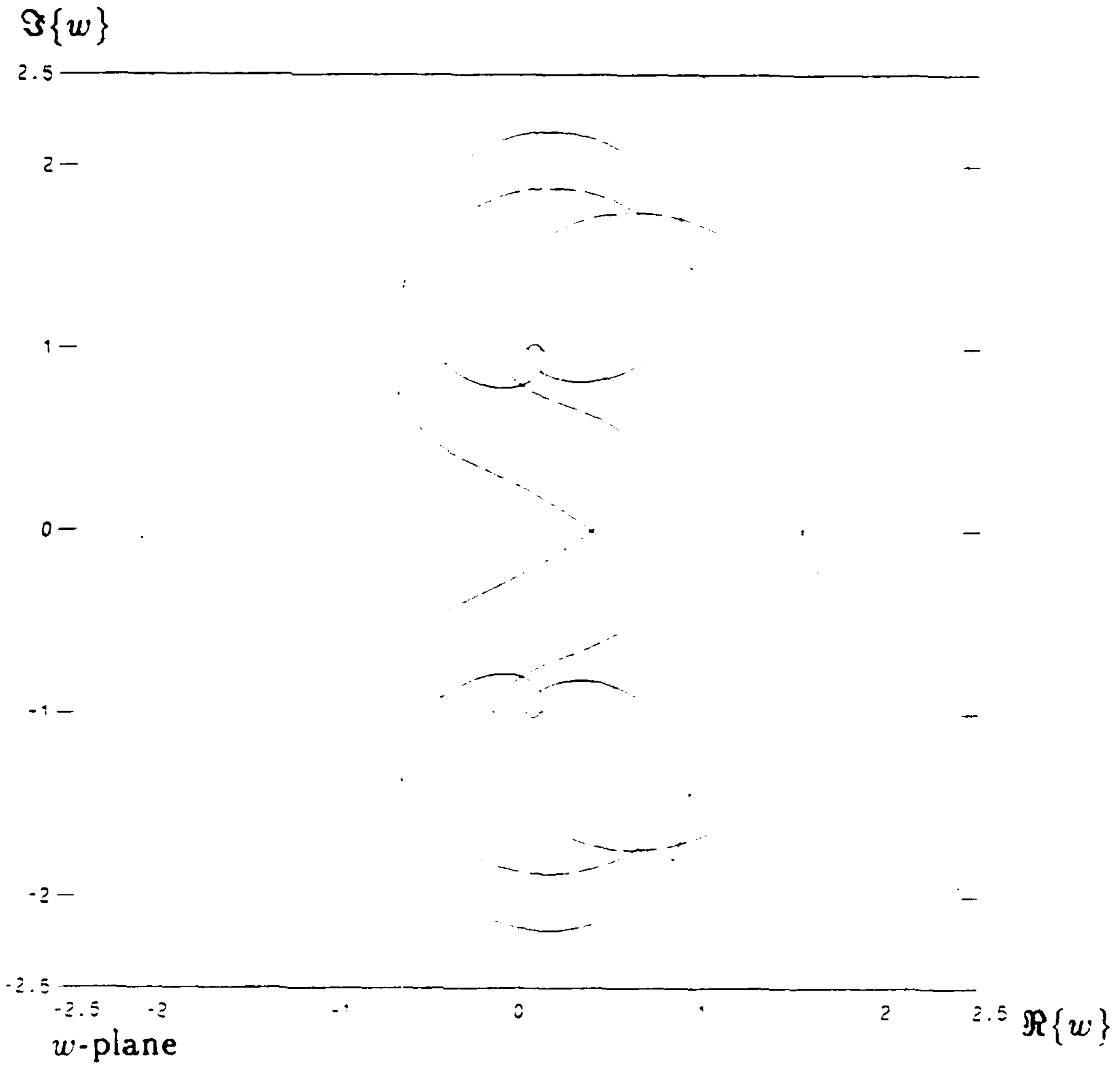


Figure 5.13: Zero paths of the polynomial associated with the autocorrelation function of the example. Zeros of solution polynomial are shown as solid, those corresponding to the mirror polynomial are dashed.

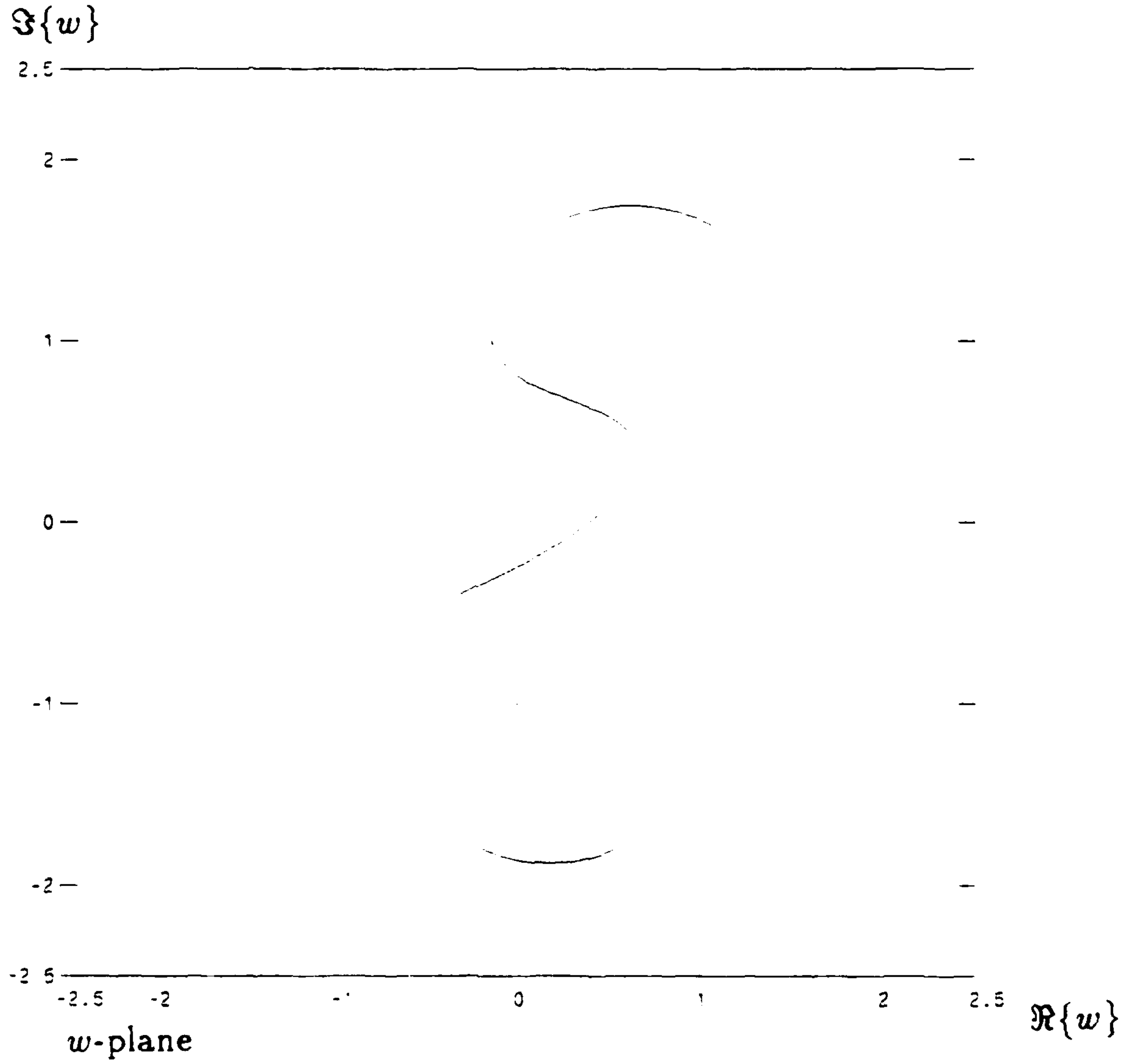


Figure 5.14: Single zero path of $p_r(w, z)$ calculated by algorithm.

a computer. In the section below we discuss these considerations and how they have been incorporated into the final algorithm.

The first consideration relates to the proper choice for a path $z(t)$. Recall that in the description of the algorithm, $z(t)$ is allowed to be arbitrary, as long as it is composed strictly of ordinary points of the algebraic function defined by the polynomial associated with the autocorrelation function, $p_r(w, z)$. Theoretically, $z(t)$ could be a short line segment in the z-plane. In this case, the resulting set of zero samples would be representative of a small region in the total set of zeros of this factor. When one then tries to solve the set of linear equations given by (5.22) to extract the coefficients of this factor, the equation coefficients will be almost the same, inviting numerical problems. The objective then is to generate a path $z(t)$ which is as varied as possible so that the resulting samples of the zeros of the factor polynomial reflect in an accurate manner the total set of zeros of this factor.

A second consideration concerns the evaluation of high degree polynomials. In evaluating $p_r(w, z)$, z has to be raised to a high power, namely the degree of $p_r(w, z)$ in z. For an N by N image, this degree is $2(N - 1)$. If the magnitude of z is large or small, this will lead to severe loss in numerical precision. This consideration then implies that the magnitude of $z(t)$ for all t should be in the proximity of one.

Although we would like to have the magnitude of $z(t)$ be close to unity, the unit circle itself has to be avoided. Consider a signal $x[m, n]$ with Fourier transform magnitude which becomes zero at $(exp(ju), exp(jv))$. Then $z = exp(ju)$ and $w = exp(jv)$ is a zero of *both* $p_x(w, z)$ and $\tilde{p}_x(w, z)$, i.e., $z = exp(ju)$ is a singular point. Thus, for some images there are singular points on the unit circle in the z-plane. An adequate strategy is to pick a path $z(t)$ where the magnitude of $z(t)$ is some number close to but not equal to one. Once $z(t)$ and the initial value w_0 have been specified, the resulting path $w(t)$ is completely determined. For this reason, it cannot be assured that the magnitude of each sample $w(t)$ will be close to unity even if the magnitude of $z(t)$ is close to unity. At first glance this seems like a severe problem, and in fact it can lead to numerical problems in some cases. However, in the great majority of cases we considered, this problem did not arise. There are two explanations for the well-behavedness of $w(t)$; first, $w(t)$ will become large only if $z(t)$ passes near the type of singular point where the algebraic function $w(z)$ becomes infinite. We have already shown that there are few such points. Similarly, one can show that $w(t)$ may become zero only if $z(t)$ crosses one of a finite number of points in the z-plane. As long as the $z(t)$ path does not come close to this set of points, we can expect that $w(t)$ will be well-behaved. Second, it is well known that the roots of large polynomials with arbitrary coefficients tend to cluster around the unit circle [41]. This behavior has indeed been observed in the numerical experiments.

With the above considerations in mind, the path shown in Figure 5.15 was chosen. The path $z(t)$ travels counterclockwise in a full circle of radius slightly larger than 1 (ranging from 1.1 to 1.2) and then travel clockwise in a circle of radius slightly less than 1 (ranging from .8 to .9). At this point, the path pair is "exchanged". Suppose that at the end of the depicted path, the zero pair is $(w_0, .9)$. At this point, a new path is begun using w as the independent variable. This path begins at w_0, goes to $w = 1.1$ and then travels in the same trajectory as $z(t)$, Figure 5.16. As long as the exchange occurs at an ordinary point of $z(w)$, and a path consisting of only ordinary

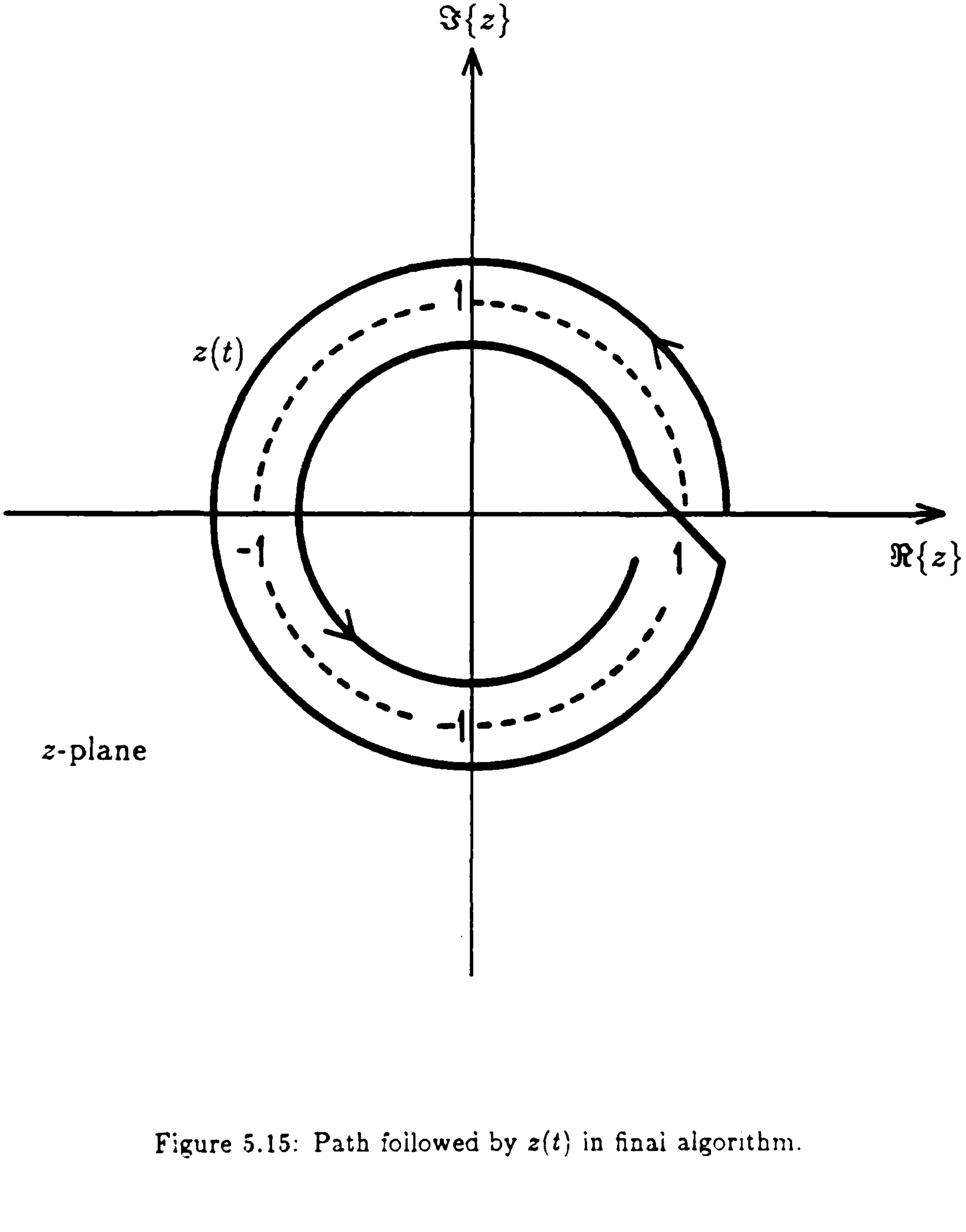
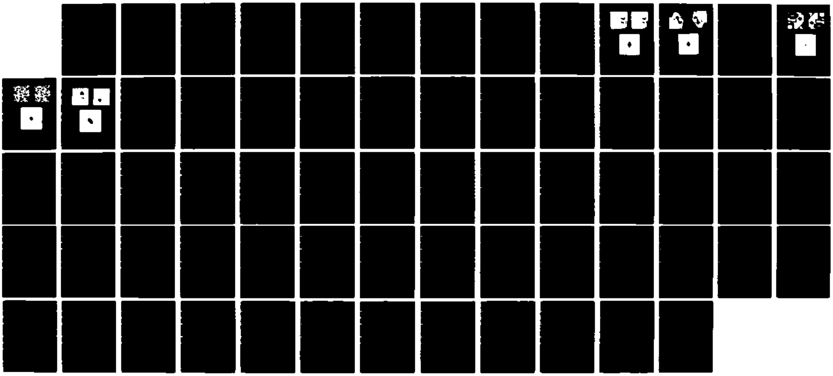
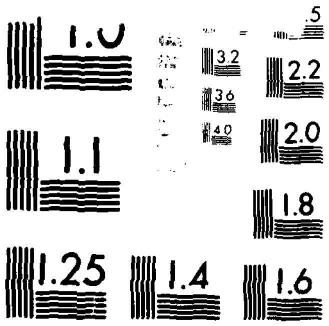


Figure 5.15: Path followed by $z(t)$ in final algorithm.

AD-A168 792 RECONSTRUCTION OF TWO-DIMENSIONAL SIGNALS FROM THE 2/2
FOURIER TRANSFORM MAGNITUDE(U) MASSACHUSETTS INST OF
TECH CAMBRIDGE RESEARCH LAB OF ELECTRON..
UNCLASSIFIED D IZRAELEVITZ JUN 86 TR-517 F/G 9/3 NL




1.0
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

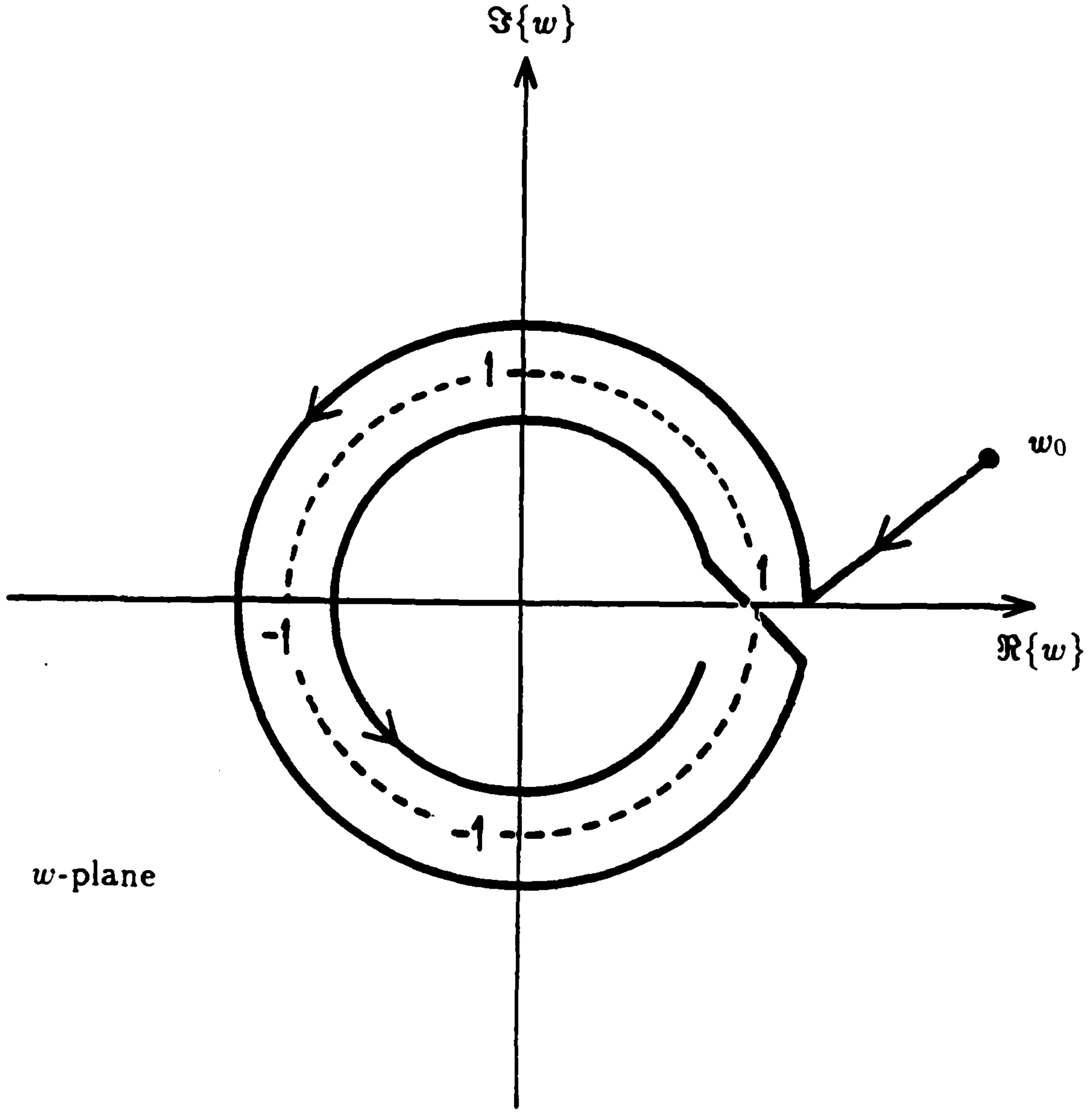


Figure 5.16: Path followed by $w(t)$ after variable swapping.

points is used, the resulting zeros of $p_r(w,z)$ calculated will continue to correspond to a single irreducible factor of $p_r(w,z)$. Intuitively, this "exchange" between w and z leads to an equal consideration of the vertical and horizontal orientations of the image and thus presents a more representative sampling of the zeros of one factor of $p_x(w,z)$ or $\tilde{p}_x(w,z)$. To demonstrate the complexity of the paths involved, in Figures 5.17 and 5.18 we show a portion of a typical path $z(t)$ and the resulting path $w(t)$, for a 20 by 20 image.

The strategy described in the previous paragraph was not successful in all trials. We have applied our algorithm to about 40 images of sizes 20 by 20 and larger, and have found that for this range of image size and one out of every ten images, the path picked was close enough to a singular point that the zeros calculated changed from one factor to another.

One way to avoid this is to calculate where all the singular points are via the discriminant polynomial discussed in Appendix B, and then generate a path which is far from any singular point. An alternate strategy is to keep monitoring the partial derivative of $p_r(w,z)$ with respect to w. If this derivative becomes zero, then the trajectory may be approaching a singular point. Then, the location of the singular point can be located and circumvented. We have not implemented either of these strategies which would have significantly increased the computational requirements of the program. Since this situation occurred so seldomly, a more efficient strategy is to restart the algorithm using a slightly different path. In all our experiments it was never found necessary to use more than two trial paths.

The next consideration, now that the choice of the path $z(t)$ has been discussed, is

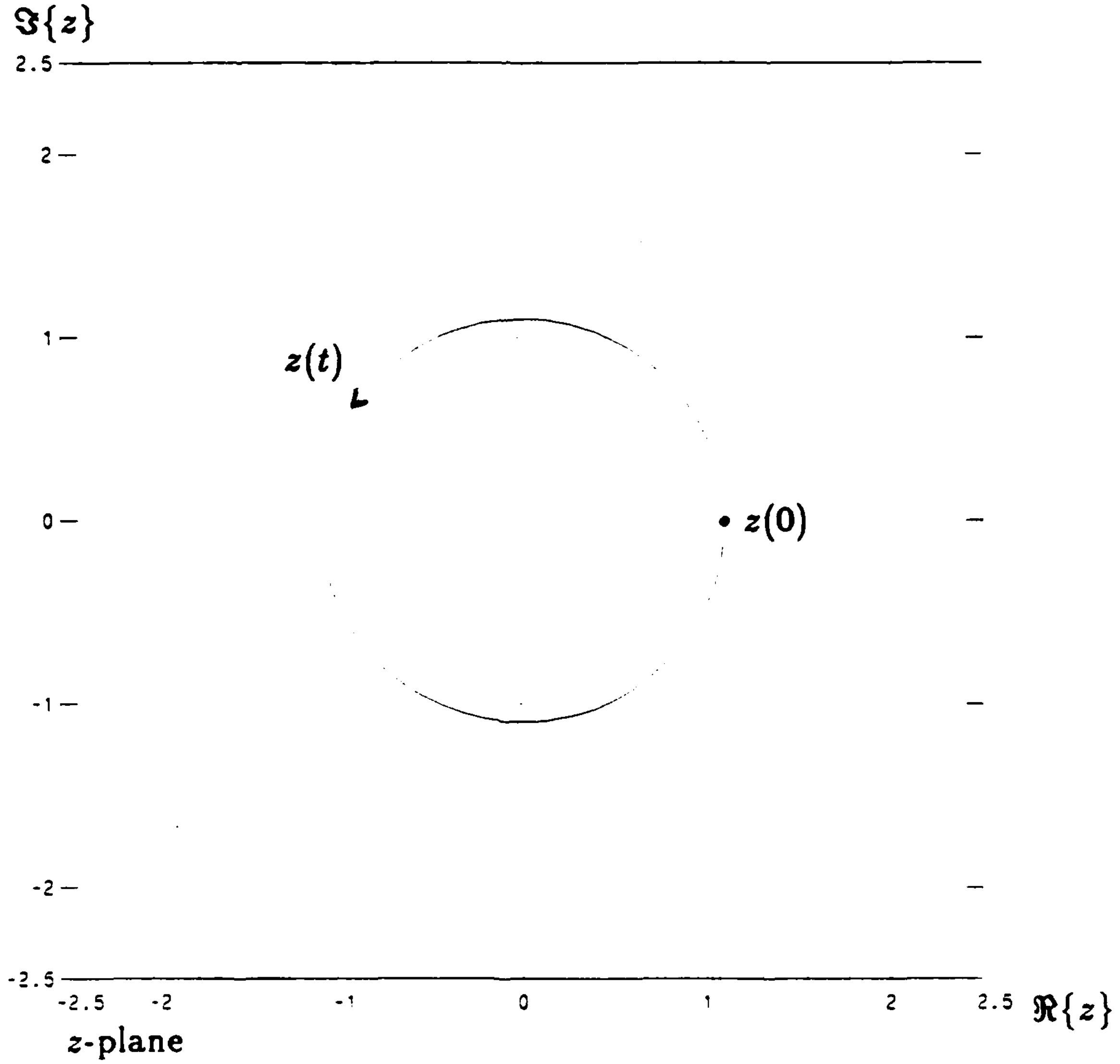


Figure 5.17: Portion of a path in the z-plane.

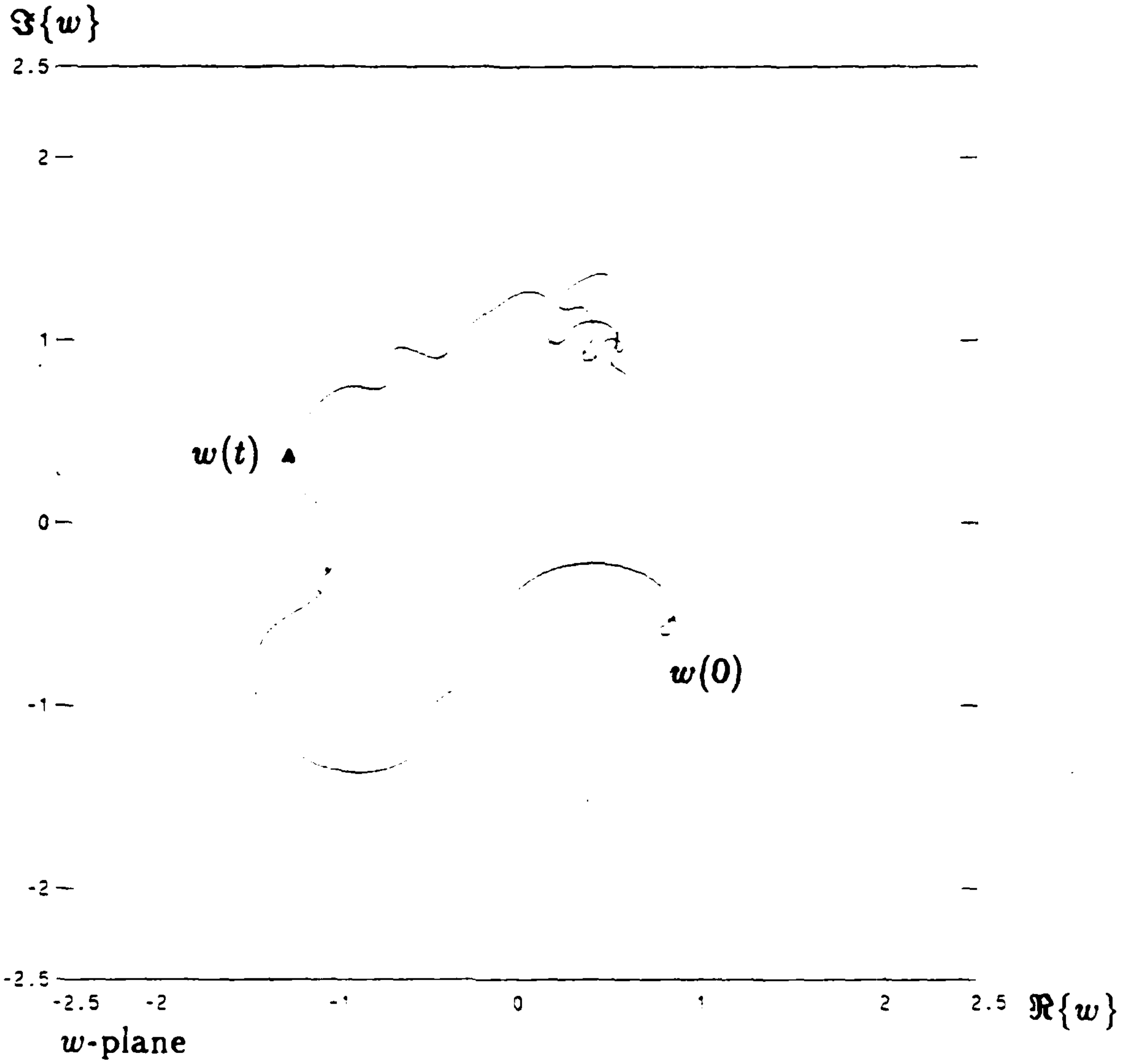


Figure 5.18: Typical path in the w-plane for 20 by 20 image.

the solution of the differential equation,

$$\frac{dw(t)}{dt} = -\frac{p_{r,z}(w,z)}{p_{r,w}(w,z)}\frac{dz(t)}{dt} \tag{5.31}$$

This equation is converted into a system of two simultaneous ordinary differential equations by considering the real and imaginary parts separately. Equation (5.31) then becomes a (nonlinear) vector initial value problem with a governing differential equation of first order. This problem has been studied extensively and several algorithms have been developed which use variable step size techniques to achieve a pre-specified accuracy [42]. In practice, we have found it more computationally efficient to use the differential equation solver to find a few digits of accuracy for w given z and then use the following version of Newton's method to calculate the zero of $p_r(w,z)$ accurately. Suppose that an estimate of a zero of $p_r(w,z)$ for a given z has been found, $\hat{w}_0$. This estimate can then be improved via the iteration,

$$\hat{w}_{i+1} = \hat{w}_i - \frac{p_r(\hat{w}_i, z)}{p_{r,w}(\hat{w}_i, z)} \tag{5.32}$$

One last issue to be resolved is the proper computation of one solution to the homogeneous set of equations (5.22). Recall that the associated matrix , say M, has rank deficiency of 1, i.e., it has a unique (to a constant) non-trivial null vector. The approach we have taken is to use the QR algorithm with column pivoting [43]. This algorithm calculates the following factorization of an m by $n \leq m$ matrix M of rank $n-1$:

$$M = QR\Pi \tag{5.33}$$

where Q is an m by m orthogonal matrix, R is of the form

$$R = \begin{array}{cc} \begin{bmatrix} R_{1,1} & \mathbf{r} \\ \mathbf{0} & 0 \end{bmatrix} & \begin{array}{c} n-1 \\ 1 \end{array} \\ \begin{array}{cc} n-1 & 1 \end{array} & \end{array} \tag{5.34}$$

where $R_{1,1}$ is non-singular upper triangular, $\mathbf{r}$ is a column vector, $\mathbf{0}$ is a row vector consisting of all zeros, and Π is a permutation matrix. Once we have the factorization above, we can find a solution to $Mx = 0$ as

$$y = -R_{1,1}^{-1}\mathbf{r} \tag{5.35}$$

$$x = \Pi^{\top} \begin{bmatrix} y \\ 1 \end{bmatrix} \tag{5.36}$$

Chapter 6

Numerical Results

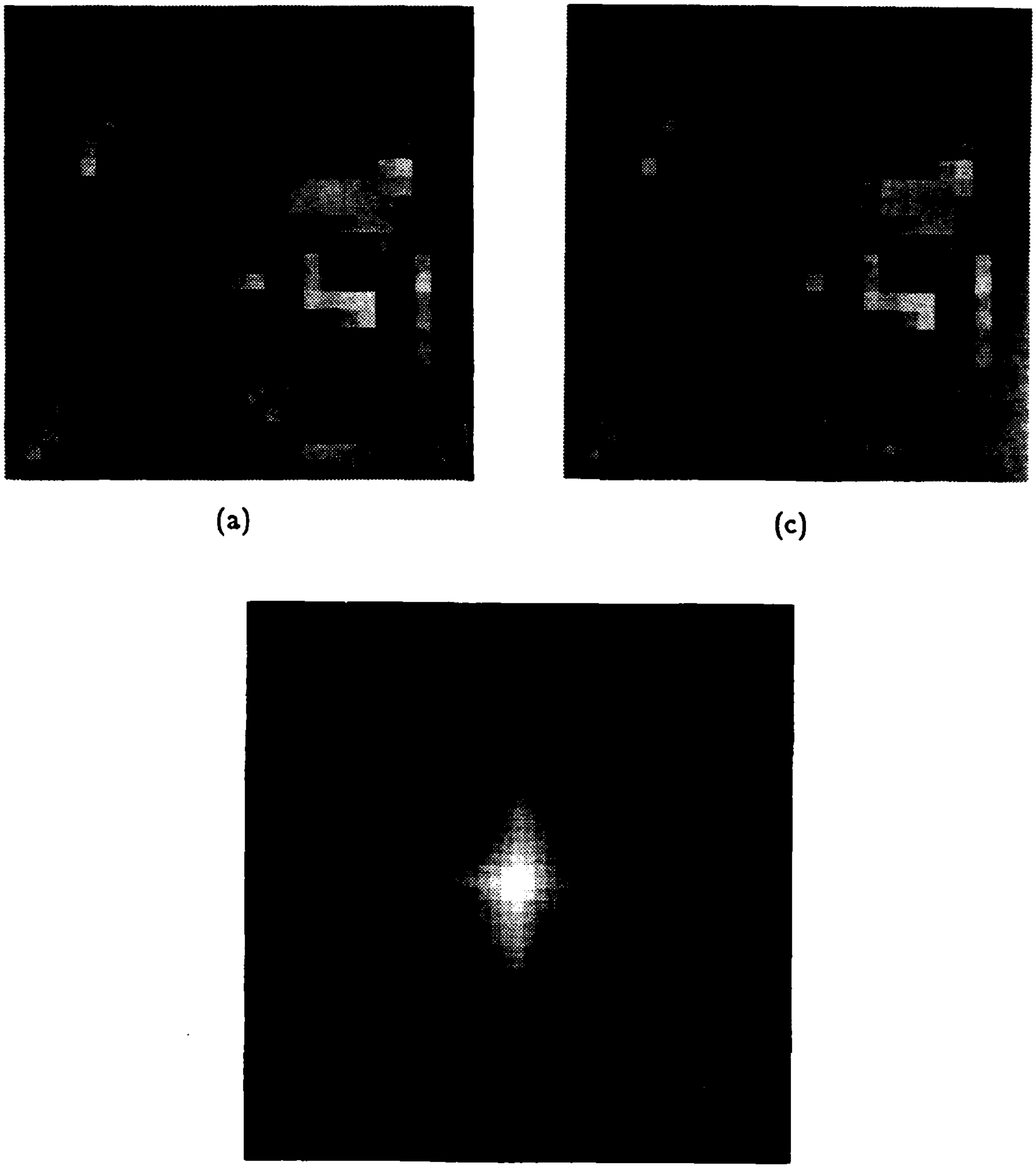
In this chapter we illustrate the performance of the new phase retrieval algorithm by applying it to several families of images. We have applied this algorithm to about 40 different realistic images of size 20 by 20 and above but we will just be showing some representative examples.

We follow these examples by a study of the effect of noisy observations on the reconstruction algorithm and a comparison of the computational requirements of the new algorithm with the Deighton algorithm.

6.1 Examples of Application of Factorization Algorithm to Phase Retrieval

In this section we present several examples of the application of the phase retrieval algorithm to several different types of images. The experimental procedure is as follows. The autocorrelation function of the original image is first calculated and the polynomial associated with this autocorrelation function formed. The phase retrieval algorithm is then used to factor this polynomial. The coefficients of the irreducible factor extracted then correspond to the pixel values of the reconstructed image. In every case it is assumed that the support of the image, or rather, the size of the smallest rectangle surrounding the support of the original image, is known. In order to emphasize the fact that we are actually reconstructing an image from its autocorrelation function, we display the original image, its autocorrelation function, and the reconstruction. In all examples, it is assumed that the unknown image corresponds to a set of intensities which must be non-negative. This requirement is certainly not necessary for our algorithm to be applicable; it only removes the sign ambiguity.

In Figure 6.1 we applied the phase retrieval algorithm to reconstruct a 25 by 25 pixel image. This size image is the largest that could be accommodated in our Vax/11-750 computer system storage capability without excessive memory paging. In this case, the reconstruction has the same orientation as the original. In the second example, another 25 by 25 pixel image, Figure 6.2 was reconstructed. In this example, the resulting reconstruction is rotated by 180 degrees. As mentioned earlier since for the case of rectangular support, an image and its 180 degree rotated version have the same autocorrelation function, it is impossible to restore the absolute orientation of the image. Figures 6.3 and 6.4 show the result of applying the reconstruction algorithm to different families of square images. Figure 6.3 is a high contrast image of size 20 by 20 pixels which may be representative of an application in astronomy. Note that in this case, the support of the image is not assumed known, since it is not rectangular; however, it is assumed that the smallest rectangle surrounding the support is known. The image in Figure 6.4 consists of a 20 by 20 image composed of independent random noise uniformly distributed between 0 and 1. This example is especially interesting since the autocorrelation function of images with random coefficients are very similar.



(a)

(c)

(b)

Figure 6.1: Girl image. (a) Original Image, (b) Autocorrelation function, (c) Reconstruction.

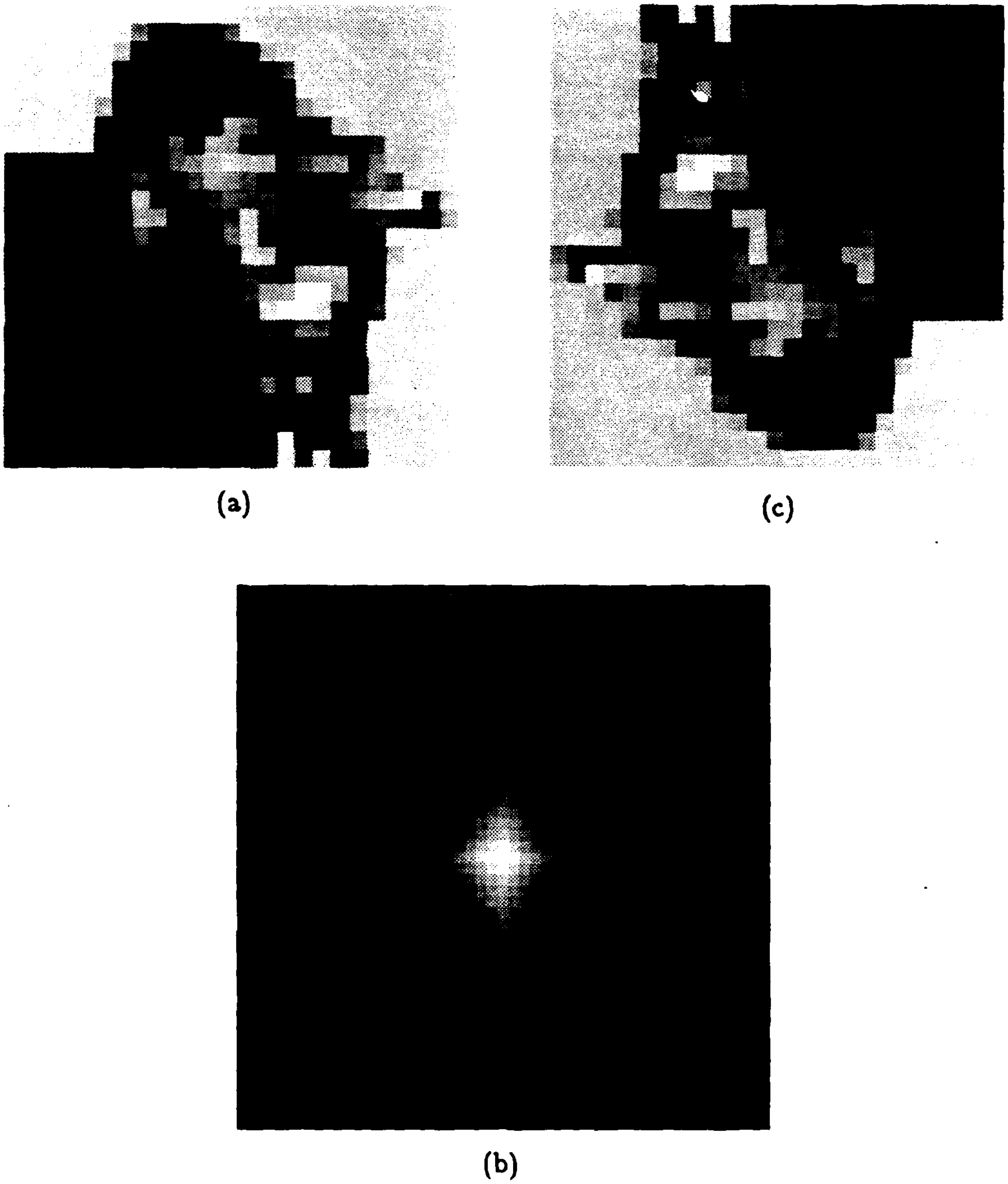


Figure 6.2: C-Man image. (a) Original Image, (b) Autocorrelation function, (c) Reconstruction.

This is due to the fact that most of the autocorrelation function content is concentrated at $r[0,0]$. This is also the family of objects for which the iterative algorithms performed very poorly.

If the support of the image is non-symmetric so that the image rotated by 180 degrees would not "fit" in the known support constraint, then the rotation ambiguity is removed. However, the algorithm described in this thesis does not use this information. Implicit in our development is that extracting either $p_z(w,z)$ or $\tilde{p}_z(w,z)$ is adequate. In the situation of a triangular support, $\tilde{p}_z(w,z)$ is associated with a signal which does not satisfy the support constraint. However, our algorithm is just as likely to return such an invalid signal. For example, in Figure 6.5 we show the result of applying our algorithm to a 20 by 20 pixel image with triangular support. The reconstruction (by chance) has the wrong support, since it is rotated by 180 degrees. This effect however, is of minor concern since it is a trivial task to extract the correct reconstruction from such a rotated version. This is a second example where only a bound on the support is assumed known, rather than the exact bound.

6.2 Effect of Noise on Reconstruction

In an effort to characterize the susceptibility of the algorithm to errors in the known data, we studied the effect of adding independent gaussian noise to the known autocorrelation function of a real image, and then applying the phase retrieval algorithm to this noisy autocorrelation function. Adding noise in this domain is equivalent in some sense to adding noise to the Fourier transform magnitude squared. Since most applications of phase retrieval measure either the autocorrelation function or the Fourier transform

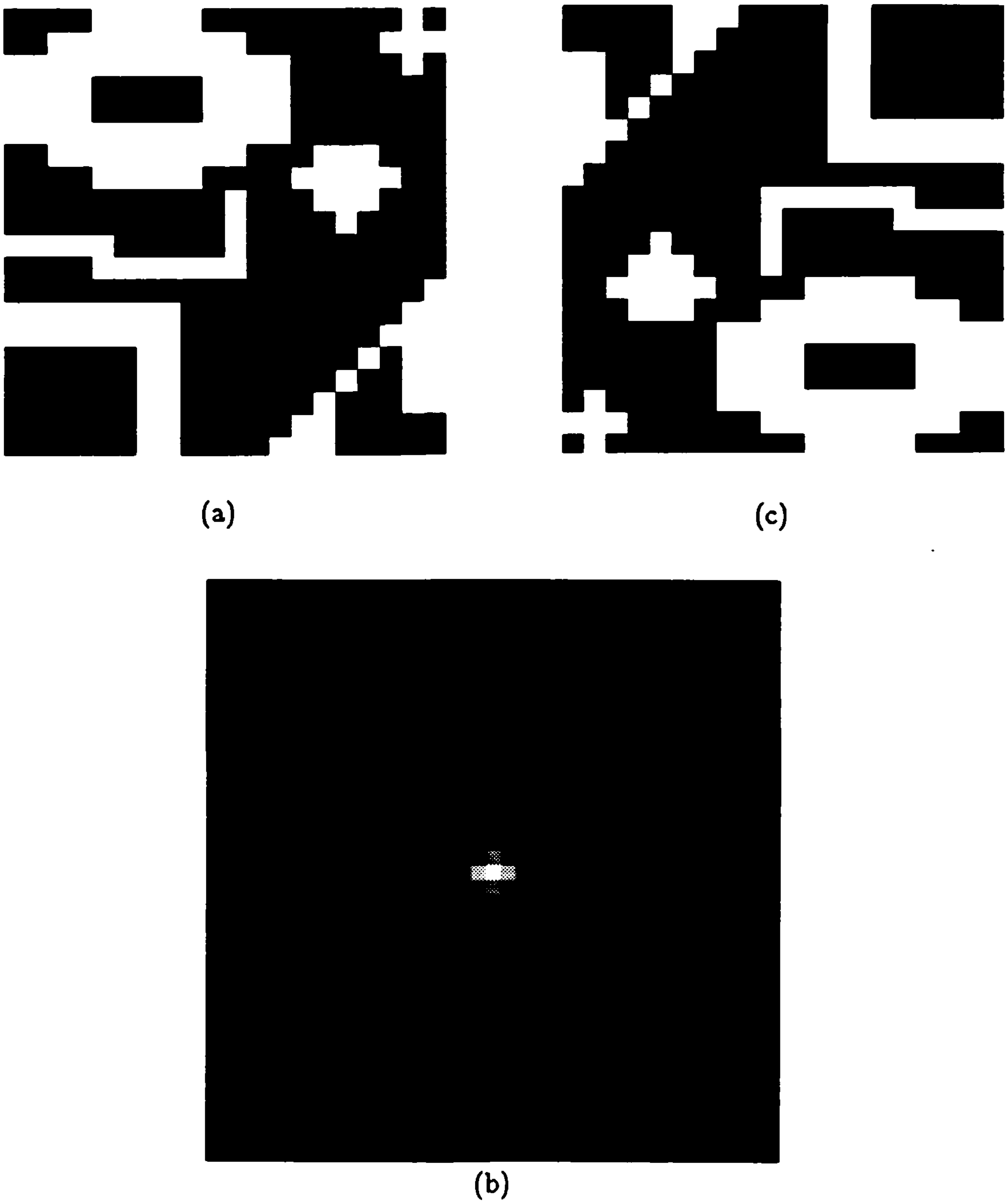


Figure 6.3: Astronomical object. (a) Original Image, (b) Autocorrelation function, (c) Reconstruction.

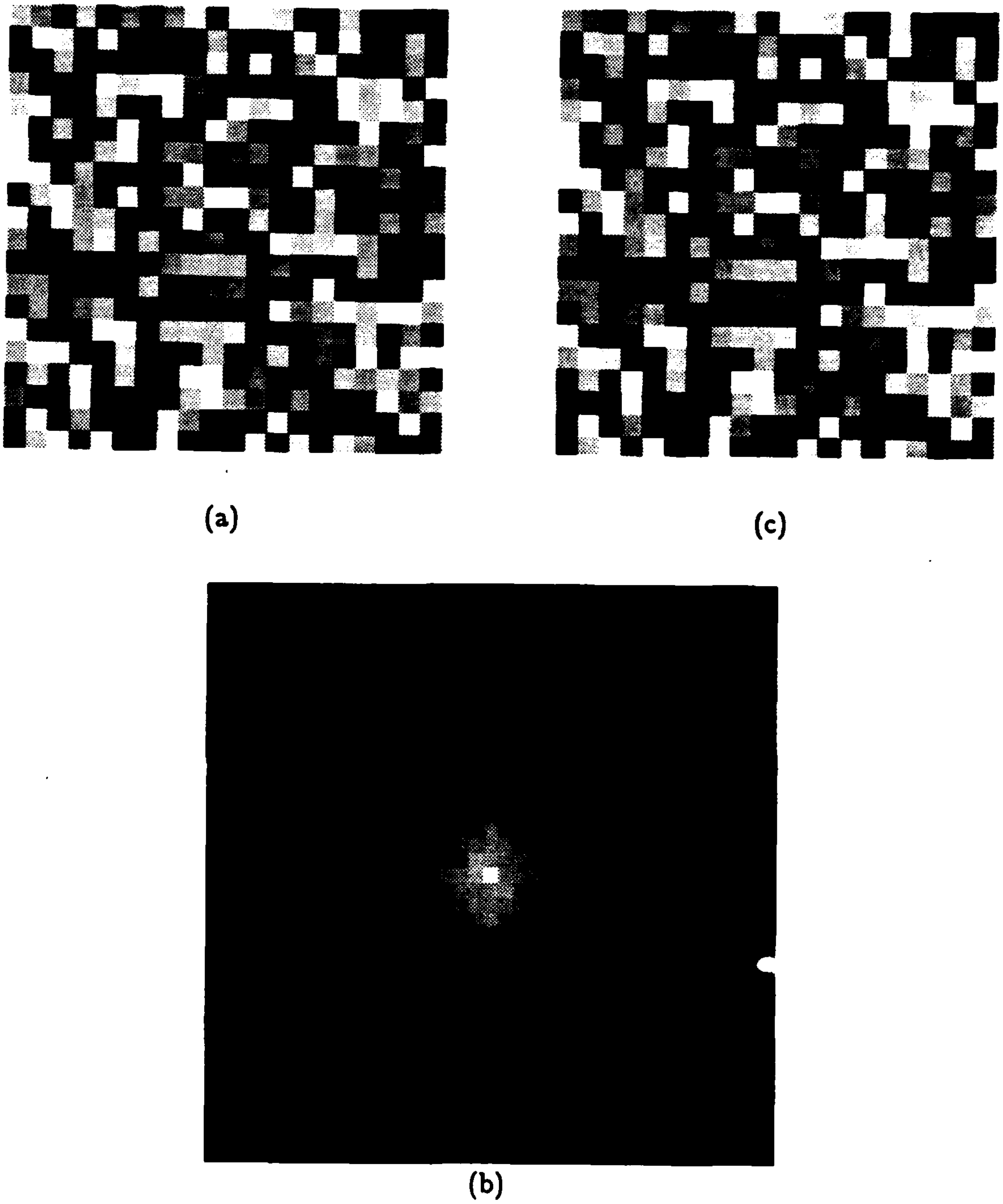


Figure 6.4: Noise image. (a) Original Image, (b) Autocorrelation function, (c) Reconstruction.

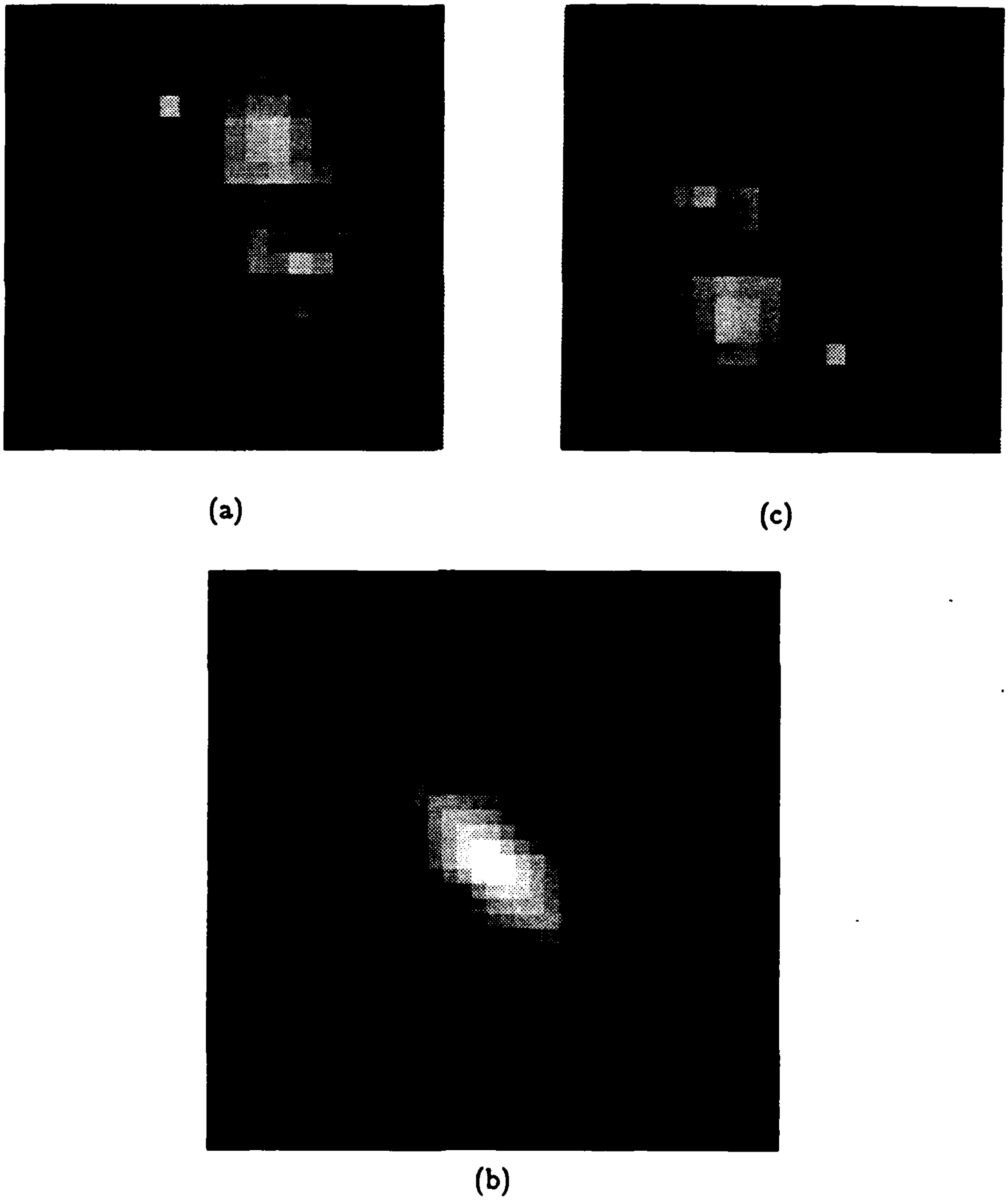


Figure 6.5: (a) Original Image, (b) Autocorrelation function, (c) Reconstruction

magnitude squared, it was felt that this form of degradation is more pertinent than adding noise to the Fourier transform magnitude directly.

Consider the autocorrelation function $r[m,n]$ of an unknown image $x[m,n]$. A noisy version of $r[m,n]$ was calculated, $\hat{r}[m,n]$ via,

$$\hat{r}[m,n] = r[m,n] + w[m,n] \tag{6.1}$$

The signal $w[m,n]$ consisted of simulated gaussian random noise of variance σ_i^2 with the added condition that $w[-m,-n] = w[m,n]$. Thus, it is assumed that only non-redundant measurements of the autocorrelation function were made.

The resulting autocorrelation function $\hat{r}[m,n]$ is thus symmetric about the origin. Thus, this degradation assumes that the signal is known to be real and furthermore, that the support of the image, i.e., the support of the true autocorrelation function, is known exactly. Given a correlation signal to noise value, SNR_i, the variance of the added noise was calculated as

$$\sigma_i^2 = \frac{\sum_{m,n} r[m,n]^2}{SNR_i} \tag{6.2}$$

The phase retrieval algorithm was then applied using this noisy autocorrelation function. The reconstruction signal to noise ratio, SNR_o, was calculated as,

$$SNR_o = \frac{\sum_{m,n}(x[m,n] - \hat{x}[m,n])^2}{\sum_{m,n} x[m,n]^2} \tag{6.3}$$

where $x[m,n]$ is the original image and $\hat{x}[m,n]$ is the reconstructed image. The SNR_o was maximized over a rotation of the reconstruction by 180 degrees.

The experiments were performed for two image sizes, 20 by 20, and 16 by 16. For each image size, two images were used, denoted here as "1" and "2", and two sets of pseudo-random noise fields, "A" and "B". The magnitude of the noise was changed until SNR_o defined in (6.3) was in the range of 10. The results of these experiments are displayed in Figures 6.6 and 6.7.

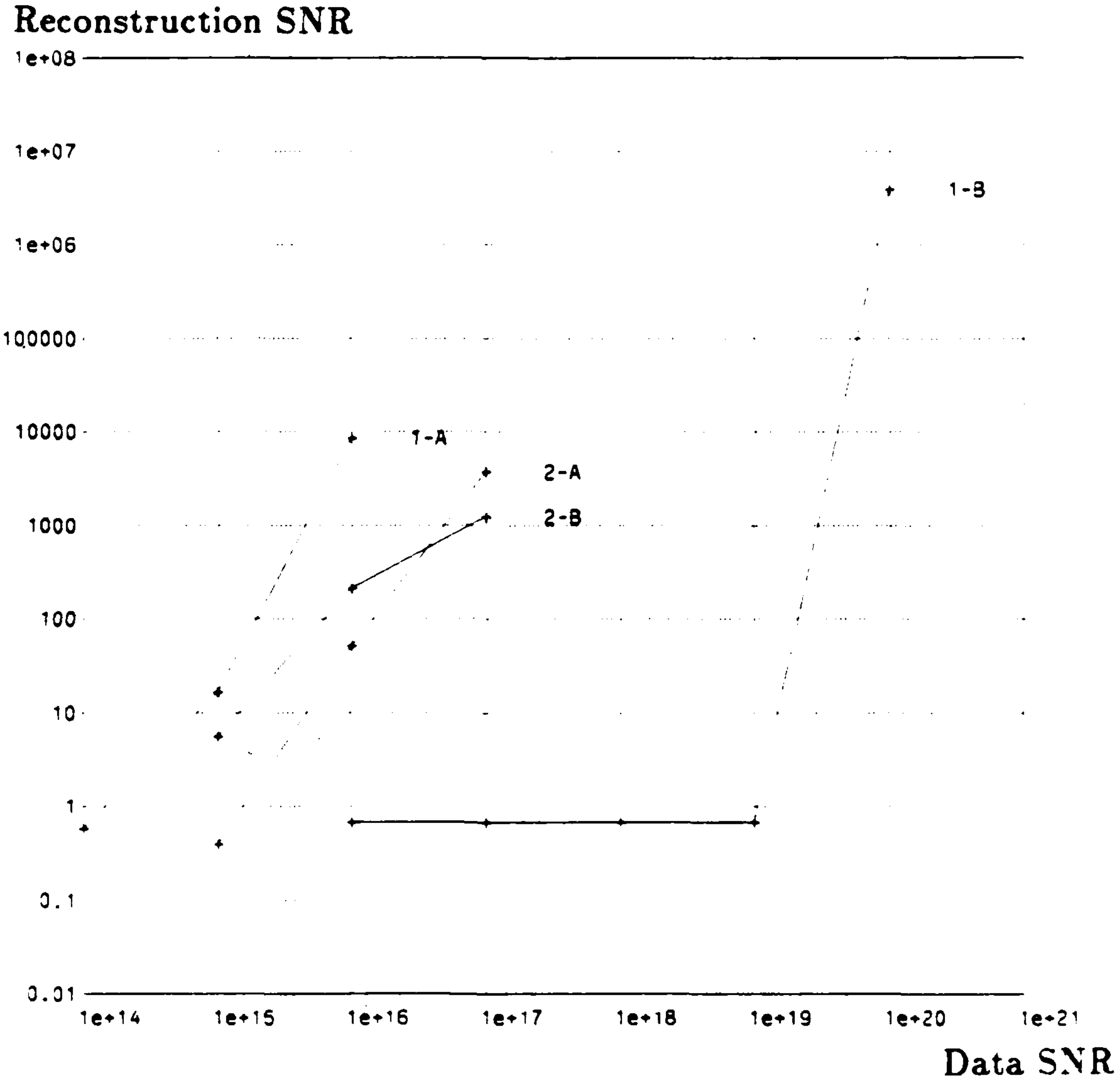


Figure 6.6: Effect of noise on reconstruction for 16 by 16 pixel images. Graphs shown correspond to the application of noise signals "A" and "B" to images "1" and"2".

We note that the algorithm is very sensitive to noise. It is difficult to specify the sensitivity of the algorithm since it varies dramatically between images and noise

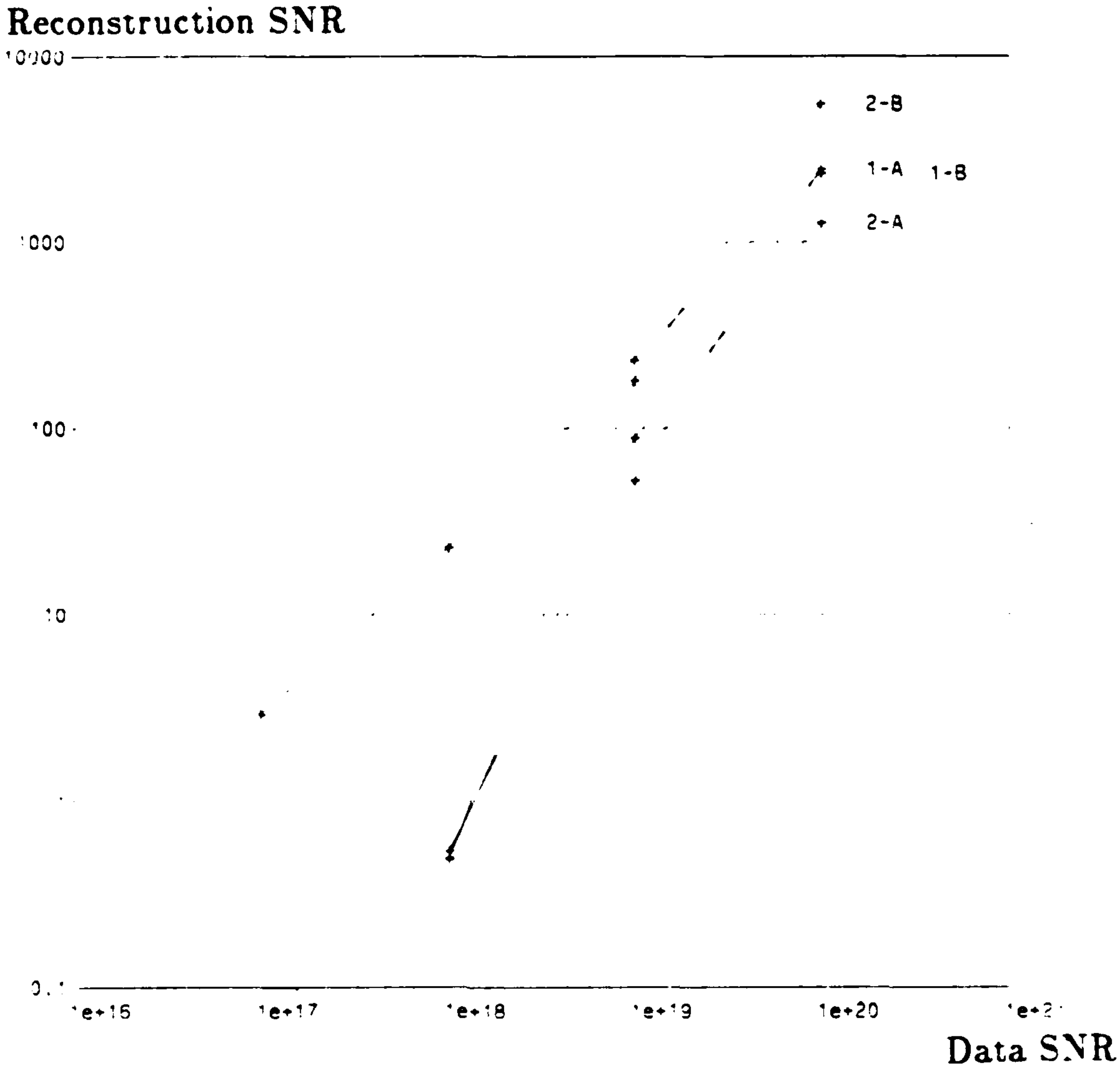


Figure 6.7: Effect of noise on reconstruction for 20 by 20 pixel images. Graphs shown correspond to the application of noise signals "A" and "B" to images "1" and"2".

signals, (see for example 1-A and 1-B in Figure 6.6). However, it seems that for 16 by 16 images, a SNR_i of better than 10^{16} is required for faithful reconstruction. For 20 by 20 images, a SNR_i of 10^{19} seems to be required for most images.

The sensitivity of the algorithm to small amounts of noise can be explained with the help of the discussion in Section 3.2.1 regarding reducible and irreducible polynomials. A small perturbation to the true associated polynomial $p_r(w, z)$ will result in an irreducible polynomial. As the perturbation increases, $p_r(w, z)$ will no longer be even "approximately" reducible. This will result in difficulty in finding a good approximation to the factors of the original factorable polynomial $p_r(w, z)$. A more formal argument supporting this observation is presented in [44].

6.3 Computational Requirements

In order to compare the computational requirements of the new algorithm with Deighton's algorithm, we measured the number of CPU seconds which are required to reconstruct a typical image using the new algorithm. The solid plot in Figure 6.8 is a display of the number of CPU seconds required per reconstruction plotted against image size. We see that for small images the Deighton algorithm is much more efficient. However, for image sizes larger than about 11, the Deighton algorithm becomes computationally more expensive.

We have found experimentally that as the image size increases, the QR decomposition becomes a dominant part of the computation in the new algorithm. This leads us to hypothesize that the algorithm asymptotically requires computations in the order of

N^6 for reconstructing an N by N image. In contrast, the Deighton algorithm requires computations which grow exponentially with N. A more efficient implementation of either the Deighton or our algorithm would certainly affect the point at which our algorithm performs better than the Deighton algorithm. However, the important thing to note is the rate of growth of each algorithm rather than the actual computer usage.

We can also compare the storage requirements of the Deighton algorithm against the present algorithm. For the Deighton algorithm, the lists which need to be compared have sizes which depend exponentially on image size. In the case of the new algorithm, the storage requirements are set predominantly by the set of linear equations to be solved. The storage requirements for this matrix are on the order of $2N^4$ for an N by N image. As the image size increases, the storage requirements for the Deighton algorithm increase much faster than for the new algorithm.

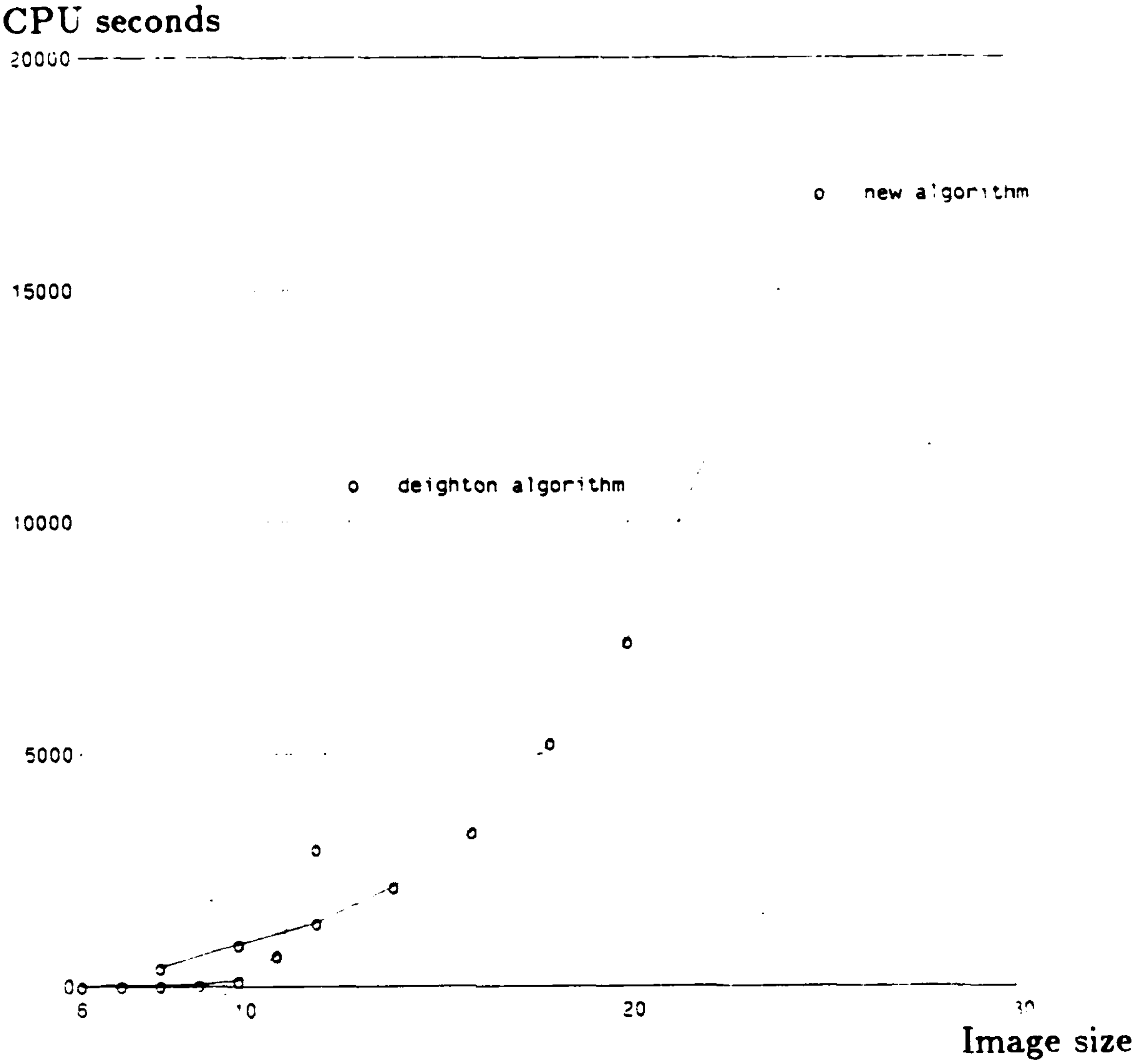


Figure 6.8: Comparison of CPU usage for Deighton vs. new algorithm.

Chapter 7

Other Applications

There are other applications which may benefit from the ideas developed in this thesis. We discuss here two such situations. The first involves the general problem of factorization of polynomials in two variables, while the second addresses two-dimensional recursive filter stability testing.

7.1 Application to General Bivariate Polynomial Factorization

As an example of where such an algorithm may be applicable, consider a two-dimensional linear shift-invariant system which is specified by the difference equation [45],

$$\sum_k \sum_l b_{kl} y[m-k, n-l] = \sum_k \sum_l a_{kl} x[m-k, n-l] \qquad (7.1)$$

where $y[m,n]$ is the system response to an excitation $x[m,n]$. The z-transform of the response of this system to a unit impulse can be expressed as,

$$H(w,z) = \frac{\sum_k \sum_l a_{kl} w^{-k} z^{-l}}{\sum_k \sum_l b_{kl} w^{-k} z^{-l}} = \frac{A(w,z)}{B(w,z)} \tag{7.2}$$

We see that $A(w,z)$ and $B(w,z)$ can be expressed as two bivariate polynomials in the variables w^{-1}, z^{-1}. In characterizing $H(w,z)$ one has to answer the question of whether the representation given in (7.2) is minimal, i.e., whether there are some polynomials $A'(w,z)$ and $B'(w,z)$ of total degree less than $A(w,z)$ and $B(w,z)$ respectively which satisfy,

$$H(w,z) = \frac{A'(w,z)}{B'(w,z)} \tag{7.3}$$

This is equivalent to asking whether $A(w,z)$ and $B(w,z)$ are such that

$$A(w,z) = P(w,z)A'(w,z) \tag{7.4}$$

$$B(w,z) = P(w,z)B'(w,z) \tag{7.5}$$

where $P(w,z)$ is a non-constant polynomial. In other words, one may ask whether $A(w,z)$ and $B(w,z)$ are relatively prime (coprime). Several algorithms have been developed to test whether two polynomials are relatively prime [46,47]. However, these tests are very complicated for large polynomials, since effectively, they generate the resultant $R_{AB}(z)$ described in Appendix B.

A necessary condition for the required $P(w,z)$ to exist is that $A(w,z)$ and $B(w,z)$ be factorable. Thus, by using a polynomial bivariate factorization algorithm, we can check whether coprimeness is satisfied. If either $A(w,z)$ or $B(w,z)$ is irreducible, then there is no need to use the more complicated coprimeness algorithms.

There are some modifications required from the algorithm presented in Chapter 5. These modifications are due to the fact that, unlike the phase retrieval problem, one does not know a priori the degree of the factor isolated by the zero-tracking procedure. Thus, although one can set a bound on the maximum number of homogeneous equations which need to be solved, one does not know a priori the number of unknowns, i.e., the number of coefficients in the polynomial to be isolated. In Appendix C, we discuss two methods for estimating the total degree of the isolated factor. Here we demonstrate the use of the second algorithm discussed in Appendix C.

To illustrate the performance of our factorization algorithm, we have generated a test polynomial $p(w,z)$ given by,

$$\begin{array}{cl} & \overset{w}{\longrightarrow} \\ z\downarrow & \begin{array}{cccccc} 2 & 8 & 13 & 23 & 30 & 12 \\ 4 & 13 & 18 & 38 & 48 & 17 \\ 3 & 12 & 23 & 48 & 55 & 16 \\ 3 & 14 & 25 & 44 & 40 & 10 \\ 2 & 8 & 17 & 25 & 19 & 4 \\ 1 & 5 & 7 & 10 & 6 & 1 \end{array} \end{array} \tag{7.6}$$

which can be expressed as the product of the following two polynomials,

$$\begin{array}{cl} & \overset{w}{\longrightarrow} \\ z\downarrow & \begin{array}{ccc} 2 & 2 & 1 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{array} \end{array} \tag{7.7}$$

and

$$\begin{array}{cc} & \overset{w}{\longrightarrow} \\ z\downarrow & \begin{array}{cccc} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 4 & 3 & 2 & 1 \end{array} \end{array} \tag{7.8}$$

In this example, the total degree of $p(w,z)$ is 10. From Bezout's theorem, we know that any other polynomial of total degree less than or equal to 10 which is prime relative to $p(w,z)$ can have at most 100 zeros in common with $p(w,z)$. Thus we need to calculate at most 101 zeros along a path pair $(w(t), z(t))$ consisting of ordinary points. This way we are assured that these zeros correspond to a single irreducible factor of $p(w,z)$. The following matrix is then generated,

$$M = \begin{bmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^{10} & z_1w_1 & \cdots & z_1w_1^9 & \cdots & w_1^{10} \\ 1 & z_2 & z_2^2 & \cdots & z_2^{10} & z_2w_2 & \cdots & z_2w_2^9 & \cdots & w_2^{10} \\ & & & & & & & & & \\ 1 & z_{101} & z_{101}^2 & \cdots & z_{101}^{10} & z_{101}w_{101} & \cdots & z_{101}w_{101}^9 & \cdots & w_{101}^{10} \end{bmatrix} \quad (101 \text{ by } 66) \tag{7.9}$$

Note that although the degree of $p(w,z)$ is known, we need to assume a triangular support for the coefficients of $p(w,z)$. This is because we can only bound the *total degree* of the factor of $p(w,z)$, not necessarily its degree in each variable. Thus although $p(w,z)$ has 36 non-zero coefficients, M has 66 columns. According to Appendix C, M will have a rank deficiency, i.e., $66 - r(M)$, equal to

$$(10 - totdeg(d) + 1)(10 - totdeg(d) + 2)/2 \tag{7.10}$$

where $d(w,z)$ is the irreducible factor of $p(w,z)$ isolated by our procedure. In order to calculate the rank of M, we performed a QR decomposition of M; Table 7.1 shows some of the diagonal elements of the triangular matrix R.

diagonal element	value
48	0.00534025
49	0.00507684
50	-0.00145097
51	-0.00059735
52	4.45087e-15
53	-1.2814e-14
54	8.82197e-15
55	5.67471e-15

Table 7.1: Diagonal elements of triangular matrix after QR decomposition for calculating total degree of factor.

From this table we see that the rank of M is 51. Therefore, the total degree of the factor $d(w, z)$ isolated from $p(w, z)$ is 6. Once the total degree of $d(w, z)$ has been determined, we can regenerate M with fewer columns,

$$M = \begin{bmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^6 & z_1 w_1 & \cdots & z_1 w_1^3 & \cdots & w_1^6 \\ 1 & z_2 & z_2^2 & \cdots & z_2^6 & z_2 w_2 & \cdots & z_2 w_2^3 & \cdots & w_2^6 \\ & & & & & & & & & \\ 1 & z_{101} & z_{101}^2 & \cdots & z_{101}^6 & z_{101} w_{101} & \cdots & z_{101} w_{101}^3 & \cdots & w_{101}^6 \end{bmatrix} \quad (101 \text{ by } 15) \tag{7.11}$$

A new QR decomposition of this matrix yields a single non-trivial null vector (to a constant) which can then be associated with the coefficients of $d(w, z)$, given below,

$$\begin{array}{cccccccc} & \overset{w}{\longrightarrow} & & & & & & \\ z \downarrow & -.25 & -.25 & 2.6e-16 & -.25 & -2.3e-17 & -8.3e-17 & -3.0e-17 \\ & -.25 & 2.1e-16 & -.25 & -.25 & 5.9e-16 & 1.3e-16 & \\ & -.5 & -.5 & -.5 & -.5 & -9.7e-17 & & \\ & -1 & -.75 & -.5 & -.25 & & & \\ & -2.1e-16 & 5.7e-16 & 4.0e-16 & & & & \\ & -2.8e-16 & -2.1e-16 & & & & & \\ & 2.9e-16 & & & & & & \end{array} \tag{7.12}$$

We see that to a scale constant, the retrieved factor is equal to one of the generating polynomials.

The second example below shows how the algorithm can be used to test irreducibility.

For this example we consider the polynomial,

$$\begin{array}{cccccc} & \overset{w}{\longrightarrow} & & & & \\ z\downarrow & 1 & 0 & 1 & 0 & 1 \\ & 0 & 1 & 0 & 1 & 0 \\ & 1 & 0 & 1 & 0 & 1 \\ & 0 & 1 & 0 & 1 & 0 \\ & 1 & 0 & 1 & 0 & 1 \end{array} \tag{7.13}$$

In this case the total degree of this polynomial is 8, so a 65 by 45 matrix M of the form given by (7.9) is generated. Table 7.2 shows the last 5 diagonal elements of the triangular matrix calculated via a QR decomposition of M for this case.

diagonal element	value
41	-0.119653
42	-0.11235
43	0.110724
44	0.0845307
45	4.17397e-13

Table 7.2: Diagonal elements of triangular matrix after QR decomposition for irreducibility test.

Since the rank deficiency of M is seen to be 1 in this case, we conclude that the polynomial shown in (7.13) is irreducible.

7.2 Application of Root-tracking to Two-Dimensional Recursive Filter Stability Testing

A further application of the ideas presented in this thesis lies in the field of stability theory of two-dimensional recursive filters. It is well known [45] that the question of whether a filter with rational z-transform transfer function,

$$H(w,z) = \frac{A(w,z)}{B(w,z)} \tag{7.14}$$

is stable or not depends on the location of the zeros of the denominator polynomial $B(w,z)$ [1]. Specifically, it has been shown that if all branches of $w(z)$ for $z = exp(jv)$, called root maps in the literature, are of magnitude less than one, then the filter described in (7.14) is bounded-input, bounded-output stable. The most difficult part of this test is showing that the branches of $w(z)$ do not cross the unit circle in the w-plane, i.e., that $A(exp(ju), exp(jv))$ is never zero.

A computationally attractive algorithm for testing for filter stability relies on the fact that the two-dimensional unwrapped phase must be continuous and periodic if all root maps are always less than one in magnitude [48]. This procedure is fast and efficient, as long as all the root maps are far from the unit circle. However, as the root maps get close to the unit circle, the phase becomes almost discontinuous, rendering a phase unwrapping algorithm helpless.

At this point it becomes more convenient to look at the root maps themselves, rather than consider their effect on the phase. Shaw [48] proposes detecting when the unwrapped phase is almost discontinuous and then doing a constrained search for a minimum of the magnitude of $A(exp(ju), exp(jv))$. However, for complicated filters, the magnitude function is likely to have local minima leading to an erroneous answer or

[1] Strictly speaking, it is necessary to consider nonessential singularities of the second kind where $A(w,z)$ and $B(w,z)$ are zero simultaneously on the unit bicircle. However, this is extremely rare and is usually ignored in stability testing procedures.

almost flat regions, which results in extremely slow convergence of the search algorithm. In fact, in Shaw's example, the search for a local minimum took 14 times longer than the phase unwrapping itself.

As an alternative, we propose that the root maps be tracked directly for those frequency regions where the phase is almost discontinuous. In other words, calculating accurately the branch of $w(z)$ for $z = exp(v_0)$ to $z = exp(v_1)$ in the range of v, $v_0 \leq v \leq v_1$ for which the phase is almost discontinuous. Such a procedure would provide exact knowledge of whether the filter is stable, and if it is stable, how close are the zeros to the unit circle, thus providing a margin of stability.

The actual calculation of the root maps for all values of $z = exp(jv)$ is also useful as an aid for testing stability by itself. It provides a way of graphically portraying the margin of stability of a filter, especially for small filters, such as 8 by 8.

As an illustration of the possible application of this algorithm to filter stability testing, we used the differential equation method of Chapter 5 to calculate all the root maps of the following filter,

$$\begin{array}{cccc} & w & & \\ & \longrightarrow & & \\ z \downarrow & 1.000000 & -1.11870 & .355 \\ & -1.000030 & 1.55704 & -.550496 \\ & .400020 & -.659803 & .222204 \end{array} \tag{7.15}$$

The polynomial corresponding to $A(w, 1)$ was calculated and its roots determined. Each of these roots provided an initial condition for the differential equation solver Samples of the root map were thus calculated and are displayed in Figure 7.1.

From the samples of the root maps calculated it was inferred that a root map may cross the unit circle. The root map was recalculated in a smaller interval centered

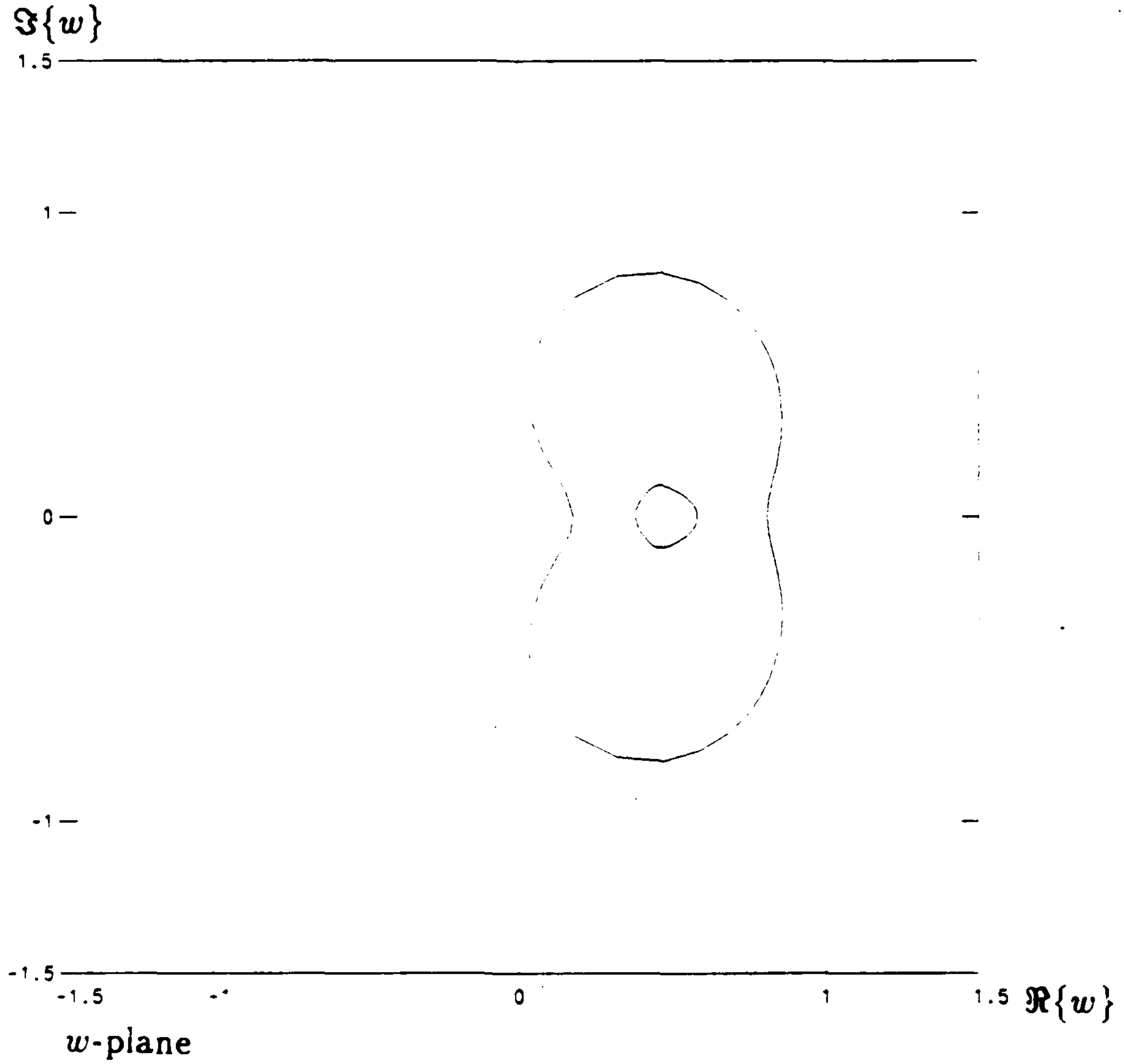


Figure 7.1: Root maps of two-dimensional recursive filter.

around the region where the root map may cross the unit circle. The resulting root map is shown in Figure 7.2. Since this precise root map does not cross the unit circle,

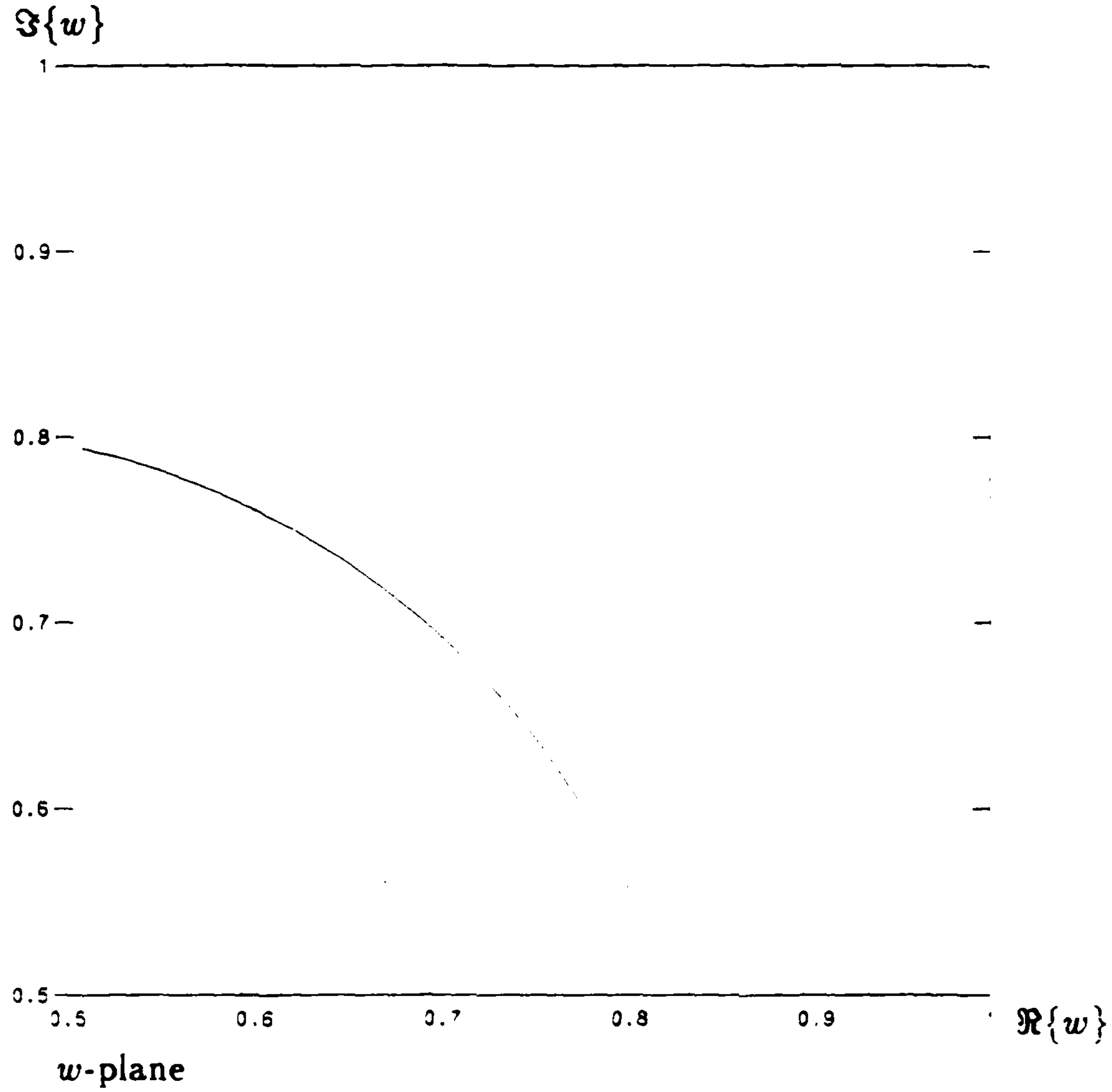


Figure 7.2: Single root map of two-dimensional recursive filter over small region.

we can be confident that the filter is stable.

Chapter 8

Summary and Suggestions for Future Research

The object of this thesis has been to develop a better algorithm for reconstruction of two-dimensional discrete signals from the Fourier transform magnitude. To this effect we have developed a new closed-form algorithm for reconstruction which is guaranteed to yield a correct solution given accurate data and which is computationally much more efficient than previous closed-form algorithms, both in terms of time of execution and memory requirements. The algorithm has been applied to a large number of images successfully.

We have noted however, that memory requirements preclude its use for images larger than 25 by 25 in the computer facilities available to us. Furthermore, the algorithm is extremely sensitive to noise in the given Fourier transform magnitude information.

Many algorithms have been proposed in the mathematical literature for solving large sets of linear equations in a memory efficient manner; for example, the conjugate gradients method. Such an approach may be fruitful in reducing the memory requirements of our phase retrieval procedure. This would allow the reconstruction of much larger images.

The problem of noise sensitivity is an important one. It is not clear at this point whether the sensitivity problem is particular to our algorithm only or is shared by other closed-form algorithms. In a sense, this property is characteristic of the phase retrieval problem since once noise is added to the Fourier transform magnitude or the autocorrelation signal, the reconstruction problem is replaced by an *approximation* problem, where no solution will satisfy the data exactly. In this case we need to decide on a criterion of goodness of match between a candidate solution and the measured data. Such a criterion should be amenable to analysis and the development of an appropriate closed-form algorithm. The GSF algorithm of Fienup does reduce an error norm; however, it may not drive the norm to its minimum value. In our opinion, solving the *phase retrieval approximation* problem should be a major focus of future research.

In developing our phase retrieval algorithm we have considered it as strictly a polynomial factorization problem. We made little use of the fact that the factors of $p_r(w, z)$, $p_x(w, z)$ and $\tilde{p}_x(w, z)$, are not independent but are intricately related. The use of this symmetry may prove useful in advancing the ideas presented here.

Finally, we have presented a few possible applications of the new factorization algorithm and root-tracking procedure to the area of two-dimensional filter design. It would be very interesting to develop other applications of these ideas.

Appendix A

Convergence of Iterative Algorithms for Phase Retrieval

In this appendix we conduct a study of the convergence properties of the Fienup algorithms for reconstruction from Fourier transform magnitude. To achieve this aim we have conducted a Monte Carlo study, i.e., the application of each algorithm to a large number of randomly generated images. The desire for a large sample size dictated that a small image size be used. However, such a study does provide a qualitative idea of the behavior of the algorithms in general.

Each trial consisted of the generation of two random images; the first image was used as the target image to be reconstructed, while the second was used as an initial estimate. There were four sets of experiments, studying the effect of a symmetric vs. non-symmetric support constraint and image size. The parameters of each of the four experiments is summarized in Table A.1.

In the first two experiments, "A" and "B". the images consisted of a 4 by 4 field of independently generated random numbers uniformly distributed between 0 and 1. It has been noted that the iterative algorithms perform better when a nonsymmetric

experiment	image size	support	max. iterations	times repeated
A	4 × 4	rect.	600	100
B	4 × 4	triang.	600	100
C	8 × 8	rect.	400	50
D	8 × 8	triang.	400	50

Table A.1: Summary of parameters used in convergence study of iterative phase retrieval algorithms.

region of support is known, for example, if the support is triangular. To study the effect of a nonsymmetric support, experiment "A" used a rectangular support, while experiment "B" used a triangular support by setting the the lower triangular half of the image to zero. The target image and the initial estimate were generated independently and no effort was made to generate a good initial estimate, other than satisfying the known spatial domain constraints. Thus, we are only interested in the case where the only a priori information is the Fourier transform magnitude, the signal support, and possibly, that the signal is non-negative. From the target image, zero padded to an 8 by 8 image, the discrete Fourier transform was calculated and the magnitude used as information to the reconstruction algorithms. Each algorithm was run with the same target and initial estimate images. For each algorithm, an error measure was calculated between the estimated image at the midpoint iteration and the target image. The algorithm was then continued and the error measure was recalculated at the final iteration. The trials were repeated, with different target and initial estimates, 100 times. The error measure used was the normalized mean squared error (NMSE) between the estimate $x_i[m,n]$ and the target image $x[m,n]$.

$$NMSE = \frac{\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}(x_i[m,n] - x[m,n])^2}{\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}(x[m,n])^2} \tag{A.1}$$

The NMSE was minimized over all trivial associates of $x_i[m,n]$, i.e., the estimate was successively multiplied by -1, and/or rotated by 180 degrees.

In order to study the effect of image size on the quality of reconstruction, the experiments described above were repeated for images consisting of a rectangular or triangular field of size 8 by 8, resulting in experiments "C" and "D". In this case, a total of 400 iterations were used and the experiment was repeated 50 times. Again the NMSE was recorded both at the midpoint and final iteration.

After conducting these experiments we were able to discern certain properties of the iterative algorithms studied. Calculating the NMSE at the middle iteration as well as the final iteration allows us to discern if the algorithm has progressed substantially in the intervening iterations. A clear way to compare the performance of these algorithms at the midpoint and final iteration is to draw a scatter plot where the abscissa of each point is the NMSE value at the midpoint iteration while the ordinate is the NMSE value at the final iteration. Points which lie close to the main diagonal indicate that no change has occurred in the intervening iterations while points substantially below the main diagonal indicate substantial improvement between the midpoint and final iterations.

Figures A.1 to A.3 show scatter plots of the NMSE at the midpoint and final iteration for each algorithm during experiment "A". A similar set of scatter plots are displayed in Figures A.4 to A.6 for experiment "B".

In the above figures, we see that for NMSE greater than 10^{-4}, there is no change between the midpoint and final iteration in almost all trials. Therefore, we can assume that for large NMSE (greater than 10^{-4}) the algorithms have reached a fixed point of the iterative procedure by the final iteration, and thus cannot be expected to make any

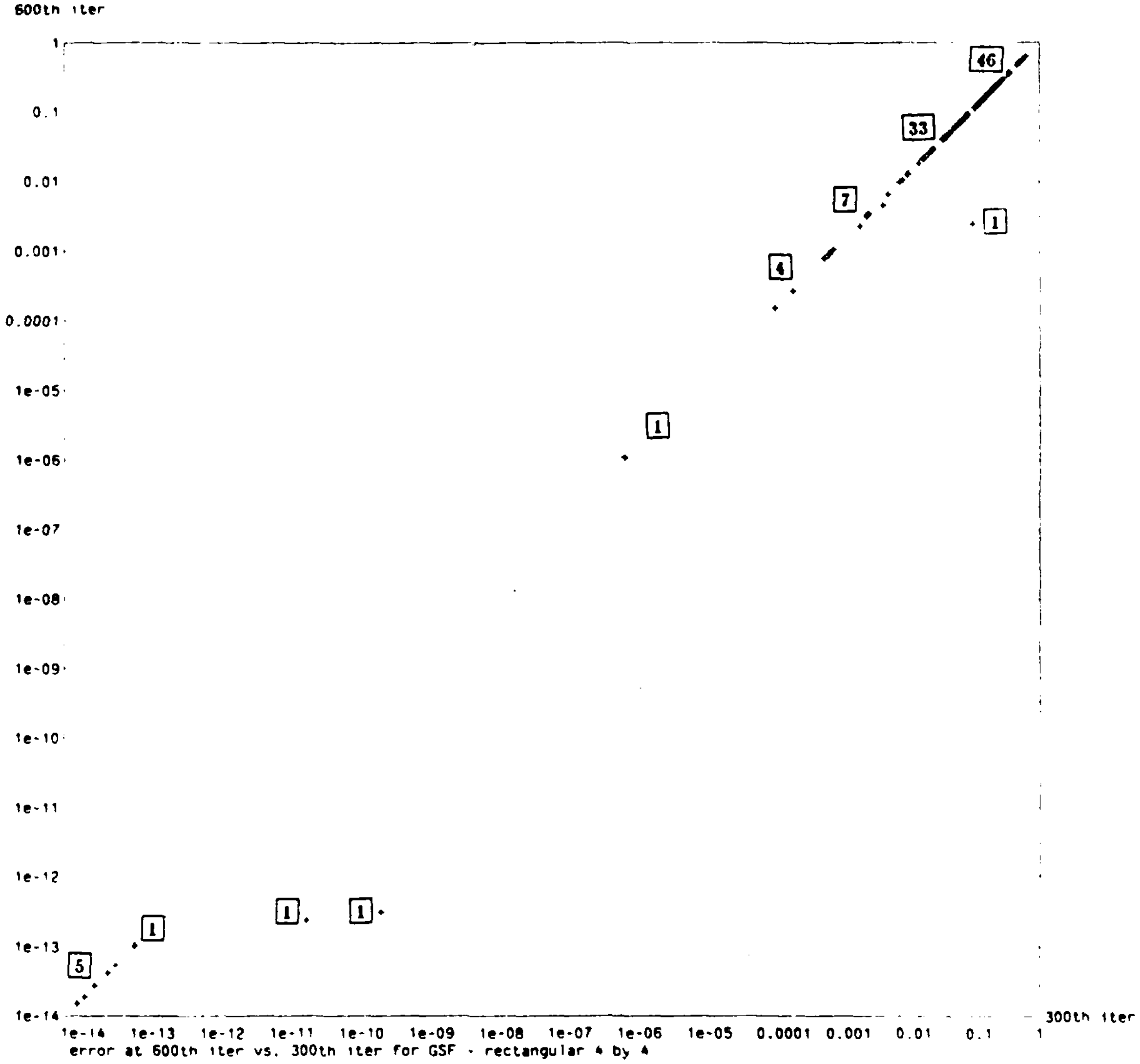


Figure A.1: Error at 600th vs. 300th iteration for GSF - rectangular 4 by 4. Number in box denotes number of trials which fell in that square.

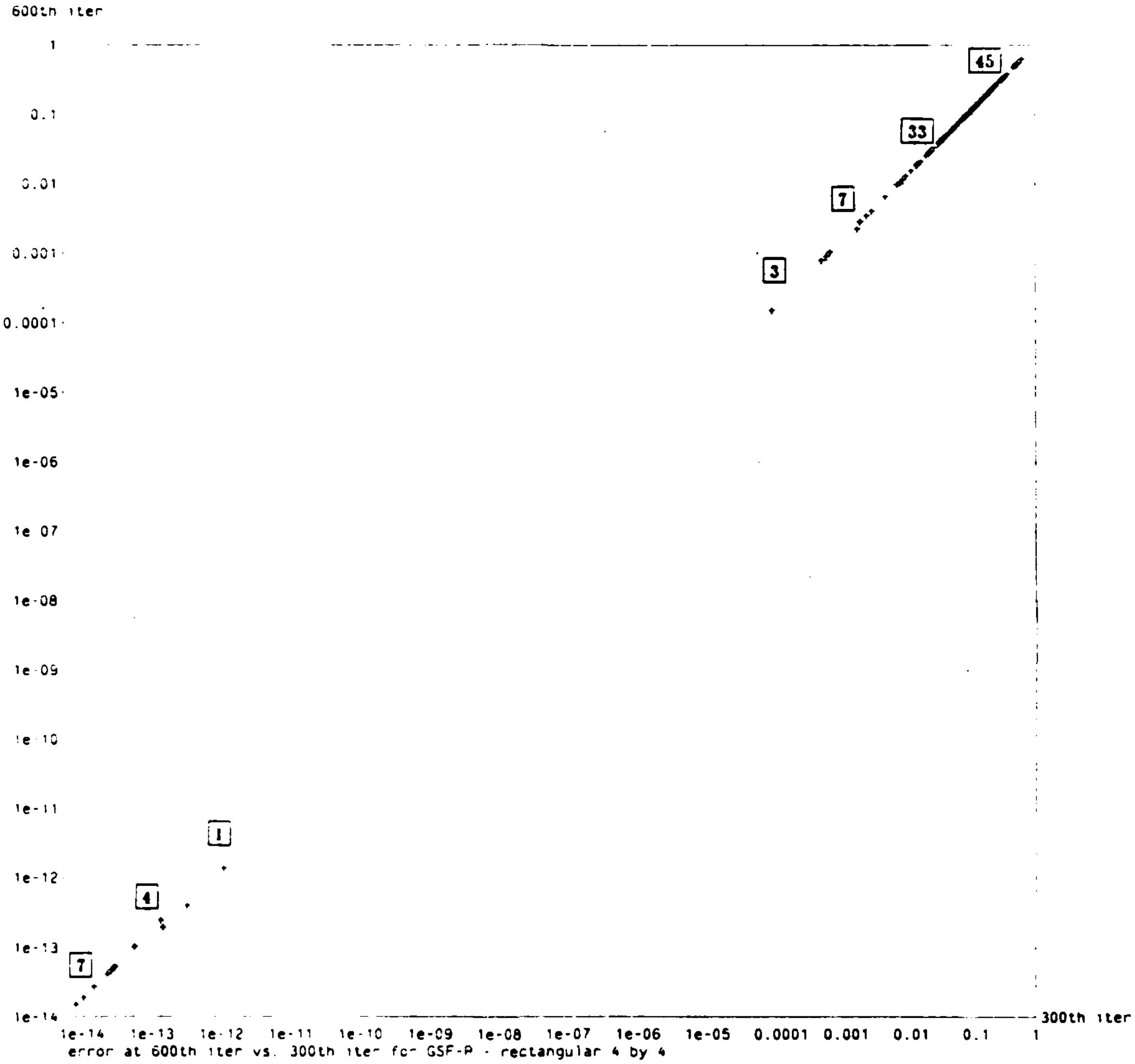


Figure A.2: Error at 600th vs. 300th iteration for GSF-P - rectangular 4 by 4

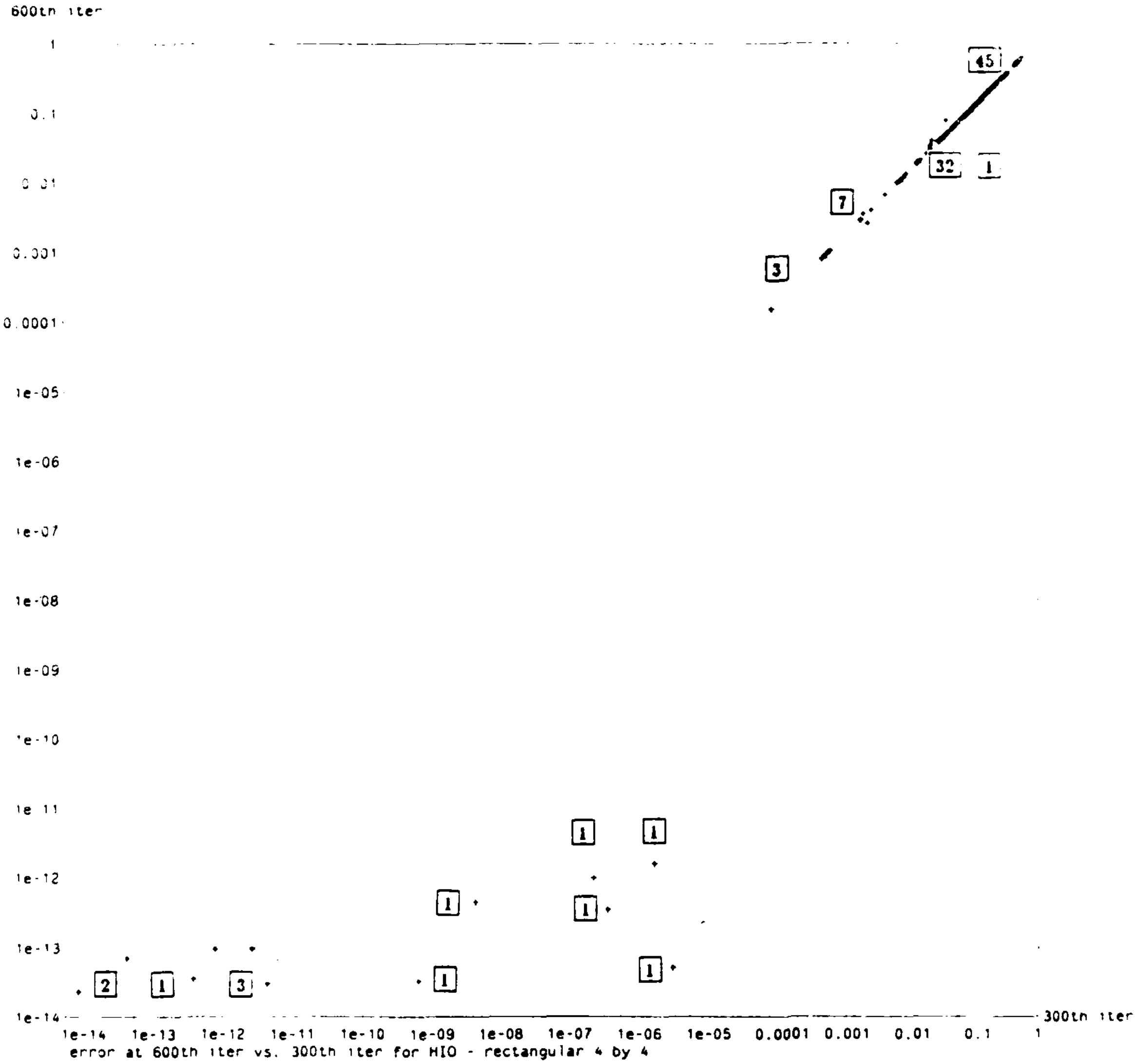


Figure A.3: Error at 600th vs. 300th iteration for HIO - rectangular 4 by 4

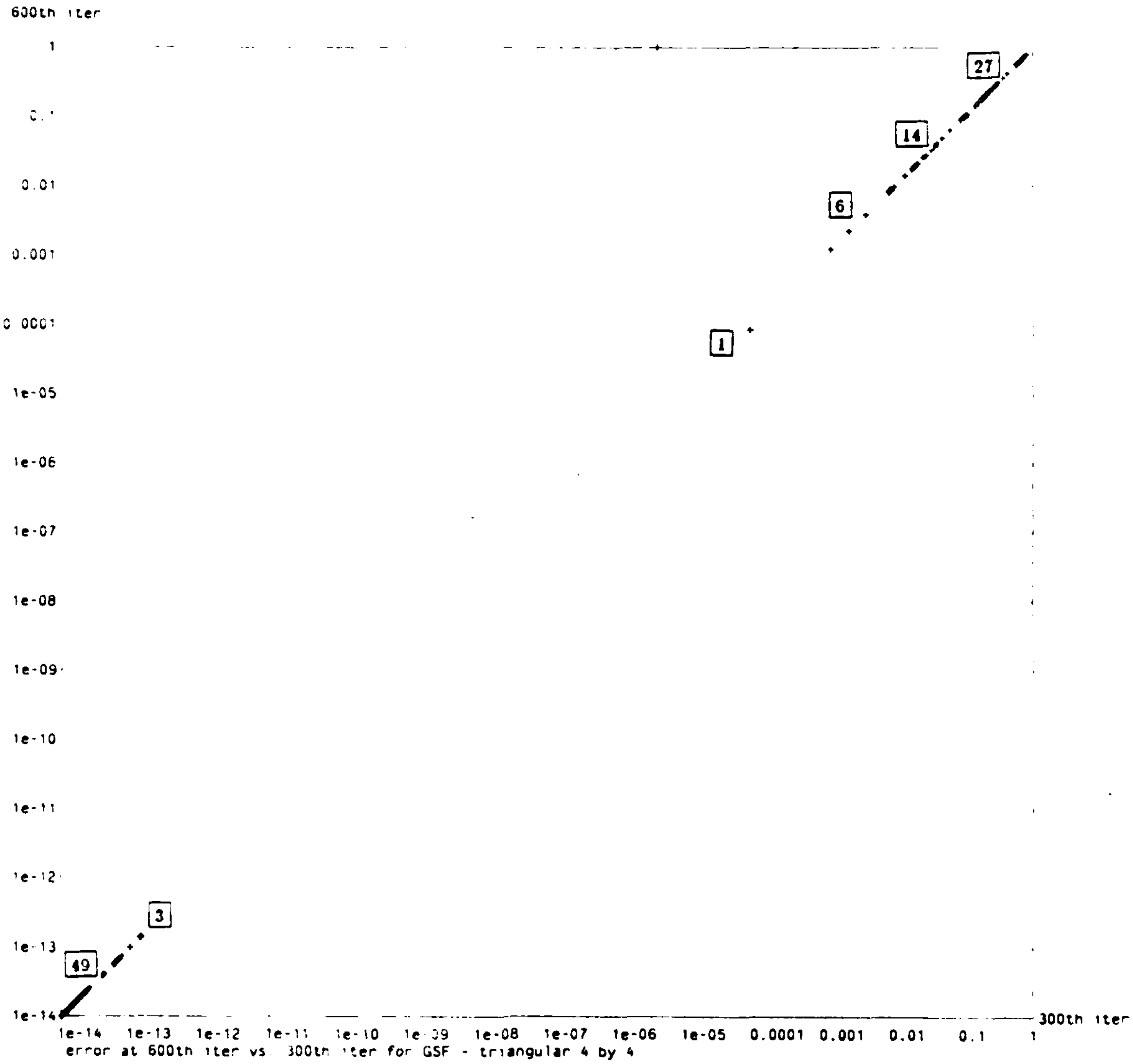


Figure A.4: Error at 600th vs. 300th iteration for GSF - triangular 4 by 4

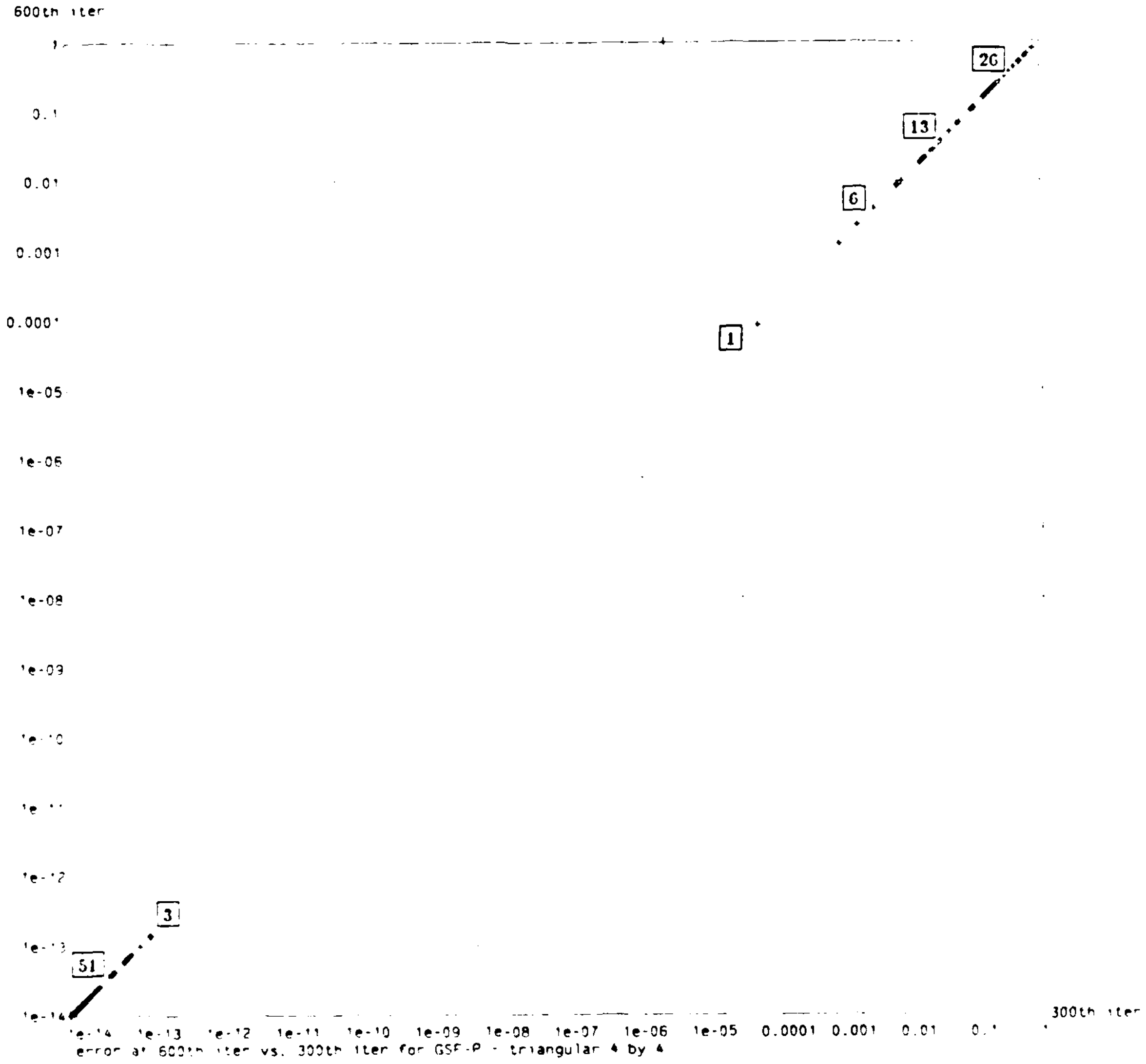


Figure A.5: Error at 600th vs. 300th iteration for GSF-P - triangular 4 by 4

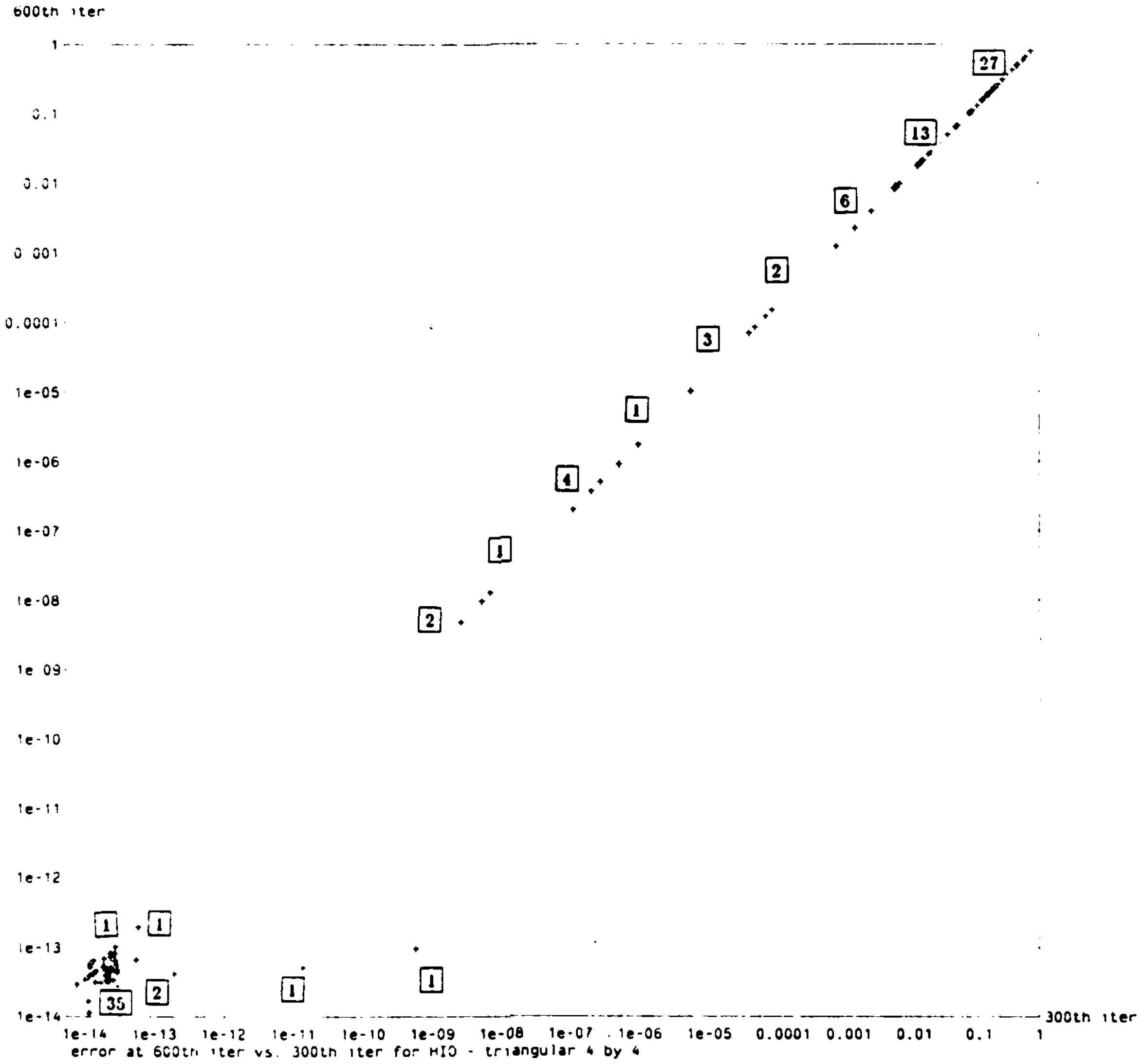


Figure A.6: Error at 600th vs. 300th iteration for HIO - triangular 4 by 4

more progress in subsequent iterations. The above observations are consistent with the results of previous authors [19,2,21] who have noted the slow progress of the GSF and GSF-P algorithms.

Confidence that a fixed point of the iterative procedure has been reached allows us to evaluate the success rate of each algorithm. For this purpose, we need to use a criterion for success. Although an appropriate cutoff value for an acceptable NMSE is difficult to assess, especially for small images as those considered in this paper, we have used a cutoff of 10^{-2} and 10^{-4}, which correspond to approximately one and two significant figures for each pixel value, respectively. For this cutoff, the success rate of each algorithm for 4 by 4 images is noted in Table A.2.

Success rate of convergence (percent)				
support type	rectangular		triangular	
cutoff	10^{-2}	10^{-4}	10^{-2}	10^{-4}
GSF	21	9	59	53
GSF-P	22	12	60	55
HIO	22	12	60	52

Table A.2: Summary of algorithm convergence as a function of support and criterion for 4 by 4 image.

The experiments described above were repeated for 8 by 8 images, experiments "C" and "D". Scatter plots of the results are shown in Figures A.7 to A.12. These plots allow us to again conclude that for the vast majority of trials, a fixed point of the iteration has been reached by the final iteration. The success rate of the iterative algorithms for 8 by 8 images is summarized in Table A.3.

Conclusions derived from Tables A.2 and A.3 are discussed in the main text.

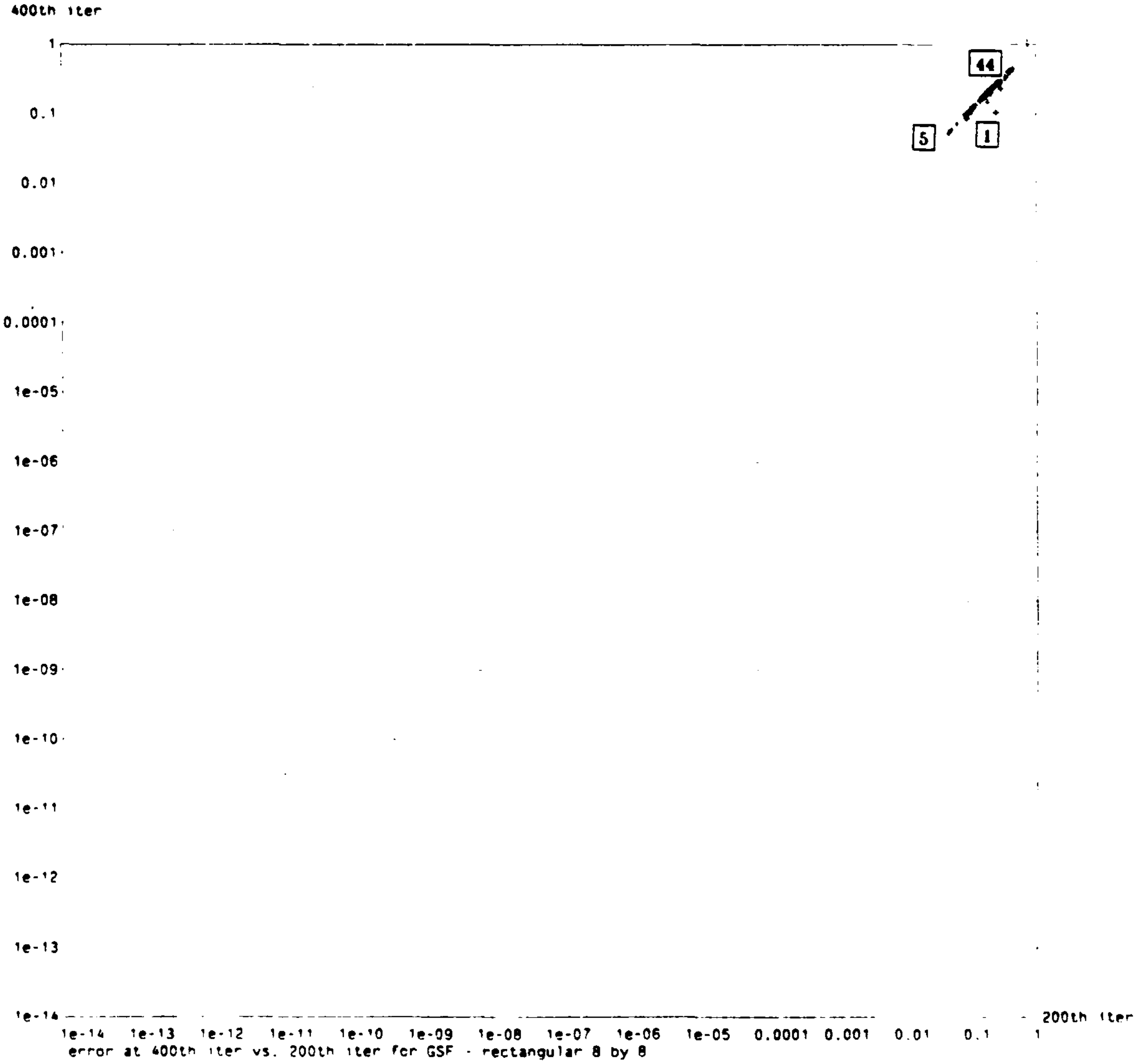


Figure A.7: Error at 400th vs. 200th iteration for GSF - rectangular 8 by 8

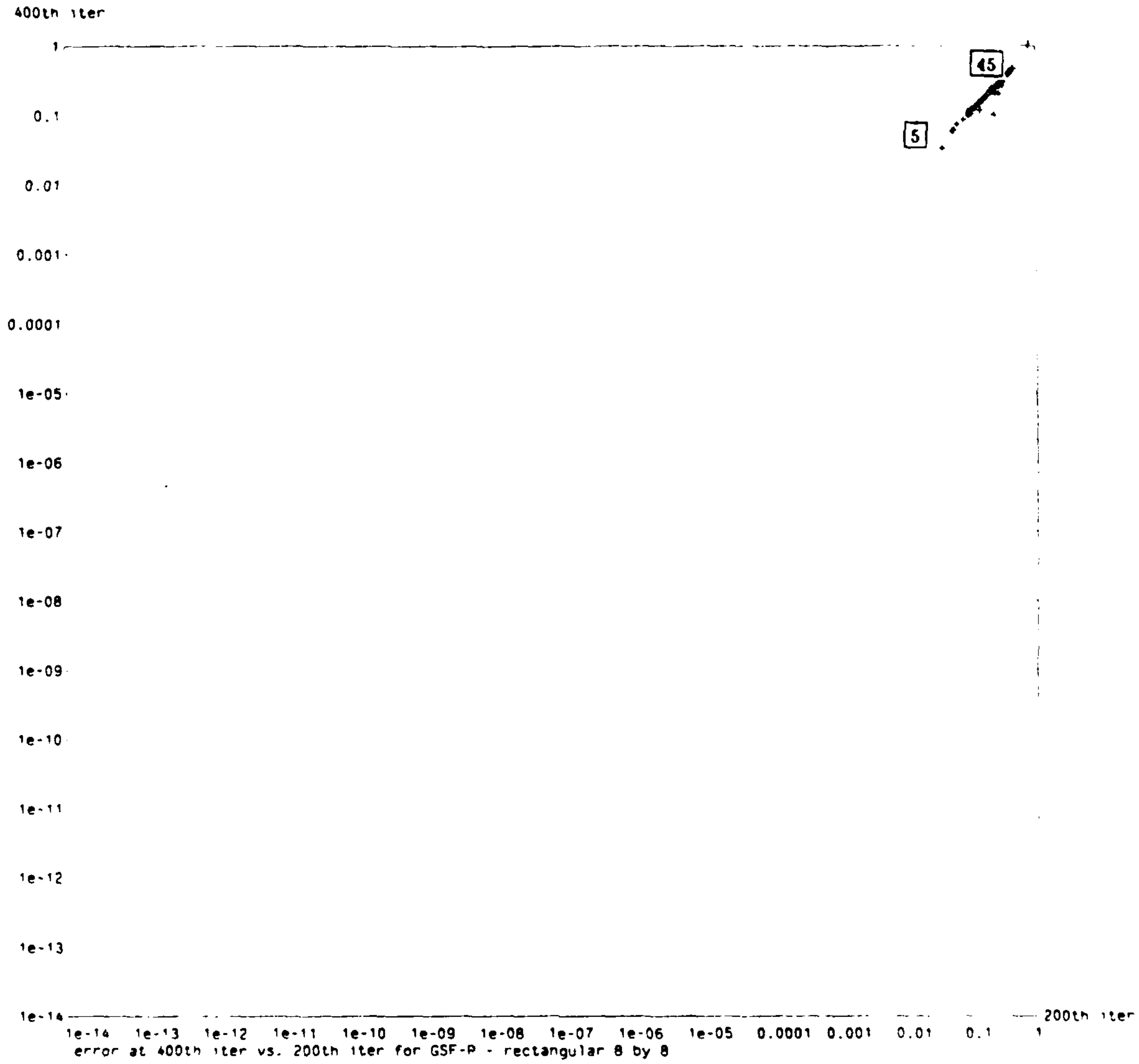


Figure A.8: Error at 400th vs. 200th iteration for GSF-P - rectangular 8 by 8

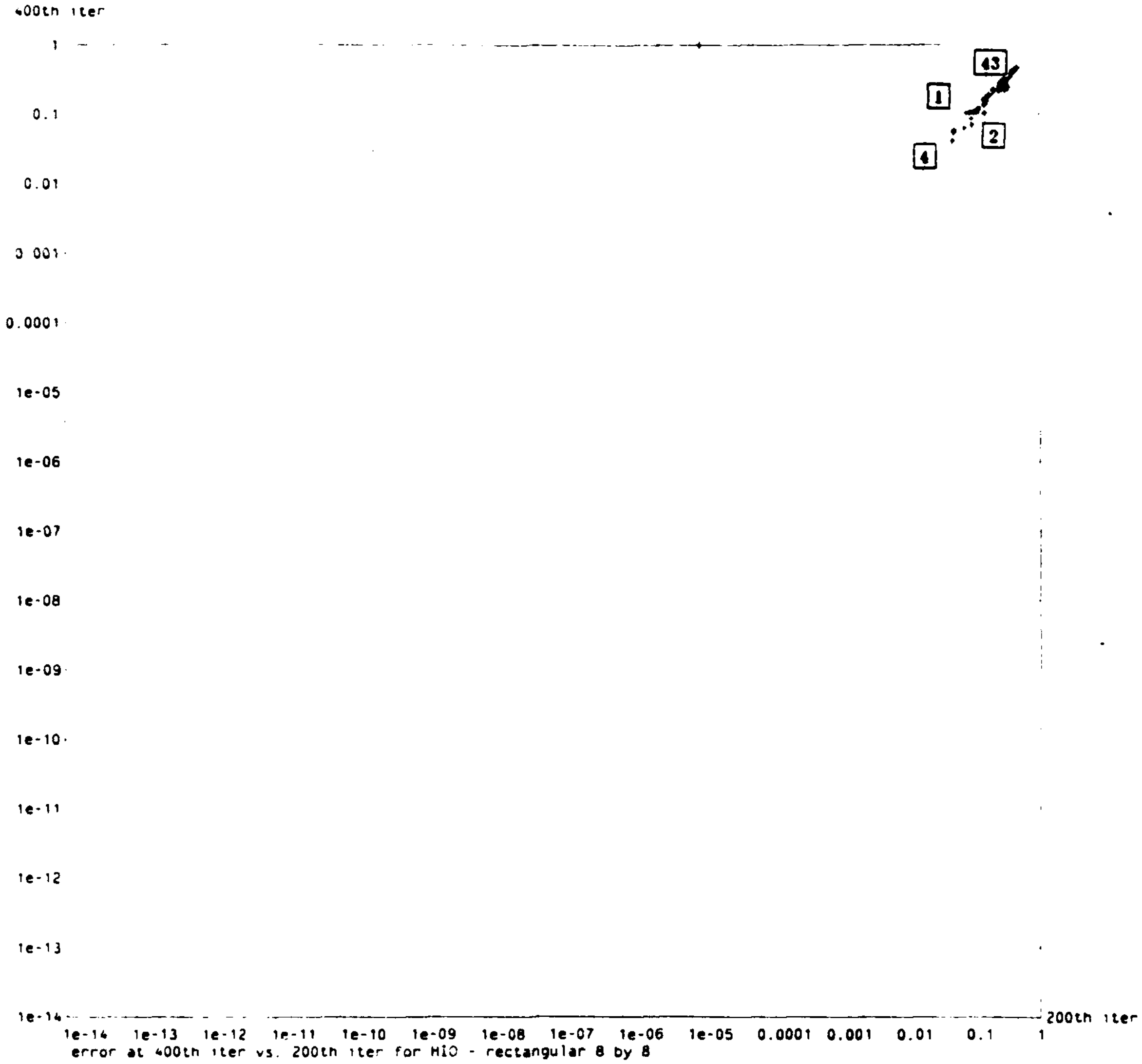


Figure A.9: Error at 400th vs. 200th iteration for HIO - rectangular 8 by 8

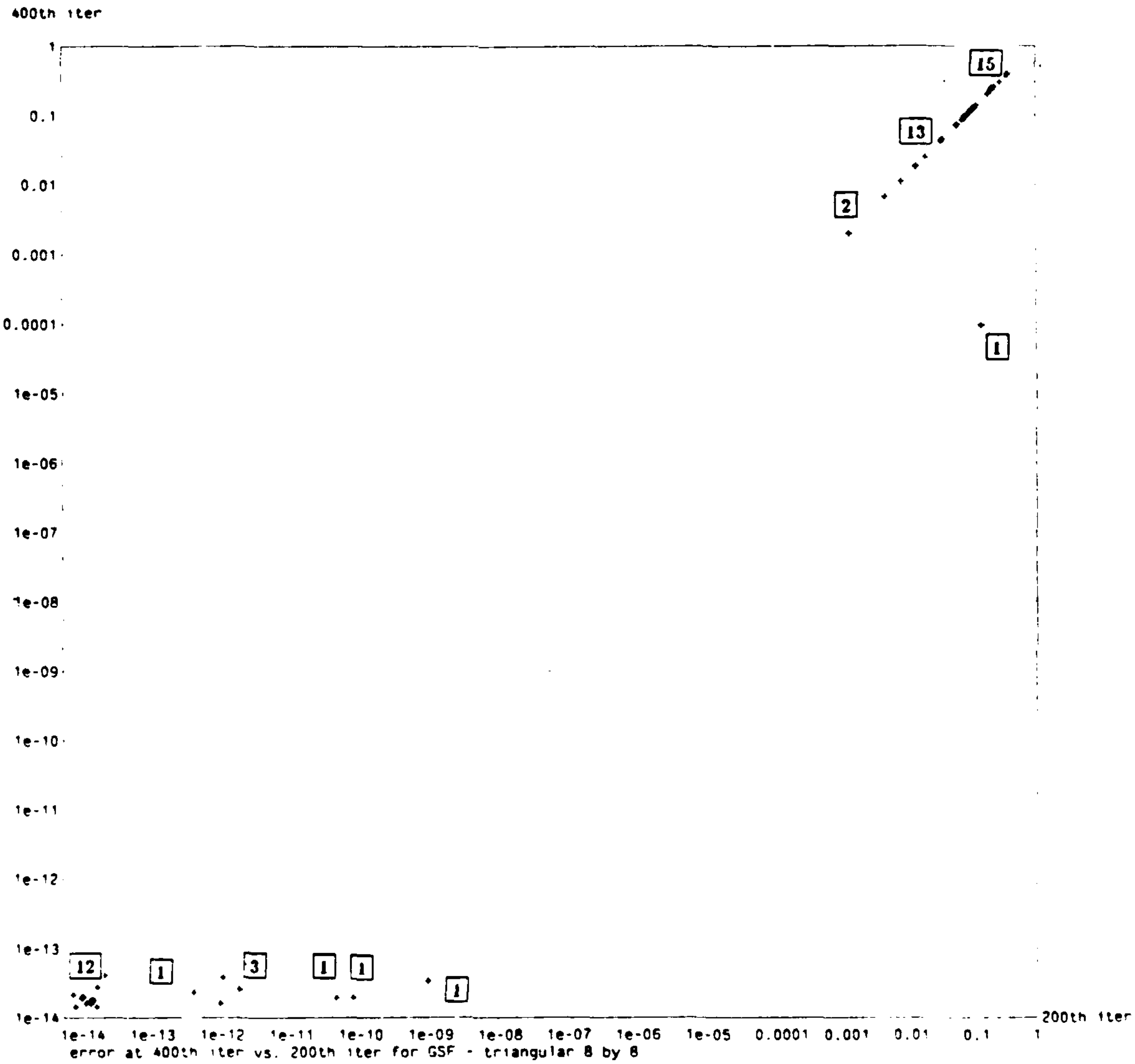


Figure A.10: Error at 400th vs. 200th iteration for GSF - triangular 8 by 8

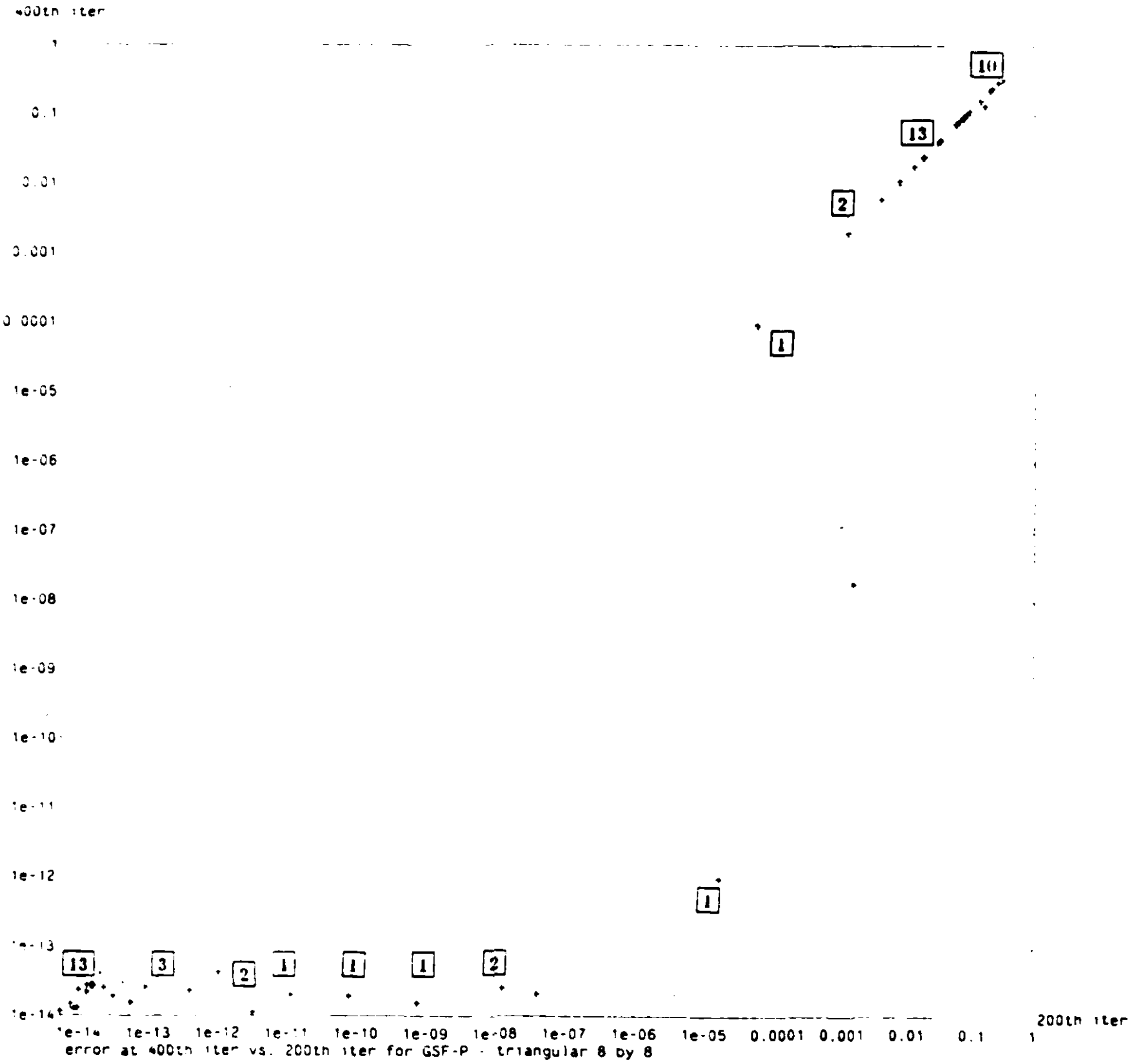


Figure A.11: Error at 400th vs. 200th iteration for GSF-P - triangular 8 by 8

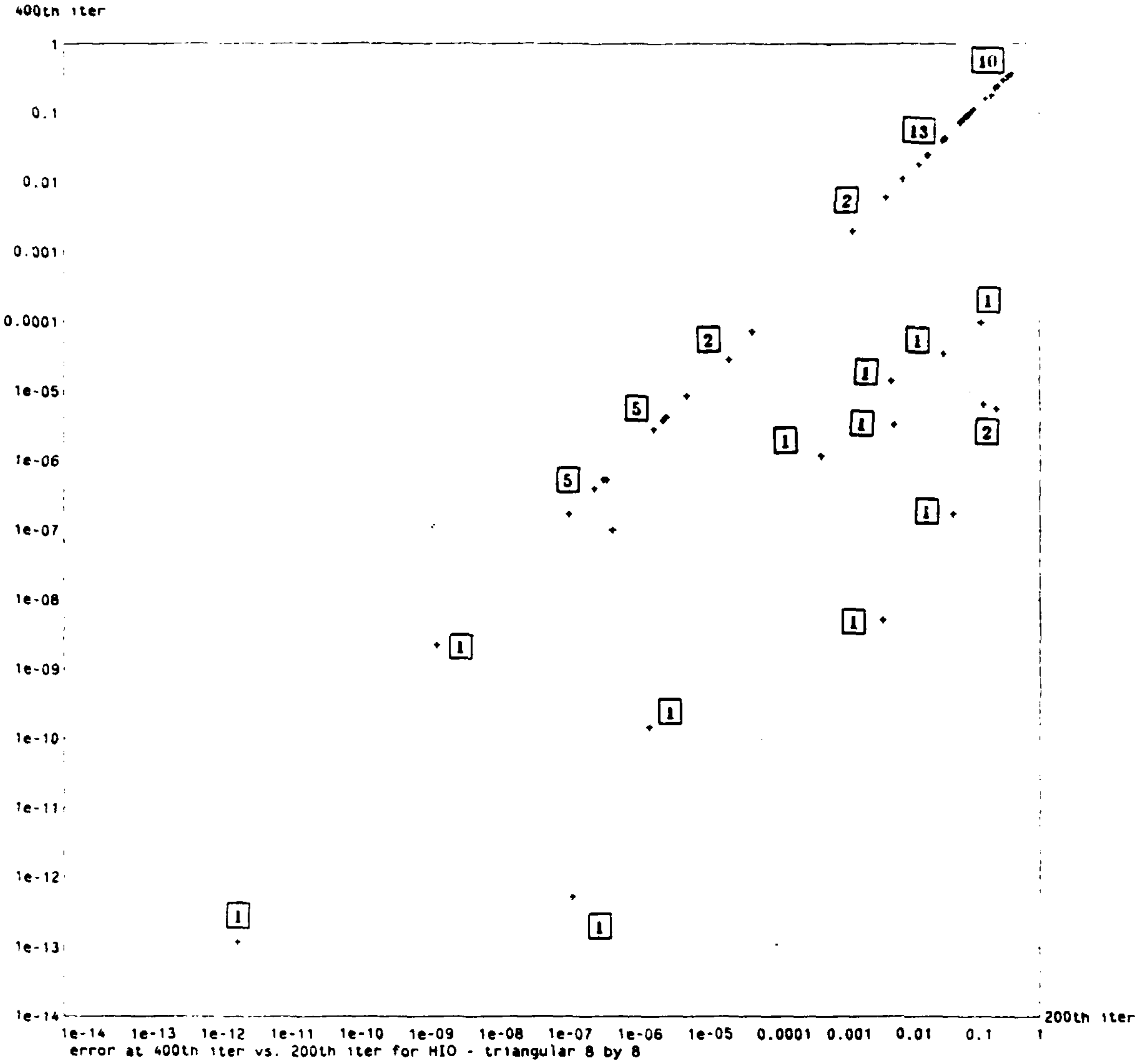


Figure A.12: Error at 400th vs. 200th iteration for HIO - triangular 8 by 8

Success rate of convergence (percent)				
support type	rectangular		triangular	
cutoff	10^{-2}	10^{-4}	10^{-2}	10^{-4}
GSF	0	0	44	40
GSF-P	0	0	54	50
HIO	0	0	54	48

Table A.3: Summary of algorithm convergence as a function of support and criterion for 8 by 8 image.

Appendix B

A Bound on the Number of Finite Common Zeros Based on Polynomial Degree [1]

Recall that Bezout's theorem is concerned with determining the number of common zeros of two bivariate polynomials. It is restated here for convenience,

Theorem B.1 *If $p(w,z)$ and $q(w,z)$ are bivariate polynomials of total degree r and s respectively with no common factors, then there are at most rs distinct pairs (w,z) where*

$$p(w,z) = q(w,z) = 0 \tag{B.1}$$

Since Bezout's theorem is concerned with *total degree* as opposed to degree in each variable, it pertains most generally to polynomials whose coefficients have triangular support as shown in Figure B.1.

In our case of reconstruction from Fourier transform magnitude, this corresponds to an image which has a triangular support. On the other hand, one is many times interested in images with square or rectangular support as shown in Figure B.2.

For the case when the polynomials under consideration have rectangular support,

[1] The derivation presented here is due to a collaboration between A. Zakhor and the author and is also described in [49].

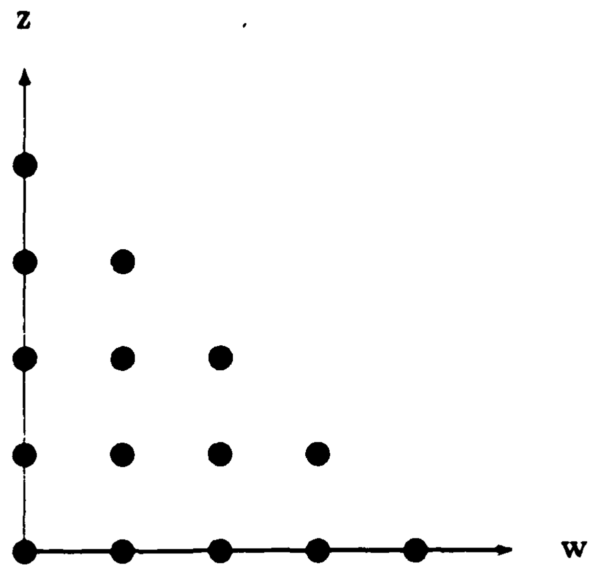


Figure B.1: Non-zero coefficients for a polynomial of total degree 4.

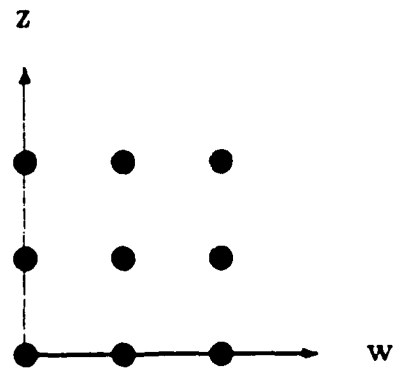


Figure B.2: Non-zero coefficients for a polynomial of degree (2,2).

we are able to lower the bound on the number of common finite zeros from the bound set by Bezout's theorem. Specifically if the two relatively prime polynomials $p(w,z)$ and $q(w,z)$ are given by

$$p(w,z) = \sum_{m=0}^{M_p} \sum_{n=0}^{N_p} p_{m,n} w^m z^n \tag{B.2}$$

and

$$q(w,z) = \sum_{m=0}^{M_q} \sum_{n=0}^{N_q} q_{m,n} w^m z^n \tag{B.3}$$

the upper bound on the number of common finite zeros set by Bezout's theorem is $(N_p + M_p)(N_q + M_q)$. Our objective is to establish a tighter upper bound on the number of common finite zeros of $p(w,z)$ and $q(w,z)$.

Before proceeding, we need to review several results concerning the *resultant* of polynomials in one or two variables. The resultant R_{pq} of two one-dimensional polynomials $p(w)$ and $q(w)$

$$p(w) = \sum_{n=0}^{M_p} p_n w^n \tag{B.4}$$

$$q(w) = \sum_{n=0}^{M_q} q_n w^n \tag{B.5}$$

$$\tag{B.6}$$

is defined [50] as the determinant of the $(M_p + M_q)$ by $(M_p + M_q)$ matrix

$$\begin{bmatrix}
p_0 & p_1 & \cdot & \cdot & \cdot & \cdot & \cdot & p_{M_p} & 0 & 0 & \cdot & \cdot & 0 \\
0 & p_0 & p_1 & \cdot & \cdot & \cdot & \cdot & p_{M_p-1} & p_{M_p} & 0 & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & \cdot & p_0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & p_{M_p} \\
q_0 & q_1 & \cdot & \cdot & \cdot & q_{M_q} & 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\
0 & q_0 & q_1 & \cdot & \cdot & q_{M_q-1} & q_{M_q} & 0 & \cdot & 0 & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & \cdot & \cdot & \cdot & q_0 & \cdot & \cdot & \cdot & \cdot & \cdot & q_{M_q}
\end{bmatrix} \tag{B.7}$$

A basic property of resultants is stated in the following theorem [50],

Theorem B.2 *When the polynomials $p(w)$ and $q(w)$ have numerical coefficients, a necessary and sufficient condition that they shall have a finite or infinite common root is that $R_{pq} = 0$.*

Consider now the two relatively prime bivariate polynomials $p(w, z)$ and $q(w, z)$ defined in (B.2) and (B.3) expressed as polynomials in w with coefficients which are polynomials in z,

$$p(w, z) = p_0(z) + p_1(z)w + \cdots + p_{M_p}(z)w^{M_p} \tag{B.8}$$

$$q(w, z) = q_0(z) + q_1(z)w + \cdots + q_{M_q}(z)w^{M_q} \tag{B.9}$$

We can define the resultant of $p(w, z)$ and $q(w, z)$ with respect to w as the determinant of the $(M_p + M_q)$ by $(M_p + M_q)$ matrix, $M(z)$, with polynomial entries:

$$M(z) =$$
$$\left[\begin{array}{cccccccccccc}
p_0(z) & p_1(z) & \cdot & \cdot & \cdot & \cdot & \cdot & p_{M_p}(z) & 0 & 0 & \cdot & 0 \\
0 & p_0(z) & p_1(z) & \cdot & \cdot & \cdot & \cdot & p_{M_p-1}(z) & p_{M_p}(z) & 0 & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\
0 & 0 & \cdot & \cdot & p_0(z) & & & \cdot & \cdot & \cdot & \cdot & p_{M_p}(z) \\
q_0(z) & q_1(z) & \cdot & \cdot & & q_{M_q}(z) & 0 & 0 & \cdot & \cdot & \cdot & 0 \\
0 & q_0(z) & q_1(z) & \cdot & \cdot & q_{M_q-1}(z) & q_{M_q}(z) & 0 & \cdot & 0 & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & \cdot & & \cdot & & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & \cdot & \cdot & \cdot & q_0(z) & \cdot & \cdot & \cdot & \cdot & q_{M_q}(z)
\end{array}\right] \tag{B.10}$$

This resultant is a function of the remaining variable z and is denoted by $R_{pq}(z)$. Expanding the determinant of the above matrix, and taking into account that each $p_i(z)$ and $q_i(z)$ is of degree at most N_p and N_q respectively, we can conclude that $R_{pq}(z)$ is a polynomial of degree $N_pM_q + N_qM_p$ or less. It can be shown [8], that if

$p(w,z)$ and $q(w,z)$ are relatively prime then $R_{pq}(z)$ is not identically zero. Moreover, if (w_0, z_0) is a common zero of $p(w,z)$ and $q(w,z)$ then $R_{pq}(z_0) = 0$. Thus the zero sets of $p(w,z)$ and $q(w,z)$ have at most $N_pM_q + N_qM_p$ values of z in common.

As our argument stands, we have not yet placed any tight limit on the number of intersections of $p(w,z)$ and $q(w,z)$ since for each z_i which is a root of $R_{pq}(z)$ there could be a large number of w_j, such that for each j,

$$p(w_j, z_i) = q(w_j, z_i) = 0 \tag{B.11}$$

In order to specify the number of w_j for each z_i, we need to study the behavior of $R_{pq}(z)$ in the vicinity of each z_i.

Theorem B.3 *If at each z_0 there are k values of w, w_j, such that*

$$p(w_j, z_0) = q(w_j, z_0) = 0 \textit{ for } j = 1, \cdots, k \tag{B.12}$$

then $R_{pq}(z)$ has a zero of multiplicity k at z_0.

The above theorem implies that $p(w,z)$ and $q(w,z)$ as defined in equation (B.2) and (B.3) have at most $N_pM_q + N_qM_p$ zeros in common.

In order to prove Theorem B.3 we need to review some results on matrices with polynomial entries, relating to the Smith Normal Form [51],

Theorem B.4 *Let $A(x)$ be an n by n polynomial matrix of normal rank r. We can find unimodular matrices $\{P(x), Q(x)\}$, such that*

$$B(x) = P(x)A(x)Q(x) \tag{B.13}$$

and

1. *$B(x)$ is a diagonal polynomial matrix called the Smith Normal Form of $A(x)$.*

2. The first r diagonal elements of $B(x)$ are monic polynomials $p_1(x)$, $p_2(x)$, $\cdots$, $p_r(x)$.

3. The remaining diagonal elements, if any, are zero.

4. $p_i(x)$ divides $p_{i+1}(x)$ for $i = 1, \cdots, r-1$.

Normal rank in the above theorem is defined in [51]. The ***unimodular*** polynomial matrices of the above theorem are defined to have nonzero constant determinant independent of x. Therefore, if $r = n$, i.e., if $A(x)$ has *full normal rank* then,

$$\det(B(x)) = \prod_{i=1}^{n} p_i(x) \tag{B.14}$$

Also, from part (4) of Theorem B.4 we can conclude that if $p_i(x) = 0$ then $p_k(x) = 0$ for $k \geq i$. From the above theorem, we can derive the following,

Theorem B.5 *Let $A(x)$ be a polynomial matrix of full normal rank. If $A(x_0)$ has rank deficiency of k then* $\det(A(x))$, *has a zero of multiplicity k at x_0.*

Proof: Using Theorem B.4 we can find $B(x)$, the Smith normal form of $A(x)$. Since $P(x)$ and $Q(x)$ are unimodular, $B(x)$ is of full normal rank. Furthermore, the ranks of $B(x)$ and $A(x)$ at each value of x, including x_0, are equal. Therefore $B(x_0)$ has rank deficiency of k. This means that $p_n(x_0) = p_{n-1}(x_0) = \cdots = p_{n-k+1}(x_0) = 0$. Therefore, from (B.14) $\det(B(x))$ has a zero of multiplicity k at x_0. Since the determinant of $A(x)$ is within a constant factor of that of $B(x)$, $A(x)$ also has a zero of multiplicity k at x_0.

□

Using Theorem B.5, we can return to Theorem B.3,

Proof: (Theorem B.3) Suppose that for z_0 there are k common finite zeros w_j, $j = 1, \cdots, k$ between $p(w, z)$ and $q(w, z)$. Then the matrix $M(z_0)$ defined by (B.10) must have k linearly independent null vectors given by:

$$[1, w_j, w_j^2, \cdots, w_j^{M_p+M_q-1}]^T \tag{B.15}$$

for $j = 1, \cdots, k$. Thus $M(z_0)$ has rank deficiency of k. Furthermore, since $p(w, z)$ and $q(w, z)$ have no common factors, $R_{pq}(z)$ is not identically zero, so $M(z)$ has full normal rank. From Theorem B.5 then, $R_{pq}(z)$ has a zero of multiplicity k at z_0. □

From Theorem B.3, and the fact that $R_{pq}(z)$ is a polynomial of degree $N_pM_q + N_qM_p$, we get immediately, the main result of this appendix,

Theorem B.6 *Let $p(w, z)$ and $q(w, z)$ be two polynomials of degree (M_p, N_p) and (M_q, N_q) with no common factors, then there are at most $N_pM_q + N_qM_p$ pairs of finite complex numbers (w, z) such that*

$$p(w, z) = q(w, z) = 0 \tag{B.16}$$

Appendix C

Calculating the Degree of an Irreducible Factor

In this appendix we want to consider the situation where the degree of an irreducible factor of the polynomial $p(w, z)$ is not known. From the path-tracking algorithm developed in Chapter 5, we have generated a large number of zeros of the polynomial $p(w, z)$. We denote these zeros by (w_k, z_k) for $k = 1, \cdots, K$. We will see later that for this case K should be picked to be the maximum allowable by Bezout's theorem, $totdeg(p)^2 + 1$. From our earlier arguments, these zeros all correspond to a single irreducible factor of $p(w, z)$, which we will denote by $d(w, z)$. It is this factor which we seek to isolate. Although the total degree of $p(w, z)$ is known, we only know that the total degree of $d(w, z)$ can be at most $totdeg(p)$. In this appendix we describe two mechanisms for calculating $totdeg(d)$.

The first mechanism is the simplest conceptually. Assume that $totdeg(d) = 1$. Then for all zeros (w_k, z_k), there must be constants d_{00}, d_{01}, d_{10} which are not all zero such that

$$d_{00} + d_{01}w_k + d_{10}z_k = 0 \tag{C.1}$$

for all k. Equivalently, we need that the matrix

$$M_1 = \begin{bmatrix} 1 & w_1 & z_1 \\ 1 & w_2 & z_2 \\ \vdots & \vdots & \vdots \\ 1 & w_K & z_K \end{bmatrix} \tag{C.2}$$

have a non-trivial nullspace, i.e., a non-zero vector x such that $M_1 x = 0$. Thus, the first step of this procedure is to calculate the rank of M_1, which we will symbolize by $r(M_1)$. If it is less than 3, then $totdeg(d) = 1$. On the other hand, if M_1 has full rank, then we generate M_2 defined by,

$$M_2 = \begin{bmatrix} 1 & w_1 & w_1^2 & z_1 & w_1 z_1 & z_1^2 \\ 1 & w_2 & w_2^2 & z_2 & w_2 z_2 & z_2^2 \\ \vdots & & \vdots & & \vdots & \\ 1 & w_K & w_K^2 & z_K & w_K z_K & z_K^2 \end{bmatrix} \tag{C.3}$$

Again, the rank of M_2 is calculated. If it is less than 6, then $totdeg(d) = 2$, otherwise, we calculate M_3. Thus the algorithm proceeds by successively calculating the rank of M_n and checking if the rank of M_n is less than $(n+1)(n+2)/2$. The smallest n for which this condition is true is the total degree of $d(w, z)$.

For large polynomials, the procedure described above becomes very inefficient in some cases. For example, if the polynomial $p(w, z)$ is itself irreducible, $totdeg(p)$ different matrices and rank determinations would be required.

It turns out that by examining the rank of only the last matrix M_n for $n = totdeg(p)$ we can calculate the degree of $d(w, z)$. For simplicity, we will call this M_n simply M for the rest of the discussion. Since by assumption $d(w_k, z_k) = 0$ for all k, the coefficients of $d(w,z)$, with appropriately padded zero coefficients, form a null vector of M. In fact, the coefficients of any polynomial of total degree $totdeg(p)$ which has $d(w,z)$ as a factor form a null vector of M. Conversely, with any null vector of M, we can associate a "null" polynomial with $d(w,z)$ as a factor by virtue of Bezout's theorem. The problem now is to find the dimension of this nullspace. Consider the set of polynomials,

$$d(w,z), zd(w,z), wd(w,z), \cdots, w^m z^n d(w,z) \tag{C.4}$$

for all m, n such that the total degree of $w^m z^n d(w,z)$ is less than or equal to $totdeg(p)$. Then all the null polynomials can be expressed as a linear combination of these polynomials. In fact the "coordinates" of a null polynomial $q(w,z) = d(w,z)a(w,z)$ are exactly the coefficients of $a(w,z)$. Thus the set of polynomials in (C.4) span the nullspace of M. We have not shown however, that these polynomials are linearly independent. Suppose that they form a linearly dependent set, then we can generate a set of coefficients $c_{i,j}$ not all zero such that

$$\sum_{i,j} c_{i,j} w^i z^j d(w,z) \equiv 0 \tag{C.5}$$

In other words, there exists a polynomial $c(w,z)$ such that $c(w,z)d(w,z) = 0$ for all w, z. However, since $d(w,z)$ is not identically zero, $c(w,z)$ must be zero. Thus the polynomials in (C.4) form a linearly independent set which spans the nullspace of M. From this argument we see that the dimension of the nullspace is simply the number of polynomials in the set described in (C.4), which in turn is simply the maximum number of coefficients in the polynomial $c(w,z)$ in (C.5). The total degree of $c(w,z)$ is $totdeg(p) - totdeg(d)$. A polynomial of total degree n has $(n+1)(n+2)/2$ coefficients;

hence, the dimension of the nullspace of M is

$$(totdeg(p) - totdeg(d) + 1)(totdeg(p) - totdeg(d) + 2)/2 \tag{C.6}$$

Note that not all possible values for the rank of M are possible, since some values for the rank of M lead to non-integer values of $totdeg(d)$. The procedure to calculate $totdeg(d)$ can be summarized as follows:

1. Generate M consisting of $(totdeg(p) + 1)(totdeg(p) + 2)/2$ columns and $K = totdeg(p)^2 + 1$ rows.

2. Calculate the rank of M, $r(M)$.

3. The value of $totdeg(d)$ is given by

$$totdeg(d) = \frac{2totdeg(p) + 3 + \sqrt{(2totdeg(p) + 3)^2 - 4r(M)}}{2} \tag{C.7}$$

Bibliography

[1] Yu. M. Bruck and L. G. Sodin. On the ambiguity of the image reconstruction problem. *Optics Communications*, 30:304–308, 1979.

[2] M. H. Hayes. The reconstruction of a multidimensional sequence from the phase or magnitude of the Fourier transform. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-30:140–154, 1982.

[3] J. R. Fienup. Space object imaging through the turbulent atmosphere. *Optical Engineering*, 18:529–534, 1979.

[4] A. Labeyrie. Attainment of diffraction limited resolution in large telescopes by Fourier speckle patterns in star images. *Astron. and Astrophys.*, 6:85–87, 1970.

[5] R. W. Gerchberg and W. O. Saxton. Phase determination from image and diffraction plane pictures in the electron microscope. *Optik*, 34:275–284, 1971.

[6] G. N. Ramachandran and R. Srinivasan. *Fourier Methods in Crystallography.* Wiley-Interscience, New York, 1970.

[7] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice Hall, Englewood Cliffs, NJ, 1975.

[8] A. Mostowski and M. Stark. *Introduction to Higher Algebra.* Pergamon Press, New York, 1964.

[9] M. H. Hayes. *Signal Reconstruction from Phase or Magnitude.* PhD thesis, Massachusetts Institute of Technology, 1981.

[10] G. A. Bliss. *Algebraic Functions.* American Mathematical Society, New York, 1933.

[11] E. J. Akutowicz. On the determination of the phase of a Fourier integral. *Trans. American Mathematical Society*, 83:179–182, 1956.

[12] A. Walther. The question of phase retrieval in optics. *Optica Acta*, 10:41–49, 1963.

[13] E. M. Hofstetter. Construction of time-limited functions with specified autocorrelation functions. *IEEE Trans. Information Theory*, IT-8:119–126, 1964.

[14] M. H. Hayes and J. H. McClellan. Reducible polynomials in more than one variable. *Proceedings of the IEEE,* 70:197–198, 1982.

[15] J. L. C. Sanz, T. S. Huang, and F. Cukierman. Stability of unique Fourier transform phase reconstruction. *Journal of the Optical Society of America,* 73:1442–1445, 1983.

[16] A. M. J. Huiser and P. van Toorn. Ambiguity of the phase-reconstruction problem. *Journal of the Optical Society of America,* 46:407–420, 1976.

[17] J. L. C. Sanz and T. S. Huang. Unique reconstruction of a band-limited multidimensional signal from its phase or magnitude. *Journal of the Optical Society of America,* 73:1446–1450, 1983.

[18] J. R. Fienup. Reconstruction of an object from the modulus of its Fourier transform. *Optics Letters,* 3:27–29, 1978.

[19] J. R. Fienup. Phase retrieval algorithms: A comparison. *Applied Optics,* 21:2758–2769, 1982.

[20] R. W. Gerchberg and W. O. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik,* 35:237–246, 1972.

[21] J. L. C. Sanz, T. S. Huang, and T-. F. Wu. A note on iterative Fourier transform phase reconstruction from magnitude. *IEEE Trans. Acoustics, Speech, and Signal Processing,* ASSP-32:1251–1254, 1984.

[22] A. Levi and H. Stark. Image restoration by the method of generalized projections with application to restoration from magnitude. *Journal of the Optical Society of America - A,* 1:932–943, 1984.

[23] M. C. Won, D. Mnyama, and R. H. T. Bates. Improving initial phase estimates for phase retrieval algorithms. *Optica Acta,* 32:377–396, 1985.

[24] J. R. Fienup. Comments on 'The reconstruction of a multidimensional sequence from the phase or magnitude of the Fourier transform'. *IEEE Trans. Acoustics, Speech, and Signal Processing,* ASSP-31:1256–1262, 1983.

[25] R. H. T. Bates. Fourier phase problems are uniquely solvable in more than one dimension. I. Underlying theory. *Optik,* 61:247–262, 1982.

[26] K. L. Garden and R. H. T. Bates. Fourier phase problems are uniquely solvabie in more than one dimension. II. One-dimensional considerations. *Optik,* 62:131–142, 1982.

[27] W. R. Fright and R. H. T. Bates. Fourier phase problems are uniquely solvable in more than one dimension. III. Computational examples for two dimensions. *Optik,* 62:219–230, 1982.

[28] R. H. T. Bates and W. R. Fright. Composite two-dimensional phase-restoration procedure. *Journal of the Optical Society of America*, 73:358–365, 1983.

[29] N. Canterakis. Magnitude-only reconstruction of two-dimensional sequences with finite regions of support. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-31:1256–1262, 1983.

[30] H. V. Deighton, M. S. Scivier, and M. A. Fiddy. Solution of the two-dimensional phase-retrieval problem. *Optics Letters*, 10:250–251, 1985.

[31] H. M. Berenyi, H. V. Deighton, and M. A. Fiddy. The use of bivariate polynomial factorization algorithms in two-dimensional phase problems. *Optica Acta*, 32:689–701, 1985.

[32] M. Marden. *Geometry of Polynomials.* American Mathematical Society, Providence, Rhode Island, 1966.

[33] A. I. Markushevich. *Theory of Functions of a Complex Variable.* Volume II, Prentice-Hall, Englewood Cliffs, NJ, 1965.

[34] E. Kaltofen. Factorization of polynomials. *Computing Supplement*, 4:95–113, 1982.

[35] E. Kaltofen. A polynomial-time reduction from bivariate to univariate integral polynomial factorization. In *Proc. 23rd Symposium on Foundations of Computer Science, IEEE*, pages 57–64, 1982.

[36] J. Lipson. Newton's method: a great algebraic algorithm. *Proc 1976 Symposium on Symbolic and Algebraic Computation*, 260–270, 1982.

[37] D. Y. Y. Yun. Hensel meets Newton - algebraic constructions in an analytic setting. In J. F. Traub, editor, *Algebraic Computational Complexity*, Academic Press, New York, 1976.

[38] P. J. Napier and R. H. T. Bates. Inferring phase information from modulus information in two-dimensional aperture synthesis. *Astron. Astrophys. Supplement*, 15:427–430, 1974.

[39] S. R. Curtis, J. S. Lim, and A. V. Oppenheim. Signal reconstruction from one bit of Fourier transform phase. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-33:643–657, 1985.

[40] S. R. Curtis. *Reconstruction of Multidimensional Signals from Zero Crossings.* PhD thesis, Massachusetts Institute of Technology, 1985.

[41] K. Steiglitz and B. Dickinson. Phase unwrapping by factorization. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-30:984–991, 1982.

[42] G. Dahlquist and A. Bjorck. *Numerical Methods.* Prentice-Hall, Englewood Cliffs, NJ, 1974.

[43] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1983.

[44] J. L. C. Sanz and T. S. Huang. Polynomial system of equations and its application to the study of the effect of noise on multidimensional Fourier transform phase retrieval from magnitude. *IEEE Trans. Acoustics, Speech, and Signal Processing,* ASSP-33:997–1004, 1985.

[45] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1984.

[46] N. K. Bose. *Applied Multidimensional System Theory.* Van Nostrand Reinhold, New York, 1982.

[47] E. Emre and O. Huseyin. Relative primeness of multivariate polynomials. *IEEE Trans. Circuits and Systems,* 22:56–57, 1975.

[48] G. A. Shaw. *Design, stability and performance of two-dimensional recursive digital filters.* PhD thesis, Georgia Institute of Technology, 1979.

[49] A. Zakhor and D. Izraelevitz. A note on the sampling of zero-crossings of two-dimensional signals. Accepted for publication, March 1986.

[50] R. J. Walker. *Algebraic Curves.* Dover Publications, New York, 1962.

[51] T. Kailath. *Linear Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1980.

DISTRIBUTION LIST

	DODAAD Code	
Director Defense Advanced Research Project Agency 1400 Wilson Boulevard Arlington, Virginia 22209 Attn: Program Management	HX1241	(1)
Head Mathematical Sciences Division Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	N00014	(1)
Administrative Contracting Officer E19-628 Massachusetts Institute of Technology Cambridge, Massachusetts 02139	N66017	(1)
Director Naval Research Laboratory Attn: Code 2627 Washington, D.C. 20375	N00173	(6)
Defense Technical Information Center Bldg 5, Cameron Station Alexandria, Virginia 22314	S47031	(12)
Dr. Judith Daly DARPA / TTO 1400 Wilson Boulevard Arlington, Virginia 22209		(1)

END

DTIC

7 — 86