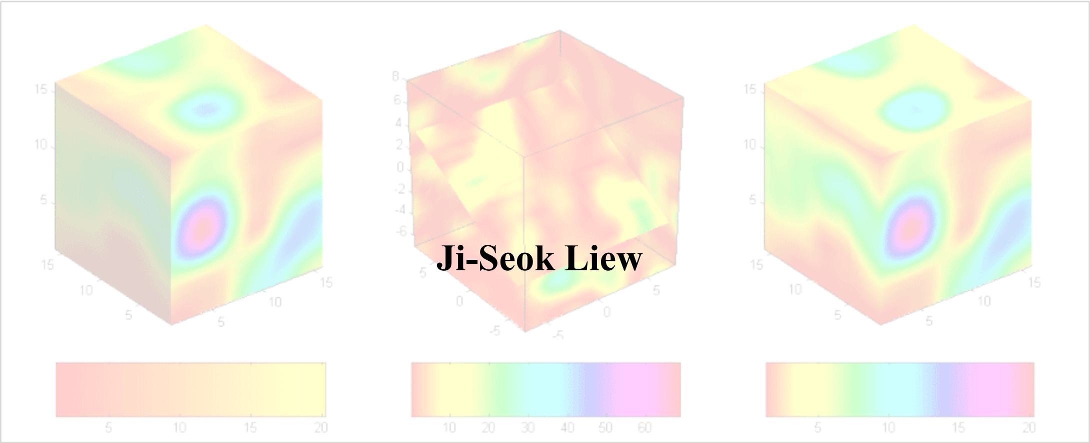
MULTI-DIMENSIONAL PARAMETRIC ESTIMATION: TWO-DIMENSIONAL SHARPENING BY PREDICTIVE BANDWIDTH EXTRAPOLATION AND FAST ALGORITHMS FOR THREE-DIMENSIONAL AUTOREGRESSIVE ESTIMATION



Ji-Seok Liew

Degree of Doctor of Philosophy (Ph.D.)
in Electrical and Computer Engineering presented on June 6, 2006.

Adviser: Dr. S. Lawrence Marple, Jr.

AN ABSTRACT OF THE THESIS OF

Ji Seok Liew for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on June 6, 2006.

Title: Multi-Dimensional Sharpening Based on Predictive Bandwidth Extrapolation

Abstract approved: ____

S. Lawrence Marple, Jr.

Achieving sharpened (enhanced detail) features of a multi-dimensional data source using the linear prediction (LP) bandwidth extrapolation (BWE) technique in the transform domain is the main objective of this research. The evolution of sensor technology has provided acquisition scenarios in which the data format is inherently multi-dimensional, including hyperspectral imaging (HSI), interferometric synthetic aperture radar (IF-SAR) imaging, and radar space-time adaptive processing (STAP). In all these applications, fully multi-dimensional signal processing that has the capability (1) to enhance the resolution of the final multi-dimensional analysis result, and (2) to provide reduced-dimension parametric features of the multi-dimensional data for purposes of data encoding/compression is highly desirable.

This thesis provides algorithmic techniques that achieve both capabilities using a novel 2-D BWE approach and its associated fast computational algorithms. Furthermore, 3-D LP fast algorithms were developed, as part of this research, that estimate 3-D AR parameters of the original 3-D sensor domain data, which are then used to produce high resolution 3-D AR spectral estimates. These new 3-D algorithms will become components of future research in 3-D BWE algorithms that sharpen features in 3-D data sources.

Multi-Dimensional Parametric Estimation:
Two-Dimensional Sharpening by Predictive Bandwidth Extrapolation and Fast Algorithms for Three Dimensional Autoregressive Estimation

by

Ji Seok Liew

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented June 6, 2006
Commencement June 2007

Doctor of Philosophy thesis of Ji Seok Liew presented on June 6, 2006.

APPROVED:

Major Professor, representing School of Electrical Engineering and Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Ji Seok Liew, Author

ACKNOWLEDGEMENTS

I appreciate my advisor, S. Lawrence Marple, Jr., for his patience and valuable guidance throughout my graduate study. His criticism and advice made me grow not only as an engineer, but also as a person. I would like to thank the members of my thesis committee Professor Ben Lee, Huaping Liu and Luca Lucchese for their helpful feedback. I would also like to thank to my lab mate Lin Wei for her patience and accessibility whenever I had technical questions.

DEDICATION

I’d like to thank my family for their endless support. I would like to thank my parents for patiently listening to my thoughts on my future and advising me. I would like to thank my wife Jung-Eun for conversing with me about many different issues of my life. Whenever I look at Naomi, my first baby in the world, her cute face stimulates me to work hard.

Finally, I’d like to thank my God for always being there and listening to my countless prayers in my good times and bad times. I could not have completed my graduate study without Him.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (CONTINUED)

Page

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

Figure | Page

LIST OF FIGURES (CONTINUED)

Figure Page

LIST OF FIGURES (CONTINUED)

Figure Page

LIST OF SYMBOLS

Symbol		Page
$\delta[n_1, n_2]$	2-D unit impulse function	16
$h[n_1, n_2]$	2-D system impulse response sequence	16
$x[n_1, n_2]$	2-D input data	16
p_1, p_2	2-D orders along each dimension	18
$a[i, j]$	2-D autoregressive (AR) coefficient	18
$T_1,\ T_2$	2-D sampling intervals along the each dimension	18
f_1, f_2	2-D frequencies along each dimension	18
X_{DTFT}	2-D discrete-time Fourier transform	18
$r_{xx}[k, l]$	2-D autocorrelation sequence	19
$P_{xx}[f_1, f_2]$	2-D power spectrum density (PSD)	21
$\boldsymbol{P}_{\mathrm{xx}}(z_1, z_2)$	2-D Z-transform of the 2-D PSD	20
$\boldsymbol{H}(z_1, z_2)$	2-D Z-transform of the system response function	20
$P_{ARMA}(f_1, f_2)$	PSD of 2-D autoregressive-moving average process	20
ρ_w	White noise variance	20
$b[i, j]$	2-D moving average (MA) coefficient	20
$e[i, j]$	2-D linear prediction error	20
$\underline{\mathbf{a}}_{p_1 p_2}$	2-D block vector AR parameters	22
$\underline{\boldsymbol{\rho}}_{p_1 p_2}$	2-D block vector variance of the errors.	22
$\underline{\mathbf{R}}_{p_1 p_2}$	2-D doubly-block-Toeplitz autocorrelation matrix	22
$\mathbf{a}_{p_1 p_2}$	2-D block vector AR parameters	22

LIST OF SYMBOLS (CONTINUED)

Symbol		Page
$\rho_{p_1 p_2}$	2-D block vector variance of the errors……………………….	22
$\mathsf{R}_{p_1 p_2}$	2-D block-Toeplitz autocorrelation matrix……………………	22
$\mathsf{0}$	2-D block all-zero vector…………………………………..	22
J	2-D reflection matrix…………………………………………	23
I	2-D block identity matrix…………………………………….	24
A_p	2-D block linear prediction parameter matrix………………..	24
P_p	2-D block linear prediction covariance matrix……………….	24
K_p	2-D forward block reflection matrix………………………….	25
$\mathsf{A}_p^{\mathsf{C}}$	2-D column order LP parameter………………………………	29
X^{C}	2-D column order Toeplitz data matrix……………………….	29
$\mathsf{A}_p^{\mathsf{R}}$	2-D row order LP parameter………………………………….	32
$\underline{\mathsf{X}}^{\mathsf{R}}$	2-D row order Toeplitz data matrix…………………………..	32
$x[n_1, n_2, n_3]$	3-D input data………………………………………………..	49
$w[n_1, n_2, n_3]$	3-D white noise………………………………………………	49
$a[i, j, k]$	3-D LP/AR coefficients………………………………………	49
ρ_{LP}	3-D minimized error variance…………………………………	49
p_1, p_2, p_3	3-D orders along each dimension…………….………………….	51
$\underline{\underline{\mathsf{a}}}_{p_1 p_2 p_3}$	3-D super-block vector AR parameters……………………….	55
$\underline{\underline{\rho}}_{p_1 p_2 p_3}$	3-D super-block vector variance of the errors………………...	55
$\underline{\underline{\mathsf{x}}}_{p_1 p_2 p_3}$	3-D super-block vector data matrix…………………………..	55

LIST OF SYMBOLS (CONTINUED)

Symbol		Page
$\underline{\mathsf{J}}$	3-D super-block reflection matrix……………………………	63
$\underline{\mathsf{I}}$	3-D super-block identity matrix………………………………	64
$\underline{\mathsf{K}}_{p}$	3-D super-block reflection coefficient matrix………………..	67
$\underline{\mathsf{A}}_{p}$	3-D super-block linear prediction parameter matrix………….	67
$\underline{\mathsf{P}}_{p}$	3-D super-block linear prediction error covariance matrix……	67
$\underline{\mathsf{e}}^{\mathsf{b}}_{p_1p_2p_3}$	3-D block backward linear prediction error matrices…………	67
$\underline{\mathsf{e}}^{\mathsf{f}}_{p_1p_2p_3}$	3-D block forward linear prediction error matrix…………….	67
$\underline{\mathsf{X}}$	3-D super block Toeplitz data matrix…………………………	68
$\underline{\mathsf{P}}^{\mathsf{f}}_{p_1p_2p_3}$	3-D forward linear prediction squared error matrix…………..	68
$\underline{\mathsf{P}}^{\mathsf{b}}_{p1p2p3}$	3-D backward linear prediction squared error matrix…………	71
$tr\{\underline{\mathsf{P}}^{\mathsf{f+b}}_{p+1}\}$	Trace of 3-D linear prediction squared errors along diagonal...	71

MULTI-DIMENSIONAL PARAMETRIC ESTIMATION: TWO DIMENSIONAL SHARPENING BY PREDICTIVE BANDWIDTH EXTRAPOLATION AND FAST ALGORITHMS FOR THREE DIMENSIONAL AUTOREGRESSIVE ESTIMATION

1. INTRODUCTION

1.1. Research Contribution Overview

Multi-dimensional predictive bandwidth extrapolation (BWE) can be applied either theoretically or experimentally to multi-dimensional data sources, achieving a super-resolution capability by combining multi-dimensional autoregressive (AR) algorithms with a multi-dimensional BWE technique as shown in Figure 1. Examples are shown for one-dimensional (1-D), two-dimensional (2-D), and three-dimensional (3-D) cases of multi-dimensional data. One-dimensional techniques based on BWE have been developed [2,4] to add predicted high frequency content to 1-D signals, where these frequencies were missing, and extended in this thesis to the 2-D case for handling imagery. The effect of the techniques is to sharpen the temporal content (1-D signals) or the spatial content (2-D images). In the past 10 years, we have seen the development of high-resolution 2-D spectral analysis techniques, such as 2-D linear prediction (LP) [1], 2-D autoregressive (AR) [10,22], and 2-D minimum variance (MV) methods [19] with associated fast computational algorithms. These algorithms estimate 2-D LP/AR/MV parameters in the original 2-D signal/image domain in order to produce high-resolution results in the transform domain [1,3,10,22]. The new 2-D BWE, the first research contribution of this thesis, switches the domain of application

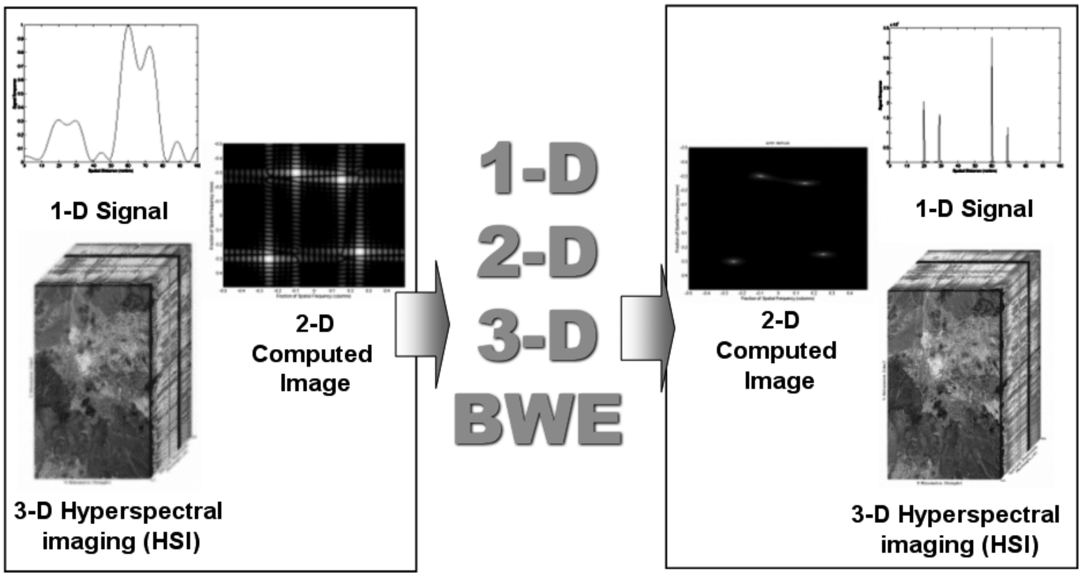


Figure 1. Sharpened and enhanced resolution of various 1-D, 2-D, and 3-D data sources using the BWE technique.

of the 2-D high-resolution techniques from the original signal or image domain to the transform domain in order to produce sharpened results in the original domain. For example, an image would be 2-D transformed, a 2-D LP applied to the spatial frequencies, and then a transform-domain spectral analysis would be performed to produce a sharpened result. Our research contribution to the new 2-D BWE technique [31] is shown in Figure 2. Using existing tools in 2-D, it is possible to branch off into either BWE using a 2-D AR power spectral density (PSD) approach (path #1) or into 2-D LP BWE of the 2-D transform of 2-D data (path #2). For this research, using the existing 2-D tools that were developed by Marple [3,20], it was possible to develop new 2-D BWE technique (path #2).

The second emphasis area of this thesis is an extension of the 2-D BWE technique for computed synthetic aperture radar (SAR) imagery. The 2-D BWE

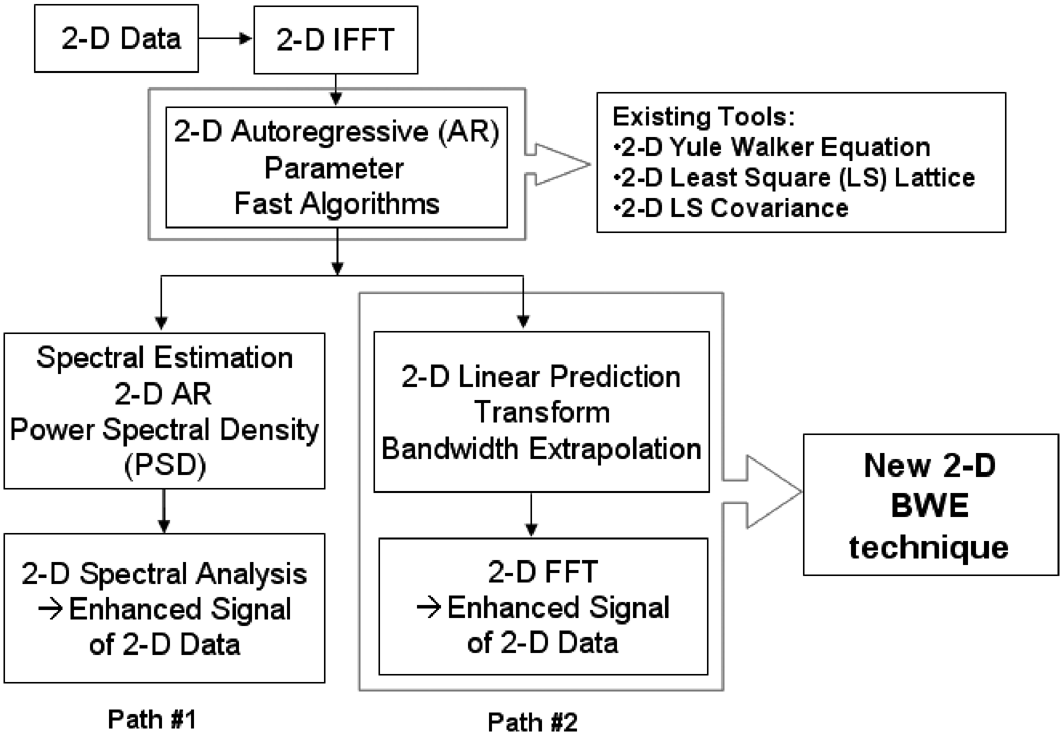

2-D Data
2-D IFFT
2-D Autoregressive (AR) Parameter Fast Algorithms
Existing Tools:
•2-D Yule Walker Equation
•2-D Least Square (LS) Lattice
•2-D LS Covariance
Spectral Estimation 2-D AR Power Spectral Density (PSD)
2-D Spectral Analysis → Enhanced Signal of 2-D Data
2-D Linear Prediction Transform Bandwidth Extrapolation
2-D FFT → Enhanced Signal of 2-D Data
New 2-D BWE technique
Path #1
Path #2

Figure 2. Research contributions for the new 2-D BWE technique are shown as a branch involving 2-D LP Transform BWE (Path #2) in conjunction with existing 2-D AR algorithms.

technique in this case is applied to data already collected in the transform domain. The research approach for application of 2-D BWE techniques to computed synthetic radar imagery is shown in Figure 3. Using radar data that is inherently a Fourier transform in the manner that it is collected, along with existing 2-D algorithms (2-D Yule Walker, 2-D least squares (LS) Lattice, and 2-D LS Covariance) [1,3,19,22] and the newly developed 2-D BWE techniques of this thesis, a novel 2-D synthetic imagery resolution enhancement technique was developed.

The last major emphasis of this thesis is the development of algorithmic

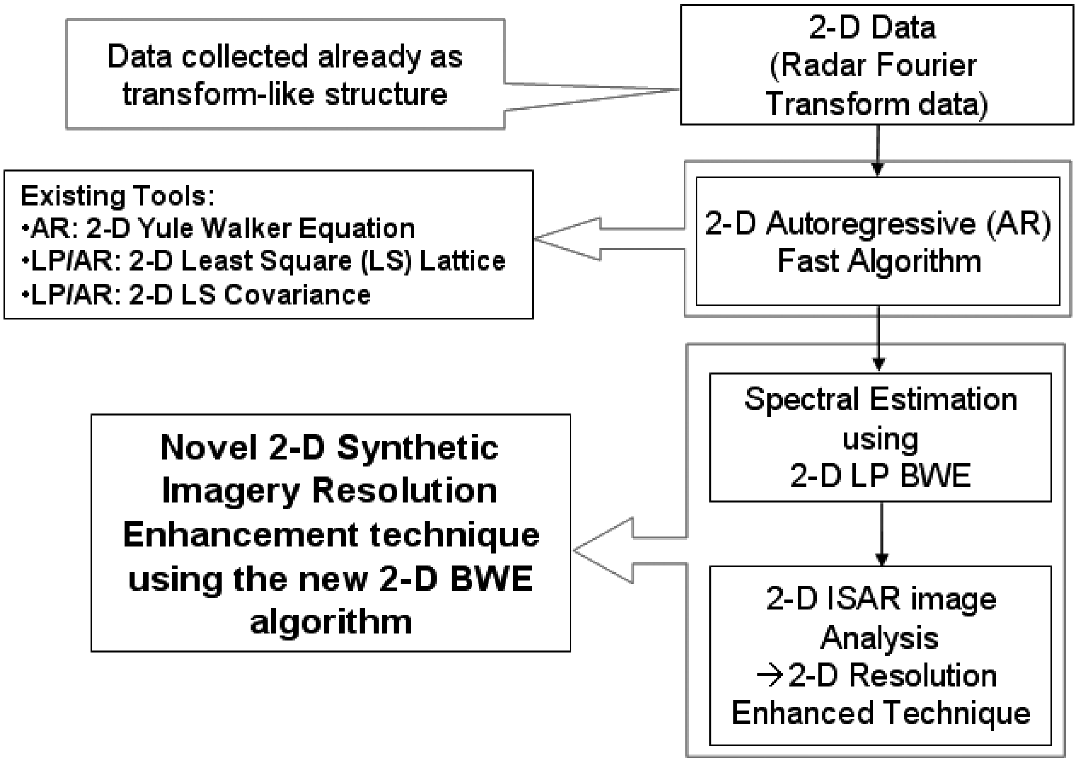

Data collected already as transform-like structure
2-D Data (Radar Fourier Transform data)
Existing Tools:
•AR: 2-D Yule Walker Equation
•LP/AR: 2-D Least Square (LS) Lattice
•LP/AR: 2-D LS Covariance
2-D Autoregressive (AR) Fast Algorithm
Spectral Estimation using 2-D LP BWE
Novel 2-D Synthetic Imagery Resolution Enhancement technique using the new 2-D BWE algorithm
2-D ISAR image Analysis
→2-D Resolution Enhanced Technique

Figure 3. Research contributions using the 2-D BWE technique for computed synthetic radar imagery. Using data collected as transform-like structure with existing 2-D algorithms and the newly developed 2-D BWE techniques, a novel 2-D synthetic imagery resolution enhancement technique was developed.

techniques for fully three-dimensional (3-D) signal processing [28,29,30], which will become a component of a future 3-D BWE technique that will enhance the resolution of a 3-D data source (e.g., HSI imagery). This work has to be developed with a fast algorithm because of the high computational overhead in 3-D signal processing. Without fast algorithms, some of the more exotic methods of spectral estimation cannot be considered for practical applications. Furthermore, this thesis proposes two new 3-D AR fast algorithms that will be components of a future 3-D BWE technique. We shall demonstrate the use of the algorithms for 3-D spectral analysis with the result

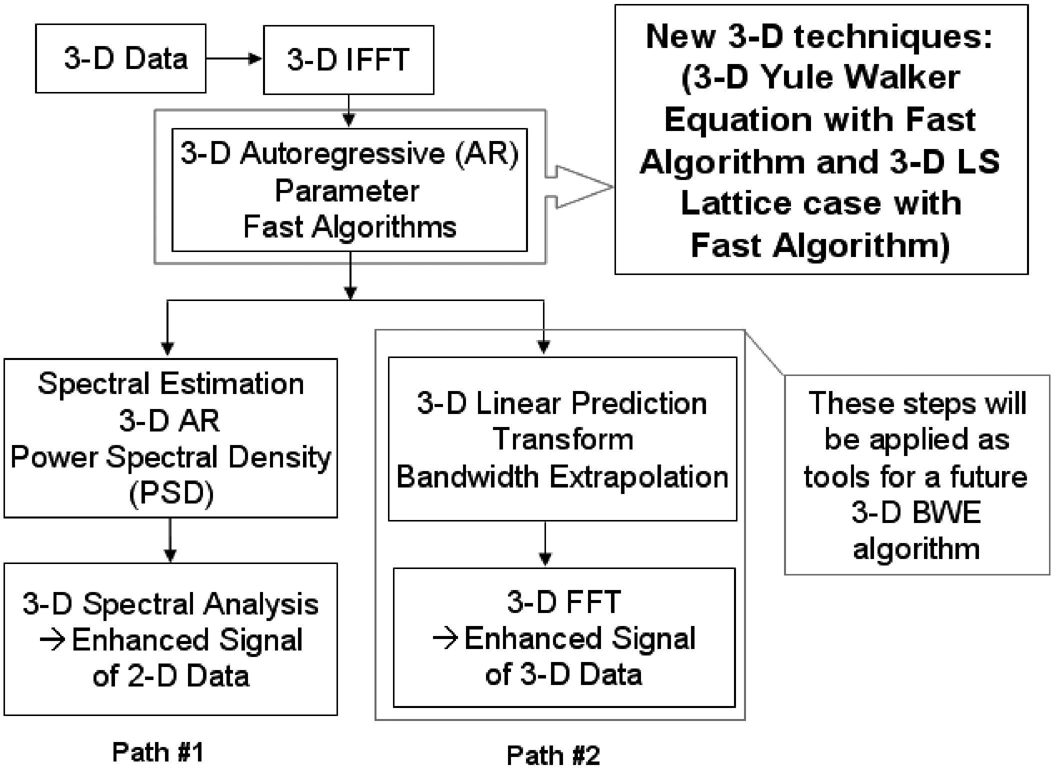

3-D Data
3-D IFFT
3-D Autoregressive (AR) Parameter Fast Algorithms
New 3-D techniques: (3-D Yule Walker Equation with Fast Algorithm and 3-D LS Lattice case with Fast Algorithm)
Spectral Estimation 3-D AR Power Spectral Density (PSD)
3-D Spectral Analysis → Enhanced Signal of 2-D Data
3-D Linear Prediction Transform Bandwidth Extrapolation
3-D FFT → Enhanced Signal of 3-D Data
These steps will be applied as tools for a future 3-D BWE algorithm
Path #1
Path #2

Figure 4. Research contribution for two new 3-D AR fast algorithms for future 3-D BWE technique using 3-D AR fast algorithms. It is possible to develop future work in 3-D LP BWE and the algorithms were tested with quality results in spectral estimation of 3-D AR PSD (Path #1)

that enhanced spectra resolution is achieved over classic 3-D Fourier spectral estimates. This research will contribute to the use in two new 3-D techniques with fast algorithm for a future 3-D BWE technique, as illustrated in Figure 4. Using either the 3-D Yule Walker equation set or the 3-D LS Lattice equation set, each with a fast algorithm, it will be possible to develop the 3-D BWE method in future research. The 3-D LP-based BWE method will sharpen and enhance the resolution of original 3-D signals.

This thesis is organized as follows. The first chapter reviews research contributions, motivations, and concepts of the BWE technique. Some examples of data sources to which the BWE techniques have been applied will be presented in Chapter 1. Related techniques, such as prior 1-D sharpening techniques, 2-D signal processing concepts, and 2-D LP/AR/MV spectral estimation techniques with fast algorithms [1,3,19,22] will be introduced in Chapter 2. Since the adaptation of Marple's techniques to determine the prediction estimator (which is the core of 2-D BWE) is an important part of this thesis, this chapter will be dedicated to an explanation of the process in detail. The concepts of Chapter 2 are crucial since the 3-D AR spectral estimation techniques in Chapter 5 are based on the basic equations and efficient calculation methods found in 2-D AR algorithms. In Chapter 3, an explanation of the process of 2-D sharpening by the new predictive bandwidth extrapolation technique, together with test results, will be presented. In Chapter 4, a variation of the 2-D BWE technique for application to computed synthetic aperture radar (SAR) imagery will be introduced. In Chapter 5, new 3-D AR fast algorithms, which include the 3-D Yule-Walker (YW) method and the 3-D lattice LP parameter estimation method, will be presented. Finally, Chapter 6 will summarize the contributions of the research reported in this thesis.

1.2. Motivation and Concept of Bandwidth Extrapolation (BWE)

One-dimensional BWE extrapolates the bandwidth of the signal in the transform domain to enhance and sharpen the resolution of the original signal in the original domain [4]. This technique extends the 1-D transform of the collected data

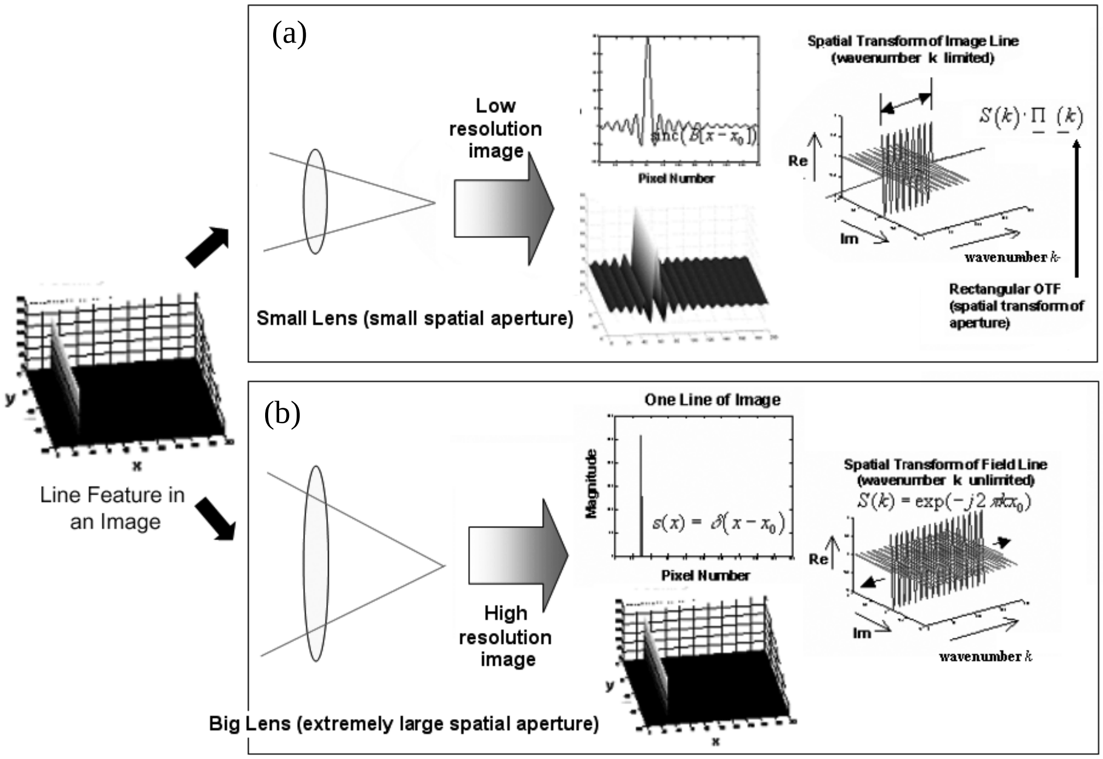


Figure 5. (a) In the case of a small spatial aperture, when analog Fourier transform is applied to this image that shows a sidelobe pattern representing distortion, its spectrum will be bandlimited. (b) If the original image has an extremely sharp line feature with no sidelobe pattern representing distortion as in the case of the ideal big lens (extremely large spatial aperture), its spectrum would have an infinite bandwidth [2].

beyond the observed bandwidth by using a linear prediction model [6,11-18] in the transform domain. Since prediction and estimation techniques are central to the BWE technique, it will be shown that BWE is different from traditional high-pass filtering methods of sharpening a signal [23]. An example of the BWE sharpening approach is shown in Figure 5. An image has bandlimited spatial frequencies (wavenumbers) k when collected from an optical imaging system with finite aperture. In the case of a small spatial aperture, the image through the lens shows a sidelobe pattern

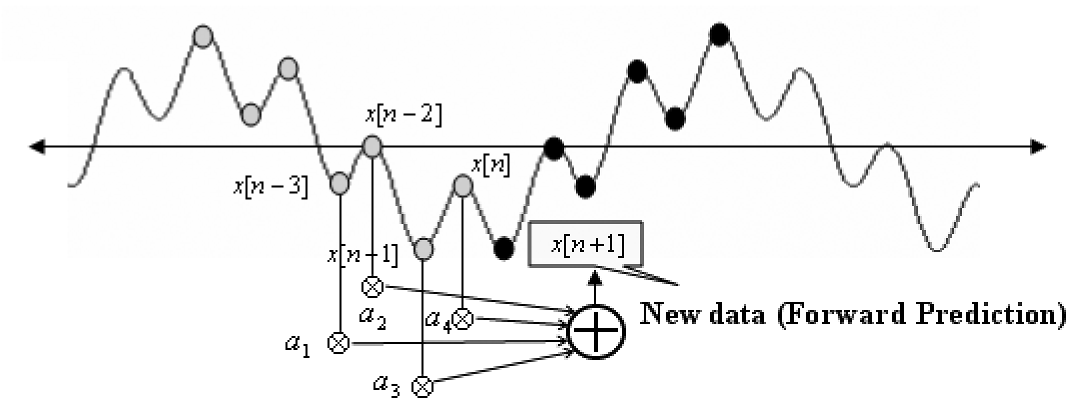


Figure 6. Bandwidth extrapolation scheme (fourth-order prediction). The gray dots represent existing data points, while the black dots represent the forward predicted extension data.

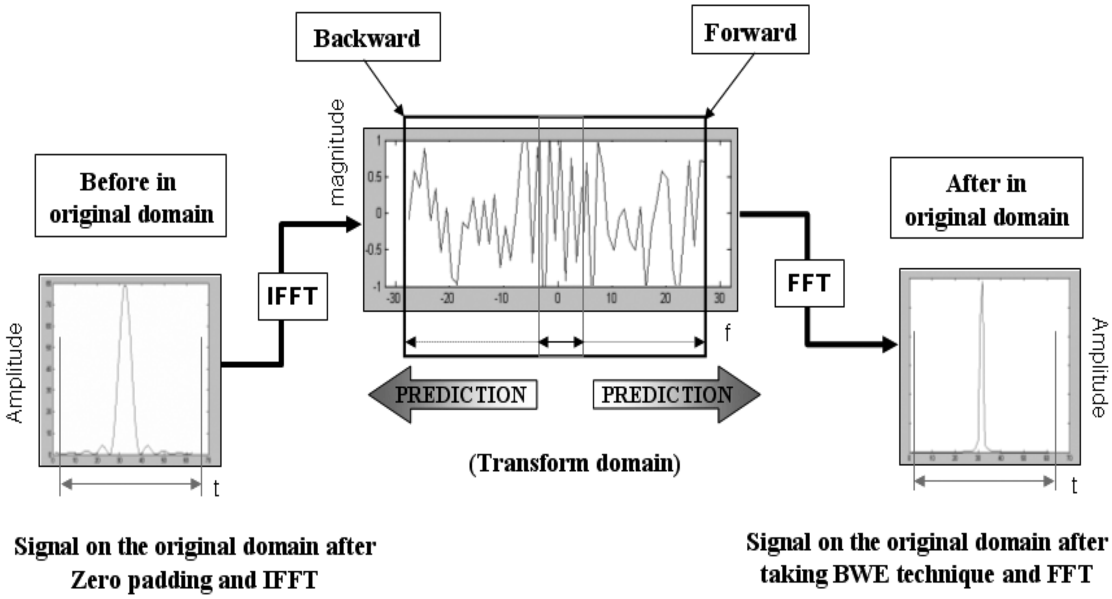

Backward
Forward
magnitude
Before in original domain
After in original domain
IFFT
FFT
Amplitude
Amplitude
1
0.5
0
-0.5
-1
-30
-20
-10
0
10
20
30
f
t
t
PREDICTION
PREDICTION
(Transform domain)
Signal on the original domain after Zero padding and IFFT
Signal on the original domain after taking BWE technique and FFT

Figure 7. A synthetic generated bandlimited data record is used to achieve sharpened 1-D signal. IFFT is first applied to the signal from the original domain, then, using the predictive algorithm, the IFFT signal is extended in the backward and forward direction. A FFT is then applied to the extended transform, which results in an enhanced resolution.

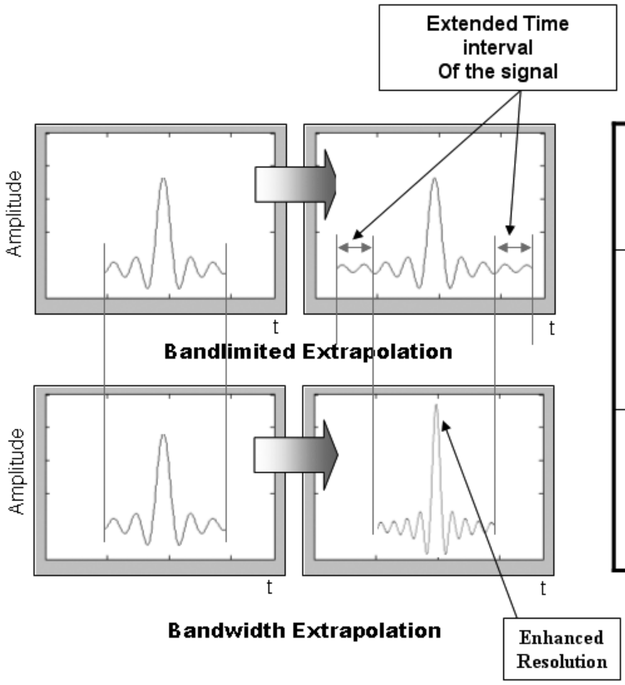
representing distortion, since its spectrum (Fourier transform) will be spatially bandlimited due to the finite aperture. If the original image has an extremely sharp line feature with no sidelobe as in the case of the ideal big lens (extremely large spatial aperture), its spectrum would have an infinite bandwidth. Based on this observation, if a predictive coefficient a can be estimated, then the bandlimited signal *S(k)* can be bandwidth extended in Δk increments $S(k+\Delta k)$ as expressed in Eq (1).

$$S(k+\Delta k) = \exp(-j\pi\Delta k x_0)S(k) = aS(k) \qquad (1)$$

Figure 6 illustrates one example of the forward extrapolation procedure [11]. The gray dots represent existing data points, while the black dots represent the forward predicted extension data. The order of prediction determines the number of past data points needed to determine the new data. For instance, in this figure, a fourth -order prediction scheme is shown, where $x[n+1]$ data is determined using the four previous data $x[n-3]$, $x[n-2]$, $x[n-1]$, and $x[n]$.

In order to present the effect of the high-resolution transform-domain-based 1-D BWE technique, a synthetic idealized bandlimited data record is used to achieve a sharpened 1-D signal, as illustrated in Figure 7. After forming the inverse transform of the original signal, prediction techniques are applied in order to extrapolate the bandlimited transform data. A linear prediction model of the transform data is combined with the original data to synthesize extrapolated transform data beyond the original transform bandlimited extent by using a BWE technique.

Note that our *bandwidth extrapolation* technique should not be confused with

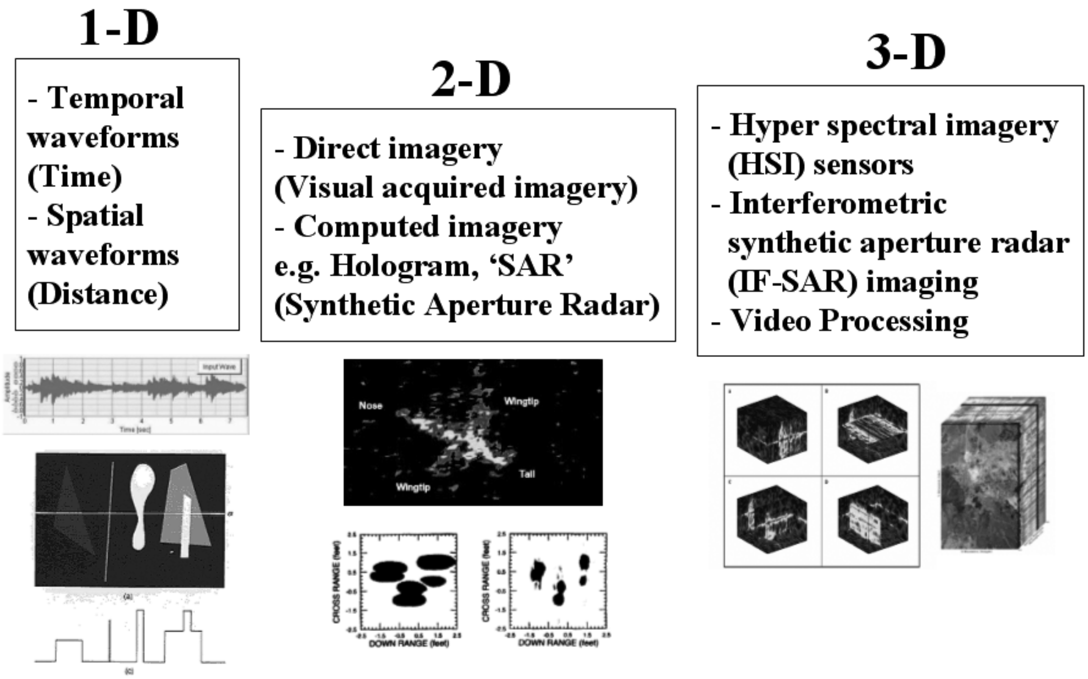


	Change in Time Domain	Change in Frequency Domain
BLE	Extended Time interval of the signal	Enhanced Resolution
BWE	Enhanced Resolution	Extended Bandwidth of The signal

Figure 8. In this figure, changes in either the original domain or the frequency domain are indicated to distinguish between BLE and BWE [11,17]. In BLE, the signal in the original domain is extended which results in the frequency domain having enhanced resolution. In BWE, the frequency domain is extended, which results in the time domain having enhanced resolution.

bandlimited extrapolation concepts of prior literature [11,13,14] that simply extend the time-limited original signal in the time domain without any modification of the frequency bandwidth. The BWE technique, on the other hand, extrapolates the bandwidth of the signal in the frequency domain (transform domain) to enhance and sharpen the resolution of the original signal in the time domain (original domain) without any modification of its time interval [4,12], as shown in the Figure 8. This 1-D BWE concept is extended in this thesis to the new 2-D BWE technique by incorporating the 2-D AR model estimation method of Chapter 3 [3].

Examples of data sources which can be BWE are not limited to time domain



1-D

- Temporal waveforms (Time)
- Spatial waveforms (Distance)

2-D

- Direct imagery (Visual acquired imagery)
- Computed imagery e.g. Hologram, ‘SAR’ (Synthetic Aperture Radar)

3-D

- Hyper spectral imagery (HSI) sensors
- Interferometric synthetic aperture radar (IF-SAR) imaging
- Video Processing

Figure 9. Three categories that represent possible sources of data for predictive application of BWE. In 1-D, temporal waveforms and single array snapshots can be used as data sources [14]. In 2-D direct imagery and computed imagery are available [4,9], while in 3-D HSI, IF-SAR, and video data can be applied [21,24]. In each case, the application of predictive BWE results in enhanced resolution of the original data source.

data. In the 1-D case, temporal waveforms and spatial waveforms can be used. In 2-D, direct imagery, such as visually acquired imagery, and computed imagery, such as SAR, can be used. In the 3-D case, hyperspectral imaging sensors, interferometric synthetic aperture radar imagery, and video data sources can be used with a future 3-D BWE technique [21]. Figure 9 shows the 3 categories that represent possible sources of data for application of prediction-based BWE. A pulsed Doppler radar example application offers insight to understanding distinctions among three possible 1-D signal sharpening techniques [2]: conventional Fourier spectral processing, parametric

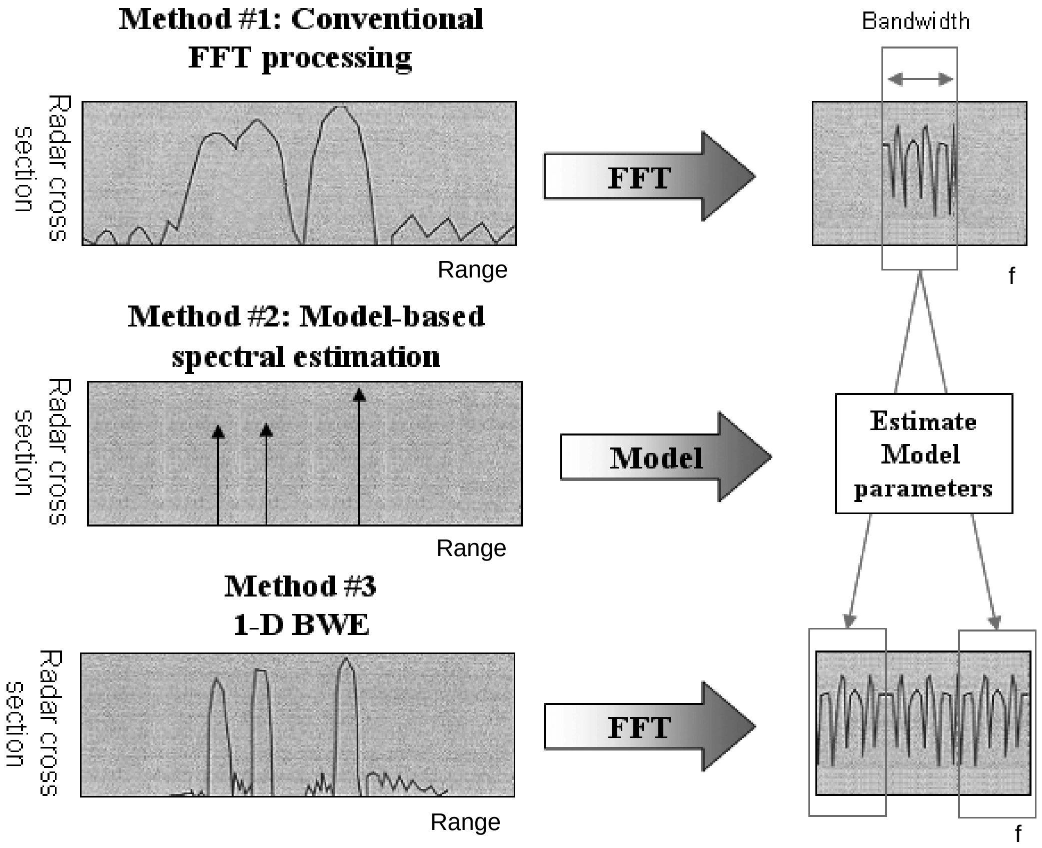
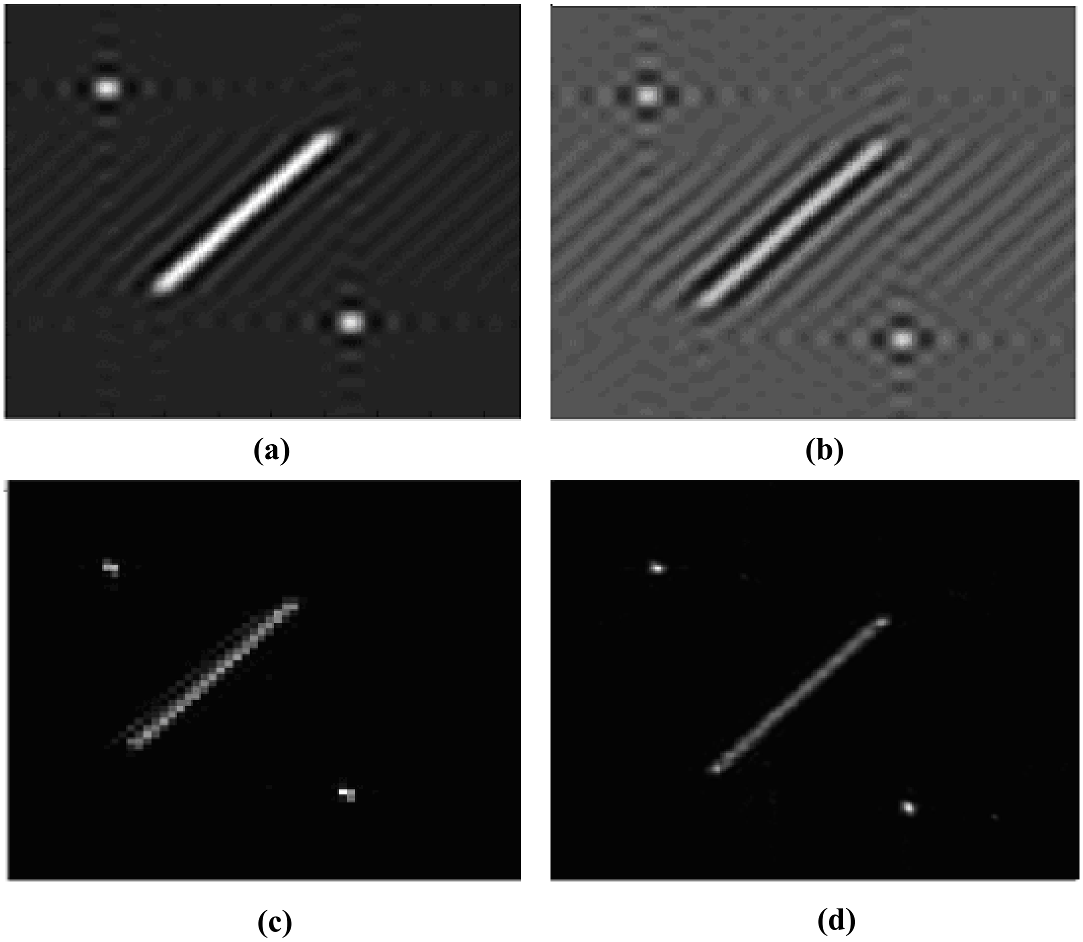


Figure 10. Moore, Zuerndorfer, and Burt from [Lincoln Laboratory Journal [4]] have shown three techniques to determine information regarding target position signal; conventional FFT processing, model-based spectral estimation, and 1-D BWE. The conventional FFT method makes it impossible to separate all the target scatterers. Using the parametric-model method, it is possible to determine all scatterer positions, however the result is limited in information. In the 1-D BWE method, it is possible to separate and determine all scatterer positions, it can also determine all their amplitudes. This is possible due to increased predictive bandwidth in the frequency domain.

spectral estimation techniques, and bandwidth extrapolation. As shown in Figure 10, the top row illustrates the limited range information available from conventional Fourier processing. The model-based technique provides the most information on frequency location while the BWE technique provides both frequency-model data and amplitude data. The middle row illustrates how the conventional model-based



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

Figure 11. Comparison of 2-D image sharpening techniques; (a) Original image. (b) Traditional 2-D highpass convolution filtering increasing the intensity of every high frequency part of the objects (main lobes and side lobes). (c) Row line by row line 1-D based 2-D BWE. (d) New 2-D BWE technique presented in this thesis.

approach primarily yields enhanced target scatterer position information, and the bottom row shows how a more conventional target position profile is constructed by using an extrapolation approach combined with conventional Fourier processing. The model-based technique provides the most information on frequency location while the BWE technique provides both frequency-model data and amplitude data. This 1-D BWE concept is extended in this thesis to the new 2-D BWE technique by combining the 2-D AR model estimation methods and fast algorithms, as will be shown in Chapter 3.

Four examples are shown in Figure 11 that present a comparison of 2-D techniques for image sharpening. Traditional 2-D highpass convolution filtering [23] is the least effective of the three methods. Due to the highpass filtering process, the intensity of all high frequency portions of objects, including both the mainlobe and the sidelobes, are increased. In the case of 1-D BWE applied only along each row line, the enhancement is achieved along only a single dimension, so it is not a true 2-D BWE, and results are not as sharp in both dimensions as a true 2-D BWE. Figure 11 (d) shows the result of the true 2-D BWE technique that is presented in Chapter 3. It is obvious from the figure shown that there is a significant resolution improvement along both dimension, in comparison with Figure 11 (c). In the post-sharpened image, it is now easier to locate the true location of the two points and the diagonal line, since the sidelobes from the previous image have been reduced significantly.

1.3. LP/AR/MV Fast Computational Algorithms

The thesis presents methods for extrapolating bandwidth in the transform domain utilizing mathematical prediction algorithms. We should note that the transform-domain step LP fast algorithms are very important because calculating precise predictors rapidly is important to the BWE technique for realizing the sharpened signals in the original domain. The BWE technique requires a heavy computational burden especially for real-time processing, and reducing the amount of computation is important. Some existing 1-D AR fast algorithms [2,4], which can also provide accurate LP parameter models, can be extended to 2-D techniques. However, extending the 1-D techniques to 2-D has faced certain difficulties due to the differences in the 1-D and 2-D linear systems theory. One problem in particular has been the lack of a capability to factor a 2-D polynomial into lower degree polynomials. The 1-D concept of isolated poles, zeros, and roots does not extend, with certain exceptions, to the 2-D case. Also the computational burden of some advanced 2-D spectral estimation methods has restricted testing and application to small 2-D data sets with simple signal scenarios.

Currently, several techniques, such as the 2-D periodogram and the 2-D AR spectral density estimation have overcome those difficulties and have seen practical application to large 2-D data sets [7-10,19,22]. The study of 2-D high-resolution spectral estimation has been presented in various reports, such as the 2-D lattice LP parameter estimation method [3] and the 2-D modified covariance method of LP [22]. The resulting fast recursive 2-D algorithms for these methods offer significant computational reductions and the techniques are useful for high resolution 2-D spectral analysis applications. These 2-D spectral analysis can be used in space-space data arrays (e.g., image processing), space-time data arrays (e.g., sonar, seismic, and synthetic aperture radar processing), or even time-time data arrays (e.g., analysis of radar pulse repetition interval vs. arrival time). The next chapter, in particular, will present the 2-D method advanced by Marple [1,3,22], which resolves the computational difficulties associated with the existing 2-D methods.

2. 2-D AUTOREGRESSIVE AND LINEAR PREDICTION RANDOM PROCESSES AND FAST ALGORITHMS

2.1. Two-Dimensional Signal Processing

2.1.1. Classical 2-D Linear System and Transform Theory

A 2-D sequence, or array, of real or complex numbers of any function $x[n_1, n_2]$ that is defined for the ordered pair of integers n_1 and n_2 over $-\infty < n_1 < \infty$ and $-\infty < n_2 < \infty$ is a discrete 2-D signal. The 2-D unit impulse function has the following definition,

$$\delta[n_1, n_2] = \delta[n_1]\delta[n_2], \tag{2}$$

and

$$\delta[n_1]\delta[n_2] = \begin{Bmatrix} 1 & for \;\; n_1 = n_2 = 0 \\ 0 & otherwise \end{Bmatrix} \tag{3}$$

The 2-D discrete convolution sum for a 2-D linear shift invariant system is

$$y[n_1, n_2] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h[n_1 - i, n_2 - j] x[i, j] \tag{4}$$

for input $x[n_1, n_2]$ and output $y[n_1, n_2]$. The 2-D sequence $h[n_1, n_2]$, defined over an infinite range in each dimension, is the 2-D system impulse response sequence. It is obtained as the output sequence when the input is $x[n_1, n_2] = \delta[n_1, n_2]$. A 2-D linear shift-invariant system is assumed to be stable if the output sequence remains bounded for all bounded input sequences, but unlike 1-D system stability, the stability of 2-D systems is considerably more difficult to test. Also causality is not inherent for 2-D signals, in which one or both dimensions of the signal are functions of space rather

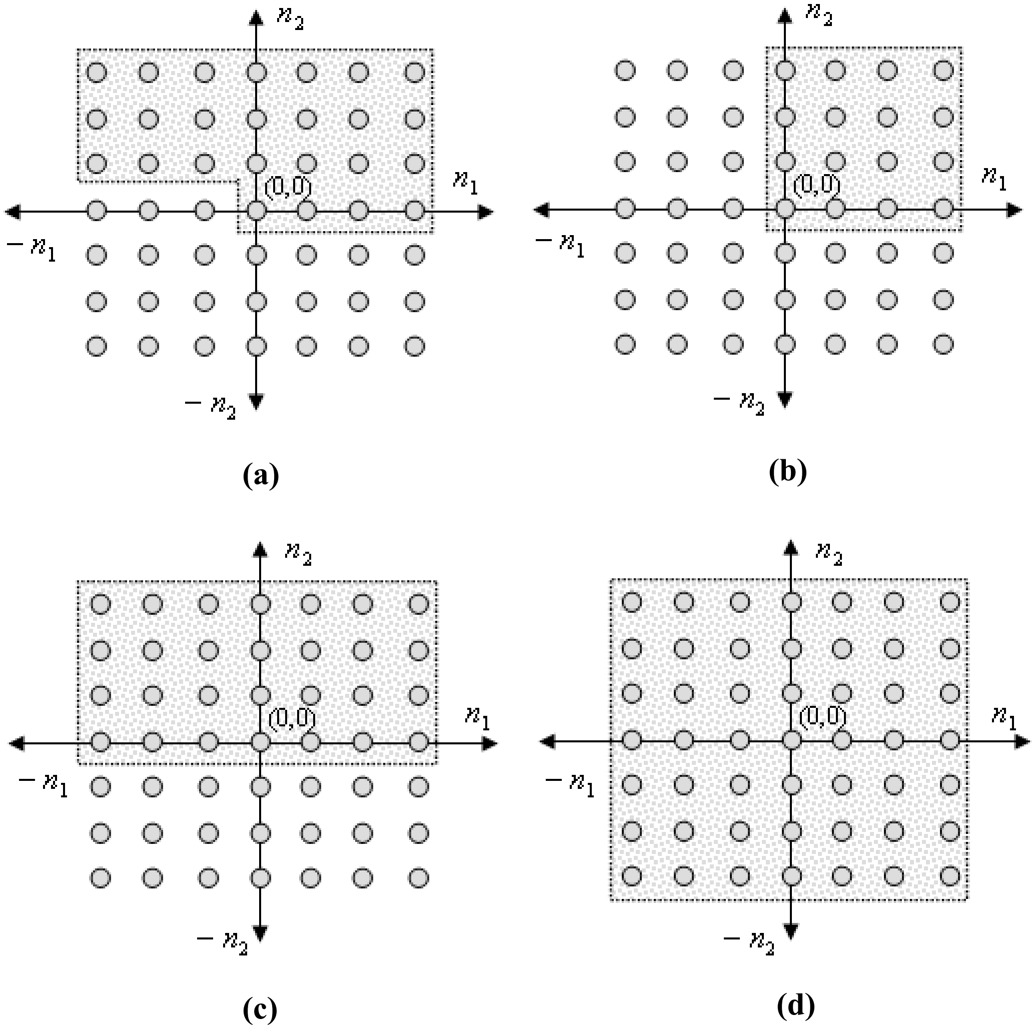


Figure 12. Three regions of support for discrete system sequence $h[n_1, n_2]$ (nonzero in shaded area). (a) Nonsymmetric half plane (NSHP) is that causal. (b) Quarter plane that is causal (QP). (c) Symmetric half plane (SHP) that is noncausal (d) Full plane (FP) that is noncausal.

than time. As presented in Figure 12, one region of support for the 2-D discrete impulse response sequence is the nonsymmetric half plane (NSHP), which can be considered as a causal 2-D system by following certain rastering operations. The symmetric half plane (SHP) is considered to be noncausal since rastering operations do not allow element sequencing in a “causal” manner. If 2-D impulse response is restricted to the quarter plane (QP) as a special case region of support, it too can form a causal 2-D system. The QP case will be the region of support for the purpose of 2-D spectral analysis. A 2-D recursive difference equation relating the input to the output of a 2-D system is

$$\sum_i \sum_j a[i,j]x[n_1 - i, n_2 - j] = \sum_i \sum_j b[i,j]y[n_1 - i, n_2 - j] \qquad (5)$$

By assuming $a[0,0] = 1$, Eq (5) can be rewritten as

$$x[n_1, n_2] = -\sum_i \sum_j a[i,j]x[n_1 - i, n_2 - j] + \sum_i \sum_j b[i,j]u[n_1 - i, n_2 - j] \qquad (6)$$

The range of the various summations will denote the order of the 2-D difference equation. For example, the order of array $a[i,j]$ would be $p_1 \times p_2$, if its region of support was the rectangular QP spanning $0 \le i \le p_1$ and $0 \le j \le p_2$.

Assuming that a continuous signal $x(t_1, t_2)$ is sampled at intervals of T_1 and T_2 along its dimensions to create the sampled sequence $x[n_1, n_2] = x(n_1T_1, n_2T_2)$, then the 2-D discrete-time Fourier transform (2-D DTFT) of the sampled 2-D signal is expressed by

$$X_{DTFT}(f_1, f_2) = T_1T_2 \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x[n_1, n_2]\exp(-j2\pi[f_1n_1T_1 + f_2n_2T_2]). \qquad (7)$$

The squared magnitude of the 2-D discrete-time Fourier transform may be interpreted as the 2-D energy spectral density:

$$Energy = T_1T_2\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}\left|x[n_1,n_2]\right|^2 = \int_{-1/2T_1}^{1/2T_1}\int_{-1/2T_2}^{1/2T_2}\left|X_{DTFT}(f_1,f_2)\right|^2 df_1 df_2 \qquad (8)$$

2.1.2. Classical 2-D Random Process and Spectral Estimation

The concept of 2-D PSD requires 2-D array samples from a 2-D random process to be wide-sense stationary, as in the 1-D case. The 2-D autocorrelation sequence (ACS) will then be a function of only the differential distance between the two sampled points in the 2-D plane. The application of the time average of a single realization of the 2-D random process to yield the 2-D autocorrelation sequence can be justified under the assumption of ergodicity of the 2-D ACS:

$$r_{xx}[n_1,n_2] = \lim_{\substack{N_1\to\infty\\ N_2\to\infty}}\frac{1}{2N_1+1}\frac{1}{2N_2+1}\sum_{i=-N_1}^{N_1}\sum_{j=-N_2}^{N_2}x[i+n_1,\,j+n_2]\,x^*[i,\,j] \qquad (9)$$

Using a similar approach to the one used in the 1-D case, it can be shown [1] that two equivalent representations of the 2-D PSD function are either

$$P_{xx}[f_1,f_2] = \lim_{\substack{N_1\to\infty\\ N_2\to\infty}} \mathrm{E}\left\{\frac{T_1}{2N_1+1}\frac{T_2}{2N_2+1}\left|\sum_{n_1=-N_1}^{N_1}\sum_{n_2=-N_2}^{N_2}x[n_1,n_2]\exp(-j2\pi[f_1n_1T_1+f_2n_2T_2])\right|^2\right\} \qquad (10)$$

or, defined as a 2-D Z-transform of the 2-D ACS $r_{xx}[k,l]$,

$$\boldsymbol{P}_{xx}(z_1,z_2) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}r_{xx}[k,l]\,z_1^{-k}z_2^{-l} \qquad (11)$$

Assuming $\boldsymbol{P}_{yy}(z_1,z_2)$ has a similar Z-transform of $r_{yy}[k,l]$, then the following relationship between $\boldsymbol{P}_{xx}(z_1, z_2)$ and $\boldsymbol{P}_{yy}(z_1, z_2)$ exists,

$$\boldsymbol{P}_{yy}(z_1, z_2) = \boldsymbol{H}(z_1, z_2)\,\boldsymbol{H}^*(1/z_1^*, 1/z_2^*)\,\boldsymbol{P}_{xx}(z_1, z_2) \tag{12}$$

in which $\boldsymbol{H}(z_1, z_2)$ is the Z-transform of the system response function. If the input process is white and the filter has a 2-D transfer function, given by a rational function of 2-D polynomials, then the output process is a 2-D autoregressive-moving average process (2-D ARMA) with PSD of

$$P_{ARMA}(f_1, f_2) = T_1 T_2 \rho_\omega \left| \frac{B(f_1, f_2)}{A(f_1, f_2)} \right|^2 \tag{13}$$

in which

$$\begin{aligned} A(f_1, f_2) &= \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} a[n_1, n_2] \exp(-j2\pi[f_1 n_1 T_1 + f_2 n_2 T_2] \\ B(f_1, f_2) &= \sum_{n_1=-N_1}^{N_1} \sum_{n_1=-N_2}^{N_2} b[n_1, n_2] \exp(-j2\pi[f_1 n_1 T_1 + f_2 n_2 T_2] \end{aligned} \tag{14}$$

are defined over the region of support. If $A(f_1, f_2) = 1$ for all (f_1, f_2), then the PSD represents a 2-D moving average process (2-D MA), and if $B(f_1, f_2) = 1$, then the PSD represents a 2-D autoregressive process (2-D AR).

2.2. Two-Dimensional Autoregressive Spectral Estimation with Fast Algorithms

First popularized by Burg [1], the one-dimensional (1-D) lattice (reflection coefficient) technique of parameter estimation has been extended [2,31] to the two-dimensional (2-D) case and is summarized in this section. The resulting fast recursive 2-D algorithm is a considerable computational simplification over and an estimation improvement on previous attempts to extend the 1-D Burg linear prediction algorithm to 2-D by exploiting some newly discovered matrix structures. The presented technique is useful for high resolution 2-D spectral analysis applications. Furthermore, the newly acquired 2-D recursively updated coefficients can be used as part of the 2-D BWE technique, which will be explained further in Chapter 4.

2.2.1. Two-Dimensional Yule-Walker Equations

Operating directly on the 2-D quarter-plane (QP) LP parameters, the 2-D Yule-Walker equations for a casual 2-D AR process are expressed as [1]:

$$\sum_{i}\sum_{j} a[i,j]\,r_{xx}[k-i,l-j] = \left\{ \begin{matrix} \rho_w & for\ \ [k,l]=[0,0] \\ 0 & otherwise \end{matrix} \right\} \tag{15}$$

Of interest here are the 2-D linear prediction parameters $a[i,j]$ with quarter-plane support. The summation ranges can be chosen to be any one of the four quadrant plane regions. In anticipation of the fast computational algorithm to be presented, we shall assume that subscript $p_1 p_2$ means p_1, the point on the 'n_1' axis, is a variable order parameter. Then, the p_2 point on the 'n_2' axis is assumed to be a fixed order parameter. An alternative representation of the first quadrant Yule-Walker equation is

$$\underline{\boldsymbol{\rho}}^{1}_{p1p2} = \underline{\mathbf{R}}_{p1p2}\,\underline{\mathbf{a}}^{1}_{p1p2}\,. \tag{16}$$

The block vector $\underline{\mathbf{a}}^{1}_{p_1p_2}$ has a superscript 1 to designate it as set as a first quadrant AR parameter vector. It is comprised of (p_1+1) vectors, each vector of which has dimension (p_2+1). The block vector $\underline{\boldsymbol{\rho}}^{1}_{p_1p_2}$ is a 2-D variance of the errors that has all zero entries, with the exception of the top entry expressed in Eq (17).

$$\underline{\boldsymbol{\rho}}^{1}_{p_1p_2} = [\boldsymbol{\rho}^{1\;T}_{p_1p_2} \quad \mathbf{0} \cdots \mathbf{0}\,]^T \tag{17}$$

Eq (17) is also defined in terms of the *vector elements* which are

$$\mathbf{a}^{1}_{p_1p_2}[i] = [\,a^{1}_{p_1p_2}[i,0] \;\; a^{1}_{p_1p_2}[i,1] \;\cdots\; a^{1}_{p_1p_2}[i,p_2]\,]^T \tag{18}$$

and

$$\boldsymbol{\rho}^{1}_{p_1p_2} = [\rho^{1}_{w} \quad 0 \cdots 0\,]^T . \tag{19}$$

The 2-D autocorrelation matrix $\underline{\mathbf{R}}_{p_1p_2}$ has a $(p_1+1)\times(p_1+1)$ block-Toeplitz structure

$$\underline{\mathbf{R}}_{p_1p_2} = \begin{bmatrix} \mathbf{R}_{p_1p_2}[\;0\;] & \cdots & \mathbf{R}_{p_1p_2}[p_1] \\ \vdots & & \vdots \\ \mathbf{R}_{p_1p_2}[-p_1] & \cdots & \mathbf{R}_{p_1p_2}[\;0\;] \end{bmatrix} \tag{20}$$

with $(p_2+1)\times(p_2+1)$ matrix elements $\mathbf{R}_{p_1p_2}[i]$, which also have Toeplitz structure. Thus, Eq (20) can be expressed alternately as

$$\underline{\mathbf{R}}_{p_1p_2}[0] =$$

$$\begin{bmatrix} \begin{bmatrix} r[0,0] & r[0,1] & \cdots & r[0,p_2] \\ r[0,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[0,-p_2] & \cdots & \cdots & r[0,0] \end{bmatrix} \cdots \begin{bmatrix} r[p_1,0] & r[p_1,1] & \cdots & r[p_1,p_2] \\ r[p_1,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[-p_1,-p_2] & \cdots & \cdots & r[p_1,0] \end{bmatrix} \\ \vdots \\ \begin{bmatrix} r[-p_1,0] & r[-p_1,1] & \cdots & r[-p_1,p_2] \\ r[-p_1,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[-p_1,-p_2] & \cdots & \cdots & r[-p_1,0] \end{bmatrix} \cdots \begin{bmatrix} r[0,0] & r[0,1] & \cdots & r[0,p_2] \\ r[0,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[0,-p_2] & \cdots & \cdots & r[0,0] \end{bmatrix} \end{bmatrix} \tag{21}$$

Note that $\mathbf{J}$ is a $(p_1+1)(p_2+1)\times(p_1+1)(p_2+1)$ reflection matrix and $\underline{\mathbf{R}}_{p_1p_2}$ is complex centrosymmetric

$$\mathbf{J}\,\underline{\mathbf{R}}_{p_1p_2}\,\mathbf{J} = (\underline{\mathbf{R}}_{p_1p_2})^* . \tag{22}$$

The above property follows as the result of $\underline{\mathbf{R}}_{p_1p_2}$ being hermitian and Toeplitz. This property allows us to express Eq (16) as

$$\mathbf{J}(\underline{\mathbf{R}}_{p_1p_2})^*\,\mathbf{J}\mathbf{J}\underline{\mathbf{a}}^3_{p_1p_2} = \mathbf{J}\,\underline{\boldsymbol{\rho}}^3_{p_1p_2} \tag{23}$$

Observing that $\mathbf{J}\,\underline{\boldsymbol{\rho}}^3_{p_1p_2} = \underline{\boldsymbol{\rho}}^1_{p_1p_2}$ and using the Eq (22) relationship, Eq (23) can be expressed as

$$\underline{\mathbf{R}}_{p_1p_2}(\mathbf{J}\underline{\mathbf{a}}^3_{p_1p_2})^* = (\underline{\boldsymbol{\rho}}^1_{p_1p_2})^* = \underline{\boldsymbol{\rho}}^1_{p_1p_2} \tag{24}$$

Note that $(\underline{\boldsymbol{\rho}}^1_{p_1p_2})^* = \underline{\boldsymbol{\rho}}^1_{p_1p_2}$ because the variance is real. By comparing Eq (16) and Eq (24), it can be deduced that the third quadrant AR parameters are complex conjugates of the first quadrant AR parameters. Using the same approach, we can determine that the fourth quadrant AR parameters are complex conjugates of the second quadrant AR parameters.

A fast computational algorithm for the solution is not based on the direct solution for the 2-D AR parameters, but is based on first solving a special variant of the 2-D AR algorithm with the form

$$\left[\mathbf{I}\ \ \mathbf{A}_{p1p2}[1]\cdots\cdots\mathbf{A}_{p1p2}[p_2]\right]\underline{\mathbf{R}}_{p1p2} = \left[\mathbf{P}_{p1p2}\ \ \mathbf{0}\cdots\cdots\mathbf{0}\ \right] \tag{25}$$

in which $\mathbf{I}$ is a $(p_1+1)\times(p_1+1)$ identity matrix, and block linear prediction parameter matrices $\mathbf{A}_{p_1p_2}[i]$ for $0\le i\le p_1$ and block linear prediction covariance matrix $\mathbf{P}_{p_1p_2}$ have dimensions $(p_1+1)\times(p_1+1)$. We obtain the 2-D AR parameters by solving

$$\mathbf{a}^{1}_{p_1p_2}[0]=[1\ \ 0\ \ \cdots\ \ 0][\mathbf{P}_{p_1p_2}]^{-1} \tag{26}$$

and, scaling such that $a^{1}[0,\,0]=1$, then

$$\mathbf{a}^{1}_{p_1p_2}[k]=\mathbf{a}^{1}_{p_1p_2}[0]\,\mathbf{A}_{p_1p_2}[i]\quad \mathit{for}\ 1\le i\le p_1. \tag{27}$$

Thus, by solving for matrices $\mathbf{P}_{p_1p_2}$ and $\mathbf{A}_{p_1p_2}[i]$ for $i=1$ to p_1 from Eq (25) and applying Eq (27), one can get the 2-D AR parameters. Similarly, solving for and scaling to force $a^{4}[0,\,0]=1$, one can compute the fourth quadrant parameters.

$$\mathbf{a}^{4}_{p_1p_2}[0]=[0\ \cdots\ 0\ \ 1][\mathbf{P}_{p_1p_2}]^{-1}$$

and

$$\mathbf{a}^{4}_{p_1p_2}[k]=\mathbf{a}^{4}_{p_1p_2}[0]\,\mathbf{A}_{p_1p_2}[i]\quad \mathit{for}\ 1\le i\le p_1. \tag{28}$$

2.2.2. Recursive Solution for 2-D Block LP Parameter Matrices

We can show [1] that the 2-D autocorrelation matrix $\mathbf{R}_p$ is hermitian $\mathbf{R}_p=\mathbf{R}_p^H$ and centrosymmetric $\mathbf{R}_p=\mathbf{J}\mathbf{R}_p^{*}\mathbf{J}$, since $\underline{\mathbf{R}}_p=\underline{\mathbf{R}}_p^H$ is block Toeplitz. Then we can find the 2-D reflection coefficient matrix, such that the following expression is valid

$$\begin{gathered}\left[\mathbf{I} \;\; \mathbf{A}_{p+1}[1] \cdots \mathbf{A}_{p+1}[p] \;\; \mathbf{A}_{p+1}[p+1]\right] = \\ \left[[\mathbf{I} \quad \mathbf{A}_p[1] \cdots \mathbf{A}_p[p] \quad \mathbf{0}\,]\right] + \left[\mathbf{K}_{p+1}[\mathbf{0} \;\; \mathbf{J}\mathbf{A}^*_p[p]\mathbf{J}\cdots\mathbf{J}\mathbf{A}^*_p[1]\mathbf{J} \;\; \mathbf{I}\right]\end{gathered} \tag{29}$$

If we multiply both sides on the right by $\mathbf{R}_p$ at order $p+1$, this will yield

$$[\mathbf{P}_{p+1} \;\; \mathbf{0} \cdots \mathbf{0}\;] = [\mathbf{P}_p \quad \mathbf{0} \quad \cdots \quad \mathbf{0} \quad \Delta_{p+1}] + \mathbf{K}_{p+1}[\;\mathbf{J} \;\; \underline{\Delta}^*_{p+1}\mathbf{J} \quad \mathbf{0} \cdots\cdots \mathbf{0} \quad \mathbf{J}\,\mathbf{P}^*_p\mathbf{J}\;] \tag{30}$$

where $\mathbf{K}_{p+1}$ is the forward block reflection matrix and Δ_{p+1} is

$$\Delta_{p+1} = \underline{\mathbf{R}}_p[p+1] + \sum_{i=1}^{p} \mathbf{A}_p[i]\underline{\mathbf{R}}_p[p+1-i]. \tag{31}$$

Also, Eq (29) will be balanced if we select

$$\mathbf{K}_{p+1} = \mathbf{A}_{p+1}[p+1] = -\Delta_{p+1}(\mathbf{J}\,\mathbf{P}^*\,\mathbf{J})^{-1} = -\Delta_{p+1}\,\mathbf{J}\,(\mathbf{P}^*)^{-1}\,\mathbf{J} \tag{32}$$

which creates the following order-update recursion

$$\mathbf{A}_{p+1}[i] = \mathbf{A}_p[i] + \mathbf{K}_{p+1}\,(\mathbf{J}\,\mathbf{A}^*_p[p+1-i]\mathbf{J})\,. \tag{33}$$

From Eq (30) and Eq (32), it is possible to derive the following recursion of the covariance matrix

$$\mathbf{P}_{p+1} = (\mathbf{I} - \mathbf{K}_{p+1}[\mathbf{J}\,\mathbf{K}^*_{p+1}\mathbf{J}])\mathbf{P}_p = \mathbf{P}_p(\mathbf{I} - [\mathbf{J}\,\mathbf{K}^T_{p+1}\mathbf{J}]\mathbf{K}^H_{p+1})\,. \tag{34}$$

The backward matrices $\mathbf{B} = \mathbf{J}\,\mathbf{A}^*_p\mathbf{J}$ are obtained by the pre and post multiplication of forward matrices by the reflection matrix. Therefore, only the forward block linear prediction matrix, forward block linear prediction covariance matrix, and the forward block reflection matrix need to be computed in the 2-D case. This reduces the recursive updates by half in the 2-D case.

3. 2-D SHARPENING BASED ON PREDICTIVE BWE USING 2-D LP ESTIMATION TECHNIQUE

There have been many attempts to develop different versions of the 2-D BWE technique [3-6]. However, those techniques have had computational disadvantages, because one has to alternatively take the Fourier transform of each column and row of the data matrix simultaneously. Also, the different values of the extended samples that border with each side of the original data samples and the missing data samples located at each corner from extrapolation might introduce unwanted errors. Note that such errors cause the algorithm to fail to achieve an enhanced resolution. Based on our new perspective, we propose an alternative 2-D spectrogram matrix array using linear transform operations on data samples, from which an opportunity to apply a highly sharpened 2-D spectral analysis procedure is created [31]. In this chapter, we will present a new 2-D BWE algorithm that extrapolates a finite-extent 2-D data matrix by not just a single sample at a time, but instead extrapolates multiple samples at a time off one edge of the 2-D data matrix. Furthermore, additional refinements of this procedure reduce unwanted extrapolation error, which is caused by both bordering problems and missing extended data samples. A block diagram for the new BWE processing that enhances the resolution of images is illustrated in Figure 13.

3.1. New 2-D Bandwidth Extrapolation

Once we get the 2-D coefficient set from Eq (33), the multiple samples may be predicted, or extended by a linear combination of its neighboring samples with

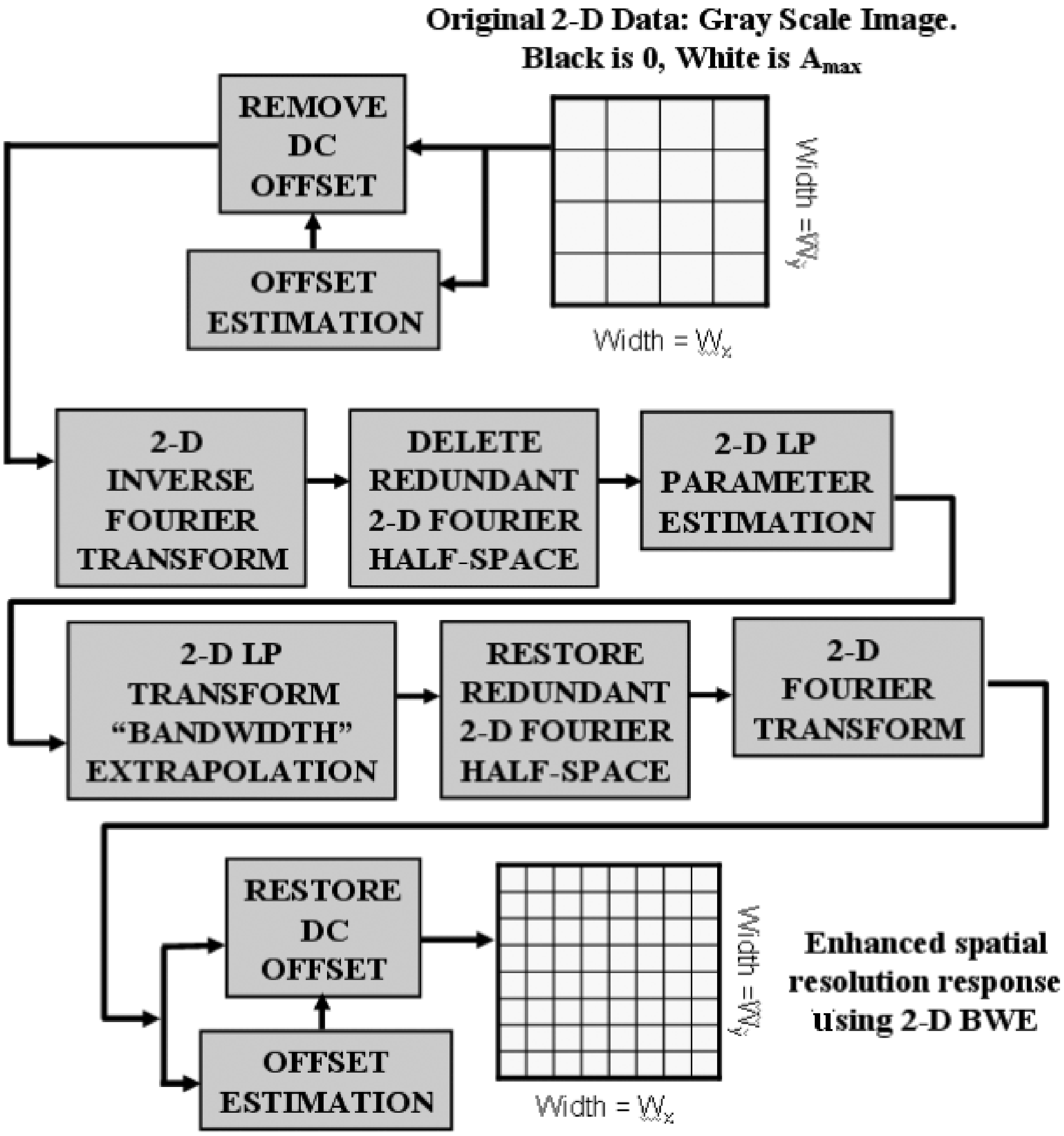

Original 2-D Data: Gray Scale Image.
Black is 0, White is A_{max}
REMOVE DC OFFSET
OFFSET ESTIMATION
Width = W_y
Width = W_x
2-D INVERSE FOURIER TRANSFORM
DELETE REDUNDANT 2-D FOURIER HALF-SPACE
2-D LP PARAMETER ESTIMATION
2-D LP TRANSFORM "BANDWIDTH" EXTRAPOLATION
RESTORE REDUNDANT 2-D FOURIER HALF-SPACE
2-D FOURIER TRANSFORM
RESTORE DC OFFSET
OFFSET ESTIMATION
Width = W_y
Width = W_x
Enhanced spatial resolution response using 2-D BWE

Figure 13. Flow chart of 2-D BWE processing with a gray scale image. First the original 2-D data is preprocessed to remove DC offset. A 2-D inverse Fourier transform is next performed, and due to its resulting conjugate symmetry, the redundant half is deleted. Next 2-D LP parameter estimation is performed, and the BWE technique is applied. Finally the missing conjugate symmetrical half is restored and a 2-D Fourier Transform is applied. Restoration of the offset to bring the image back to its original gray scale level is made, resulting in an enhanced spatial resolution image.

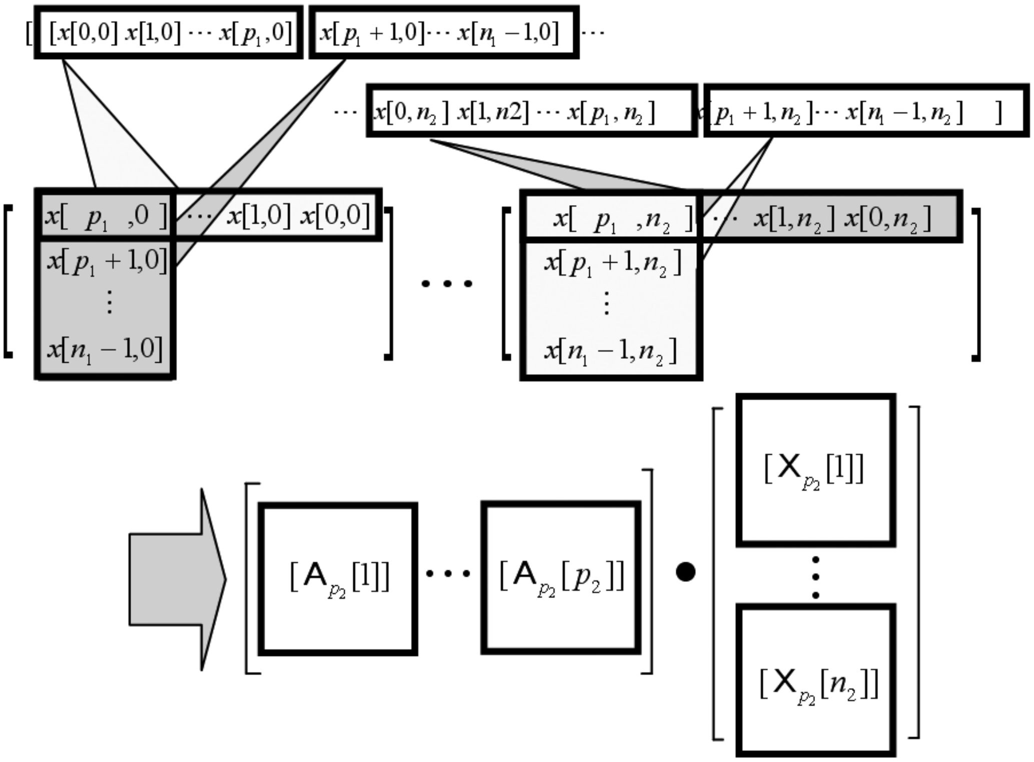

[[x[0,0] x[1,0] ⋯ x[p_1,0]] x[p_1+1,0] ⋯ x[n_1−1,0] ⋯
⋯ x[0,n_2] x[1,n2] ⋯ x[p_1,n_2] x[p_1+1,n_2] ⋯ x[n_1−1,n_2]]
x[p_1,0] ⋯ x[1,0] x[0,0]
x[p_1+1,0]
x[n_1−1,0]
x[p_1,n_2] ⋯ x[1,n_2] x[0,n_2]
x[p_1+1,n_2]
x[n_1−1,n_2]
[A_{p_2}[1]] ⋯ [A_{p_2}[p_2]]
[X_{p_2}[1]]
[X_{p_2}[n_2]]

Figure 14. The original data is in a vector form, and is partitioned and arranged into Toeplitz structured matrices

updated A_p. Using the quarter-plane model, the coefficients can be predicted by four different prediction filters of row order p_1 and column order p_2. To perform the 2-D column-order-forward-LP extrapolation on the first quadrant, the following equation is used

$$\underline{\mathsf{X}}^{\mathsf{c}}[n_2] = \sum_{j=1}^{p_2} \mathsf{A}_{p_2}^{\mathsf{c}}[j]\, \underline{\mathsf{X}}^{\mathsf{c}}[n_2 - j] \qquad (35)$$

in which p_2 has to fall between n_1 and n_2 in value, and $\mathsf{A}_{p_2}^{\mathsf{c}}$ is the original data set discussed in Chapter 2. Eq (35) for the QP support regions can be arranged, or ordered, into at least two convenient block matrix forms by constructing the 2-D parameters either by rows or by columns.

$$\left[\underbrace{\underbrace{\overbrace{\overbrace{x[0,0]\cdots x[p_1,0]}^{p_1+1}\cdots x[n_1-1,0]}^{n_1}\cdots\cdots x[n_1-1,p_2]}_{n_1\cdot(p_2+1)}\cdots\cdots x[n_1-1,n_2]}_{n_1\cdot(n_2+1)}\right] \tag{36}$$

Note that the original data set, Eq (36), is a 1-D vector of length $n_1\cdot(n_2+1)$. $\mathsf{A}^{\mathsf{c}}_{p_2}$ is a 2-D block vector of appropriate dimension to match the matrix multiplication block size of Eq (35). Eq (36) is partitioned into a set of block vectors

$$\left[\mathsf{X}^{\mathsf{c}}[0]\ \cdots\ \mathsf{X}^{\mathsf{c}}[p_2]\ \cdots\ \mathsf{X}^{\mathsf{c}}[n_2]\right], \tag{37}$$

where each block vector is constructed as depicted in Eq (38), and as illustrated in Figure 14,

$$\mathsf{X}^{\mathsf{c}}[0]=\begin{bmatrix} x[p_1,0] & \cdots & x[n_1-1,0] \\ \vdots & \ddots & \vdots \\ x[\ 0,\ 0] & \cdots & x[n_1-p_1-1,0] \end{bmatrix},$$

$$\mathsf{X}^{\mathsf{c}}[p_2]=\begin{bmatrix} x[p_1,p_2] & \cdots & x[n_1-1,p_2] \\ \vdots & \ddots & \vdots \\ x[\ 0,\ p_2] & \cdots & x[n_1-p_1-1,p_2] \end{bmatrix}, \text{ and}$$

$$\mathsf{X}^{\mathsf{c}}[n_2]=\begin{bmatrix} x[p_1,n_2] & \cdots & x[n_1-1,n_2] \\ \vdots & \ddots & \vdots \\ x[\ 0,\ n_2] & \cdots & x[n_1-p_1-1,n_2] \end{bmatrix} \tag{38}$$

starting with $\mathsf{X}^{c}[0]$, all data set values for which $p_2 = 0$ is used to construct the first column and row of each block vector, as shown in Eq (31). After constructing the first column and row of each block vector, it is possible to construct the rest of the block vector as a Toeplitz matrix as shown in Figure 14 which is linked with Eq (39). The sequential block vectors are created in the same manner for p_2 = 1, 2,.. n_2. With Toeplitz matrices, it is possible to obtain a matrix which has been extended by one new vector term in the forward column direction.

$$\begin{bmatrix} x[p_1, n_2] & \cdots & \overset{\textbf{New term}}{\boxed{x[n_1-1, n_2]}} \\ \vdots & \ddots & \vdots \\ x[\,0,\, n_2] & \cdots & x[n_1-p_1-1, n_2] \end{bmatrix} = \mathsf{A}_{p_2}[1]\,\underline{\mathsf{X}}^{c}[n_2-1] + \cdots + \mathsf{A}_{p_2}[p_2]\,\underline{\mathsf{X}}^{c}[n_2-p_2] \tag{39}$$

are defined in terms of block matrix entries A_{p_2} and $\underline{\mathsf{X}}^{c}$.

$$\overbrace{\begin{bmatrix} \overbrace{\boxed{\mathsf{A}_{p_2}[1]}}^{(p_1+1)\times(p_1+1)} & \cdots & \boxed{\mathsf{A}_{p_2}[p_2]} \end{bmatrix}}^{p_2}_{\;for\, 1 \le j \le p_2} \bullet \begin{bmatrix} \overbrace{\begin{bmatrix} x[p_1, n_2-1] & \cdots & x[n_1-1, n_2-1] \\ \vdots & \ddots & \vdots \\ x[0, n_2-1] & \cdots & x[n_1-p_1-1, n_2-1] \end{bmatrix}}^{(n_1-p_1-1)} \\ \vdots \\ \begin{bmatrix} x[p_1, n_2-p_2] & \cdots & x[n_1-1, n_2-p_2] \\ \vdots & \ddots & \vdots \\ x[0, n_2-p_2] & \cdots & x[n_1-p_1-1, n_2-p_2] \end{bmatrix} \end{bmatrix} \tag{40}$$

Note that a new term, $x[n_1-1, n_2]$, is shown in Eq (39). By taking the reverse steps to extract the relevant data set from the column extended data set, it can be shown that the new term is the missing term not recovered by other BWE techniques

[25]. Also, by using the technique of reference [2], each line does not need to be calculated individually, resulting in a fast algorithm when finding individual extended data sets. To perform the 2-D row order forward LP extrapolation on the first quadrant, similar steps can be used involving the extrapolation of the column vector, with the difference being that we use the extended column data set from the column extrapolation rather than the original data set. The following equation is used for row extrapolation operations:

$$\underline{X}^{R}[n_1] = \sum_{i=1}^{p_1} A_p^{R}[i]\, \underline{X}^{R}[n_1 - i] \tag{41}$$

where superscript ^{R} denotes *row*, and the value of p_1 has to fall between n_1 and n_2.

$$\left[\underbrace{\underbrace{\overbrace{\overbrace{x[0,0]\cdots x[0,p_2]}^{p_2+1}\cdots x[0,n_2-1]}^{n_2}\cdots\cdots x[p_1,n_2-1]}_{n_2\cdot(p_1+1)}\cdots\cdots x[n_1,n_2-1]}_{n_2\cdot(n_1+1)}\right] \tag{42}$$

Eq (41) can be rewritten into a set of block vectors ordered by rows

$$\left[X^{R}[0] \quad \cdots \quad X^{R}[p_1] \quad \cdots \quad X^{R}[n_1]\right] \tag{43}$$

where each block vector is

$$X^{R}[0] = \left[\begin{array}{ccc} x[0,p_2] & \cdots & x[0,n_2-1] \\ \vdots & \ddots & \vdots \\ x[\,0,\,0] & \cdots & x[0,n_2-p_2-1] \end{array}\right],$$

$$\mathsf{X}^{\mathsf{R}}[p_1]=\begin{bmatrix} x[p_1,p_2] & \cdots & x[p_1,n_2-1] \\ \vdots & \ddots & \vdots \\ x[p_1,0] & \cdots & x[p_1,n_2-p_2-1] \end{bmatrix}, \text{ and}$$

$$\mathsf{X}^{\mathsf{R}}[n_1]=\begin{bmatrix} x[n_1,p_2] & \cdots & x[n_1,n_2-1] \\ \vdots & \ddots & \vdots \\ x[n_1,0] & \cdots & x[n_1,n_2-p_2-1] \end{bmatrix}. \tag{44}$$

Applying the same methods that were used for column extrapolation, Eq (44) terms are derived from Eq (42) terms. However, when Eq (45) below is carried out, there are two new terms appearing in the resulting extended row data set, as shown:

$$\begin{bmatrix} x[p_1,n_2] & \cdots & \overset{\textbf{New term}}{\boxed{x[n_1-1,n_2]}} \\ \vdots & \ddots & \vdots \\ x[\,0,\,n_2] & \cdots & x[n_1-p_1-1,n_2] \end{bmatrix} = \mathsf{A}_{p_1}[1]\,\underline{\mathsf{X}}^{\mathsf{R}}[n_1-1]+\cdots+\mathsf{A}_{p_1}[p_1]\,\underline{\mathsf{X}}^{\mathsf{R}}[n_1-p_1] \tag{45}$$

$$\Rightarrow \underset{for\ 1\le i\le p_1}{\overbrace{\begin{bmatrix} \overbrace{\boxed{\mathsf{A}_{p_1}[1]}}^{(p_2+1)\times(p_2+1)} & \cdots & \boxed{\mathsf{A}_{p_1}[p_1]} \end{bmatrix}}^{p_1}} \bullet \begin{bmatrix} \overbrace{\begin{bmatrix} x[n_1-1,p_2] & \cdots & x[n_1-1,n_2] \\ \vdots & \ddots & \vdots \\ x[n_1-1,0] & \cdots & x[n_1-1,n_2-p_2] \end{bmatrix}}^{(n_2-p_2-1)} \\ \vdots \\ \begin{bmatrix} x[n_1-p_1,p_2] & \cdots & x[n_1-p_1,n_2] \\ \vdots & \ddots & \vdots \\ x[n_1-p_1,0] & \cdots & x[n_1-p_1,n_2-p_2] \end{bmatrix} \end{bmatrix}$$

As mentioned above, the two new resulting terms, $x[n_1,n_2-1]$ and $x[n_1,n_2]$, recover the missing terms that were not available in other BWE techniques [25]. Note this method can keep going forward and cover all terms in the time domain's first quadrant with the knowledge of a fixed order. Similarly, the backward extrapolation on the third quadrant is defined by column and row recursions:

$$\begin{aligned}\underline{\mathbf{X}}^{\mathbf{C}}[n_2 - p_2] &= \sum_{j=1}^{p_2} (\mathbf{JA}^{*}_{p_2}[j]\mathbf{J})\underline{\mathbf{X}}^{\mathbf{C}}[n_2 - p_2 + j] \\ &= (\mathbf{JA}^{*}_{p_2}[1]\mathbf{J})\underline{\mathbf{X}}^{\mathbf{C}}[n_2 - p_2 + 1] + \cdots + (\mathbf{JA}^{*}_{p_2}[p_2]\mathbf{J})\underline{\mathbf{X}}^{\mathbf{C}}[n_2]\end{aligned} \tag{46}$$

and

$$\begin{aligned}\underline{\mathbf{X}}^{\mathbf{R}}[n_1 - p_1] &= \sum_{i=1}^{p_1} (\mathbf{JA}^{*}_{p_1}[i]\mathbf{J})\underline{\mathbf{X}}^{\mathbf{R}}[n_1 - p_1 + i] \\ &= (\mathbf{JA}^{*}_{p_1}[1]\mathbf{J})\underline{\mathbf{X}}^{\mathbf{R}}[n_1 - p_1 + 1] + \cdots + (\mathbf{JA}^{*}_{p_1}[p_1]\mathbf{J})\underline{\mathbf{X}}^{\mathbf{R}}[n_1]\end{aligned} \tag{47}$$

where p_1 and p_2 have to be between n_1 and n_2. The steps to determine the forward column extrapolation and row extrapolations can be used to determine the backward column extrapolation and row extrapolation using Eq (40) and (41), and the 2-D fast algorithm from [3]. This allows the extrapolation of the original data in the backward column and backward row direction as shown in Figure 15.

With the combination of the forward and backward extrapolation of columns and rows, it is possible to extend the original data set in all four directions, as shown in Figure 15. This proposed BWE technique not only recovers three missing terms, which are not recovered with other proposed 2-D BWE techniques [5,25], but also provides a fast algorithm to reduce the computational operations needed to estimate forward and backward prediction coefficient matrices.

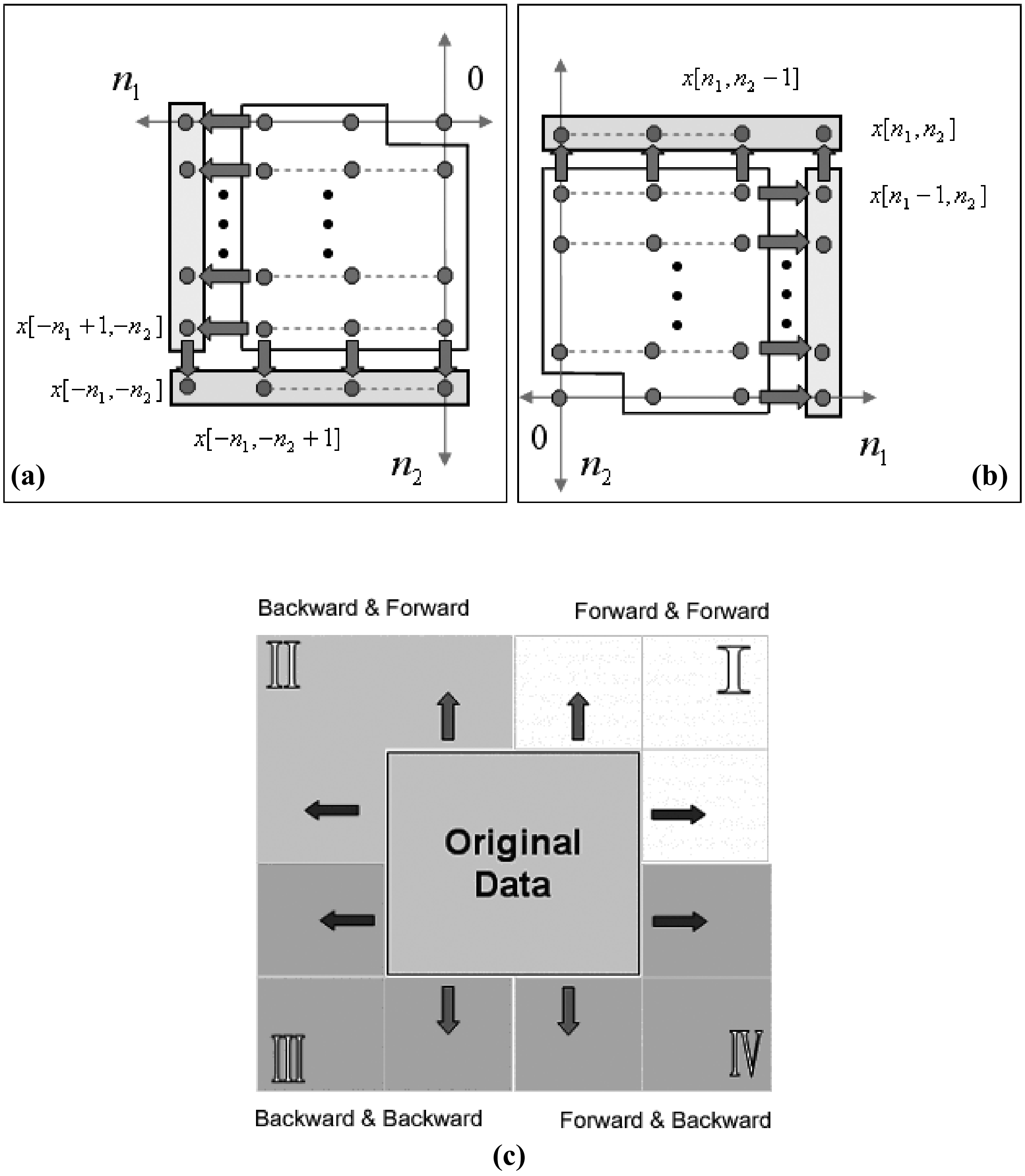


Figure 15. In previous BWE techniques, a true 2-D BWE technique was not achieved, which resulted in a poor result relative to our new method. (a) Column-backward extrapolation and row-backward extrapolation. (b) Column-forward extrapolation and row-forward extrapolation. (c) The new technique employs a true 2-D BWE technique, which can be expanded in any arbitrary direction. For this research the original data is first expanded in the column forward and backward direction, followed by row expansion in the forward and backward direction.

3.2. Experimental Results

In this section, we will apply the new 2-D BWE technique and its companion spectral estimator to simulated image data. The original data, shown in Figure 16, has two point objects and one line object that form a gray scale image for testing BWE algorithms. The synthetic gray-scale image is 128×128 pixels, and it is scaled from -0.5 to +0.5 along each axis. The image contains blurring effects (sidelobes) due to the finite optical aperture assumed to capture the image. It also has a tiny amount of noise added to ensure stable estimated 2-D LP parameter that are used in the 2-D BWE technique.

Figure 17 shows the result of the alternative 2-D BWE technique presented in this thesis using the 2-D AR parameter matrix estimated by the 2-D lattice algorithm [3]. It is obvious from the figure shown that there is a significant resolution improvement over the pre-sharpened Figure 16. In the post-sharpened image (the result of extrapolation by a factor of two (256×256 pixels)) using an eighth order prediction model, it is now easier to locate the true location of the two points and the diagonal line, since the sidelobes from the previous image have been reduced significantly.

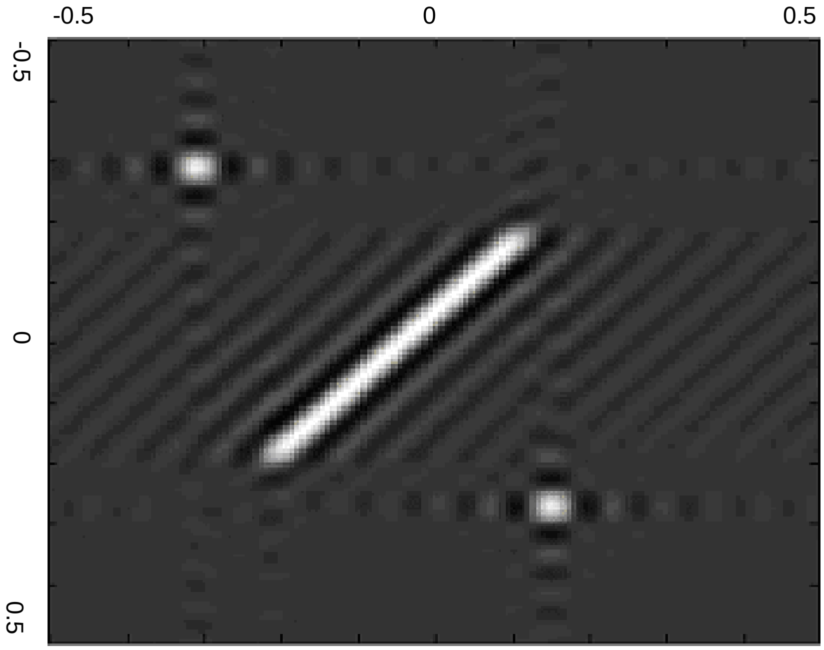


Figure 16. The original synthetic 128×128 pixel image, with unit width and height, containing two points and a line object with blurring effects (sidelobes).

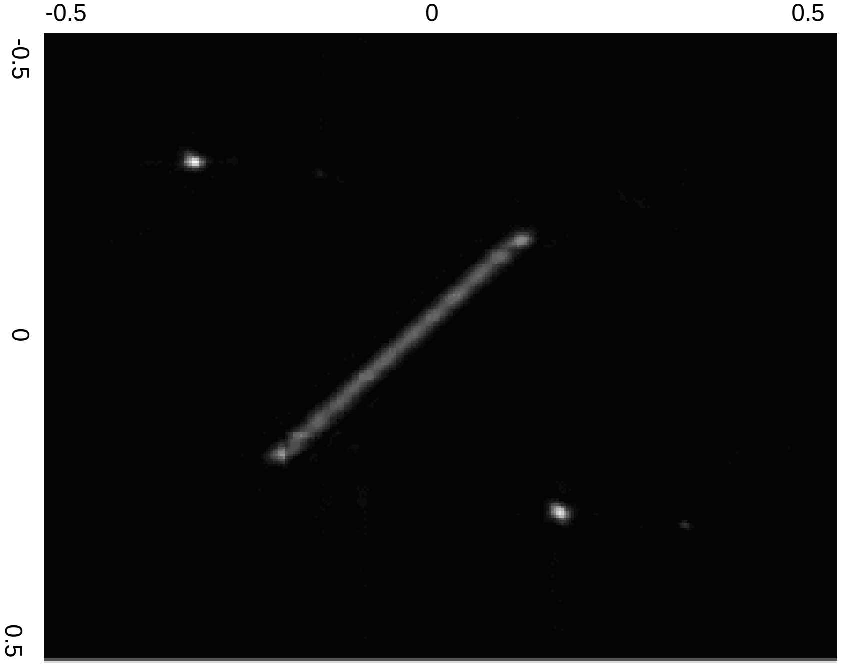


Figure 17. Post-sharpened 128×128 pixel image, with unit width and height, as a result of applying the new 2-D BWE technique.

4. NOVEL 2-D SYNTHETIC APERTURE RADAR IMAGERY RESOLUTION ENHANCEMENT TECHNIQUE

In this chapter, the 2-D BWE technique is applied to wideband polarimetric radar data. A 2-D SAR image represents the magnitude and location in both down range and cross range of each scattering center [8,9]. SAR resolution is an important factor in acquiring details of scatterers seen in the image.

The two-dimensional BWE technique can be a promising technique for enhanced-resolution SAR imagery [5,6,7]. Furthermore, the application of the new 2-D BWE technique increases the number of scattering centers detected in the SAR image in comparison with that of conventional SAR image generation that uses the 2-D FFT to form the image. A simplified block diagram for the processing steps to create SAR images is illustrated in Figure 18.

4.1. Two-dimensional Sharpening Based on SAR Radar: Missile Data

SAR images to be formatted from simulated S-band and X-band target signature data are presented here to validate the proposed 2-D BWE techniques. The synthetic S-band and X-band data represent phase history data collected from a hypothesized missile structure as though it was taken in a microwave anechoic chamber. Such data presumes that data was acquired by ~~a~~ slowly rotating the target ("a missile" in this case) within the center of the RF field of the virtual chamber. The rotation is stepped in degrees of azimuth (nose-on is the 0° reference for the azimuth). At each azimuth step, the target is illuminated with a stepped frequency sequence. At

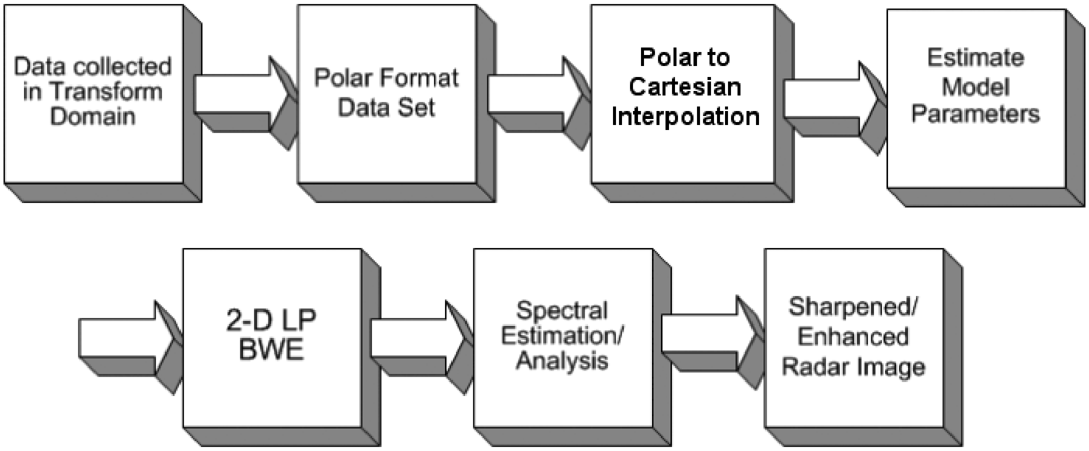

Data collected in Transform Domain
Polar Format Data Set
Polar to Cartesian Interpolation
Estimate Model Parameters
2-D LP BWE
Spectral Estimation/ Analysis
Sharpened/ Enhanced Radar Image

Figure 18. A simplified block diagram for creating SAR imagery. 2-D BWE technique can be a promising resolution enhancement technique for radar imagery.

each frequency and azimuth step, a complex value is recorded (the so-called radar cross section (RCS)) [16,24]. A stepped frequency sequence RCS response for S-band between 3.093 to 3.947 GHz in 10.67 MHz steps was synthetically generated from target microwave backscatter models. A sequence for X-band from 10.240 to 11.093 GHz in similar 10.67 MHz steps was also generated. The step interval in azimuth was 0.015° for both cases.

The data is effectively acquired in a polar format (due to the rotation) [18], and must be interpolated to a cartesian coordinate format for processing by the 2-D BWE algorithm. As shown in Figure 19, the phase history data to be processed into radar imagery has a fan-shaped polar acquisition format. To get it into a cartesian format, data points are interpolated within the fan-shaped region to a cartesian grid. In Figure 20, the missile’s nose is pointed to the top of the page, while the fin tips are pointed to the right and left of the page top.

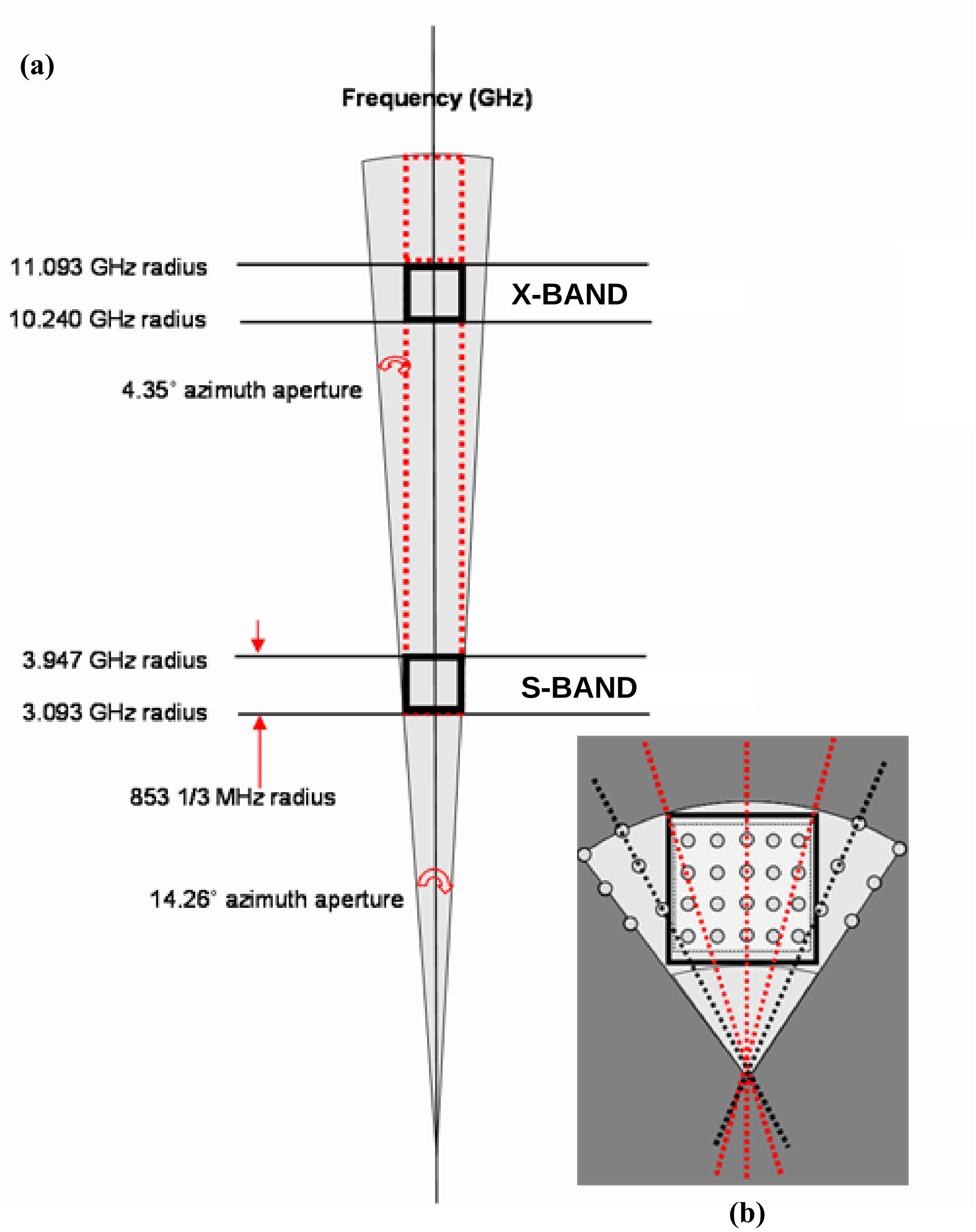
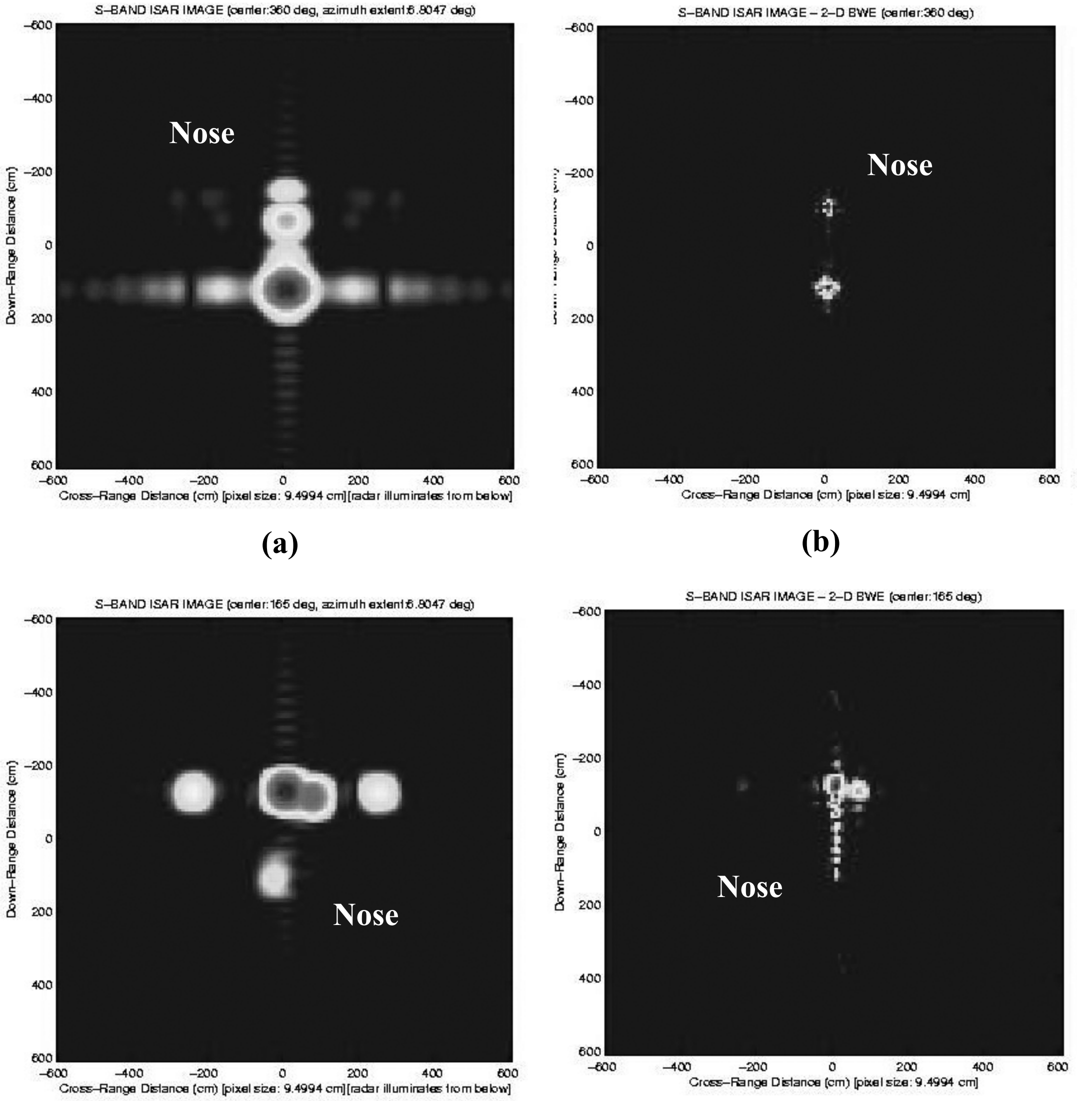


Figure 19. A variation of the 2-D BWE technique for computed synthetic radar imagery. The acquired data, shown on the left, is not originally in a cartesian format. Because of this, pre-processing interpolation is required. The fan-shaped data, shown on the right, is interpolated to attain a cartesian format for further processing [24]. (a) The original data acquired in a polar format (due to the rotation), (b) Interpolated cartesian coordinate data format.

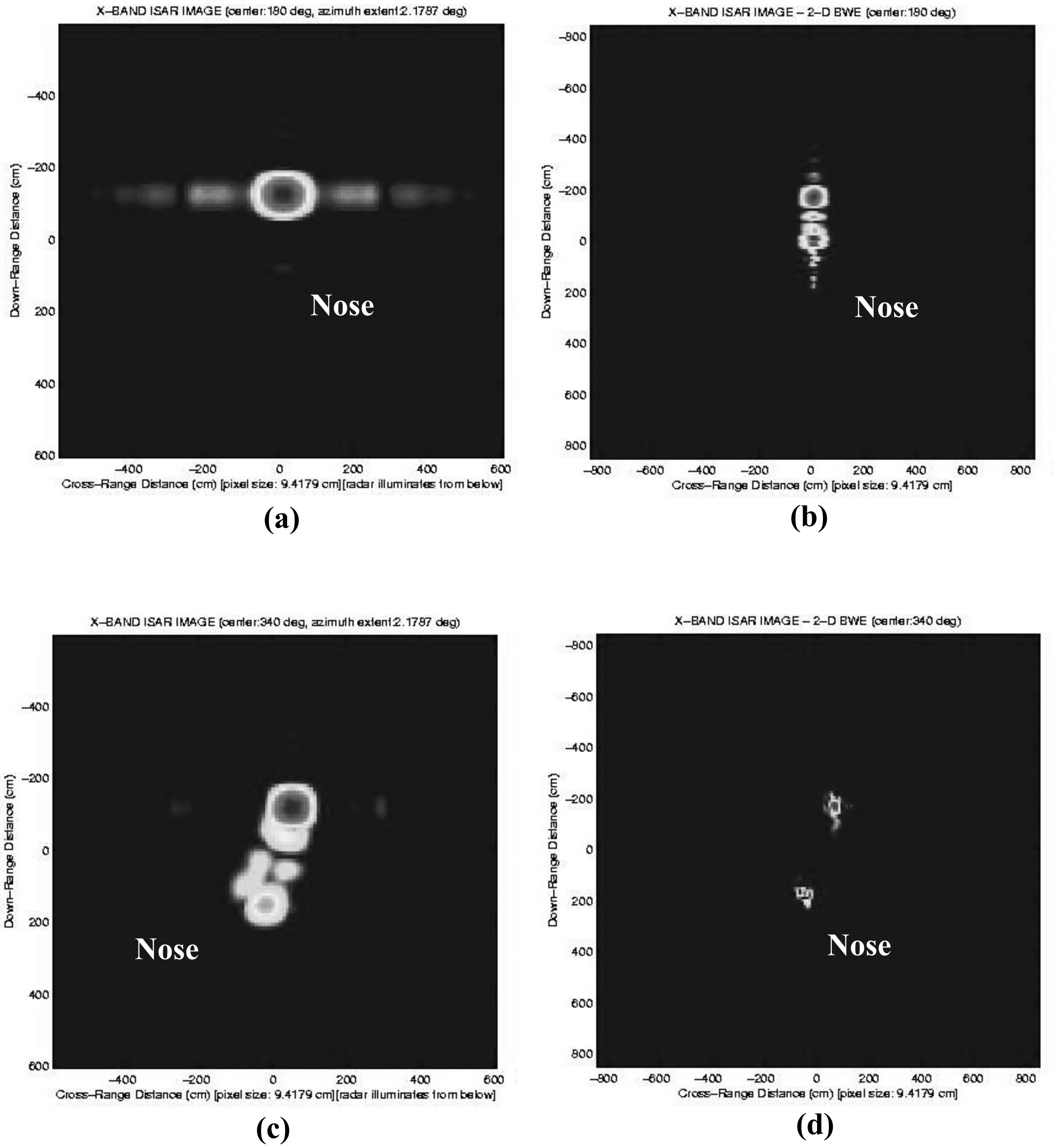


(a) (b)

(c) (d)

Figure 20. S-band SAR image with analysis center azimuth in degrees (a),(b) radar image of simulated target obtained by 2-D FFT with Hamming weights (Center azimuth: 180°, 65° respectively) (c),(d) Resolution-enhanced radar image of simulated target obtained by 2-D BWE (Center azimuth: 180°, 65° respectively)

Figure 20 and 21 are, respectively SAR images processed by either conventional Fourier-based SAR imagery operations or the new BWE method. Shown are S-band imagery with a center frequency of 3.5 GHz and X-band imagery with



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

Figure 21. X-band SAR image with analysis center azimuth in degrees (a),(b) radar image of simulated target obtained by 2-D FFT with Hamming weights (Center azimuth: 180°, 80° respectively) (c),(d) Resolution-enhanced radar image of simulated target obtained by 2-D BWE (Center azimuth: 180°, 80° respectively)

center frequency of 10 GHz, both of which used a bandwidth of 400 MHz and a measurement polarization of VV. The ‘VV channel’ represents the target response signal [24] when the vertical polarization is both transmitted and received. For the

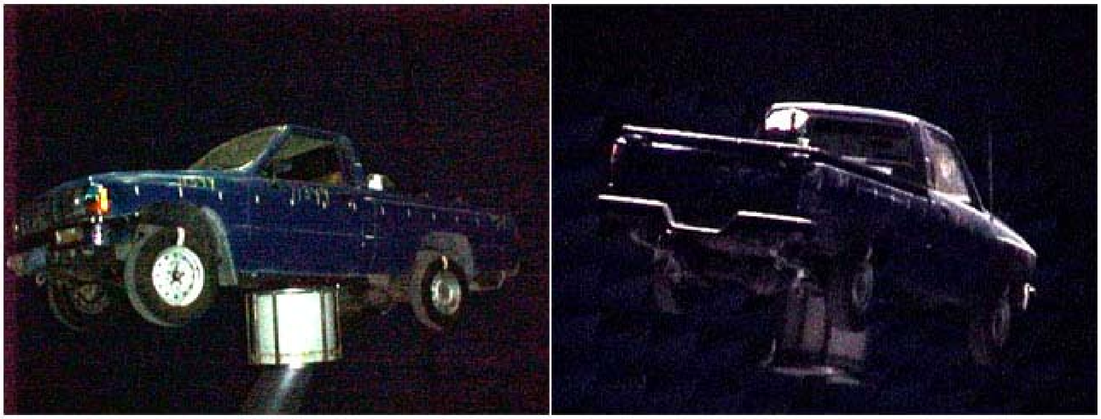


Figure 22. Photo of target on pylon in radar anechoic chamber of actual truck (The data were collected in a 10-story microwave anechoic chamber in Rancho Bernardo, CA. Data was acquired by Prof. Marple and provided for use in this research.)

original image, nominal down-range resolution and cross-range resolution was 9.4179 cm/pixel. The original image and the image enhanced by applying the 2-D BWE technique (shown in Figure 20 and 21 in (b),(d)), have higher resolution than the (a),(c) Fourier-based imagery.

4.2. Two-Dimensional Sharpening Based on SAR Radar: Truck Data

We demonstrate image-enhanced results in this section for phase history data measured from an actual truck. The data were collected in a 10-story microwave anechoic chamber located in Rancho Bernardo, CA. The data was acquired by Prof. Marple and provided to the author of the thesis. SAR imagery processed from measured S-band target data obtained in the chamber are presented to validate the proposed 2-D techniques. The rotation is stepped in degrees of azimuth (nose-on is the 0° reference for the azimuth), and at each azimuth step, the target is illuminated

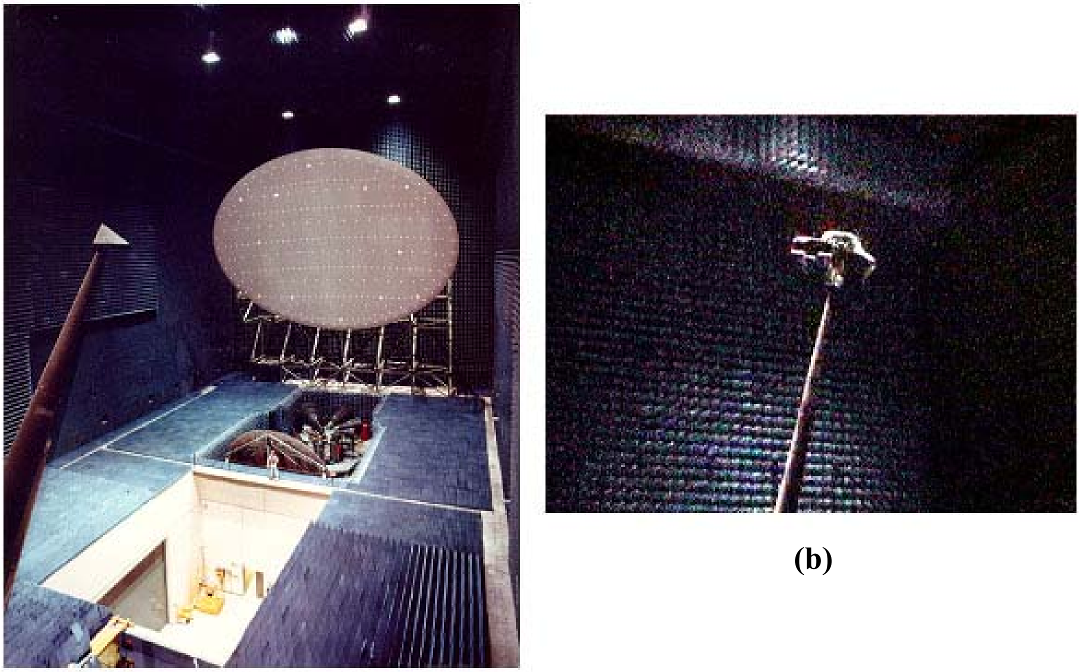
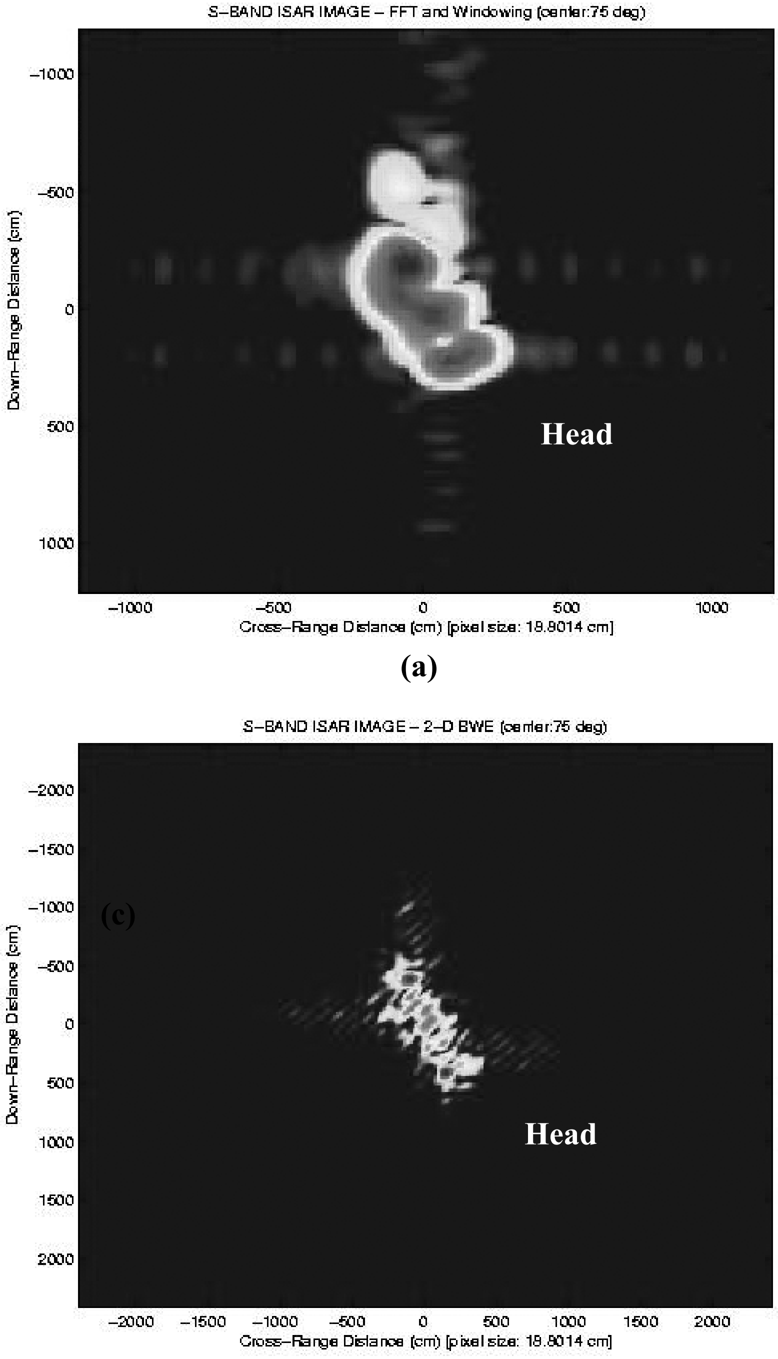


Figure 23. (a) 10-story microwave anechoic chamber (b) A target is raised on a pylon in the middle of the chamber. Data was acquired by Prof. Marple and provided for use in this research.

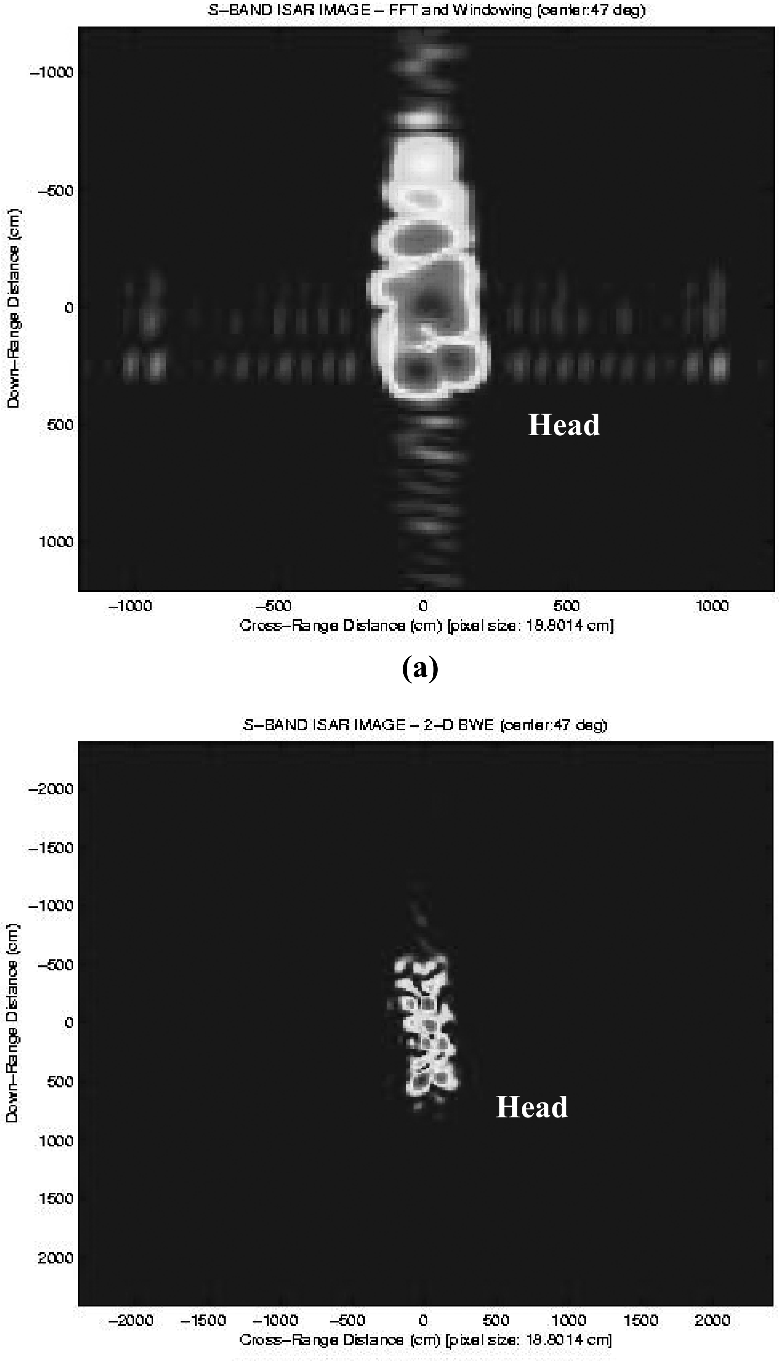
with a stepped frequency RF generator. The user provided parameters here are same as for the synthetic missile data. The acquired data is not originally in a cartesian coordinate format, and so a pre-processing interpolation must be done first to convert from polar formatted data to cartesian formatted data. As shown in Figure 19, the received data is in a fan-shaped acquisition format, and to get it into cartesian coordinate format, data points are interpolated within the fan-shaped image to form square cartesian-formatted data. The truck was raised on a pylon in the middle of the chamber and rotated on the pylon that is shown in Figure 23. The radar illuminates the rotating truck and receives an echoed signal from the truck. As the truck is rotating, the acquisition equipment



(a)

(b)

Figure 24. S-band SAR image with analysis center azimuth in degrees (a) Radar image of truck obtained by 2-D FFT with Hamming weights (Center azimuth: 75°) (b) Resolution-enhanced radar image of truck obtained by 2-D BWE (Center azimuth: 75°)



(a)

(b)

Figure 25. S-band SAR image with analysis center azimuth in degrees (a) Radar image of truck obtained by 2-D FFT with Hamming weights (Center azimuth: 47°) (b) Resolution-enhanced radar image of truck obtained by 2-D BWE (Center azimuth: 47°)

captures the signal at stepped azimuth angles and stepped frequencies. The images shown in Figure 24 and 25 are created from the truck data taken in the chamber, at center azimuth angles of 75° and 47°. These are the SAR images obtained by the conventional S-band Fourier-based SAR imagery with 3.52 GHz center frequency, 400 MHz bandwidth and the measurement polarization is VV. “VV channel” represents the target response signal when vertical polarization is transmitted and vertical polarization is received. The nominal down-range resolution and the cross-range resolution of the original images are both 18.80 cm/pixel. The original image and the enhanced image, obtained by applying 2D BWE, show that the resolution is enhanced by the 2-D BWE approach.

5. 3-D AUTOREGRESSIVE AND LINEAR PREDICTION RANDOM PROCESS THEORY AND ALGROITHMS

The evolution of sensor technology has provided acquisition scenarios today in which the data format is inherently three-dimensional (3-D). Examples include hyperspectral imaging (HSI) sensors (x-spatial dimension × y-spatial dimension × wavelength), interferometric synthetic aperture radar (IF-SAR) imaging (x-spatial dimension × y-spatial dimension × elevation/height dimension), and space-time adaptive processing (STAP) of radar (fast time [samples within a received pulse] × slow time [pulse-to-pulse sampling] × sensor array element number).

The three dimensions often involve a mixture of time, space, and wavelength dimensions. In all these applications, fully 3-D signal processing that has the capability (1) to enhance the resolution of the final 3-D analysis result (most often imagery), and (2) to provide reduced-dimension parametric features of the 3-D data for purposes of data encoding/compression is highly desirable. This chapter provides algorithmic techniques applicable to both capabilities. We also illustrate the value of our technique with a simulated 3-D synthetic aperture radar imagery application involving point targets in 3-D.

5.1. Three-Dimensional Octant-Based Autocorrelation and 3-D LP Parameter Matrices

The technique presented here is useful for estimating 3-D autoregressive (AR) parameters from 3-D autocorrelation sequence (ACS) values [28]. It operates directly in 3-D octant-space using the 3-D Yule-Walker equations with a recursive solution.

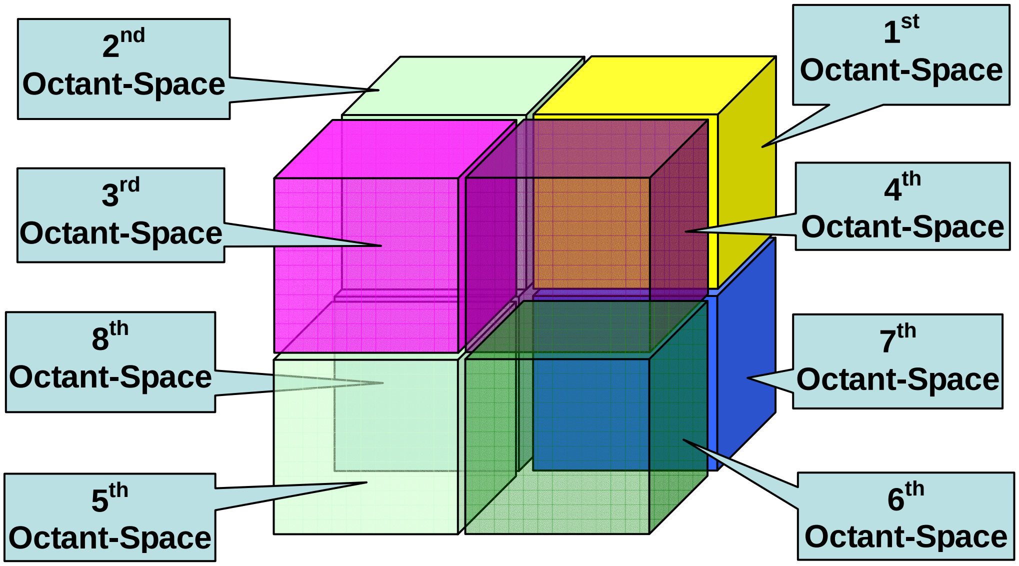


Figure 26. Eight octant-space regions of support for 3-D AR parameter arrays. While the prediction direction can be chosen arbitrarily, the indicated division of 3-D space was chosen due to its obvious extension from 2-D quadrants. The origin is located at the center of the eight octants. Detail descriptions are provided in Figures 28 and 29.

This computationally simple and fast performing algorithm, which has a close heritage to the quadrant-based 2-D algorithm [1,3] will be shown using a new approach involving recursive estimation which has a related set of triply Toeplitz block matrices. A 3-D AR sequence $x[n_1,n_2,n_3]$ is generated by driving a 3-D linear shift-invariant filter with 3-D white noise $w[n_1,n_2,n_3]$,

$$x[n_1,n_2,n_3]=-\sum_i\sum_j\sum_k a[i,j,k]\; x[n_1-i,n_2-j,n_3-k]+w[n_1,n_2,n_3]\;. \tag{48}$$

The summation ranges can be selected to be any one of the eight support regions of Eq (51) and Eq (52).

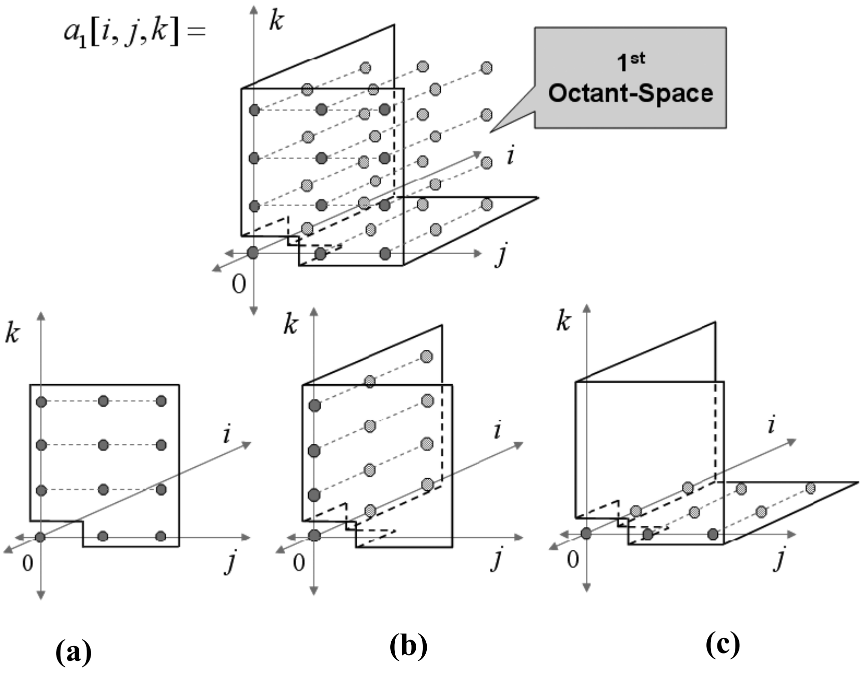


Figure 27. The top octant space, when chosen at $a^1[2,3,2]$, represents the 3-D AR parameters located in the first octant space. The number of coefficients is 2×3×2. To further explain the octant space, three examples were chosen: (a) at $a^1[2,3,0]$, (b) at $a^1[2,0,2]$ and (c) at $a^1[0,3,2]$.

A 3-D linear prediction estimate of the array sample $x[n_1, n_2, n_3]$ will have the form

$$\hat{x}[n_1, n_2, n_3] = -\sum_m \sum_n \sum_l a[i, j, k]\; x[n_1 - l, n_1 - i, n_2 - j, n_3 - k] \tag{49}$$

in which $a[i, j, k]$ are the 3-D LP/AR coefficients. If an octant region of support is selected, the 3-D linear prediction coefficients $x[n_1, n_2, n_3] - \hat{x}[n_1, n_2, n_3]$ that minimize the LP squared error

$$\rho_{LP} = \mathrm{E}\left\{ \left| x[n_1, n_2, n_3] - \hat{x}[n_1, n_2, n_3] \right|^2 \right\} \tag{50}$$

will yield a linear prediction error that is a 3-D white process only if the 3-D LP is of the same 3-D order as the 3-D AR process that created the data. The region of support for the first-, second-, third-, fourth-, fifth-, sixth-, seventh-, and eighth octant-space AR parameter arrays (d = 1,2,3,4,5,6,7 and 8th octant space) are defined below. The total number of octant-space (OS) AR parameters in any of the eight octant cases is $p_1 p_2 p_3 + p_1 p_2 + p_1 p_3 + p_2 p_3 + p_1 + p_2 + p_3$. Figures 26 and 27 illustrate the region of support for these eight OS autoregressive arrays and the prediction directions of the AR spectral estimator.

$$\left\{\begin{array}{ll} a^1[i,j,k]: & 0 < j \le p_2 \;\; for \;\; 1 \le k \le p_3 \;, 1 \le j \le p_2 \;\; and \;\; k = 0 \\ & 0 < j \le p_2 \;\; for \;\; 1 \le i \le p_1 \;, 1 \le j \le p_2 \;\; and \;\; i = 0 \\ & 0 < k \le p_3 \;\; for \;\; 1 \le i \le p_1 \;, 1 \le k \le p_3 \;\; and \;\; i = 0 \end{array}\right\}$$

$$\left\{\begin{array}{ll} a^2[i,j,k]: & - p_2 \le j < 0 \;\; for \;\; 1 \le k \le p_3 \;, - p_2 \le j \le 1 \;\; and \;\; k = 0 \\ & - p_2 \le j < 0 \;\; for \;\; 1 \le i \le p_1 \;, - p_2 \le j \le -1 \;\; and \;\; i = 0 \\ & 0 < k \le p_3 \;\; for \;\; 1 \le i \le p_1, 1 \le k \le p_3 \;\; and \;\; i = 0 \end{array}\right\}$$

$$\left\{\begin{array}{ll} a^3[i,j,k]: & - p_2 \le j < 0 \;\; for \;\; 1 \le k \le p_3 \,, - p_2 \le j \le -1 \;\; and \;\; k = 0 \\ & - p_2 \le j < 0 \;\; for \; - p_1 \le i \le -1 \,, - p_2 \le j \le -1 \;\; and \;\; i = 0 \\ & 0 < k \le p_3 \;\; for \; - p_1 \le i \le -1 \,, 1 \le k \le p_3 \;\; and \;\; i = 0 \end{array}\right\}$$

$$\left\{\begin{array}{ll} a^4[i,j,k]: & 0 < j \le p_2 \;\; for \;\; 1 \le k \le p_3 \;, 1 \le j \le p_2 \;\; and \;\; k = 0 \\ & 0 < j \le p_2 \;\; for \; - p_1 \le i \le -1 \,, 1 \le j \le p_2 \;\; and \;\; i = 0 \\ & 0 < k \le p_1 \;\; for \; - p_1 \le i \le -1 \,, 1 \le k \le p_3 \;\; and \;\; i = 0 \end{array}\right\}$$

$$\left\{\begin{array}{ll} a^5[i,j,k]: & - p_2 \le j < 0 \;\; for \; - p_2 \le k \le -1 \,, - p_2 \le j \le -1 \;\; and \;\; k = 0 \\ & - p_2 \le j < 0 \;\; for \;\; 1 \le i \le p_1 \;, - p_2 \le j \le -1 \;\; and \;\; i = 0 \\ & - p_3 \le k < 0 \;\; for \;\; 1 \le i \le p_1 \;, - p_3 \le k \le -1 \;\; and \;\; i = 0 \end{array}\right\}$$

$$\left\{\begin{array}{ll} a^6[i,j,k]: & - p_2 \le j < 0 \;\; for \; - p_3 \le k \le -1 \,, - p_2 \le j \le -1 \;\; and \;\; k = 0 \\ & - p_2 \le j < 0 \;\; for \; - p_1 \le i \le -1 \,, - p_2 \le j \le -1 \;\; and \;\; i = 0 \\ & - p_3 \le k < 0 \;\; for \; - p_1 \le i \le -1 \,, - p_3 \le k \le -1 \;\; and \;\; i = 0 \end{array}\right\} \tag{51}$$

$$\left\{\begin{array}{ll} a^7[i,j,k]: & 0 < j \le p_2 \quad for \; -p_3 \le k \le -1 \,,\, 1 \le j \le p_2 \;\; and \;\; k = 0 \\ & 0 < j \le p_2 \quad for \; -p_1 \le i \le -1 \,,\, 1 \le j \le p_2 \;\; and \;\; i = 0 \\ & -p_3 \le k < 0 \quad for \; -p_1 \le i \le -1 \,,\, -p_3 \le k \le -1 \;\; and \;\; i = 0 \end{array}\right\}$$

$$\left\{\begin{array}{ll} a^8[i,j,k]: & 0 < j \le p_2 \quad for \; -p_3 \le k \le -1 \,,\, 1 \le j \le p_2 \;\; and \; k = 0 \\ & 0 < j \le p_2 \quad for \; 1 \le i \le p_1 \,,\, 1 \le j \le p_2 \;\; and \;\; i = 0 \\ & -p_3 \le k < 0 \quad for \; 1 \le i \le p_1 \,,\, -p_3 \le k \le -1 \;\; and \;\; i = 0 \end{array}\right\} \quad (52)$$

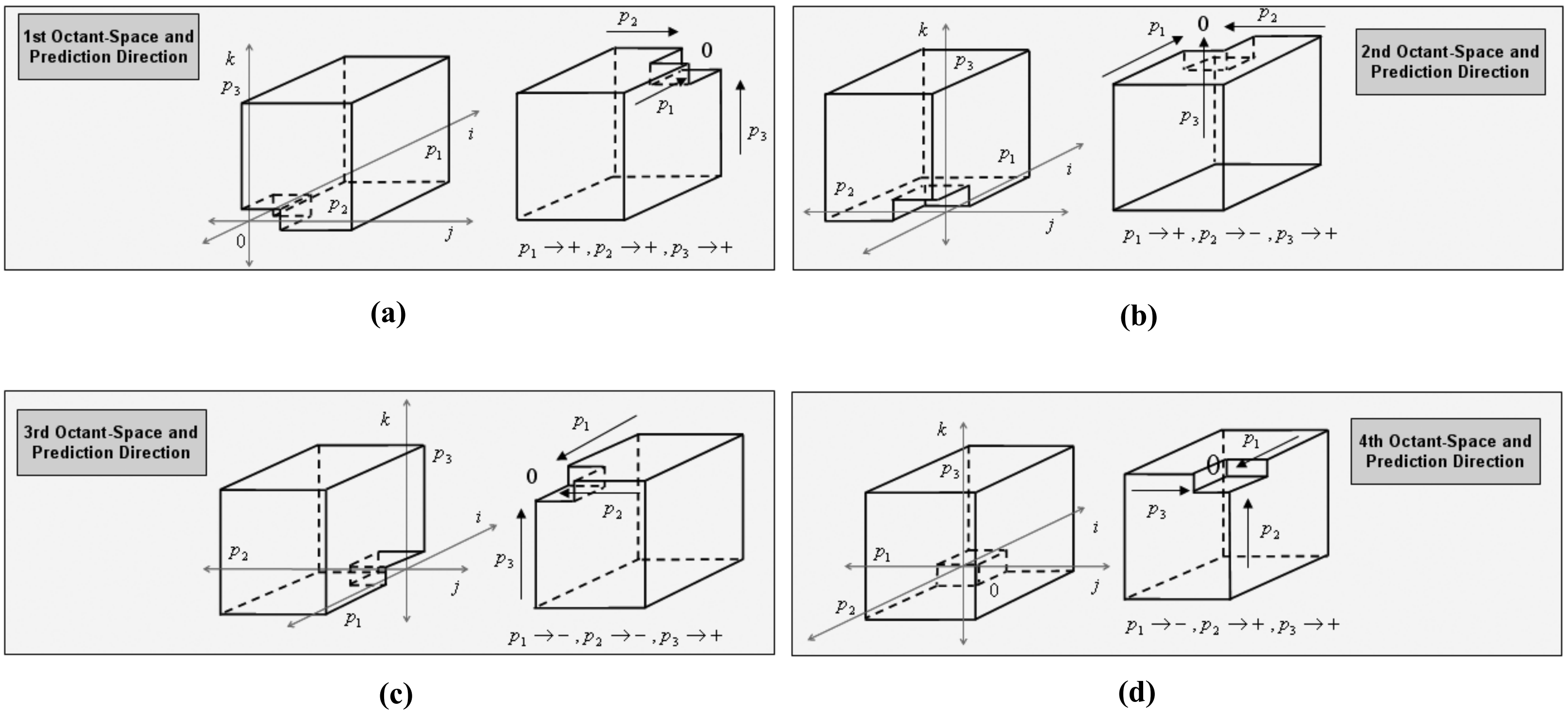
5.2. Three-Dimensional Autoregressive Spectral Estimation with Fast Algorithms

5.2.1. Three-Dimensional Yule-Walker Equations

If the 3-D autocorrelation sequence is known, then it can be shown that the parameters of each of the eight octants satisfy the following 3-D Yule-Walker (YW) equations and the corresponding octant 3-D prediction directions. The 3-D YW equations for a 3-D AR process are obtained by multiplying $x^*[n_1 - i, n_2 - j, n_3 - k]$ by Eq (48) and by taking the expectation to yield

$$\sum_i \sum_j \sum_k a[i,j,k]\; r_{xx}[n_1 - i, n_2 - j, n_3 - k] = \left\{\begin{array}{ll} \rho_\omega & for[i,j,k]=[0,0,0] \\ 0 & otherwise \end{array}\right\} \quad (53)$$

The summation ranges can be selected to be any one of the eight octants of $a[i,j,k]$. In anticipation of the fast computational algorithm to be presented, we shall assume that subscript $p_1 p_2 p_3$ means p_1 (point on the 'i' axis) is a variable order parameter and p_2 (point on the 'j' axis) and p_3 (point on the 'k' axis) in Figures 28 and 29 are assumed to be fixed order parameters. More detailed figures are presented in


1st Octant-Space and Prediction Direction
k
p_3
p_1
i
p_2
0
j
p_2
0
p_1
p_3
$p_1 \to +, p_2 \to +, p_3 \to +$
2nd Octant-Space and Prediction Direction
k
p_3
p_1
i
p_2
j
p_1
0
p_2
p_3
$p_1 \to +, p_2 \to -, p_3 \to +$
3rd Octant-Space and Prediction Direction
k
p_3
i
p_2
j
p_1
p_1
0
p_2
p_3
$p_1 \to -, p_2 \to -, p_3 \to +$
4th Octant-Space and Prediction Direction
k
p_3
i
p_1
j
0
p_2
p_1
0
p_3
p_2
$p_1 \to -, p_2 \to +, p_3 \to +$

(a)

(b)

(c)

(d)

Figure 28. Illustration of the prediction directions in the eight-octant space AR spectral estimator: (a) represents the first octant space and prediction direction of p_1 in the positive direction, p_2 in the positive direction, and p_3 in the positive direction. (b) represents the second octant space and prediction direction of p_1 in the positive direction, p_2 in the negative direction, and p_3 in the positive direction. (c) represents the third octant space and prediction direction of p_1 in the negative direction, p_2 in the negative direction, and p_3 in the positive direction. (d) represents the forth octant space and prediction direction of p_1 in the negative direction, p_2 in the positive direction, and p_3 in the positive direction.

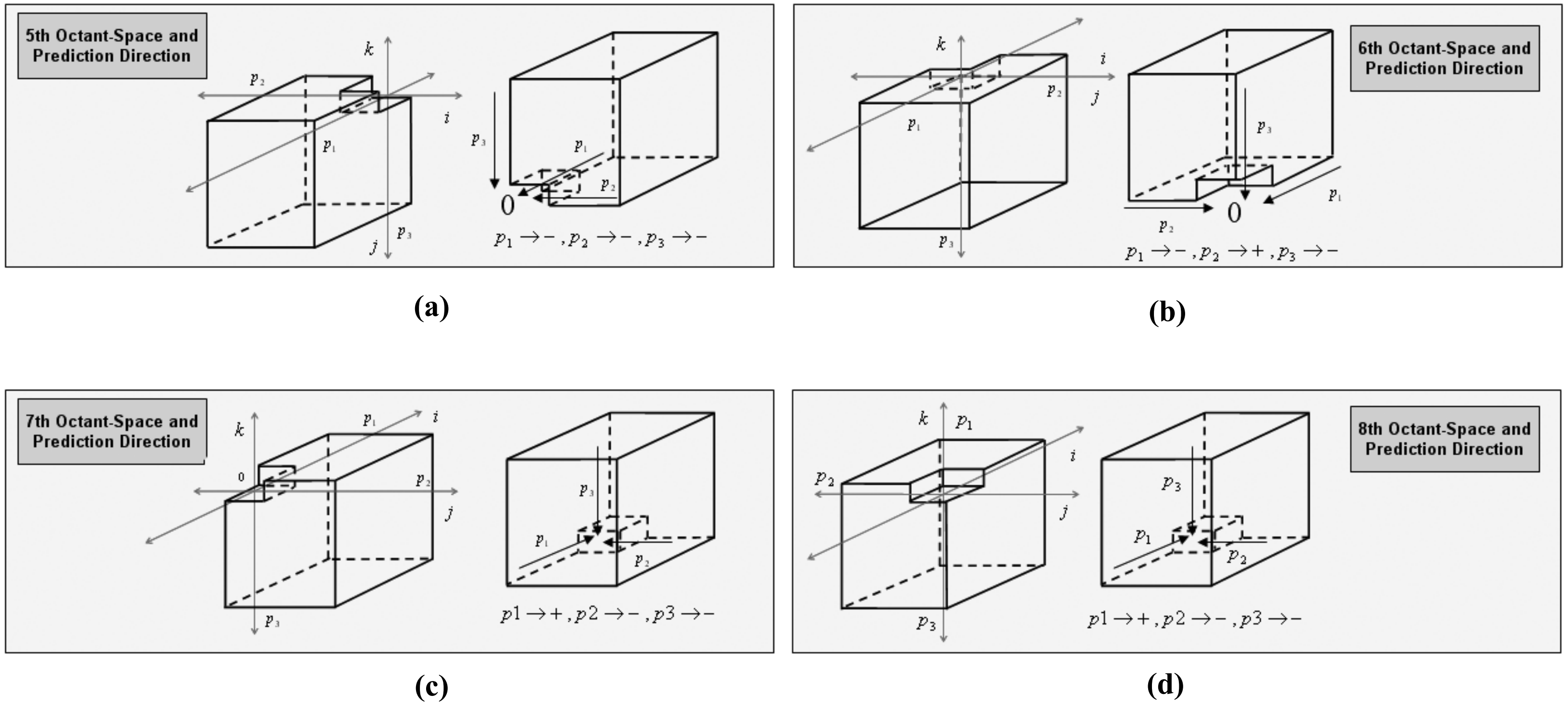

5th Octant-Space and Prediction Direction
k
i
j
p_1
p_2
p_3
0
$p_1 \rightarrow -, p_2 \rightarrow -, p_3 \rightarrow -$
(a)
6th Octant-Space and Prediction Direction
k
i
j
p_1
p_2
p_3
0
$p_1 \rightarrow -, p_2 \rightarrow +, p_3 \rightarrow -$
(b)
7th Octant-Space and Prediction Direction
k
i
j
0
p_1
p_2
p_3
$p1 \rightarrow +, p2 \rightarrow -, p3 \rightarrow -$
(c)
8th Octant-Space and Prediction Direction
k
i
j
p_1
p_2
p_3
$p1 \rightarrow +, p2 \rightarrow -, p3 \rightarrow -$
(d)

Figure 29. Illustration of the prediction directions in the eight-octant space AR spectral estimator: (a) represents the fifth octant space and prediction direction of p1 in the negative direction, p2 in the negative direction, and p3 in the negative direction. (b) represents the sixth octant space and prediction direction of p1 in the negative direction, p2 in the positive direction, and p3 in the negative direction. (c) represents the seventh octant space and prediction direction of p1 in the positive direction, p2 in the negative direction, and p3 in the negative direction. (d) represents the eighth octant space and prediction direction of p1 in the positive direction, p2 in the negative direction, and p3 in the negative direction.

Appendix A on pages 88-96. The 3-D Yule-Walker equations for the support regions can be arranged, or ordered, into at least six convenient super block vector forms

$$\underline{\underline{\mathbf{a}}}^{d}_{p_1p_2p_3}\ ,\ \underline{\underline{\mathbf{a}}}^{d}_{p_1p_3p_2}\ ,\ \underline{\underline{\mathbf{a}}}^{d}_{p_2p_1p_3}\ ,\ \underline{\underline{\mathbf{a}}}^{d}_{p_2p_3p_1}\ ,\ \underline{\underline{\mathbf{a}}}^{d}_{p_3p_1p_2}\ ,\ and\ \ \underline{\underline{\mathbf{a}}}^{d}_{p_3p_2p_1} \tag{54}$$

by ordering the 3-D AR coefficients in a three-tiered hierarchy (one order on outside, one in middle, one on the inside). An alternative block vector representation of the first octant-space Yule-Walker equation is

$$\underline{\underline{\boldsymbol{\rho}}}^{1}_{p_1p_2p_3} = \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{1}_{p_1p_2p_3} \tag{55}$$

where the data vector is also a super block vector of dimension (p_1+1).

$$\underline{\underline{\mathbf{x}}}_{p_1p_2p_3} = \left[\underline{\mathbf{x}}_{p_1p_2p_3}[i]\ \ \underline{\mathbf{x}}_{p_1p_2p_3}[i-1]\cdots\cdots\underline{\mathbf{x}}_{p_1p_2p_3}[i-p_1]\right] \tag{56}$$

The *super block vector* $\underline{\underline{\mathbf{a}}}^{1}_{p_1p_2p_3}$, with superscript 1, designates this as a set of the first-octant AR parameters. It is composed of (p_1+1) *block vectors*, each of dimension $(p_2+1)(p_3+1)\times 1$.

$$\underline{\underline{\mathbf{a}}}^{d}_{p_1p_2p_3} = \left[\underline{\mathbf{a}}^{1\ \ T}_{p_1p_2p_3}[0]\ \ \underline{\mathbf{a}}^{1\ \ T}_{p_1p_2p_3}[1]\ \cdots\ \underline{\mathbf{a}}^{1\ \ T}_{p_1p_2p_3}[p_1]\right]^T \tag{57}$$

which are further partitioned in terms of the *block vectors*

$$\underline{\mathbf{a}}^{1}_{p_1p_2p_3}[i] = \left[\mathbf{a}^{1\ \ T}_{p_1p_2p_3}[i,0]\ \ \mathbf{a}^{1\ \ T}_{p_1p_2p_3}[i,1]\ \cdots\ \mathbf{a}^{1\ \ T}_{p_1p_2p_3}[i,p_2]\right]^T \tag{58}$$

and even further partitioned in terms of the *scalar elements*

$$\mathbf{a}^{1}_{p_1p_2p_3}[i,j] = \left[a^{1\ \ T}_{p_1p_2p_3}[i,j,0]\ \ a^{1\ \ T}_{p_1p_2p_3}[i,j,1]\cdots\ a^{1\ \ T}_{p_1p_2p_3}[i,j,p_3]\right]^T \tag{59}$$

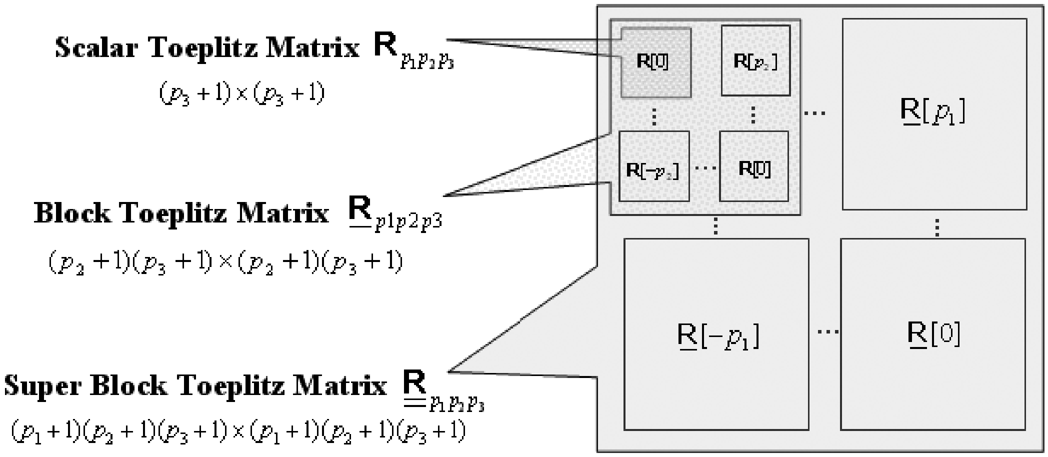

Scalar Toeplitz Matrix $\mathbf{R}_{p_1p_2p_3}$
$(p_3+1)\times(p_3+1)$
Block Toeplitz Matrix $\underline{\mathbf{R}}_{p1p2p3}$
$(p_2+1)(p_3+1)\times(p_2+1)(p_3+1)$
Super Block Toeplitz Matrix $\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}$
$(p_1+1)(p_2+1)(p_3+1)\times(p_1+1)(p_2+1)(p_3+1)$
R[0]
R[p_2]
R[$-p_2$]
R[0]
$\underline{\mathrm{R}}[p_1]$
$\underline{\mathrm{R}}[-p_1]$
$\underline{\mathrm{R}}[0]$

Figure 30. The hierarchy of the super block Toeplitz matrix in terms of structure and dimension. The super block Toeplitz matrix is composed of block Toeplitz matrices, which in turn is composed of scalar Toeplitz matrices.

The *super block vector* $\underline{\underline{\rho}}_{p_1p_2p_3}$ has all-zero *super block vectors* $\underline{\mathbf{0}}$ and one top super block entry $\underline{\rho}^1_{p_1p_2p_3}$ that has all zero entries, except for the top entry, which is the noise variance ρ_w. Note that ‘**0**’ is a vector of (p_3+1) zeros and $\underline{\mathbf{0}}$ is a column block vector of $(p_1+1)\times(p_3+1)$ zeros.

$$\underline{\underline{\rho}}^1_{p_1p_2p_3} = [\underline{\rho}^{1\ T}_{p_1p_2p_3}\quad \underline{\mathbf{0}} \cdots \underline{\mathbf{0}}]^T \tag{60}$$

which is defined in terms of the *block vectors*

$$\underline{\rho}^1_{p_1p_2p_3} = [\rho^{1\ T}_{p_1p_2p_3}\quad \mathbf{0} \cdots \mathbf{0}]^T \tag{61}$$

which is in turn defined in terms of the *scalar elements*

$$\rho^1_{p_1p_2p_3} = [\rho^1_w \quad 0 \cdots 0]^T \tag{62}$$

The data vector is a super block vector described in Eq (56) and the autocorrelation matrix is defined as

$$\mathbf{E}\left\{\underline{\underline{\mathbf{x}}}_{p_1p_2p_3}\underline{\underline{\mathbf{x}}}^{H}_{p_1p_2p_3}\right\}=\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}=$$

$$\mathbf{E}\left\{\begin{bmatrix} x[i,j,k] \\ \vdots \\ x[i,j,k-p_3] \\ x[i,j-1,k] \\ \vdots \\ x[i,j-1,k-p_3] \\ \vdots \\ \vdots \\ x[i,j-p_2,k-p_3] \\ \vdots \\ \vdots \\ x[i-p_1,j-p_2,k-p_3] \end{bmatrix} \bullet \begin{bmatrix} x^*[i,j,k]\cdots x^*[i-p_1,j-p_2,k-p_3] \end{bmatrix}\right\} \tag{63}$$

Super block matrix $\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}$ has the dimension $(p_1+1)(p_2+1)(p_3+1)\times$ $(p_1+1)(p_2+1)(p_3+1)$, where each super block consists of $(p_2+1)\times(p_2+1)$ block Toeplitz matrices (each block element is Toeplitz) and each Toeplitz block consist of $(p_3+1)\times(p_3+1)$ scalar elements. Therefore, matrix $\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}$ is said to be triply Toeplitz or super block Toeplitz as shown in Figure 30.

A subscript $p_1p_2p_3$ is used to remind the reader that '*variable order* p_1' and '*fixed orders* p_2 *and* p_3' ordering has been used for Eq (54). We can describe the matrix $\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}$ in terms of the *super block matrices,*

$$\underline{\underline{\mathsf{R}}}_{p_1p_2p_3} = \begin{bmatrix} \underline{\mathsf{R}}_{p_1p_2p_3}[\ 0\ ] & \cdots & \underline{\mathsf{R}}_{p_1p_2p_3}[p_1]] \\ \vdots & & \vdots \\ \underline{\mathsf{R}}_{p_1p_2p_3}[-p_1] & \cdots & \underline{\mathsf{R}}_{p_1p_2p_3}[\ 0\ ] \end{bmatrix} \tag{64}$$

which are defined in terms of matrix elements with dependence on the variable order p_1. For example, the one for index 0 is,

$$\underline{\mathsf{R}}_{p_1p_2p_3}[0] =$$

$$\begin{bmatrix} \begin{bmatrix} r[\ 0,\ 0,\ 0] & r[\ 0,\ 0,\ 1] & \cdots & r[\ 0, 0,\ p_3] \\ r[\ 0,\ 0,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[\ 0, 0,-p_3] & \cdots & \cdots & r[\ 0,\ 0,\ 0] \end{bmatrix} & \cdots & \begin{bmatrix} r[\ 0,\ p_2,\ 0] & r[0, p_2, 1] & \cdots & r[0, p_2, p_3] \\ r[\ 0,-p_2 -1] & & & \vdots \\ \vdots & & & \vdots \\ r[0,-p_2,-p_3] & \cdots & \cdots & r[0,\ p_2,\ 0] \end{bmatrix} \\ \vdots & & \vdots \\ \begin{bmatrix} r[0,-p_2,\ 0] & r[0, p_2,\ 1] & \cdots & r[0, p_2, p_3] \\ r[0,-p_2,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[0,-p_2,-p_3] & \cdots & \cdots & r[0,-p_2, 0] \end{bmatrix} & \cdots & \begin{bmatrix} r[\ 0,\ 0,\ 0] & r[\ 0,\ 0,\ 1] & \cdots & r[0, 0, p_3] \\ r[0,\ 0,\ -1] & & & \vdots \\ \vdots & & & \vdots \\ r[0, 0,-p_3] & \cdots & \cdots & r[0,\ 0,\ 0] \end{bmatrix} \end{bmatrix} \tag{65}$$

At index p_1, the block matrix has the form,

$$\underline{\mathsf{R}}_{p_1p_2p_3}[p_1] =$$

$$\begin{bmatrix} \begin{bmatrix} r[\ p_1,\ 0,\ 0] & r[\ _1, 0, 1] & \cdots & r[p_1, 0,\ p_3] \\ r[\ p_1,\ 0,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[p_1,\ 0,-p_3] & \cdots & \cdots & r[\ p_1, 0, 0] \end{bmatrix} & \cdots & \begin{bmatrix} r[p_1,\ p_2, 0] & r[p_1, p_2, 1] & \cdots & r[p_1, p_2, p_3] \\ r[p_1,-p_2,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[p_1,-p_2,-p_3] & \cdots & \cdots & r[p_1,\ p_2, 0] \end{bmatrix} \\ \vdots & & \vdots \\ \begin{bmatrix} r[p_1,-p_2, 0] & r[p_1, p_2, 1] & \cdots & r[p_1, p_2, p_3] \\ r[p_1,-p_2,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[p_1,-p_2,-p_3] & \cdots & \cdots & r[p_1,-p_2, 0] \end{bmatrix} & \cdots & \begin{bmatrix} r[\ p_1, 0, 0] & r[p_1, 0, 1] & \cdots & r[\ p_1, 0,\ p_3] \\ r[\ p_1, 0,-1] & & & \vdots \\ \vdots & & & \vdots \\ r[p_1, 0,-p_3] & \cdots & \cdots & r[\ p_1, 0, 0] \end{bmatrix} \end{bmatrix} \tag{66}$$

Eq (67) below has previously been shown [1] and used with computationally efficient algorithms for the 2-D spectral estimate. The structuring has been selected based on the fact that the 2-D data matrix will have Toeplitz block structure [3,10].

$$\underline{\rho}^{1}_{p_1p_2p_3} = \underline{\mathbf{R}}_{p_1p_2p_3}\underline{\mathbf{a}}^{1}_{p_1p_2p_3} \qquad (67)$$

If one ignores the values of p_1 in Eq (64), the 3-D Yule-Walker equations can be arranged or ordered in the same way as in the 2-D case. If p_2 is fixed to 0 and the second variable p_2 is changed from 0 to p_2, $\underline{\mathbf{R}}_{p_1p_2p_3}$ matrices will look like

$$\underline{\mathbf{R}}_{p_1p_2p_3} = \begin{bmatrix} \mathbf{R}_{p_1p_2p_3}[\ 0\ ] & \cdots & \mathbf{R}_{p_1p_2p_3}[p_2] \\ \vdots & & \vdots \\ \mathbf{R}_{p_1p_2p_3}[-p_2] & \cdots & \mathbf{R}_{p_1p_2p_3}[\ 0\ ] \end{bmatrix} \qquad (68)$$

and this is defined in terms of the matrix elements with the variable order p_3 and a fixed order p_2,

$$\mathbf{R}_{p_1p_2p_3}[p_2] = \begin{bmatrix} r[0,\ p_2,\ 0] & r[0, p_2, 1] & \cdots & r[0, p_2, p_3] \\ r[0, p_2 - 1] & & & \vdots \\ \vdots & & & \vdots \\ r[0, p_2, -p_3] & \cdots & \cdots & r[0, p_2, 0] \end{bmatrix} \qquad (69)$$

As mentioned before, the 3-D Yule-Walker (YW) equations have six alternative super block matrix forms. For example, the YW equations with the $p_3p_2p_1$ order autocorrelation matrix for the first octant-space would be

$$\underline{\underline{\rho}}^{1}_{p_3p_2p_1} = \underline{\underline{\mathbf{R}}}_{p_3p_2p_1}\underline{\underline{\mathbf{a}}}^{1}_{p_3p_2p_1}\ . \qquad (70)$$

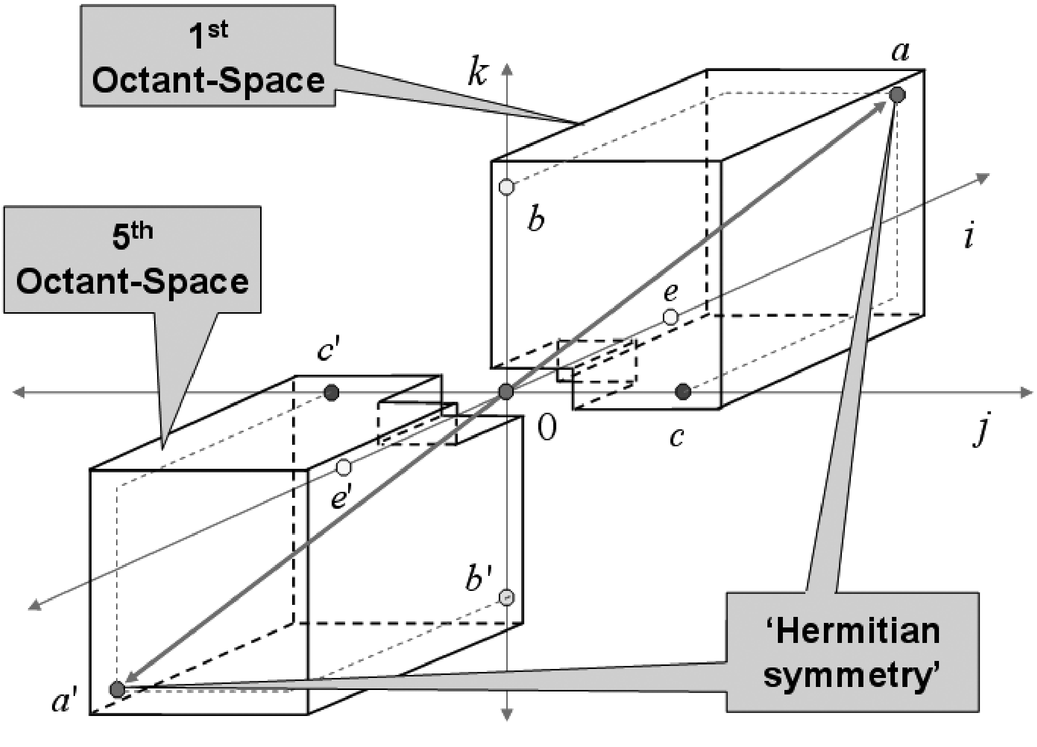


Figure 31. Illustration of the complex conjugate property in the $d = 1$ and $d = 5$ coefficients. With the origin located at $i = j = k$: $a - a'$, $b - b'$, $c - c'$, and $d - d'$ exhibit complex conjugate property.

Similarly, we can also derive $\underline{\underline{\mathsf{R}}}_{p_3 p_2 p_1}$ in terms of the super block matrices and defined in terms of the matrix elements with the variable order p_3 that equals zero.

$$\underline{\underline{\mathsf{R}}}_{p_3 p_2 p_1} = \begin{bmatrix} \underline{\mathsf{R}}_{p_3 p_2 p_1}[\ 0\ ] & \cdots & \underline{\mathsf{R}}_{p_3 p_2 p_1}[p_3] \\ \vdots & & \vdots \\ \underline{\mathsf{R}}_{p_3 p_2 p_1}[-p_3] & \cdots & \underline{\mathsf{R}}_{p_3 p_2 p_1}[\ 0\ ] \end{bmatrix} \tag{71}$$

In this case, subscript $p_3 p_2 p_1$ means 'variable order p_3' and 'fixed order p_2 and fixed order p_1. Super block matrix $\underline{\mathsf{R}}_{p_3 p_2 p_1}$ has a dimension of

$(p_1+1)(p_2+1)(p_3+1)\times\quad(p_1+1)(p_2+1)(p_3+1)$, where each super block is a $(p_2+1)\times(p_2+1)$ block Toeplitz and each Toeplitz block is a $(p_1+1)\times(p_1+1)$ scalar element with the form

$$\underline{\mathbf{R}}_{p_3p_2p_1}[p_1]=$$

$$\begin{bmatrix} \begin{bmatrix} r[p_1,\,0,\,0] & r[p_1,\,0,\,1] & \cdots & r[p_1,0,\,0] \\ r[-1,\,0,\,0] & & & \vdots \\ \vdots & & & \vdots \\ r[-p_1,0,\,0] & \cdots & \cdots & r[\,0,\,0,\,0] \end{bmatrix} & \cdots & \begin{bmatrix} r[0,p_2,0] & r[0,p_2,1] & \cdots & r[p_1,p_2,0] \\ r[-1,-p_2,0] & & & \vdots \\ \vdots & & & \vdots \\ r[-p_1,-p_2,0] & \cdots & \cdots & r[0,p_2,0] \end{bmatrix} \\ \vdots & & \vdots \\ \begin{bmatrix} r[0,p_2,\,0] & r[\,0,\,p_2,\,1] & \cdots & r[p_1,p_2,0] \\ r[-1,-p_2,0] & & & \vdots \\ \vdots & & & \vdots \\ r[-p_1,-p_2,0] & \cdots & \cdots & r[0,\,p_2,\,0] \end{bmatrix} & \cdots & \begin{bmatrix} r[\,0,\,0,\,0] & r[\,1,\,0,\,1] & \cdots & r[p_1,0,\,0] \\ r[-1,\,0,\,0] & & & \vdots \\ \vdots & & & \vdots \\ r[-p_1,0,\,0] & \cdots & \cdots & r[0\,,0,\,0] \end{bmatrix} \end{bmatrix}$$

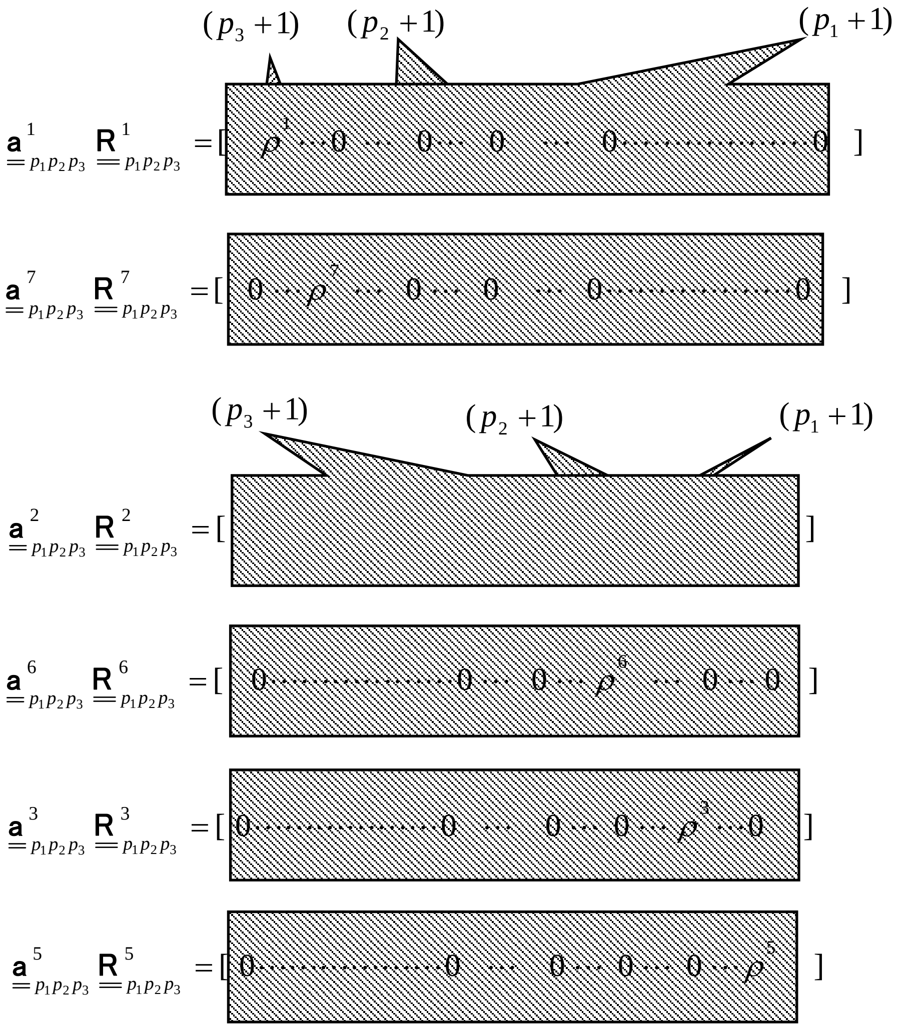
(72)

If the 3-D autocorrelation sequence is known, then it can be shown that the first eight octant-space parameters satisfy the following 3-D Yule-Walker normal equations

$$\begin{aligned}
\underline{\underline{\rho}}^{1}_{p_1p_2p_3} &= \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{1}_{p_1p_2p_3}\\
\underline{\underline{\rho}}^{2}_{p_1p_2p_3} &= \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{2}_{p_1p_2p_3}\\
\underline{\underline{\rho}}^{3}_{p_1p_2p_3} &= \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{3}_{p_1p_2p_3}\\
\underline{\underline{\rho}}^{4}_{p_1p_2p_3} &= \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{4}_{p_1p_2p_3}\\
\underline{\underline{\rho}}^{5}_{p_1p_2p_3} &= \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{5}_{p_1p_2p_3}\\
\underline{\underline{\rho}}^{6}_{p_1p_2p_{3_1}} &= \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{6}_{p_1p_2p_3}\\
\underline{\underline{\rho}}^{7}_{p_1p_2p_3} &= \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{7}_{p_1p_2p_3}\\
\underline{\underline{\rho}}^{8}_{p_1p_2p_3} &= \underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{8}_{p_1p_2p_3}\,.
\end{aligned}$$

(73)

The alternative representations are described in Eq (74).


(p_3+1)
(p_2+1)
(p_1+1)
$\underline{\underline{\mathbf{a}}}^{1}_{p_1p_2p_3}\,\underline{\underline{\mathbf{R}}}^{1}_{p_1p_2p_3} = [\ \rho^{1} \cdots 0 \cdots\ 0 \cdots\ 0\ \cdots\ 0 \cdots\cdots 0\ ]$
$\underline{\underline{\mathbf{a}}}^{7}_{p_1p_2p_3}\,\underline{\underline{\mathbf{R}}}^{7}_{p_1p_2p_3} = [\ 0 \cdots \rho^{7} \cdots\ 0 \cdots\ 0\ \cdots\ 0 \cdots\cdots 0\ ]$
(p_3+1)
(p_2+1)
(p_1+1)
$\underline{\underline{\mathbf{a}}}^{2}_{p_1p_2p_3}\,\underline{\underline{\mathbf{R}}}^{2}_{p_1p_2p_3} = [\ ]$
$\underline{\underline{\mathbf{a}}}^{6}_{p_1p_2p_3}\,\underline{\underline{\mathbf{R}}}^{6}_{p_1p_2p_3} = [\ 0 \cdots\cdots 0 \cdots\ 0 \cdots \rho^{6}\ \cdots\ 0 \cdots 0\ ]$
$\underline{\underline{\mathbf{a}}}^{3}_{p_1p_2p_3}\,\underline{\underline{\mathbf{R}}}^{3}_{p_1p_2p_3} = [\ 0 \cdots\cdots 0\ \cdots\ 0 \cdots\ 0 \cdots \rho^{3} \cdots 0\ ]$
$\underline{\underline{\mathbf{a}}}^{5}_{p_1p_2p_3}\,\underline{\underline{\mathbf{R}}}^{5}_{p_1p_2p_3} = [\ 0 \cdots\cdots 0\ \cdots\ 0 \cdots\ 0 \cdots\ 0 \cdots \rho^{5}\ ]$

(74)

The autocorrelation block matrices $\underline{\mathbf{R}}_{p_1p_2p_3}[i]$ are not hermitian [3]. However, they do satisfy the relationship

$$\underline{\mathbf{R}}_{p_1p_2p_3}[-i] = \underline{\mathbf{R}}^{H}_{p_1p_2p_3}[i] \tag{75}$$

This is sufficient to make the super block matrices $\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}$ hermitian. They are also conjugate centrosymmetric:

$$\underline{\mathbf{J}}(\underline{\underline{\mathbf{R}}}_{p_1p_2p_3})\underline{\mathbf{J}} = (\underline{\underline{\mathbf{R}}}_{p_1p_2p_3})^* \tag{76}$$

in which $\underline{\mathbf{J}}$ is a $(p_1+1)(p_2+1)(p_3+1)\times(p_1+1)(p_2+1)(p_3+1)$ reflection matrix.

$$\underline{\mathbf{J}}(\underline{\underline{\mathbf{R}}}_{p_1p_2p_3})^* \underline{\mathbf{J}}\underline{\mathbf{J}}\underline{\underline{\mathbf{a}}}^1_{p_1p_2p_3} = \underline{\mathbf{J}}\underline{\underline{\boldsymbol{\rho}}}^1_{p_1p_2p_3} \tag{77}$$

Observing that $\underline{\mathbf{J}}\underline{\underline{\boldsymbol{\rho}}}^1_{p1p2p3} = \underline{\underline{\boldsymbol{\rho}}}^5_{p1p2p3}$, Eq (70) then becomes

$$\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}(\underline{\mathbf{J}}\underline{\underline{\mathbf{a}}}^1_{p_1p_2p_3})^* = (\underline{\underline{\boldsymbol{\rho}}}^5_{p_1p_2p_3})^* = \underline{\underline{\boldsymbol{\rho}}}^5_{p_1p_2p_3} \tag{78}$$

Comparing Eq (77) with Eq (78), it can be concluded that $\underline{\underline{\mathbf{a}}}^1_{p_1p_2p_3}$ is obtained as the complex conjugate of $\underline{\underline{\mathbf{a}}}^5_{p_1p_2p_3}$. Figure 31 illustrates the coefficient sets for octant 1 and 5 ($d=1$ and $d=5$) which have the hermitian symmetry property. Conjugate pairs are

$$\begin{aligned}
a^1_{p_1p_2p_3}[i,j,k] &= a^5_{p_1p_2p_3}[-i,-j,-k]\\
a^2_{p_1p_2p_3}[i,-j,k] &= a^6_{p_1p_2p_3}[-i,j,-k]\\
a^3_{p_1p_2p_3}[-i,-j,k] &= a^7_{p_1p_2p_3}[i,j,-k]\\
a^4_{p_1p_2p_3}[-i,j,k] &= a^8_{p_1p_2p_3}[i,-j,-k]
\end{aligned} \tag{79}$$

A fast computational algorithm for the solution of $\underline{\underline{\mathbf{a}}}^1_{p_1p_2p_3}$ is not based on direct solution for the 3-D AR parameters, but it is based on solving a special variant of the 3-D AR algorithm involving the relationship

$$\left[\underline{\mathbf{I}}\ \ \underline{\mathbf{A}}_{p_1p_2p_3}[1]\cdots\underline{\mathbf{A}}_{p_1p_2p_3}[p_1]\right]\underline{\underline{\mathbf{R}}}_{p_1p_2p_3} = [\ \underline{\mathbf{P}}_{p_1p_2p_3}\ \ \underline{\mathbf{0}}\ \cdots\ \underline{\mathbf{0}}\ ] \tag{80}$$

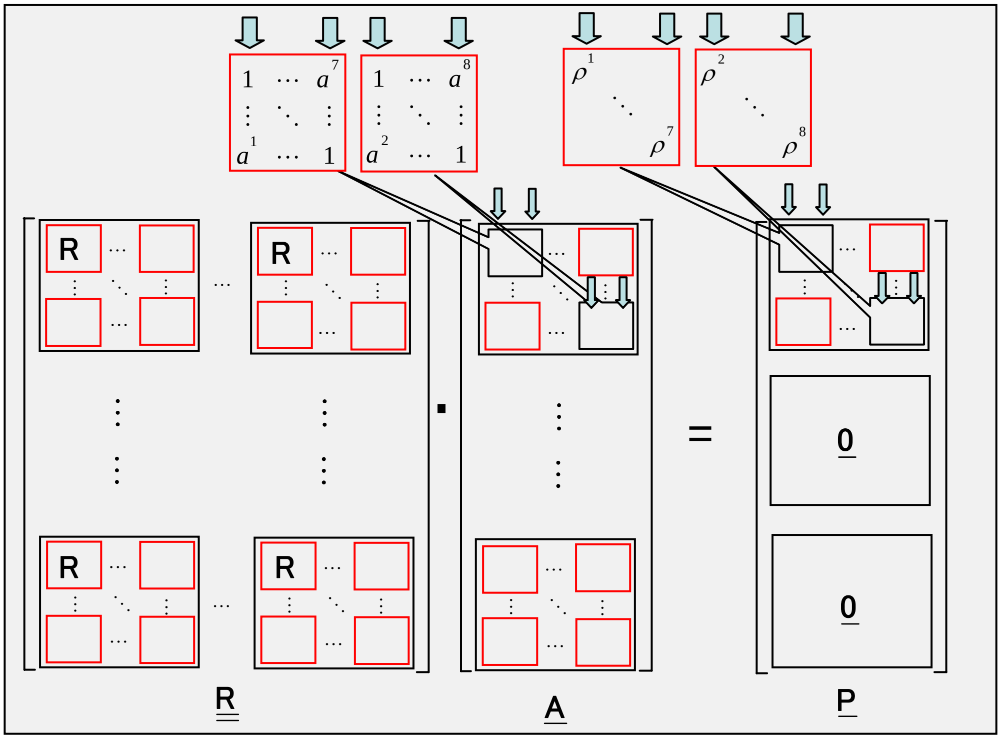


Figure 32. Using the results obtained from Eq (74), a new structure is formed by combining the *super block* linear prediction parameter matrixes $\underline{\mathsf{A}}$ and the *super block* covariance matrixes $\underline{\mathsf{P}}$.

in which $\underline{\mathsf{I}}$ is a $(p_1+1)(p_3+1)\times(p_1+1)(p_3+1)$ identity matrix. Each block linear prediction parameter matrix $\underline{\mathsf{A}}_{p_1p_2p_3}[p_1]$ and the block linear prediction covariance matrix $\underline{\mathsf{P}}_{p_1p_2p_3}$ have dimension $(p_2+1)(p_3+1)\times(p_2+1)(p_3+1)$. Multiplying both sides of Eq (80) by $\underline{\mathsf{P}}_{p_1p_2p_3}$, the result can be expressed in an alternative form

$$\underline{\underline{\mathsf{R}}}\bullet\begin{bmatrix}\underline{\mathsf{I}}*\underline{\mathsf{P}}^{-1}*\rho_1\\ \underline{\mathsf{A}}_1*\underline{\mathsf{P}}^{-1}*\rho_1\\ \vdots\\ \underline{\mathsf{A}}_p*\underline{\mathsf{P}}^{-1}*\rho_1\end{bmatrix}=\begin{bmatrix}\mathsf{P}*\underline{\mathsf{P}}^{-1}*\rho_1\\ \mathsf{0}*\underline{\mathsf{P}}^{-1}*\rho_1\\ \vdots\\ \mathsf{0}*\underline{\mathsf{P}}^{-1}*\rho_1\end{bmatrix}=\begin{bmatrix}\underline{\mathsf{I}}*\rho_1\\ \underline{\mathsf{0}}\\ \vdots\\ \underline{\mathsf{0}}\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}\rho_1&&\\&\ddots&\\&&\rho_1\end{bmatrix}\\ \underline{\mathsf{0}}\\ \vdots\\ \underline{\mathsf{0}}\end{bmatrix}\qquad(81)$$

In other words,

$$\underline{\underline{\mathbf{R}}} \bullet \underline{\mathbf{P}}^{-1} \times \rho_1 \times [\underline{\mathbf{I}}\ \underline{\mathbf{A}}_1 \ \cdots\cdots \underline{\mathbf{A}}_p]^T = \underline{\mathbf{P}}^{-1} \times \rho_1 \times [\ \underline{\mathbf{P}}\ \ \underline{\mathbf{0}} \ \cdots\cdots \ \underline{\mathbf{0}}\ ]^T . \tag{82}$$

As shown above, it is clear that we only care about the elements in the first rows of $\underline{\mathbf{A}}$ and $\underline{\mathbf{P}}$. Using the hermitian symmetry property shown in Eq (79), we consider only four octant coefficient sets and associated variances to build the new structure of the normal equation. If one now compares Eq (80) with Eq (84), we note that

$$\underline{\rho}^{1}_{p_1p_2p_3} = \underline{\mathbf{P}}_{p_1p_2p_3}\ \underline{\mathbf{a}}^{1}_{p_1p_2p_3}[0] \tag{83}$$

so that the following relationships are the result

$$\begin{aligned}
\underline{\mathbf{a}}^{1}_{p_1p_2p_3}[0] &= [1\cdots 0\ \ 0\cdots 0\ \ 0\cdots 0\ \ \cdots\ \ 0\cdots 0][\underline{\mathbf{P}}_{p_1p_2p_3}]^{-1} \\
\underline{\mathbf{a}}^{7}_{p_1p_2p_3}[0] &= [0\cdots 1\ \ 0\cdots 0\ \ 0\cdots 0\ \ \cdots\ \ 0\cdots 0][\underline{\mathbf{P}}_{p_1p_2p_3}]^{-1} \\
\underline{\mathbf{a}}^{2}_{p_1p_2p_3}[0] &= [0\cdots 0\ \ 1\cdots 0\ \ 0\cdots 0\ \ \cdots\ \ 0\cdots 0][\underline{\mathbf{P}}_{p_1p_2p_3}]^{-1} \\
\underline{\mathbf{a}}^{8}_{p_1p_2p_3}[0] &= [0\cdots 0\ \ 0\cdots 1\ \ 0\cdots 0\ \ \cdots\ \ 0\cdots 0][\underline{\mathbf{P}}_{p_1p_2p_3}]^{-1}
\end{aligned} \tag{84}$$

Eq (85) below is similar to the conjugate relationships of the forward and backward LP coefficients observed in the 1-D AR analysis. By scaling to force $a^{1}[0,\ 0,\ 0] = 1$

$$\begin{aligned}
\underline{\mathbf{a}}^{1}_{p_1p_2p_3}[i] &= \underline{\mathbf{a}}^{1}_{p_1p_2p_3}[0]\ \underline{\mathbf{A}}_{p_1p_2p_3}[i] \\
\underline{\mathbf{a}}^{7}_{p_1p_2p_3}[i] &= \underline{\mathbf{a}}^{7}_{p_1p_2p_3}[0]\ \underline{\mathbf{A}}_{p_1p_2p_3}[i] \\
\underline{\mathbf{a}}^{2}_{p_1p_2p_3}[i] &= \underline{\mathbf{a}}^{2}_{p_1p_2p_3}[0]\ \underline{\mathbf{A}}_{p_1p_2p_3}[i] \\
\underline{\mathbf{a}}^{8}_{p_1p_2p_3}[i] &= \underline{\mathbf{a}}^{8}_{p_1p_2p_3}[0]\ \underline{\mathbf{A}}_{p_1p_2p_3}[i]
\end{aligned} \tag{85}$$

Thus, having solved for matrices $\underline{\mathbf{P}}_{p_1p_2p_3}$ and $\underline{\mathbf{A}}_{p_1p_2p_3}[i]$ for $i = 1$ to p_1 from Eq (83), the relationships of Eq (85) may then be solved to four 3-D autoregressive (AR) parameter sets and associated variances.

5.2.2. Recursive Solution for 3-D LP Parameter Matrices

Since $\underline{\underline{\mathbf{R}}}_p = \underline{\underline{\mathbf{R}}}_p^H$ is super block Toeplitz (triply Toeplitz), we can show that the 3-D autocorrelation matrix is hermitian and conjugate centrosymmetric $\underline{\underline{\mathbf{R}}}_p = \mathbf{J}\underline{\underline{\mathbf{R}}}_p^*\mathbf{J}$. Reflection matrix $\underline{\mathbf{J}}$ has dimension $(p_2+1)(p_3+1)\times(p_2+1)(p_3+1)$. The triply Toeplitz structure can be exploited to develop the 3-D version of the recursive 1-D Levinson algorithm that solves Eq (80). As this chapter has already presented, Eq (80) may alternatively be expressed as Eq (86). Using the centrosymmetric property and the identity matrix $\underset{-}{\mathbf{I}}$, we can find the 3-D reflection coefficient matrix $\underline{\mathbf{K}}_{p+1}$, such that the following expression is valid

$$\left[\underset{-}{\mathbf{I}}\;\; \underline{\mathbf{A}}_{p+1}[1]\cdots\cdots\underline{\mathbf{A}}_{p+1}[p]\;\; \underline{\mathbf{A}}_{p+1}[p+1]\right]=$$

$$\left[\underset{-}{\mathbf{I}}\;\; \underline{\mathbf{A}}_p[1]\cdots\cdots\underline{\mathbf{A}}_p[p]\;\; \underline{\mathbf{0}}\right]+\left[\underline{\mathbf{K}}_{p+1}[\underline{\mathbf{0}}\;\; \underline{\mathbf{J}}\;\; \underline{\mathbf{A}}_p^*[p]\underline{\mathbf{J}}\cdots\underline{\mathbf{J}}\;\; \underline{\mathbf{A}}_p^*[1]\underline{\mathbf{J}}\;\;\; \underset{-}{\mathbf{I}}\right] \qquad (86)$$

If we multiply both sides on the right by $\underline{\underline{\mathbf{R}}}_p$ at order $p+1$, it will yield

$$[\underline{\mathbf{P}}_{p+1}\;\; \underline{\mathbf{0}}\;\cdots\;\underline{\mathbf{0}}\;\; \underline{\mathbf{0}}\;]=[\underline{\mathbf{P}}_p\;\;\; \underline{\mathbf{0}}\;\;\cdots\;\;\underline{\mathbf{0}}\;\; \underline{\Delta}_{p+1}]+\underline{\mathbf{K}}_{p+1}[\;\underline{\mathbf{J}}\;\; \underline{\Delta}_{p+1}^*\underline{\mathbf{J}}\;\;\; \underline{\mathbf{0}}\cdots\underline{\mathbf{0}}\;\; \underline{\mathbf{J}}\,\underline{\mathbf{P}}_p^*\underline{\mathbf{J}}] \qquad (87)$$

where

$$\underline{\Delta}_{p+1}=\underline{\underline{\mathbf{R}}}_p[p+1]+\sum_{i=1}^{p}\sum_{j=1}^{p}\underline{\mathbf{A}}_p[j]\underline{\underline{\mathbf{R}}}_p[i+1-j] \qquad (88)$$

We can apply the recursive relationship concept from the original Levinson algorithm.

Also, Eq (69) will be balanced if we select

$$\underline{\mathbf{K}}_{p+1} = \underline{\mathbf{A}}_{p+1}[p+1] = -\underline{\Delta}_{p+1}(\underline{\mathbf{J}}\ \underline{\mathbf{P}}^{*}_{p}\ \underline{\mathbf{J}})^{-1} = -\underline{\Delta}_{p+1}\ \underline{\mathbf{J}}\ (\underline{\mathbf{P}}^{*}_{p})^{-1}\ \underline{\mathbf{J}} \tag{89}$$

which creates the following order-update recursion

$$\underline{\mathbf{A}}_{p+1}[i] = \underline{\mathbf{A}}_{p}[i] + \underline{\mathbf{K}}_{p+1}(\underline{\mathbf{J}}\ \underline{\mathbf{A}}^{*}_{p}[p+1-i]\underline{\mathbf{J}}) \tag{90}$$

From Eq (87) and (89), it is possible to derive the following recursion of the covariance matrix

$$\underline{\mathbf{P}}_{p+1} = (\underline{\mathbf{I}} - \underline{\mathbf{K}}_{p+1}[\underline{\mathbf{J}}\ \underline{\mathbf{K}}^{*}_{p+1}\underline{\mathbf{J}}])\underline{\mathbf{P}}_{p} = \underline{\mathbf{P}}_{p}(\underline{\mathbf{I}} - [\underline{\mathbf{J}}\ \underline{\mathbf{K}}^{T}_{p+1}\underline{\mathbf{J}}])\underline{\mathbf{K}}^{T}_{p+1}\,. \tag{91}$$

This concludes the recursive updates, showing that the centrosymmetry introduced by the pre and post multiplication of reflection matrices has provided a further reduction in computations.

5.2.3. Three-Dimensional Lattice Linear Prediction Parameter Estimation Method

Assuming that a finite data record of 3-D data samples $x[i, j, k]$ has been acquired for $0 \le i \le n_1 - 1$, $0 \le j \le n_2 - 1$ and $0 \le k \le n_3 - 1$, the 3-D block forward and 3-D block backward linear prediction error matrices of order $p_1 p_2 p_3$ and dimension $(n_1 - p_1)(n_2 - p_2)\mathrm{X}(p_1 + 1)(p_2 + 1)$ are then defined as

$$\begin{gathered}\underline{\mathbf{e}}^{\mathbf{b}}_{p_1p_2p_3}[n_1] = \underline{\mathbf{X}}\ [n_1 - p_1] + \sum_{i=1}^{p_1}(\underline{\mathbf{J}}\underline{\mathbf{A}}^{*}_{p_1p_2p_3}[i]\underline{\mathbf{J}})\ \underline{\mathbf{X}}\ [n_1 - p_1 + i] \\ \underline{\mathbf{e}}^{\mathbf{f}}_{p_1p_2p_3}[n_1] = \underline{\mathbf{X}}\ [n_1] + \sum_{i=1}^{p_1}\underline{\mathbf{A}}_{p_1p_2p_3}[i]\underline{\mathbf{X}}\ [n_1 - i]\end{gathered} \tag{92}$$

In which $\underline{\mathsf{X}}$ has dimension $(n_3 - p_3)(n_2 - p_2) \times (p_3 + 1)(p_2 + 1)$

$$\underline{\mathsf{X}} = \begin{bmatrix} \mathsf{X}[p_2] & \cdots & \underline{\mathsf{X}}[n_2 - 1] \\ \vdots & \ddots & \vdots \\ \mathsf{X}[n_2 - p_2 - 1] & \cdots & \mathsf{X}[p_2] \\ \vdots & \ddots & \vdots \\ \mathsf{X}[0] & \cdots & \mathsf{X}[n_2 - p_2 - 1] \end{bmatrix} \tag{93}$$

and this is a super block Toeplitz data matrix as shown in Eq (94).

$$\underline{\underline{\mathsf{X}}} = \begin{bmatrix} \underline{\mathsf{X}}[p_1] & \cdots & \underline{\mathsf{X}}[n_1 - 1] \\ \vdots & \ddots & \vdots \\ \underline{\mathsf{X}}[n_1 - p_1 - 1] & \cdots & \underline{\mathsf{X}}[p_1] \\ \vdots & \ddots & \vdots \\ \underline{\mathsf{X}}[0] & \cdots & \underline{\mathsf{X}}[n_1 - p_1 - 1] \end{bmatrix} \tag{94}$$

The 3-D forward and backward linear prediction squared error matrices are defined as

$$\underline{\mathsf{P}}^{\mathsf{f}}_{p_1 p_2 p_3} = \frac{1}{n_1 - p_1} \sum_{i=p_1}^{n_1-1} \underline{\mathsf{e}}^{\mathsf{f}}_{p_1 p_2 p_3}[i] (\underline{\mathsf{e}}^{\mathsf{f}}_{p_1 p_2 p_3}[i])^H \tag{95}$$

$$\underline{\mathsf{P}}^{\mathsf{b}}_{p_1 p_2 p_3} = \frac{1}{n_1 - p_1} \sum_{i=p_1}^{n_1-1} \underline{\mathsf{e}}^{\mathsf{b}}_{p_1 p_2 p_3}[i] (\underline{\mathsf{e}}^{\mathsf{b}}_{p_1 p_2 p_3}[i])^H \tag{96}$$

Note that $E\{\underline{\mathsf{P}}^{\mathsf{f}}_{p_1 p_2 p_3}\} = E\{\ \underline{\mathsf{J}}(\underline{\mathsf{P}}^{\mathsf{b}}_{p_1 p_2 p_3})^*\ \underline{\mathsf{J}}\} = \underline{\mathsf{P}}_p$. That the recursive 3-D lattice forms a 3-D linear prediction error filter relationship between the 3-D forward and backward linear prediction matrix errors may be seen by using Eq (90). Substituting Eq (91) into Eq (92) yields

$$\begin{gathered} \underline{\mathsf{e}}^{\mathsf{f}}_{p+1}[n] = \underline{\mathsf{e}}^{\mathsf{f}}_{p}[n] + \underline{\mathsf{K}}_{p+1}\, \underline{\mathsf{e}}^{\mathsf{b}}_{p}[n-1] \\ \underline{\mathsf{e}}^{\mathsf{b}}_{p+1}[n] = \underline{\mathsf{e}}^{\mathsf{b}}_{p}[n-1] + (\underline{\mathsf{J}}\, \underline{\mathsf{K}}^*_{p+1}\, \underline{\mathsf{J}}\ )\ \underline{\mathsf{e}}^{\mathsf{f}}_{p}[n]\,. \end{gathered} \tag{97}$$

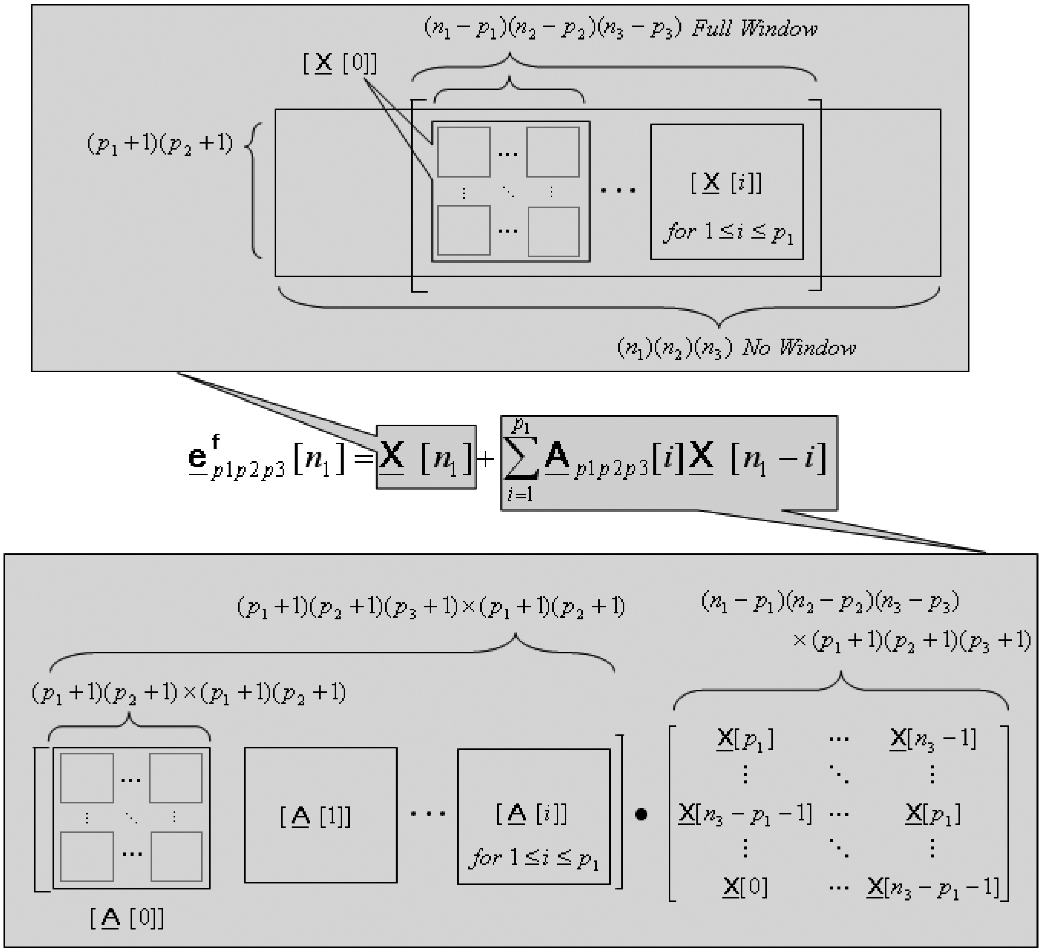

$(n_1-p_1)(n_2-p_2)(n_3-p_3)$ Full Window
$[\underline{\mathbf{X}}[0]]$
$(p_1+1)(p_2+1)$
$[\underline{\mathbf{X}}[i]]$
for $1 \le i \le p_1$
$(n_1)(n_2)(n_3)$ No Window
$\underline{\mathbf{e}}^{\mathbf{f}}_{p1p2p3}[n_1] = \underline{\mathbf{X}}[n_1] + \sum_{i=1}^{p_1} \underline{\mathbf{A}}_{p1p2p3}[i]\underline{\mathbf{X}}[n_1-i]$
$(p_1+1)(p_2+1)(p_3+1)\times(p_1+1)(p_2+1)$
$(n_1-p_1)(n_2-p_2)(n_3-p_3)$
$\times(p_1+1)(p_2+1)(p_3+1)$
$(p_1+1)(p_2+1)\times(p_1+1)(p_2+1)$
$[\underline{\mathbf{A}}[1]]$
$[\underline{\mathbf{A}}[i]]$
for $1 \le i \le p_1$
$[\underline{\mathbf{A}}[0]]$
$\underline{\mathbf{X}}[p_1]$
$\underline{\mathbf{X}}[n_3-1]$
$\underline{\mathbf{X}}[n_3-p_1-1]$
$\underline{\mathbf{X}}[p_1]$
$\underline{\mathbf{X}}[0]$
$\underline{\mathbf{X}}[n_3-p_1-1]$

Figure 33. Super block vector representations of the 3-D linear prediction error equation

Rather than estimating the 3-D octant-space parameters directly from the data, it is more advantageous (for both computational reduction and estimation performance) to estimate the 3-D super block linear prediction parameter matrices. Using the relationships cited before in this section, it is possible to obtain the estimates of the octant-space linear prediction parameters. Based on the property, $E\{\underline{\mathbf{P}}^{\mathbf{f}}_p\} = E\{\underline{\mathbf{J}}(\underline{\mathbf{P}}^{\mathbf{b}}_p)^*\underline{\mathbf{J}}\} = \underline{\mathbf{P}}_p$, we can form an estimate $\hat{\underline{\mathbf{P}}}_p$ of $\underline{\mathbf{P}}_p$ by taking half of the sum of the forward and backward 3-D linear prediction squared errors

$$\frac{\underline{\mathbf{P}}_{p}^{\mathbf{f+b}}}{2} + \frac{(\underline{\mathbf{P}}_{p}^{\mathbf{f}} + \underline{\mathbf{J}}\,(\underline{\mathbf{P}}_{p+1}^{\mathbf{b}})^{*}\,\underline{\mathbf{J}})}{2} = \hat{\underline{\mathbf{P}}}_{p+1} \tag{98}$$

which is a newer formulation of the squared error sum than those previously cited in the literature [20]. Since the above sum has positive diagonal elements [3], which was formed from squared errors, an approach to obtain a least squares estimate can be obtained by minimizing the sum of the forward and backward 3-D linear prediction squared errors along the diagonal (that is, the trace), min $tr\{\underline{\mathbf{P}}_{p+1}^{\mathbf{f+b}}\}$, as shown in Figure 34.

Let us first define an estimate of the crosscorrelation between the block forward and backward 3-D linear prediction errors

$$\underline{\mathbf{P}}_{p_1p_2p_3}^{\mathbf{fb}} = \frac{1}{n_1 - p_1}\sum_{i=p_1}^{n_1-1} \underline{\mathbf{e}}_{p_1p_2p_3}^{\mathbf{f}}[i]\,(\underline{\mathbf{e}}_{p_1p_2p_3}^{\mathbf{f}}[i])^{H} \tag{99}$$

Substituting Eq (97) into Eq (96) and (98) and combining the partial correlation matrix $\underline{\Delta}_p = \underline{\mathbf{P}}_{p}^{\mathbf{fb}} + (\underline{\mathbf{J}}\underline{\mathbf{P}}_{p}^{\mathbf{fb}}\,\underline{\mathbf{J}})^{T}$, the result be expressed as

$$\underline{\mathbf{P}}_{p+1}^{\mathbf{f+b}} = \underline{\mathbf{P}}_{p}^{\mathbf{f+b}} + \underline{\mathbf{K}}_{p+1}\underline{\Delta}_{p}^{H} + \underline{\Delta}_{p}\,\underline{\mathbf{K}}_{p+1}^{H} + \underline{\mathbf{K}}_{p+1}(\underline{\mathbf{J}}\,\underline{\mathbf{P}}_{p}^{\mathbf{f+b}}\,\underline{\mathbf{J}})^{*}\,\underline{\mathbf{K}}_{p+1}^{H} \tag{100}$$

By completing the square, Eq (100) may be expressed in the form

$$\underline{\mathbf{P}}_{p+1}^{\mathbf{f+b}} = \underline{\mathbf{P}}_{p}^{\mathbf{f+b}} + \underline{\Delta}_{p}\,[\underline{\mathbf{J}}\,\underline{\mathbf{P}}_{p}^{\mathbf{f+b}}\,\underline{\mathbf{J}}]^{-1}\,\underline{\Delta}_{p} + \underline{\mathbf{C}}_{p+1}\,[\underline{\mathbf{J}}\,(\underline{\mathbf{P}}_{p}^{\mathbf{f+b}})^{*}\,\underline{\mathbf{J}}]\,\underline{\mathbf{C}}_{p+1}^{H} \tag{101}$$

where $\underline{\mathbf{C}}_{p+1} = \underline{\mathbf{K}}_{p+1} + \underline{\Delta}_{p}\,(\underline{\mathbf{J}}\,\underline{\mathbf{P}}_{p}^{\mathbf{f+b}}\,\underline{\mathbf{J}})^{-1}$. To minimize the $tr\{\underline{\mathbf{P}}_{p+1}^{\mathbf{f+b}}\}$ with respect to the

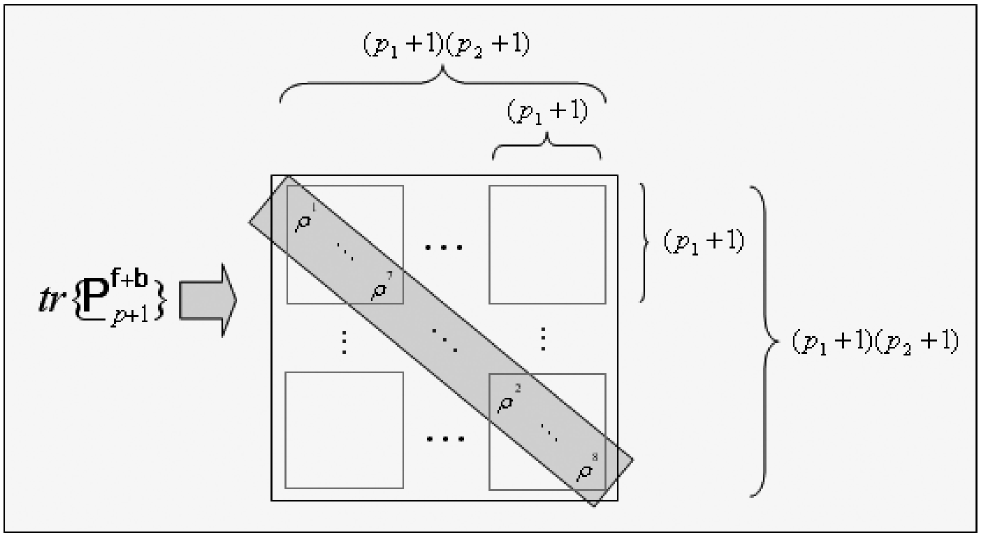


Figure 34. 3-D linear prediction squared errors along the diagonal.

single unknown matrix $\underline{\mathsf{K}}_{p+1}$, we note that only the third term of Eq (101) depends on $\underline{\mathsf{K}}_{p+1}$. Because this third term is a quadratic matrix product that can only contribute positive values to the diagonal elements of $\underline{\mathsf{P}}^{\mathsf{f+b}}_{p+1}$, the only choice that will minimize the trace is $\underline{\mathsf{C}}_{p+1} = 0$, or (102)

$$\underline{\mathsf{K}}_{p+1} = \underline{\Delta}_p [\underline{\mathsf{J}}\, \underline{\mathsf{P}}^{\mathsf{f+b}}_p \underline{\mathsf{J}}]^{-1}$$

By selecting $\underline{\mathsf{K}}_{p+1}$ as indicated in Eq (102), the sum of forward and backward 3-D squared errors that minimize the trace may be obtained by inspection of Eq (101) as

$$\underline{\mathsf{P}}^{\mathsf{f+b}}_{p+1} = \underline{\mathsf{P}}^{\mathsf{f+b}}_p + \underline{\mathsf{K}}_{p+1} \underline{\Delta}^H_p \tag{103}$$

In the 3-D case, we need only compute the forward block linear prediction parameter matrix $\underline{\mathsf{A}}$, the forward block linear prediction covariance matrix $\underline{\mathsf{P}}$, and forward block reflection matrix $\underline{\mathsf{K}}$. The backward matrices are obtained by pre- and post-multiplication of the forward matrix by the reflection matrix $\underline{\mathsf{J}}$. Thus, the centrosymmetry property has provided a further reduction in computation.

5.3. Application Example

The 3-D Yule-Walker algorithm and lattice LP parameter estimation method from the thesis have been implemented in a MATLAB script functions *levin_3D, yulwk_3D, and lattice_3D,* available from www.engr.orst.edu/~liewj/mdata.

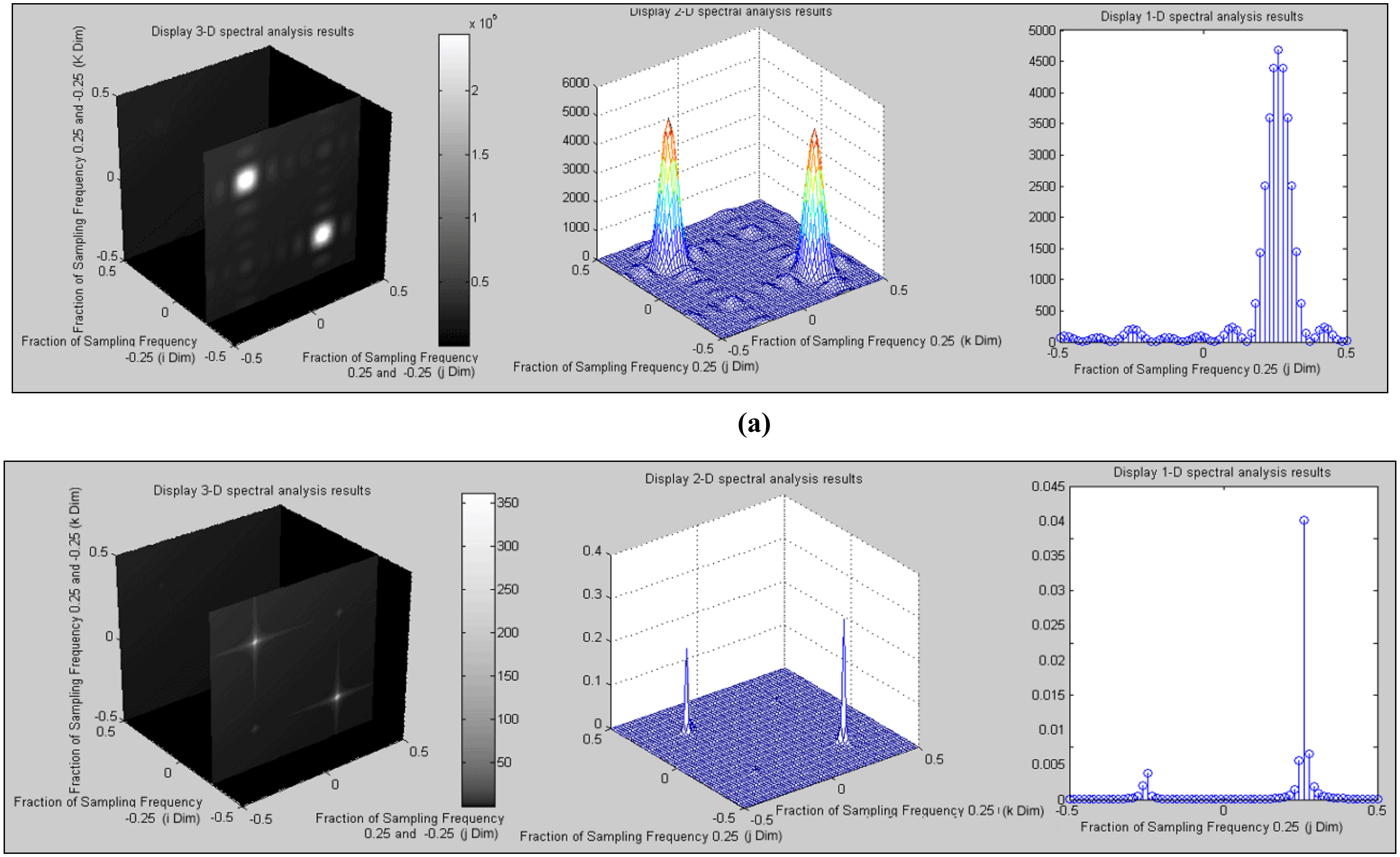
The spectrum of the 1-D and 2-D plots shown in Figure 35 are well known [1,3,10]. In the 2-D plot, the spectrum's peaks are located using the x and y axes, and the z axis represents the magnitude.

In order to visualize the performance of 3-D spectral estimation, the magnitude of the 3-D power spectral density is illustrated in 3-D space with the points whose coordinates are comprised of the i, j, and k elements as shown in Figure 35 (top left and bottom left). A vertical strip on the right side of the plots indicates the mapping color (gray scale) of the magnitude of the 3-D PSD. For example, the brightest spot in the 3-D plot indicates the highest peak point in the 2-D plot. The 2-D plot is a planar slice from the volume space taken at a specific frequency of interest, where the gray scale from the 3-D plots now indicate the amplitude. The 1-D plot is line slice from the 2-D plot, taken at a specific frequency of interest. These slices are useful to show where interesting regions exist in the space of the spectral estimation volume.

5.3.1. Three-Dimensional Yule-Walker Spectral Estimation

Figure 35 illustrates the resolution and accuracy of three dimensional (3-D) spectrum analysis for five complex 3-D sinusoidal signals, embedded in low level white noise (yielding a signal-to-noise ratio (SNR) of about 30 dB), using two spectral estimation methods: one based on 3-D FFT spectral estimation and one based on the 3-D Yule-Walker spectral estimation approach. A 32×32×32-point 3-D data set of noisy sinusoidal samples is generated with MATLAB to implement the algorithm. The estimates are plotted as fractions of the sampling frequencies (#1: i =0.00, j =0.00, k =0.00; #2: i =-0.25, j =0.25, k =-0.25; #3: i =0.25, j =-0.25, k = -0.25; #4: i =0.25, j =-0.25, k =0.25; #5: i =-0.25, j =0.25, k =-0.25) and show five spots which have their highest magnitudes at the exact sinusoidal frequencies. In this plot, the color bar indicates the amplitude. The figures in the middle columns of Figure 35 show the 2-D AR estimated spectra using one slice of the original 3-D PSD matrix. The sliced 2-D matrix shows the plane at a fractional sampling frequency i =0.25. The figures in the last column of Figure 35 illustrate 1-D AR spectral estimation results using one column vector of the 2-D PSD matrix. The figures in the last column are a 1-D plot of the plane of j =0.25.

We conclude that the spectral estimate based on 3-D Yule-Walker solution provides a much better spectral estimate than the estimate given by 3-D FFT-based estimate in Figure 35. The plots shown in Figure (a) look markedly worse than Figure 35 (b). They have not only bigger mainlobes (bigger spheres), but also several sidelobes (several small spheres).



(a)

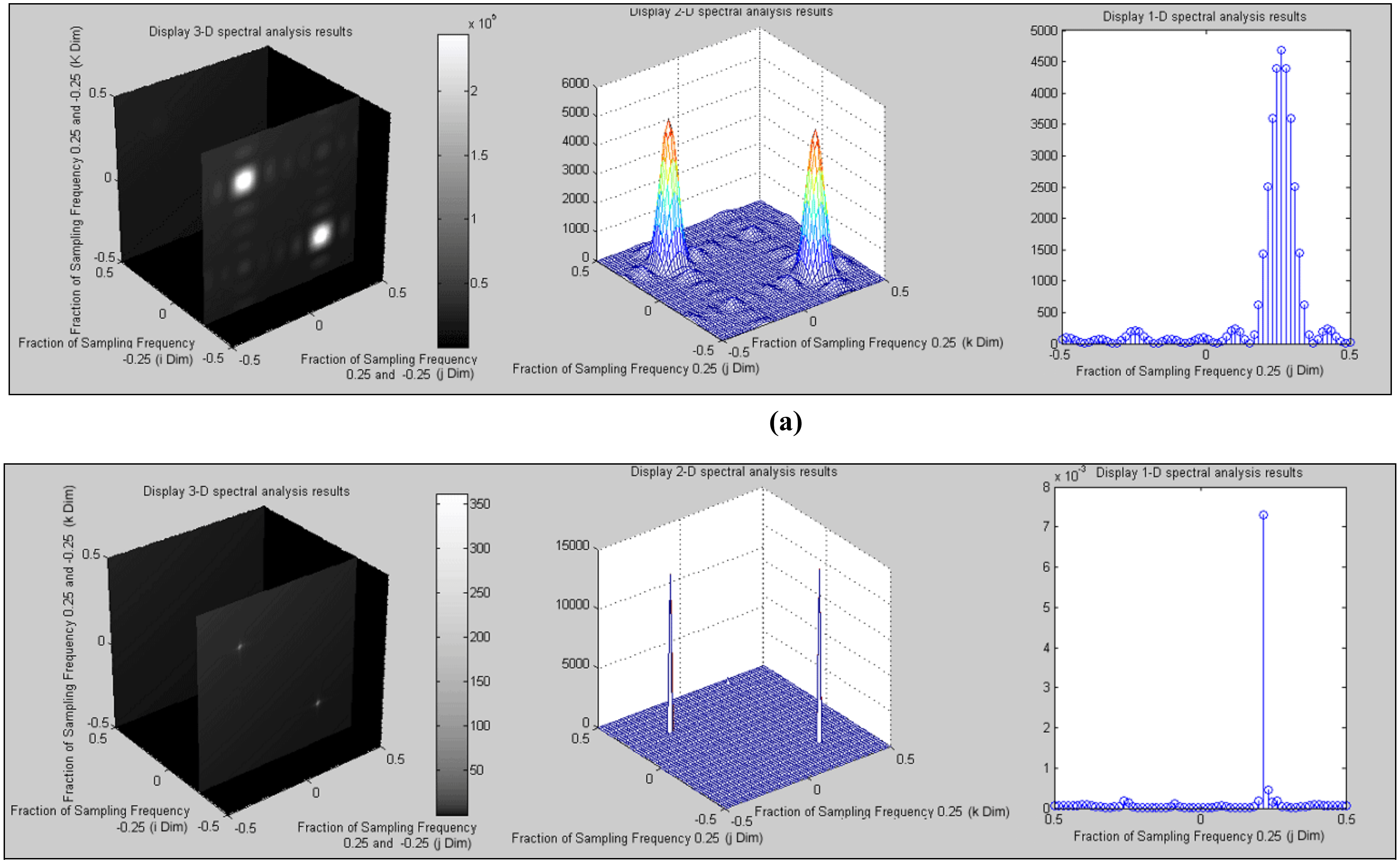
(b)

Figure 35. Three-Dimensional autoregressive spectral estimates of five 3-D sinusoidal signals in white noise: (a) Classical FFT-based estimate (the peak spots at i_1=0.00, j_1=0.00, k_1=0.00; i_2=-0.25, j_2=0.25, k_2=-0.25; i_3=0.25, j_3=-0.25, k_3=-0.25; i_4=0.25, j_4= -0.25, k_4=0.25; i_5=-0.25, j_5=0.25, k_5=-0.25). (b) AR estimate by the 3-D Yule-Walker algorithm (the peak spots are same as (a))

5.3.2. Three-Dimensional Lattice LP Parameter Spectral Estimation

Figure 36 illustrates the resolution and accuracy of three dimensional (3-D) spectrum analysis for five complex 3-D sinusoidal signals, embedded in low level white noise (yielding a signal-to-noise ratio (SNR) of about 30 dB), using two spectral estimation methods: one based on 3-D FFT spectra estimation and one based on the 3-D Yule-Walker spectral estimation approach. A 32×32×32-point 3-D data set of noisy sinusoidal samples is generated with MATLAB to implement the algorithm. The estimates are plotted as fractions of the sampling frequencies (#1: i =0.00, j =0.00, k =0.00; #2: i =-0.25, j =0.25, k =-0.25; #3: i =0.25, j =-0.25, k = -0.25; #4: i =0.25, j =-0.25, k =0.25; #5: i =-0.25, j =0.25, k =-0.25) and show five spots which have their highest magnitudes at the exact sinusoidal frequencies. In this plot, the color bar indicates the amplitude. The figures in the middle columns of Figure 36 shows the 2-D AR estimated spectrum using one slice of the original 3-D PSD matrix. The sliced 2-D matrix shows the plane at fractional sampling frequency i =0.25. The figures in the last column in Figure 36 illustrate the 1-D AR spectral estimation results using one column vector of 2-D PSD matrix. The figures in the last column are a 1-D plot of the plane of j =0.25.

We conclude that the spectral estimate based on 3-D Lattice case gives a much better spectral estimate than the estimate given by 3-D FFT-based estimate as shown in Figure 36. In the 3-D Yule-Walker case as shown in Figure 36 (b), the plots look markedly different than Figure 36 (a), which have not only bigger mainlobes (bigger spheres), but also several sidelobes (several small spheres).



(a)

(b)

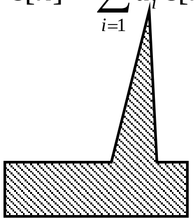
Figure 36. Three-Dimensional autoregressive spectral estimates of five 3-D sinusoidal signals in white noise: (a) Classical FFT-based estimate(the peak spots at i_1=0.00, j_1=0.00, k_1=0.00; i_2=-0.25, j_2=0.25, k_2=-0.25; i_3=0.25, j_3=-0.25, k_3=-0.25; i_4=0.25, j_4= -0.25, k_4=0.25; i_5=-0.25, j_5=0.25, k_5=-0.25). (b) 3-D Lattice LP parameter estimation method (the peak spots are same as (a))

5.3.3. Three-dimensional LP-based coding/compression

The use of new 3-D LP fast algorithms is not limited to the examples discussed so far. For example, by segmenting data into blocks and applying mathematical predictive extrapolation, data processing such as compression of digital audio (1-D case), image processing (2-D case), and video coding (3-D case) are possible. The new 3-D techniques introduced in this thesis will have a major impact on future coding/compression technique developments.

One-dimensional linear prediction (LP)-based coding/compression has long been used to compress 1-D signals [26]. It should be possible to extend to 3-D concepts/theory the currently successful 1-D LP-based techniques. The new 3-D techniques (3-D Yule-walker equation [29] and 3-D LS Lattice case with Fast Algorithm [30]) presented in this thesis can be applied to future 3-D LP-based data compression algorithms for efficient transmission and storage of 3-D data sets. If the data are well characterized by their 3-D LP coefficients, 3-D LP-based compression techniques could be used to reduce the number of bits to save or to transmit 3-D data. The quality of the predictive coefficients will directly affect the overall compression performance and the 3-D algorithm quality; the results of the 3-D spectral analysis in previous sections suggest that the quality will likely be very good.

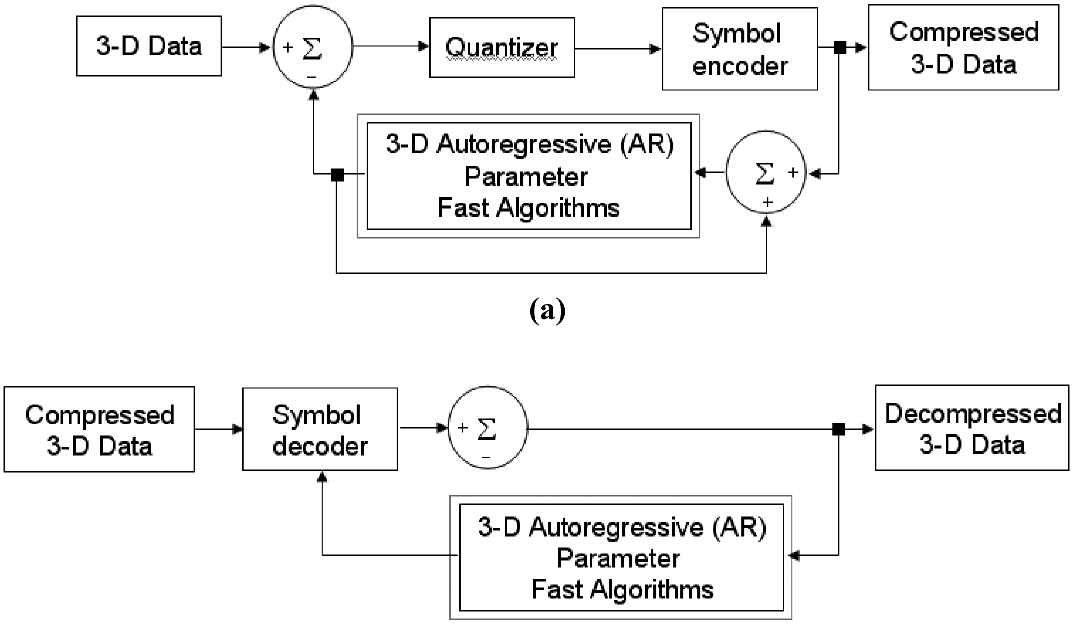
There have been many attempts in 1-D to develop different versions of the LP-based data compression algorithms [26]. Linear predictive coding (LPC) is an appropriate example to explain how LP coefficients are used for purposes of data compression, such as speech compression (1-D temporal domain algorithm) based on linear predictive modeling [26,27]. Since speech signals are not stationary, we are typically interested in the similarities over short time intervals. For vowel sounds which have a high degree of similarity within a selected sampling interval, the parametric modeling produces an efficient representation of the sound [26]. The autocorrelation $r_{ss}[i]$ which describes the redundancy in the signal $s[n]$ is computed only over a few time samples $i = 0,1,\cdots p$ and the compressed signal $s\overset{\wedge}{[n]}$ then can be defined as



$$s\overset{\wedge}{[n]} = \sum_{i=1}^{p} a_i \; s[n-i]$$

$$= \begin{bmatrix} r_{ss}[0] & r_{ss}[1] & \cdots & r_{ss}[p-1] \\ r_{ss}[1] & r_{ss}[0] & \cdots & \vdots \\ \vdots & & \ddots & \vdots \\ r_{ss}[p-1] & \cdots & \cdots & r_{ss}[0] \end{bmatrix} \cdot \begin{bmatrix} r_{ss}[1] \\ r_{ss}[2] \\ \vdots \\ r_{ss}[p] \end{bmatrix}. \qquad (104)$$

The basic idea of linear prediction coding/compression is to record the p LP coefficients as a compressed file in lieu of the actual data samples. Suppose we want to construct a prediction for $a(i)$ as a linear combination of its previous p samples called a p th order linear predictor. Obviously, compression can be accomplished only if one chooses a number p less than the number n of all data samples. If a compression ratio (p/n) is obtained nominally in the 1-D case, we would expect to achieve a minimum of $(p/n)^2$ or $(p/n)^3$ for 2-D and 3-D data sets with the 2-D and 3-D LP algorithms of this thesis.



(a)

(b)

Figure 37. New 3-D LP algorithms presented in this thesis would become foundational blocks in future 3-D LP-based compression algorithms. (a) Compression (b) Decompression.

In the case of a 2-D signal (typically an image), various efficient 2-D LP-based coding techniques have been developed and applied to the transmission of 2-D signals [26]. However, these techniques remap the neighboring pixels of a central analysis pixel into a 1-D vector so that Eq (104) is directly used [32], so it is not a true 2-D LP-based coding/compression technique. Furthermore, there has not been a 3-D LP-based coding/compression technique even though many 3-D data sets including hyperspectral imaging (HSI), interferometric synthetic aperture radar (IF-SAR) imaging, radar space-time adaptive processing (STAP), and high-definition television (HDTV) can be the source of data for processing by a 3-D compression technique.

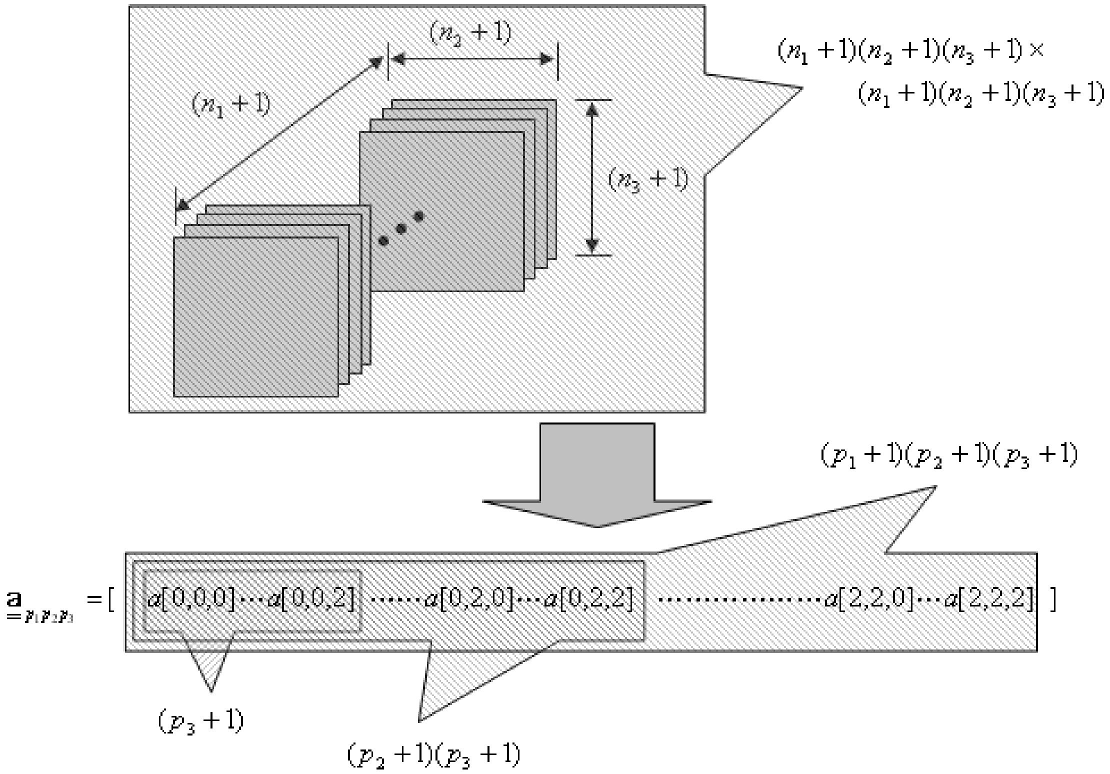

(n_2+1)
(n_1+1)
(n_3+1)
$(n_1+1)(n_2+1)(n_3+1)\times$
$(n_1+1)(n_2+1)(n_3+1)$
$(p_1+1)(p_2+1)(p_3+1)$
$\underline{\mathbf{a}}_{p_1p_2p_3} = [\ a[0,0,0]\cdots a[0,0,2]\ \cdots\cdots a[0,2,0]\cdots a[0,2,2]\ \cdots\cdots\cdots\cdots\cdots a[2,2,0]\cdots a[2,2,2]\ ]$
(p_3+1)
$(p_2+1)(p_3+1)$

Figure 38. 32×32×32 data samples. Only an order 2×2×2 LP coefficient super block vector (about 7 scalar values) is needed to represent all the 32^3 data values if the 3-D signal were simply a 3-D sinusoidal. The compression factor is would then be $(27/32)^3$, which is a very great savings, in which the 3-D LP coefficient data is saved in lieu of the original 32^3 sinusoidal data values.

Figure 37 shows how the new 3-D LP algorithms presented in this thesis could become foundational blocks in future 3-D LP-based compression/decompression algorithms.

The coding/compression efficiency factor would likely be the 1-D savings, only squared or cubed (for 2-D and 3-D data sets respectively). The new 3-D algorithms presented in Chapter 5 could be used to reduce the number of bits over that required to save the original 2-D or 3-D data samples. For example, an extreme case is a 3-D complex sinusoid of 32×32×32 data samples. If 3-D LP analysis is performed, only an order 2×2×2 LP coefficient *super block vector* [28] (about 27 scalar values) is needed to represent all the 32^3 data values as shown below in Eq (105).

$$\underline{\underline{\mathbf{a}}}_{p_1,p_2,p_3=1,1,1} = \left[\underline{\mathbf{a}}_{p_1,p_2,=1,1}[0]\ \underline{\mathbf{a}}_{p_1,p_2,=1,1}[1]\ \underline{\mathbf{a}}_{p_1,p_2,=1,1}[2]\right] \tag{105}$$

which is defined in terms of the *block vectors*

$$\underline{\mathbf{a}}_{p_3p_2p_1} = \left[\mathbf{a}_{p_3p_2p_1}[0] \quad \mathbf{a}_{p_3p_2p_1}[1] \quad \mathbf{a}_{p_3p_2p_1}[2]\right], \tag{106}$$

which is in turn defined in terms of the *scalar elements*

$$\mathbf{a}_{p_3p_2p_1}[0] = \left[a_{p_3p_2p_1}[0,0,0] \quad a_{p_3p_2p_1}[0,0,1] \quad a_{p_3p_2p_1}[0,0,2]\right],$$

$$\mathbf{a}_{p_3p_2p_1}[1] = \left[a_{p_3p_2p_1}[0,1,0] \quad a_{p_3p_2p_1}[0,1,1] \quad a_{p_3p_2p_1}[0,1,2]\right],$$

and

$$\mathbf{a}_{p_3p_2p_1}[2] = \left[a_{p_3p_2p_1}[0,2,0] \quad a_{p_3p_2p_1}[0,2,1] \quad a_{p_3p_2p_1}[0,2,2]\right]. \tag{107}$$

Therefore, the compression factor would be $(27/32)^3$, which is a very great compression ratio, saving the LP coefficient data in lieu of the original 32^3 sinusoidal data values as shown in Figure 38.

6. SUMMARY OF RESEARCH CONTRIBUTIONS

In this last chapter, the implications of the research contributions presented in this thesis are summarized and suggestions for future work are provided. The contributions of this thesis are divided into three main sections: (1) 2-D sharpening based on predictive BWE using 2-D AR estimation fast algorithms (2-D Yule Walker, 2-D LS Lattice, and 2-D LS Covariance), (2) 2-D synthetic imagery resolution enhancement techniques using the extensions of concepts introduced for a 2-D BWE technique, and (3) new tools for future 3-D BWE techniques (3-D Yule-Walker Equations and 3-D lattice LP parameter estimation method, and fast algorithms for both).

The first research contribution we introduced was the new 2-D BWE technique [31] based on the fast 2-D LP algorithm that operated in conjunction with existing 2-D spectral estimation techniques. This new alternative super-sharpening BWE method, presented in this thesis, is capable of preserving the full dynamic range of features in complicated signals that are being analyzed.

The second research contribution was a variation of the 2-D BWE technique for application to computed synthetic radar imagery. Using radar data collected already in a transform-like data structure, novel 2-D synthetic imagery resolution enhancement techniques were developed.

The last research contribution is comprised of novel 3-D spectral analysis techniques, particularly new 3-D autoregressive spectral estimation algorithms [28,29,30]. Both the 3-D Yule-Walker equations, for the known 3-D autocorrelation sequence (ACS) case and the 3-D lattice linear prediction parameter estimation method for unknown correlation case were developed, each with a fast recursive solution operating directly in a 3-D octant-space support region. These techniques provided enhanced 3-D spectral resolution results, particularly spectral estimation using the 3-D AR power spectral density technique. These new algorithms will form the critical component of a future 3-D sharpening algorithm, which will be the subject of future work and research challenges for implementing a full 3-D BWE technique.

Furthermore, fully 3-D signal processing that has the capability to provide reduced-dimension parametric features of the multi-dimensional data for purposes of data coding/compression is highly desirable. The new 3-D algorithms presented in this thesis could be used to reduce the number of bits to save or to transmit a 3-D (which would be a new use of 3-D LP coding and compression) data set relative to the number of bits for just sending the original 3-D data without LP coding/compression in future research.

BIBLIOGRAPHY

[1] S. Lawrence Marple Jr., *Digital Spectral Analysis with Applications*, Prentice Hall, Englewood Cliffs, NJ, 1987.

[2] S. Lawrence Marple Jr., "Sharpening and Bandwidth Extrapolation Techniques for Radar Micro-Doppler Feature Extraction," *2003 Proceedings of the International Conference on Radar*, pp. 166-169, 3-5 September 2003, Australia.

[3] S. Lawrence Marple Jr., "Two-dimensional Lattice Linear Prediction Parameter Estimation Method and Fast Algorithm," *IEEE Signal Processing Letters*, vol. 7, pp. 164-168, June 2000.

[4] Thomas G. Moore, Brian W. Zuerndorfer, and Earl C. Burt, "Enhanced Imagery Using Spectral-Estimation-Based Techniques," *Lincoln Laboratory Journal*, vol. 10, number 2, pp. 171-186, 1997.

[5] Kei Suwa and Masafumi Iwamoto, "Two Dimensional Bandwidth Extrapolation Technique for Polarimetric Radar Images," *2004 SICE Annual Conference*, August 4-6 2004, Hokkaido Institute of Technology, Japan.

[6] Inder J. Gupta, Mark J. Beak, and Ali Moghaddar, "Data Extrapolation for High Resolution Radar Imaging," *IEEE Transactions on Antennas and Propagation*, vol. 42, issue 11, pp. 1540 – 1545, November 1994.

[7] K.-T. Kim, S.-W. Kim, and H.-T. Kim, "Two-dimensional ISAR imaging using full polarization and super-resolution processing techniques," *IEE Proceedings Radar, Sonar, and Navigation*, vol. 145, pp. 240-246, August 1998.

[8] Inder J. Gupta, "High-resolution radar imaging using 2-D linear prediction," *IEEE Transactions on Antennas and Propagation*, vol. 42, issue 1, pp. 31-37, January 1994.

[9] Xiaojian Xu and Ram M. Narayanan, "Enhanced Resolution in SAR/ISAR Imaging Using Iterative Sidelobe," *IEEE Transactions on Image Processing*, vol. 14, number 4, April 2005.

[10] S. Lawrence Marple Jr. and C. Marino, "High-Resolution M-Channel, Two-Dimensional Lattice Linear Prediction Algorithm," *2003 Asilomar Conference on Systems, Signals and Computers*, vol. 2, pp. 2129-2133, November 2003.

[11] Athanasios Papoulis, “A New Algorithm in Spectral Analysis and Band-Limited Extrapolation,” *IEEE Transactions on Circuits and Systems*, vol. 22, issue 9, pp. 735-742, September 1975.

[12] J. Cadzow, “An extrapolation procedure for band-limited signals,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, issue 1, pp. 4-12, February 1979.

[13] K. Abend, “Spectrum analysis and resolution enhancement by band-limited extrapolation,” *1980 IEEE International Conference on ICASSP*, vol. 5, pp. 603-606, April 1980.

[14] B. Sullivan and Bede Liu, “On the use of singular value decomposition and decimation in discrete-time band-limited signal extrapolation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 32, issue 6, pp. 1201-1212, December 1984.

[15] Hsueh-Jyh Li Farhat and N. Yuhsyen Shen, “A new iterative algorithm for extrapolation of data available in multiple restricted regions with application to radar imaging,” *IEEE Transactions on Antennas and Propagation*, vol. 35, issue 5, pp. 581-588, May 1987.

[16] Yuanxun Wang and Hao Ling, “A frequency-aspect extrapolation algorithm for ISAR image simulation based on two-dimensional ESPRIT,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, issue 4, part 1, pp. 1743-1748, July 2000.

[17] L. Sanz and T. Huang, “Discrete and continuous bandlimited signal extrapolation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, number 5, pp. 1276-1285, October 1983.

[18] Kyung-Tae Kim, Ji-Hoon Bae, and Hyo-Tae Kim, “Effect of AR Model-Based Data Extrapolation on Target Recognition Performance,” *IEEE Transactions on Antennas and Propagation*, vol. 51, issue 4, pp. 912-914, April 2003.

[19] S. Lawrence Marple Jr., "Minimum Variance Spectral Estimator Fast Algorithms Based on Covariance and Modified Covariance Methods of Linear Prediction," *2001 Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 711-714, 4-7 November 2001.

[20] Andreas Jakobsson, S. Lawrence Marple Jr., and Petre Stoica, “Computationally Efficient Two-Dimensional Capon Spectrum Analysis,” *IEEE Transactions on Signal Processing*, vol. 48, issue 9, pp. 2651-2661, September 2000.

[21] Byoung Seon Choi, “An order-recursive algorithm to solve the 3-D Yule-Walker equations of causal 3-D AR models,” *IEEE Transactions on Signal Processing*, vol. 47, issue 9, pp. 2491-2502, September 1999.

[22] S. Lawrence Marple Jr., "Fast Algorithm for the Two-dimensional Modified Covariance Method of Linear Prediction," *1998 IEEE Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems & Computers*, vol. 2, pp. 1452-1455, 1-4 November 1998.

[23] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Prentice Hall, second edition, Menlo Park, CA, 2001.

[24] Mehrdad Soumekh, *Synthetic Aperture Radar Signal Processing with MATLAB Algorithms*, Wiley-Interscience, Hoboken, NJ, April 1999.

[25] I. Erer, M. Kartal and A. H. Kayran, “2-D data extrapolation for high resolution radar imaging using autoregressive lattice modeling,” *IEE Proceedings Radar, Sonar and Navigation*, vol. 148, issue 5, pp. 277-283, October 2001.

[26] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition. Englewood Cliffs*, Prentice-Hall. NJ, 1993.

[27] S. Singhal, “High quality audio coding using multipulse LPC,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.2, pp. 1101-1104, 3-6 April 1990.

[28] Jiseok Liew and S. Lawrence Marple Jr., “Three-dimensional Fast Algorithm Solution For Octant-Based Three-dimensional Yule-Walker Equations,” *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 889-892, 18-23 March 2005.

[29] Jiseok Liew and S. Lawrence Marple Jr., “Fast Algorithm Solution for Octant-Based Three-Dimensional Yule-Walker Equations,” submitted for publication

[30] Jiseok Liew and S. Lawrence Marple Jr., “3-D Least-square based lattice linear prediction parameter estimation method,” submitted for publication.

[31] Jiseok Liew and S. Lawrence Marple Jr., “Two-Dimensional Sharpening Based on Predictive Bandwidth Extrapolation Using Two-Dimensional AR Estimation Fast Algorithm,” submitted for publication.

APPENDIX

Appendix treated herein is derived from section 5.2.1 pages on 55-58. The eight examples represent the first-, second-, third-, fourth-, fifth-, sixth-, seventh-, and eighth-octant space respectively and prediction directions of $p_1, p_2,$ and p_3 $(p_1 = 1 , p_2 = 2 \; p_3 = 1)$ are shown in Figures A-H. In anticipation of the fast computational algorithm to be presented, we shall assume that subscript $p_1 p_2 p_3$ means p_1 (point on the 'i' axis) is a variable order parameter and p_2 (point on the 'j' axis) and p_3 (point on the 'k' axis) are assumed to be fixed order parameters.

Super block matrix and *vector* representations of the first-, second-, third-, fourth-, fifth-, sixth-, seventh-, and eighth-octant space Yule-Walker equations are respectively

$$\begin{aligned}
\underline{\underline{\rho}}^{1}_{p_1p_2p_3} &= \underline{\underline{\mathsf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathsf{a}}}^{1}_{p_1p_2p_3} \; , \\
\underline{\underline{\rho}}^{2}_{p_1p_2p_3} &= \underline{\underline{\mathsf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathsf{a}}}^{2}_{p_1p_2p_3} \; , \\
\underline{\underline{\rho}}^{3}_{p_1p_2p_3} &= \underline{\underline{\mathsf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathsf{a}}}^{3}_{p_1p_2p_3} \; , \\
\underline{\underline{\rho}}^{4}_{p_1p_2p_3} &= \underline{\underline{\mathsf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathsf{a}}}^{4}_{p_1p_2p_3} \; , \\
\underline{\underline{\rho}}^{5}_{p_1p_2p_3} &= \underline{\underline{\mathsf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathsf{a}}}^{5}_{p_1p_2p_3} \; , \\
\underline{\underline{\rho}}^{6}_{p_1p_2p_3} &= \underline{\underline{\mathsf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathsf{a}}}^{6}_{p_1p_2p_3} \; , \\
\underline{\underline{\rho}}^{7}_{p_1p_2p_3} &= \underline{\underline{\mathsf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathsf{a}}}^{7}_{p_1p_2p_3} \; , \\
\underline{\underline{\rho}}^{8}_{p_1p_2p_3} &= \underline{\underline{\mathsf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathsf{a}}}^{8}_{p_1p_2p_3}
\end{aligned}$$

as described in Eqs (A-H).

The First Octant-Space

- Block vector representation of the first octant-space Yule-Walker equation $\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\underline{\underline{\mathbf{a}}}^1_{p_1p_2p_3} = \underline{\underline{\boldsymbol{\rho}}}^1_{p_1p_2p_3}$ as described in Eq (A).

$$\begin{bmatrix} 1 \\ a^1[0,0,1] \\ a^1[0,0,2] \\ a^1[0,1,0] \\ a^1[0,1,1] \\ a^1[0,1,2] \\ a^1[1,0,0] \\ a^1[1,0,1] \\ a^1[1,0,2] \\ a^1[1,1,0] \\ a^1[1,1,1] \\ a^1[1,1,2] \end{bmatrix} \bullet \begin{bmatrix} r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] & r[1,1,2] \\ r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] \\ r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] \\ r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] \\ r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] \\ r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] \\ r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] \\ r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] \\ r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] \\ r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] \\ r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] \\ r[-1,-1,-2] & r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] \end{bmatrix} = \begin{bmatrix} \rho^1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (A)$$

- This example ($p_1 = 1$, $p_2 = 2$ $p_3 = 1$) represents the first octant space and prediction direction of p_1 in the positive direction, p_2 in the positive direction, and p_3 in the positive direction as shown in Figure A.

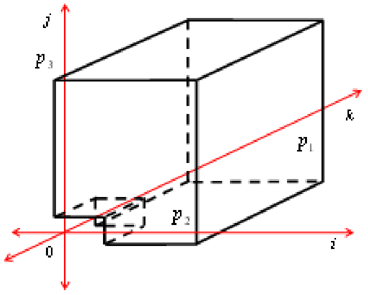


Figure A. Illustration of the prediction directions

The Second Octant-Space

- Block vector representation of the first octant-space Yule-Walker equation $\underline{\underline{\mathrm{R}}}_{p_1 p_2 p_3} \underline{\underline{\mathrm{a}}}^2_{p_1 p_2 p_3} = \underline{\underline{\rho}}^2_{p_1 p_2 p_3}$ as described in Eq (B).

$$\begin{bmatrix} a^2[0,-1,0] \\ a^2[0,-1,1] \\ a^2[0,-1,2] \\ 1 \\ a^2[0,0,1] \\ a^2[0,0,2] \\ a^2[1,-1,0] \\ a^2[1,-1,1] \\ a^2[1,-1,2] \\ a^2[1,0,0] \\ a^2[1,0,1] \\ a^2[1,0,2] \end{bmatrix} \bullet \begin{bmatrix} r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] & r[1,1,2] \\ r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] \\ r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] \\ r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] \\ r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] \\ r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] \\ r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] \\ r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] \\ r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] \\ r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] \\ r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] \\ r[-1,-1,-2] & r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \rho^2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{(B)}$$

- This example ($p_1 = 1$, $p_2 = 2$ $p_3 = 1$) represents the second octant space and prediction direction of p_1 in the negative direction, p_2 in the positive direction, and p_3 in the positive direction as shown in Figure B.

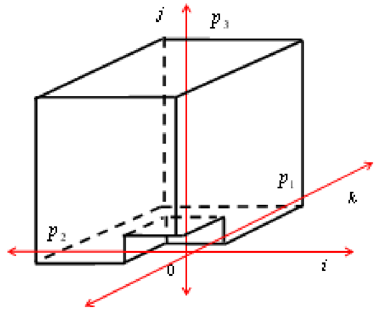


Figure B. Illustration of the prediction directions

The third Octant-Space

- Block vector representation of the first octant-space Yule-Walker equation $\underline{\underline{\mathrm{R}}}_{p_1p_2p_3}\underline{\underline{\mathbf{a}}}^3_{p_1p_2p_3} = \underline{\underline{\rho}}^3_{p_1p_2p_3}$ as described in Eq (C).

$$\begin{bmatrix} a^3[-1,-1,0] \\ a^3[-1,-1,1] \\ a^3[-1,-1,2] \\ a^3[-1,0,0] \\ a^3[-1,0,1] \\ a^3[-1,0,2] \\ a^3[0,-1,0] \\ a^3[0,-1,1] \\ a^3[0,-1,2] \\ 1 \\ a^3[0,0,1] \\ a^3[0,0,2] \end{bmatrix} \bullet \begin{bmatrix} r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] & r[1,1,2] \\ r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] \\ r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] \\ r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] \\ r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] \\ r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] \\ r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] \\ r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] \\ r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] \\ r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] \\ r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] \\ r[-1,-1,-2] & r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho^3 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(C)}$$

- This example ($p_1 = 1$, $p_2 = 2$ $p_3 = 1$) represents the second octant space and negative direction of p_1 in the negative direction, p_2 in the positive direction, and p_3 in the positive direction as shown in Figure C.

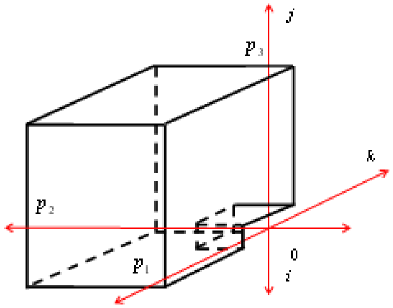


Figure C. Illustration of the prediction directions

The fourth Octant-Space

- Block vector representation of the first octant-space Yule-Walker equation $\underline{\underline{\mathrm{R}}}_{p_1p_2p_3}\underline{\underline{\mathbf{a}}}^4_{p_1p_2p_3}=\underline{\underline{\boldsymbol{\rho}}}^4_{p_1p_2p_3}$ as described in Eq (D).

$$\begin{bmatrix} a^4[-1,0,0] \\ a^4[-1,0,1] \\ a^4[-1,0,2] \\ a^4[-1,1,0] \\ a^4[-1,1,1] \\ a^4[-1,1,2] \\ 1 \\ a^4[0,0,1] \\ a^4[0,0,2] \\ a^4[0,1,0] \\ a^4[0,1,1] \\ a^4[0,1,2] \end{bmatrix} \bullet \begin{bmatrix} r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] & r[1,1,2] \\ r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] \\ r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] \\ r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] \\ r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] \\ r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] \\ r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] \\ r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] \\ r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] \\ r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] \\ r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] \\ r[-1,-1,-2] & r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho^4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(D)}$$

- This example ($p_1 = 1$, $p_2 = 2$ $p_3 = 1$) represents the second octant space and negative direction of p_1 in the positive direction, p_2 in the positive direction, and p_3 in the positive direction as shown in Figure D.

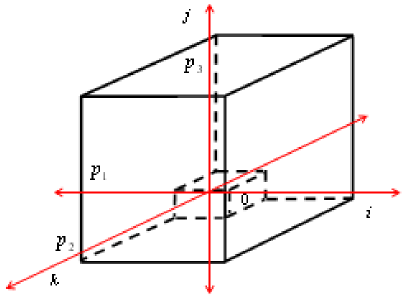


Figure D. Illustration of the prediction directions

The fifth Octant-Space

- Block vector representation of the first octant-space Yule-Walker equation $\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\,\underline{\underline{\mathbf{a}}}^{5}_{p_1p_2p_3} = \underline{\underline{\boldsymbol{\rho}}}^{5}_{p_1p_2p_3}$ as described in Eq (E).

$$\begin{bmatrix} a^5[-1,-1,-2] \\ a^5[-1,-1,-1] \\ a^5[-1,-1,0] \\ a^5[-1,0,-2] \\ a^5[-1,0,-1] \\ a^5[-1,0,0] \\ a^5[0,-1,-2] \\ a^5[0,-1,-1] \\ a^5[0,-1,0] \\ a^5[0,0,-2] \\ a^5[0,0,-1] \\ 1 \end{bmatrix} \bullet \begin{bmatrix} r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] & r[1,1,2] \\ r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] \\ r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] \\ r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] \\ r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] \\ r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] \\ r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] \\ r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] \\ r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] \\ r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] \\ r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] \\ r[-1,-1,-2] & r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho^5 \end{bmatrix} \quad \text{(E)}$$

- This example ($p_1 = 1$, $p_2 = 2$ $p_3 = 1$) represents the second octant space and negative direction of p_1 in the negative direction, p_2 in the negative direction, and p_3 in the positive direction as shown in Figure E.

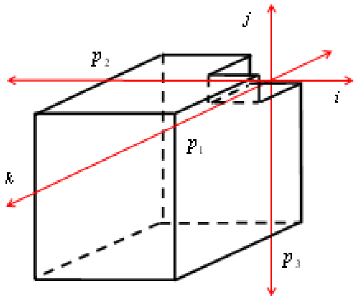


Figure E. Illustration of the prediction directions

The sixth Octant-Space

- Block vector representation of the first octant-space Yule-Walker equation $\underline{\underline{\mathrm{R}}}_{p_1p_2p_3}\underline{\underline{\mathrm{a}}}^{6}_{p_1p_2p_3} = \underline{\underline{\rho}}^{6}_{p_1p_2p_3}$ as described in Eq (F).

$$\begin{bmatrix} a^6[-1,0,-2] \\ a^6[-1,0,-1] \\ a^6[-1,0,0] \\ a^6[-1,1,-2] \\ a^6[-1,1,-1] \\ a^6[-1,1,0] \\ a^6[0,0,-2] \\ a^6[0,0,-1] \\ 1 \\ a^6[0,1,-2] \\ a^6[0,1,-1] \\ a^6[0,1,0] \end{bmatrix} \bullet \begin{bmatrix} r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] & r[1,1,2] \\ r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] \\ r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] \\ r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] \\ r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] \\ r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] \\ r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] \\ r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] \\ r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] \\ r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] \\ r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] \\ r[-1,-1,-2] & r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho^6 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{(F)}$$

- This example ($p_1 = 1$, $p_2 = 2$ $p_3 = 1$) represents the second octant space and negative direction of p_1 in the positive direction, p_2 in the negative direction, and p_3 in the negative direction as shown in Figure F.

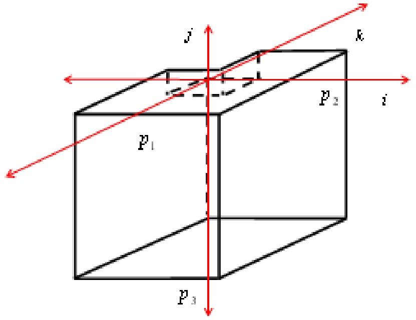


Figure F. Illustration of the prediction directions

The seventh Octant-Space

- Block vector representation of the first octant-space Yule-Walker equation $\underline{\underline{\mathbf{R}}}_{p_1p_2p_3}\underline{\underline{\mathbf{a}}}^{7}_{p_1p_2p_3} = \underline{\underline{\boldsymbol{\rho}}}^{7}_{p_1p_2p_3}$ as described in Eq (G).

$$\begin{bmatrix} a^7[0,0,-2] \\ a^7[0,0,-1] \\ 1 \\ a^7[0,1,-2] \\ a^7[0,1,-1] \\ a^7[0,1,0] \\ a^7[1,0,-2] \\ a^7[1,0,-1] \\ a^7[1,0,0] \\ a^7[1,1,-2] \\ a^7[1,1,-1] \\ a^7[1,1,0] \end{bmatrix} \bullet \begin{bmatrix} r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] & r[1,1,2] \\ r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] \\ r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] \\ r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] \\ r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] \\ r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] \\ r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] \\ r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] \\ r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] \\ r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] \\ r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] \\ r[-1,-1,-2] & r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \rho^7 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(G)}$$

- This example ($p_1 = 1$, $p_2 = 2$ $p_3 = 1$) represents the second octant space and positive direction of p_1 in the positive direction, p_2 in the negative direction, and p_3 in the negative direction as shown in Figure G.

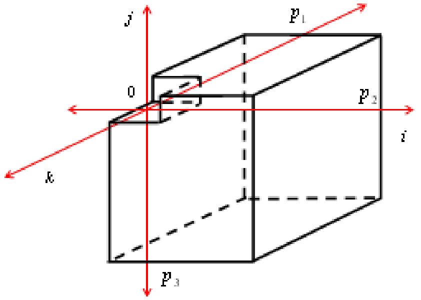


Figure G. Illustration of the prediction directions

The eighth Octant-Space

- Block vector representation of the first octant-space Yule-Walker equation $\underline{\underline{R}}_{p_1p_2p_3}\,\underline{\underline{a}}^{8}_{p_1p_2p_3} = \underline{\underline{\rho}}^{8}_{p_1p_2p_3}$ as described in Eq (H).

$$\begin{bmatrix} a^8[0,-1,-2] \\ a^8[0,-1,-1] \\ a^8[0,-1,0] \\ a^8[0,0,-2] \\ a^8[0,0,-1] \\ 1 \\ a^8[1,-1,-2] \\ a^8[1,-1,-1] \\ a^8[1,-1,0] \\ a^8[1,0,-2] \\ a^8[1,0,-1] \\ a^8[1,0,0] \end{bmatrix} \bullet \begin{bmatrix} r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] & r[1,1,2] \\ r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] & r[1,1,1] \\ r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] & r[1,1,0] \\ r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] & r[1,0,2] \\ r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] & r[1,0,1] \\ r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] & r[1,0,0] \\ r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] & r[0,1,2] \\ r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] & r[0,1,1] \\ r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] & r[0,1,0] \\ r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] & r[0,0,2] \\ r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] & r[0,0,1] \\ r[-1,-1,-2] & r[-1,-1,-1] & r[-1,-1,0] & r[-1,0,-2] & r[-1,0,-1] & r[-1,0,0] & r[0,-1,-2] & r[0,-1,-1] & r[0,-1,0] & r[0,0,-2] & r[0,0,-1] & r[0,0,0] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho^8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(H)}$$

- This example ($p_1 = 1$, $p_2 = 2$ $p_3 = 1$) represents the second octant space and positive direction of p_1 in the negative direction, p_2 in the negative direction, and p_3 in the negative direction as shown in Figure H.

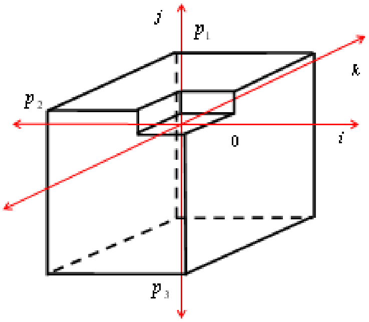


Figure H. Illustration of the prediction directions