



RECONSTRUCTION ALGORITHM CHARACTERIZATION AND
PERFORMANCE MONITORING IN LIMITED-ANGLE
CHROMOTOMOGRAPHY

THESIS

Kevin C. Gustke, Captain, USAF

AFIT/GE/ENG/04-12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright Patterson Air Force Base, Ohio

Approved for public release; distribution unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/04-12

RECONSTRUCTION ALGORITHM CHARACTERIZATION AND
PERFORMANCE MONITORING IN LIMITED-ANGLE
CHROMOTOMOGRAPHY

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Kevin C. Gustke, B.S.E.E.
Captain, USAF

March, 2004

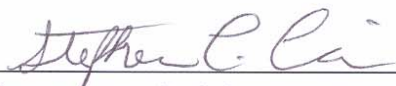
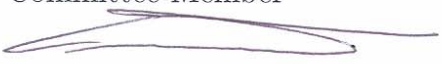
Approved for public release; distribution unlimited

RECONSTRUCTION ALGORITHM CHARACTERIZATION AND
PERFORMANCE MONITORING IN LIMITED-ANGLE
CHROMOTOMOGRAPHY

Kevin C. Gustke, B.S.E.E.

Captain, USAF

Approved:

 STEPHEN C. CAIN Ph.D. Thesis Advisor	<u>9 March 2004</u> Date
 Maj MATTHEW E. GODA Ph.D. Committee Member	<u>9 MAR 04</u> Date
 GLEN P. PERRAM Ph.D. Committee Member	<u>15 Mar 04</u> Date
 RONALD F. TUTTLE Ph.D. Committee Member	<u>11 MAR 04</u> Date

Acknowledgements

I would like to thank first and foremost my lovely wife for her patience with me and her care for my son while we have been going through the rigorous demands placed on us by AFIT. Without her support, this thesis would not be possible. I would also like to recognize the direction of my advisor, Dr. Stephen Cain, and the guidance received by committee members Maj Matthew Goda, Dr. Glen Perram, and Dr. Ronald Tuttle. Thanks to the guys in GE-04M who where always eager to extend a helpful hand when I had no clue what was going on in class; you truly are great people. Finally, I would like to thank my Lord and Savior, Jesus Christ, who is a source of comfort and strength always.

Kevin C. Gustke

Table of Contents

	Page
Acknowledgements	iii
List of Figures	vii
List of Tables	xi
Abstract	xii
 I. Introduction	 1
1.1 Background	1
1.2 Problem Definition	3
1.3 Outline	4
1.4 Scope and Limitations	4
1.5 Standards	5
1.6 Methodology	7
 II. Fundamentals of Chromotomography	 8
2.1 Characteristics of Hyperspectral Imaging	8
2.2 Instrument description	10
2.3 Basic Reconstruction: The Matrix Inverse Algorithm	10
2.4 The Missing Cone	14
2.5 Methods for Filling In the Missing Cone	15
SVD-PCA	16
SVD-POCS	17
Non-Iterative PCA	18

	Page
III. Implementation and Analysis	19
3.1 Data Preparation	19
Temperature Map	19
3.2 Implementation of Pseudo-inverse in MATLAB	25
Synthetic Data Generation	26
Pseudo-inverse Reconstruction	29
Pseudo-inverse Solution with the Wiener Inverse	31
Pseudo-inverse Solution with the Threshold Inverse	32
The Best Inversion Method	33
Impact of Signal Non-Uniformity	38
3.3 Implementation of the Iterative Improvement Algorithm	39
The SVD-POCS Algorithm	39
Performance	44
3.4 Modified SVD-POCS (MSP)	53
Negative Intensity Check	53
Cold Field Stop Update	53
Force Sum	54
Include the Mean in the ODC	54
3.5 Performance Comparison	56
Error vs Temperature	64
3.6 Monochromatic Source Test	66
IV. Iteration Monitoring	70
4.1 The Warm Field Stop	70
Error Calculation in the Warm Field Stop	70
4.2 Performance with the Warm Field Stop	72
4.3 Performance in the Warm Field Stop	77
4.4 Fireball in the Field of View	82
4.5 Absolute Radiometry in Photons	96

	Page
V. Conclusions	98
5.1 Summary	98
5.2 Lessons Learned	99
5.3 Future Work	100
Automated Model Dimension Selection	100
Warm Field Stop	100
Improving Radiometry	100
Speed Enhancement	101
Time Analysis	101
5.4 Conclusion	102
Appendix A. MATLAB Code	103
A.1 Master Routine Code	103
A.2 Get Object Code	108
A.3 Data Synthesis Code	113
A.4 Pseudo-inverse Reconstruction Code	115
A.5 SVD-POCS Code	118
Appendix B. Additional Figures	123
B.1 Warm Field Stop NVE Versus Iteration	123
Bibliography	124

List of Figures

Figure		Page
1.	Hyperspectral Data Collection Methods.	9
2.	Instrument Description.	10
3.	Simple Standard Configuration Schematic.	11
4.	Recorded Image After Prism Rotation	11
5.	Radon Transform	15
6.	Example Hyperspectral Data Cube	18
7.	Broadband Image of the Temperature Map Scene.	20
8.	Blackbody Curve for a Temperature of 295K.	21
9.	Atmospheric Transmission	22
10.	Prism Dispersion Locations.	27
11.	No Pixel Interpolation Needed	28
12.	Pixel Interpolation Needed	28
13.	Example of Interpolation	29
14.	Synthetic Data Example.	30
15.	Pseudoinverse Reconstruction Example.	31
16.	Atmospheric Transmission Curve	35
17.	Pseudoinverse Error Versus Frequency	36
18.	NVE for the Pseudoinverse.	37
19.	NMRE of the Pseudoinverse	38
20.	Normalized Mean Removed Error of the Pseudoinverse	39
21.	Singular Values for the Temperature Map Scene.	41
22.	Flowchart for SVD-POCS.	44
23.	Mean Photon Count of Original and SVD-POCS	45
24.	Reconstruction Down One Spatial Location of the Cube . . .	45
25.	NVE and NRMSE Trends Versus Dimension	47

Figure		Page
26.	NMRE Trends Versus Dimension	47
27.	NRMSE Versus Iteration for SVD-POCS.	48
28.	NMRE Versus Iteration for SVD-POCS	49
29.	SVD-POCS Reconstruction Example	50
30.	Atmospheric Absorption Curve	51
31.	NMRE for SVD-POCS	51
32.	NVE for SVD-POCS.	52
33.	Flowchart for Modified SVD-POCS (MSP)	55
34.	NRMSE of SVD-POCS and MSP Versus Dimension.	56
35.	NMRE for SVD-POCS and MSP	57
36.	Photon Count Versus Wavelength for MSP	58
37.	Atmospheric Absorption Curve	59
38.	NMRE Comparison of Iterative Techniques	59
39.	NVE Comparison of Iterative Techniques	60
40.	NRMSE Versus Iteration for SVD-POCS and MSP.	61
41.	NMRE Versus Iteration for SVD-POCS and MSP	62
42.	NRMSE Versus Frequency for SVD-POCS and MSP.	63
43.	NRMSE Versus Temperature for 290.5-300.7K.	65
44.	NRMSE Versus Temperature for 291-338K.	65
45.	Broadband View of Five Bars.	66
46.	NVE for the Five Frequency Bars.	67
47.	Monochromatic Source Test Reconstructed Bands	67
48.	Spectral Lineshape for 25 Bands	69
49.	Spectral Lineshape with 43 Bands	69
50.	Imager Schematic Showing the Warm Field Stop	71
51.	Warm Field Stop Example Images	71
52.	Atmospheric Absorption Curve	73

Figure		Page
53.	NRMSE with the Warm Field Stop	74
54.	NMRE of Each Band for All Reconstruction Types	75
55.	NVE with the Warm Field Stop.	76
56.	NRMSE of the Warm Field Stop of 300K	77
57.	NMRE for the Warm Field Stop of 300K	78
58.	NRMSE for a Warm Field Stop of 500K	79
59.	NRMSE for a Warm Field Stop of 100K	80
60.	NRMSE of Iterative Methods with a Fireball	83
61.	NMRE Versus Wavelength for the Fireball Scene	84
62.	NVE Versus Wavelength for the Fireball Scene	85
63.	NRMSE Versus Iteration for the Fireball Scene	86
64.	NMRE Versus Iteration for the 1000k 67 Pixel Diameter Fireball	87
65.	Atmospheric Absorption Curve	88
66.	Mean Photon Count for MSP with a Fireball Diameter of 6 Pixels.	88
67.	Mean Photon Count for MSP with Fireball Diameter of 32 Pixels.	89
68.	Mean Photon Count for MSP with a Fireball Diameter of 67 Pixels.	89
69.	NRMSE for MSP with Fireball Diameter of 6 Pixels.	90
70.	NRMSE for MSP with Fireball Diameter of 32 Pixels.	91
71.	NRMSE for MSP with Fireball Diameter of 67 Pixels.	91
72.	NRMSE Versus Iteration for MSP with Fireball Diameter of 6 Pixels.	92
73.	NRMSE Versus Iteration for MSP with Fireball Diameter of 32 Pixels.	93
74.	NRMSE Versus Iteration for MSP with Fireball Diameter of 67 Pixels	93
75.	NMRE for MSP with Fireball Diameter of 6 Pixels.	94

Figure		Page
76.	NMRE for MSP with Fireball Diameter of 32 Pixels.	94
77.	NMRE for MSP with Fireball Diameter of 67 Pixels.	95
78.	Mean Photon Count Versus Frequency	96
79.	Mean Number of Photons for the 1000K Fireball Scene	97
80.	NVE for the Warm Field Stop of 300K	123

List of Tables

Table		Page
1.	Wavelength Bins and Frequencies	21
2.	NRMSE Results for the Wiener Method	32
3.	NRMSE Results for the Threshold Method	33
4.	Performance of Pseudoinverse Methods.	34
5.	Performance of Iterative Methods.	57
6.	Location in Frequency of Each Bar.	67
7.	Performance of Iterative Methods Including the Warm Field Stop	72
8.	Performance of Iterative Methods With a Fireball	82

Abstract

Hyperspectral data collection and analysis is an increasing priority with the growing need to obtain greater classification precision than offered by traditional spatial imagery. In this thesis, trends in hyperspectral chromotomographic reconstruction are explored where reconstruction is performed using a series of spatial-chromatic images. Chromotomography involves capturing a series of two-dimensional images where each image is created by placing a prism in front of the focal plane array; causing spectral dispersion corresponding to a series of prism angles over a single rotation.

Before testing reconstruction, synthetic data is produced, approximating what would be produced from prism dispersion on the focal plane array. The pseudo-inverse singular matrix problem is addressed where two methods are compared to find which produces minimal error.

The standard iterative error reduction algorithm, SVD-POCS, is shown to be incapable of reconstructing the mean of the source scene, making absolute radiometry analysis impractical. However, SVD-POCS is shown to provide the least error if the goal is to perform relative radiometry analysis. Additional constraints are needed to make absolute radiometry analysis possible. The added constraints of non-negativity, spatial extent of the cold field stop, forcing the sum, and keeping the mean for each iteration improves absolute radiometric performance.

These additional constraints also allow use of a warm field stop to monitor reconstruction error for both the pseudo-inverse and iterative improvement algorithm. Error can be calculated each iteration to ascertain when a minimum has been reached in a mean square error sense. Thus, minimum mean square error of the reconstruction can be obtained with confidence.

RECONSTRUCTION ALGORITHM CHARACTERIZATION AND PERFORMANCE MONITORING IN LIMITED-ANGLE CHROMOTOMOGRAPHY

I. Introduction

1.1 Background

Initial work on chromotomography was led by Jonathan M. Mooney, formerly of AFRL/SNHI [1], in the mid to late nineties with a goal of pursuing alternatives to conventional hyperspectral imagers. It did not take Mooney long to discover a problem with his technique [2]. The system transfer matrix of the problem is singular, and therefore, reconstruction cannot be described by a unique solution. Additionally, the initial reconstruction result had large artifacts resulting in significant reconstruction error. Thus, techniques are needed in order to obtain a better reconstruction which minimizes error. This thesis will explore current reconstruction methods and adapt the iterative reconstruction improvement procedure to enable better radiometric analysis.

Chromotomography is tomography applied to hyperspectral imaging. Tomography is best known for its medical application. A series of images is recorded where each image contains a different slice of information about the same scene. For example, the medical field uses computerized axial tomography scanning (CAT scan) to obtain three dimensional images of human body parts. The imaging device captures a series of two dimensional x-ray images while rotating around the body part of interest. Computers then process the images, creating a three dimensional image of the body part in a manner which maximizes the signal-to-noise ratio.

Reconstruction in chromotomography has not been explored in the same detail as it has in medical tomography. Current reconstruction methods have various limitations. These limitations must be overcome to provide the benefits of the technology to the intelligence community. Chromotomography has improved spatial coverage and complete spectral coverage, versus conventional hyperspectral imaging techniques that are limited in either spatial or spectral coverage. Conventional hyperspectral imagers fall into three categories; those which image through a slit, view the scene through one spectral filter at a time, and those which use interference to image spectrally. Hyperspectral imagers using a slit have a limited field of view and low throughput. Hyperspectral imagers using a series of spectral filters have high optical throughput, but are poor at measuring a rapidly changing spectral event or providing good spectral resolution. Imagers using wavefront interference, such as the Michelson interferometer, can view the entire scene and spectra, but at the cost of complex instrument construction, alignment maintenance difficulty, severe vibration sensitivity, and 50% optical throughput for the standard configuration.

Hyperspectral data is used for a variety of defense related as well as commercial applications. The utility of hyperspectral images for determining the chemical content of exhaust plumes helps to identify processes involved with creation of the exhaust. Hyperspectral data is also used to find targets buried in camouflage by taking advantage of the often unique spectral signatures of man made materials as opposed to those found occurring in nature. Environmental monitoring of the atmosphere is also accomplished using hyper-spectral data in order to identify temperature profiles and aid in weather forecasting. The general utility of this type of data makes research into basic methods for collecting and processing it of potentially significant use to a wide variety of customers.

1.2 Problem Definition

The first goal of this thesis is to detail the state of the art reconstruction method and make necessary additions to investigate the performance of absolute radiometry versus relative radiometry. The second goal of this thesis is to aid AFIT in development of a chromotomographic imager by characterizing the trade-offs in the design with reconstruction complexities and implementing a reconstruction tool in MATLAB.

The objective of reconstruction is to recover a complete spatial-chromatic 3D hyperspectral scene from a series of images where chromatic information has been obtained in each two-dimensional image (or data set). The images themselves do not contain all the information necessary to recover the 3D hyperspectral scene directly, requiring an additional routine to estimate missing information. The computer based reconstruction algorithm is key to the effectiveness of the chromotomographic hyperspectral imager. The efficiency and performance of the algorithm determines how useful the technology will be to end users.

Previous research by Mooney [3], Brodzik [4], and An [5] have left several issues open with the current reconstruction algorithm. They include: What is the best inversion approximation technique? How many principal eigenspectra are needed to form the best missing cone estimate? How can convergence and scene improvement be verified in the iterative procedures? Is absolute radiometry possible?

In order to better understand the problem, a few definitions must be covered. Hyperspectral data sets consist of three dimensions: two spatial dimensions and wavelength. Data is organized in a cube consisting of the three dimensions. Each two dimensional image of the scene corresponds to the scene intensity at one particular wavelength. Time can be considered a fourth dimension, representing the data as a “movie.” The discrete Fourier transform is a mathematical operation used to show intensity relationships in the frequency domain. The discrete Fourier transform (DFT) of an $N \times M$ image is also $N \times M$ where intensities correspond to spatial

frequencies in the x and y direction. The discrete convolution of two functions f and g is a sum which expresses the amount of overlap of one of the functions with the other as it is shifted over time or space [6].

1.3 Outline

Chapter 2 provides the necessary background on hyperspectral imaging and chromotomography. Included is a description of how data recorded by the sensor is related to the imaged scene. The mathematics of the reconstruction are explained followed by a discussion of problems that occur and why additional methods are needed to reduce reconstruction error. Chapter 3 describes how data is generated for use in algorithm modeling, how the pseudo-inverse is performed, how the implementation of iterative methods is done, and investigates the effects of changing the model dimension in the SVD-POCS and modified method. Chapter 4 introduces a new technique used to monitor reconstruction error in the iterative process. Chapter 5 provides a summary of results and a conclusion while presenting areas that need to be further explored in the future.

1.4 Scope and Limitations

Research will focus on work performed at AFRL/SNHI at Hanscom, AFB by Jonathan M. Mooney, Brodzik, et al. The state of the art reconstruction method includes the pseudo-inverse solution followed by a series of iterations using a set of constraints. The primary limitation results from the inherent nature of working with a singular system transfer matrix. The solution cannot be uniquely recovered, thus a set of infinite solutions exist. Additionally, data used for reconstruction is completely synthetic since the instrument itself has yet to be built at AFIT. All modeling and processing in this research is done on a computer, which means all data is discrete. The aim of this thesis is to identify trends in chromotomographic reconstruction that are dependant on constraints and not to infer specific performance bounds. The work

presented in this paper holds when the continuous spectrum has been adequately sampled. Finally, the reconstruction for time dynamic events may be poor. If the object being imaged changes rapidly during a single prism rotation, application of the inverse system transfer matrix may cause unacceptable reconstruction error, since expected dependencies in the data will not exist. The affect on reconstruction will be an increase in the overall noise level. Time sensitive aspects are not analyzed in this thesis

1.5 Standards

The quality of image reconstruction will be determined by comparing the reconstructed images to known chromatic images of the scene. Known images are used to create synthetic data which is in turn used for reconstruction. A problem with error computation when comparing images is results are often subjective based on a human users opinion. There is no metric which can tell us "which image looks the best." But, if the goal is to perform absolute radiometry, then a mean square error metric will give an idea if reconstruction is good or not. The metric of choice for quantitative error performance is Normalized Root Mean Square Error (NRMSE). It is found by first obtaining the mean squared error (MSE) as:

$$MSE_k = \frac{\sum \sum [o_k(m, n) - c_k(m, n)]^2}{M * N}, \quad (1)$$

where $o_k(m, n)$ is the known image and $c_k(m, n)$ is the reconstructed image for chromatic band k. The region of each image corresponding to the cold field stop is not used when calculating MSE. The MSE measurement only looks at the quantitative difference between each pixel which could be quite large depending on the units of the data being evaluated. To account for this, the RMSE is divided by the mean number of photons per pixel for that image. The NRMSE is:

$$NRMSE_k = \frac{\sqrt{MSE_k} * M' * N' * K}{\sum \sum \sum [o(m, n, \lambda)]} * 100 \%, \quad (2)$$

where the non-cold field stop area of the scene is dimension $M' \times N'$ for all K bands. The normalization factor is needed to show error performance in a more meaningful way. Thus, NRMSE is a percentage of the root mean square pixel error divided by the non-cold field stop mean value of the entire scene. In practice, the NRMSE cannot be calculated because knowledge of the true hyperspectral scene content does not exist. The NRMSE metric is by no means perfect, and may lead to misleading results since the metric relies on a quantitative error measurement.

An alternative error metric, referred to as Normalized Mean Removed Error (NMRE) in this thesis, does not include the means of the scene and reconstructed data set in error calculation. Comparing the NMRE results with NRMSE provides insight on how the mean effects error. It is calculated in the same manner as NRMSE, but with the mean removed in the image difference calculation. It is found by obtaining the mean removed square error:

$$MRSE_k = \frac{\sum \sum [(o_k(m, n) - \sum \sum \frac{o_k(m, n)}{M * N}) - (c_k(m, n) - \sum \sum \frac{c_k(m, n)}{M * N})]^2}{M * N}, \quad (3)$$

where $\sum \sum \frac{o_k(m, n)}{M * N}$ and $\sum \sum \frac{c_k(m, n)}{M * N}$ are the means of the source object scene and reconstructed data respectively. The square root of MRSE is divided by the non-field stop mean of the source to give NMRE:

$$NMRE_k = \frac{\sqrt{MRSE_k} * M' * N' * K}{\sum \sum \sum [o(m, n, \lambda)]} * 100 \%. \quad (4)$$

This is similar to Normalized Mean Square Error (NMSE) from Lim [6], which uses variance differences between images, rather than intensity in an effort to remove bias effects in image comparison. According to Lim, a human will typically judge the reconstructed image with the smaller NVE as closer to the original. The NVE is obtained by:

$$NVE_k = \frac{Var[o_k(m, n) - c_k(m, n)]}{Var[o_k(m, n)]} * 100 \%, \quad (5)$$

where $Var[\cdot]$ is the variance. The NVE of the reconstruction is potentially infinite if the object scene has zero variance between pixels. NVE results are similar to NMRE but are normalized differently, which leads to a result that is hard to compare with NRMSE.

The NVE error metric will be used only to gain a idea of which images visually look the closest to the original. Both NMRE and NVE do not include the mean, but NMRE is on the same normalization scale as NRMSE. Thus the error introduced by the mean is the difference between NRMSE and NMRE. The scale of NVE is harder to comprehend, since each variance error is divided by a mean cube variance.

1.6 Methodology

In order to solve the reconstruction problem, fundamentals of how data is recorded by the imager must be understood. Thus, the first step is to take a look at the math behind the problem. The imager mathematical model is then used to create synthesized data. A known hyperspectral data cube is input to the model to create synthesized data. Having a known cube allows for an error calculation between known data and reconstructed data.

The pseudo-inverse solution produces the initial reconstruction. Trade-offs associated with the pseudo-inverse reconstruction are explored in an effort to find the pseudo-inverse solution with the least error. The next step is implementation of an iterative improvement algorithm followed by a discussion about optimal variable selection for the most improved solution. Additional constraints are considered to improve absolute radiometric performance. Finally, the effectiveness of using a warm field stop to monitor reconstruction performance is tested by comparing error in the warm field stop versus error over the scene.

II. Fundamentals of Chromotomography

2.1 Characteristics of Hyperspectral Imaging

Hyperspectral images consist of data spanning three dimensions: two spatial dimensions and wavelength. Data is organized in a cube consisting of the three dimensions. Time can be considered a fourth dimension, representing the data as a “movie.” Conventional hyperspectral imagers capture information in one of three ways. In the scanned slit method (shown in Figure 1a), light is dispersed onto the focal plane array after passing through a narrow slit. The slit direction is one spatial dimension, while the direction normal to the slit is the wavelength dimension. Sensor motion moves the slit imaging scene normal to the slit, creating the second spatial dimension component. The scanned slit method’s biggest flaw is limited viewing of the scene through a narrow slit at any given time. Any events occurring out of the field of view are not detected. Another problem is the limited amount of light the sensor receives through the slit which results in a reduced signal-to-noise ratio. Increasing exposure time allows for some compensation, but the penalty is even poorer spatial detection ability. Additionally, diffraction effects associated with the slit produce undesired spectral mixing. In the filter method (shown in Figure 1b), the scene is recorded in both spatial dimensions while a series of spectral filters are applied, creating the wavelength dimension. The filter method has excellent spatial coverage, but is very poor at measuring rapidly changing spectral events, has poor spectral resolution due to filter bandwidth limitations, and/or has limited spectral coverage. (See [1] for an expanded description.) The wave interference method, used in Fourier transform spectrometers (FTS) such as the Michelson interferometer, uses beamsplitters and mirrors to create an interference pattern on the focal plane. One of the mirrors in the FTS moves, sweeping out an interferogram which is used to find the real spectra of the scene. The FTS has spatial coverage of the filter method with spectral coverage limited by the sweep distance of the moving mirror. The

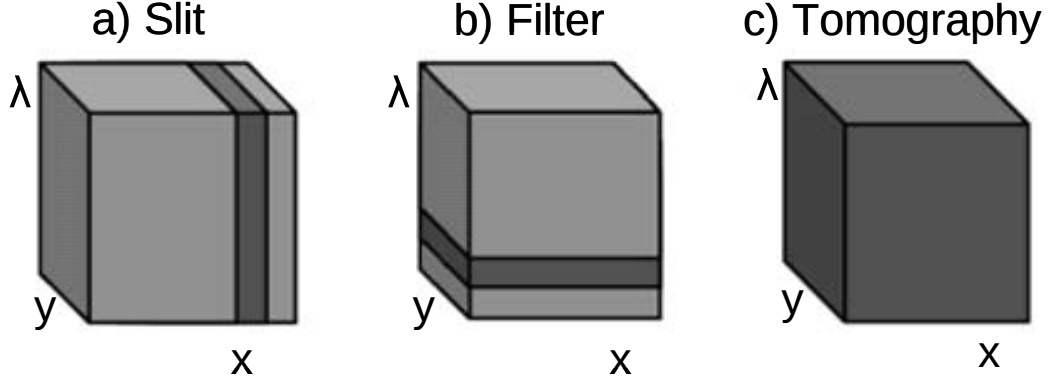


Figure 1 Hyperspectral information is typically obtained by: a) imaging through a slit and scanning, b) imaging the entire scene and filtering, or c) tomographic processing. [1]

FTS is very difficult to build and maintain compared to other methods; and is much more sensitive to vibration and noise amplification due to the multiplexing nature of imager.

In Mooney’s chromotomography design [2], a direct vision prism is placed in front of the imaging sensor. The prism disperses the intensity associated with each wavelength of the scene along a different angle while allowing a single wavelength to pass straight through. The prism then rotates a known angle and a new image is recorded. The process is repeated until the prism completes one revolution. The series of images is used by the reconstruction algorithm to obtain a 3D hyperspectral data cube approximation. The reconstructed 3D scene provides an estimate of image intensity as a function of position and wavelength. Mooney’s initial reconstruction algorithm [2] used a pseudo-inverse of a singular system transfer function matrix.

The chromotomographic sensor’s main advantages are high optical throughput, excellent spatial coverage, and continual spectral coverage as illustrated in Figure 1c). The main drawbacks are added computational complexity, non-unique reconstruction, and potential noise amplification. These problems are addressed in the literature by Brodzik [4], Mooney [3], An [5], and Chapter 3 of this thesis.

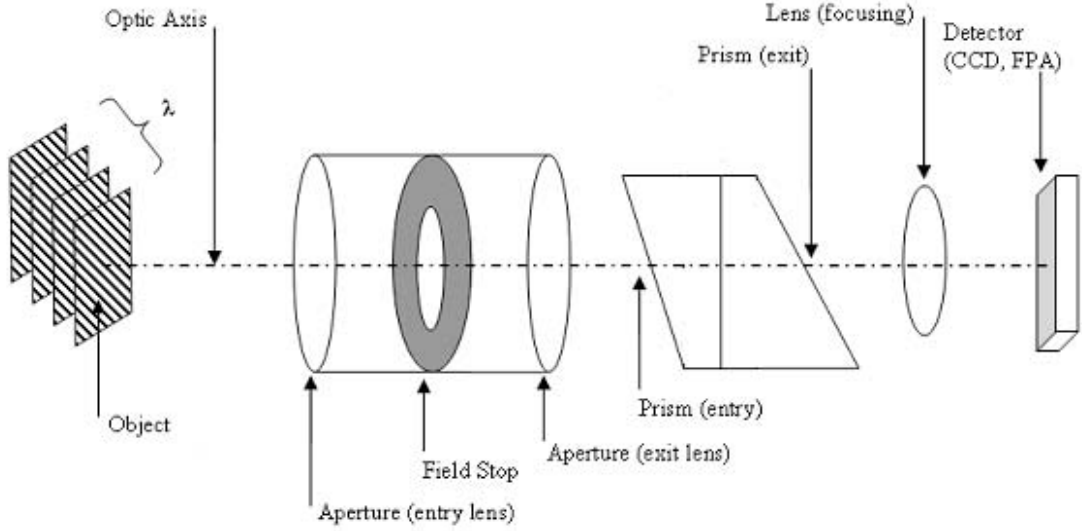


Figure 2 A photon's path from the object to the CCD will take it through two aperture lenses, a field stop, a direct view prism, and finally a focusing lens.

2.2 Instrument description

Mooney's chromotomographic imager design [2] consists of three focusing lenses, a field stop, a direct view prism, and a focal plane array (coupled charged device or CCD). A simple system diagram is shown in Figure 2. The series of spatial tomographic projection images are captured at the FPA. A computer takes the images and performs the reconstruction.

2.3 Basic Reconstruction: The Matrix Inverse Algorithm

To gain a better understanding of the reconstruction algorithm, it is useful to understand how data recorded by the camera is formatted. Mooney used an excellent illustration in [7] to describe how data is seen by the computer. Each image on the focal plane contains spectral information which has been linearly dispersed at an angle corresponding to a unique prism orientation for a single revolution. A generalized three wavelength system is shown in Figure 3. The object, which for this example only consists of three colors, is split into three spectral components by the

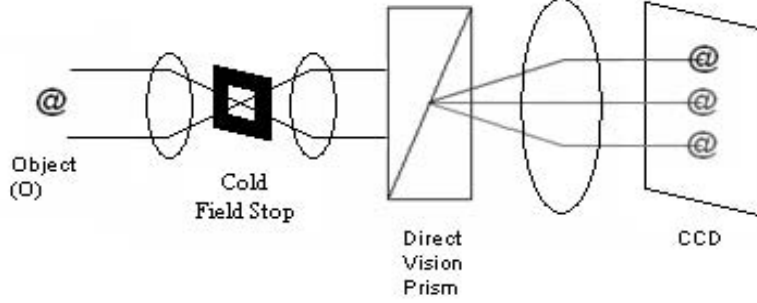


Figure 3 The object is chromatically split by the direct vision prism and dispersed on the CCD.

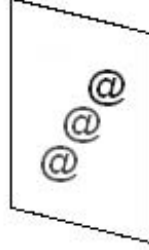


Figure 4 Subsequent images are taken at different prism rotation angles causing spectral dispersion to fall along a different line.

prism and dispersed on the CCD. The next image is taken after some known prism rotation, as shown in Figure 4. Each spectral sub image traces a circular path with a radius dependent on prism dispersion.

Each image recorded by the sensor consists of all spectral sub images convolved with a unique point spread function based on prism dispersion and rotation angle. For three spectral bands, the data recorded by the computer is:

$$d_1(m, n) = w_{11}(x, y) * o_1(x, y) + w_{12}(x, y) * o_2(x, y) + w_{13}(x, y) * o_3(x, y), \quad (6)$$

$$d_2(m, n) = w_{21}(x, y) * o_1(x, y) + w_{22}(x, y) * o_2(x, y) + w_{23}(x, y) * o_3(x, y), \quad (7)$$

$$d_3(m, n) = w_{31}(x, y) * o_1(x, y) + w_{32}(x, y) * o_2(x, y) + w_{33}(x, y) * o_3(x, y), \quad (8)$$

where $d_i(m, n)$ is the recorded image for prism rotation i , $w_{ik}(x, y)$ is the point spread function for prism rotation i at wavelength k , and $o_k(x, y)$ is the chromatic image at wavelength k . In this example, data is recorded at three different prism rotation angles.

Mooney shows in [7] that by taking the Discrete Fourier Transform (DFT) of both sides of Equations 6-8, the recorded data DFT, D , can be shown as a linear transformation of the DFT of the chromatic images by the convolution theorem. The convolution theorem states a convolution in the spatial domain is equal to multiplication in the Fourier domain. This is expressed as:

$$D_{u,v} = W_{u,v} O_{u,v}, \quad (9)$$

where $D_{u,v}$ is a three by one vector, $W_{u,v}$ is a three by three matrix, and $O_{u,v}$ is a three by one vector for the three band example. This relationship holds for each spatial frequency (u, v) meaning there exists $u * v$ (or $m * n$) different relationships represented by Equation 9 for a CCD of size (m, n) . Solving for $O_{u,v}$, we get:

$$O_{u,v} = W_{u,v}^{-1} D_{u,v}. \quad (10)$$

If $W_{u,v}$ is invertible, the final step would be to find the inverse 2D DFT of $O_k(u, v) \forall k$. The problem is $W_{u,v}$ is almost always singular. Mooney uses singular value decomposition (SVD) to find a pseudo-inverse of $W_{u,v}$. From Strang [8], the SVD of matrix W is defined as:

$$W = U \Sigma V^H, \quad (11)$$

where U and V are $M \times N$ and $N \times N$ orthogonal matrices such that

$$U^H U = U U^H = V^H V = V V^H = I, \quad (12)$$

where I is the $N \times N$ identity matrix. Σ is an $N \times N$ diagonal matrix of eigenvalues. The pseudo-inverse of $W_{u,v}$ is:

$$W_{u,v}^+ = V_{u,v} \Sigma_{u,v}^{-1} U_{u,v}^H, \quad (13)$$

and the pseudo-inverse solution is:

$$O_{u,v}^+ = V_{u,v} \Sigma_{u,v}^{-1} U_{u,v}^H D_{u,v}, \quad (14)$$

When additive noise on the CCD is considered, the recorded data is now:

$$\acute{D}_{u,v} = W_{u,v} O_{u,v} + N_{u,v}, \quad (15)$$

where $N_{u,v}$ is the spatial frequency vector resulting from the 2D DFT of the additive noise matrix. The pseudo-inverse is now:

$$\acute{O}_{u,v}^+ = V_{u,v} \Sigma_{u,v}^{-1} U_{u,v}^H \acute{D}_{u,v}, \quad (16)$$

or

$$\acute{O}_{u,v}^+ = V_{u,v} \Sigma_{u,v}^{-1} U_{u,v}^H (U_{u,v} \Sigma_{u,v} V_{u,v}^H O_{u,v} + N_{u,v}). \quad (17)$$

The diagonal values of $\Sigma_{u,v}$ are potentially zero if $W_{u,v}$ is singular, which will cause the elements of $\Sigma_{u,v}^{-1}$ to become very large and allow filtered noise to dominate the restoration. To account for this, two different inversion methods for $\Sigma_{u,v}$ can be applied, the Threshold inverse and the Wiener inverse. For the threshold inverse, values of $\Sigma_{u,v}$ less than some ε have the corresponding $\Sigma_{u,v}^{-1}$ set to zero:

$$\acute{\Sigma}_{(k,k)u,v}^{-1} = \begin{cases} \Sigma_{(k,k)u,v}^{-1}, & \Sigma_{(k,k)u,v} > \varepsilon \\ 0, & otherwise \end{cases}. \quad (18)$$

The best value of ε is SNR dependent and specific choices are covered in Section 3.2. Alternatively, the value of $\Sigma_{u,v}^{-1}$ can be set as the Wiener inverse of:

$$\dot{\Sigma}_{(k,k)u,v}^{-1} = \frac{\Sigma_{(k,k)u,v}}{(\Sigma_{(k,k)u,v}^2 + \varepsilon^2)}, \quad (19)$$

where ε is some value which balances the loss of spectral resolution and noise amplification. Mooney used the Wiener inverse with an ε of 1.5 in [2].

The pseudo-inverse solution can also be written as:

$$\dot{O}_{u,v}^+ = V_{u,v}(\dot{\Sigma}_{u,v}^{-1}\Sigma_{u,v}V_{u,v}^H O_{u,v} + \dot{\Sigma}_{u,v}^{-1}U_{u,v}^H N_{u,v}), \quad (20)$$

where it is more readily apparent that a large value of $\Sigma_{u,v}^{-1}$ will amplify the noise value $N_{u,v}$ and allow noise to dominate the reconstruction. Mooney states that even with the altered $\Sigma_{u,v}^{-1}$, the reconstruction contains severe artifacts and is particularly bad for images with low spatial or high chromatic frequencies. This result is attributed to the null space of the transfer matrix W and is a reflection of the missing cone problem.

2.4 The Missing Cone

In [3], Mooney et al describe the origins of the “cone of missing information.” It is said the cone is best understood in the context of the central slice theorem combined with the Radon transform. According to the central slice theorem, the 2D Fourier transform of a projection is equal to a plane through the origin of the 3D Fourier transform of the entire spectral object. The prism rotates around the optical axis which is normal to the focal plane. The angle between the projection plane and the optical axis, θ , remains the same for all projections. As the number of projections approach infinity, a cone will be traced out where the half angle of the cone vertex is θ . The missing cone is this area inside, where projections do not provide any information about the object.

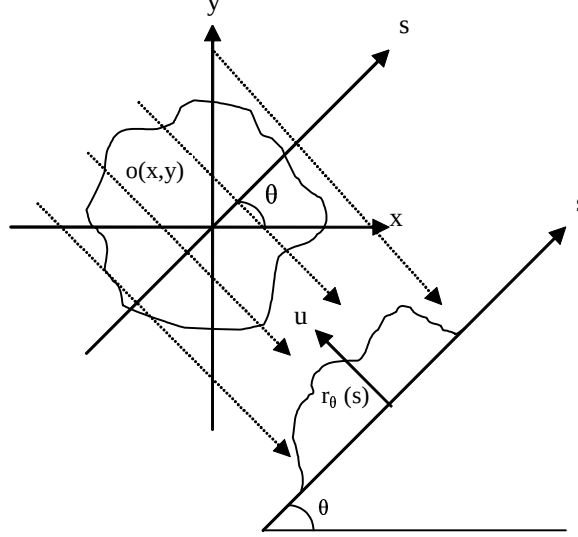


Figure 5 The Radon transform is the integration of the scene along parallel rays at an angle θ with the y-axis.

To understand the Radon transform, consider a 2D scene projected onto a 1D line [6]. The Radon transform produces the projection of the 2D scene along parallel rays onto a projection line. The transform is defined as:

$$r_{\theta}(s) = \int o(s \cos(\theta) - u \sin(\theta), s \sin(\theta) + u \sin(\theta)) du, \quad (21)$$

where $s = x \cos(\theta) + y \sin(\theta)$, $u = x \sin(\theta) - y \cos(\theta)$, θ is the projection angle, $|s|$ is the distance from the origin, and $o(x, y)$ is the 2D scene. Figure 5 graphically shows the Radon transform from two to one dimension. Imagine an additional axis normal to x and y. The projection is now 3D to 2D where R_{θ} is now the projection plane. Rotation of the prism has the affect of rotating this projection plane around the y axis.

2.5 Methods for Filling In the Missing Cone

There are three methods in current literature for filling the missing cone information covered here in chronological order of development. The first approach,

developed by Mooney, Brodzik, and An [3], uses principal component analysis in an iterative technique that relies heavily on spectral imagery redundancy. The second, developed by Brodzik and Mooney [9], uses a generalized technique based on a hybrid of a direct pseudo-inverse method and the iterative method of projections onto convex sets. Finally, the third approach, developed by An [5], is non-iterative thereby improving computational efficiency at the cost of reconstruction accuracy.

SVD-PCA. The first approach is to approximate the missing information by using redundancy inherent to hyperspectral imagery. Data inside the missing cone is forced to values consistent with data known to be outside the missing information by projecting the all eigenspectra onto the eigenspectra corresponding to the largest singular values of the previous reconstruction and then iterate. This technique is known as principal component analysis (PCA). The assumption in PCA is that the L largest eigenvectors and eigenvalues, corresponding to dimensionality L , contain the actual "signal" while the rest of the eigenvalues and eigenvectors generally contain noise. According to [10], PCA begins by first finding the K -dimensional mean vector μ and the $K \times K$ covariance matrix R_{FF} for the complete data set. Compute the eigenvectors and eigenvalues by Equation 11 and order them in descending order. Form the L largest eigenvectors into the $K \times L$ matrix A . The data can then be represented by the L largest principal components by finding:

$$Y' = A^T(Y - \mu), \quad (22)$$

where Y is the original scene and Y' is the reduced dimension scene.

Unsatisfactory solutions to the reconstruction are thereby eliminated by requiring the reconstruction meet the constraint based on the assumption incorrect data values in the missing cone generate spectra outside a subset of eigenspectra from the true spectral image. The number of eigenspectra required to describe the spectral

reconstruction should be minimized. The technique was reported to reduce artifacts by Mooney in [3] and is used as a building block for the next method, SVD-POCS.

SVD-POCS. In the second approach, the computation of a the pseudo-inverse solution is followed by the enforcement of a series of projections which enforce a set of constraints in an effort to reduce the reconstruction error. The algorithm’s convergence speed is accelerated and reconstruction performance is improved if the chosen constraint sets have a convex nature. The constraints need to add information about the scene not shared by the initial estimate, they must be computationally efficient, and should physically describe the scene to a high degree of accuracy. In [9], Brodzik listed several options for constraints existing in both the spatial and transform domains. He chose his transform domain constraint to be the “correct” spectral information known from the original reconstruction which corresponds to non-zero singular values. The spatial domain constraint, which is the fundamental missing data estimator, comes from spatial redundancy between the adjacent chromatic images. The constraint is enforced as done in SVD-PCA. Hyperspectral images are known to have a significant amount of correlation between adjacent spectral images as seen in Figure 6. This redundancy can be estimated by finding the SVD of a two dimensional data set where each row corresponds to one spectral band and each column corresponds to the same pixel location of each chromatic image. A detailed description of POCS by Combettes can be found in [11].

Figure 22 shows a flow chart for the SVD-POCS method as described in [9]. After performing the original reconstruction, the transform data of the pseudo-inverse reconstruction, O^+ , is reorganized and then subjected to two projections based on an object domain constraint and a transform domain constraint. See Section 3.3 for a detailed explanation of the algorithm. Brodzik showed that SVD-POCS does improve upon the pseudo-inverse by removing artifacts associated with the missing cone. Most of the gain in the performance was accomplished within the first 10 iter-

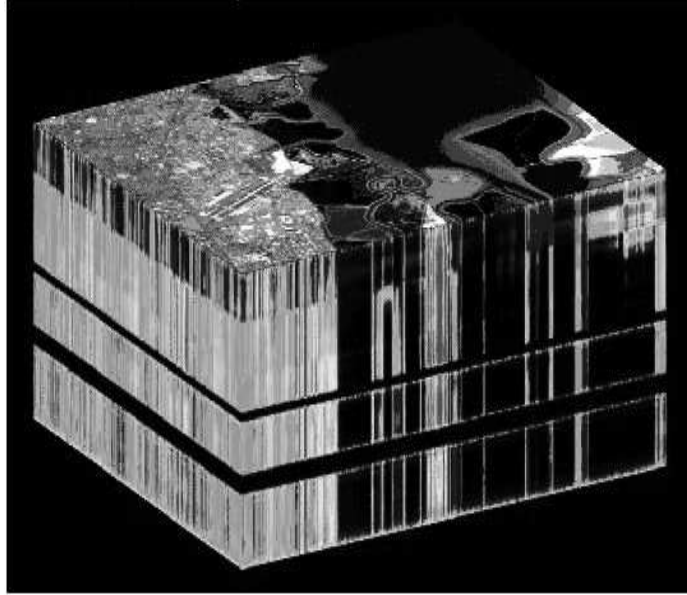


Figure 6 Shown is an example hyperspectral data cube taken using the AVIRIS instrument. Notice the strong correlation between the spectral bands. A broadband image is at the top of the cube. Source: <http://aviris.jpl.nasa.gov>

ations but the algorithm has no way to check error performance while iterating since the entire scene is unknown. Performance of SVD-POCS is analyzed in Section 3.3.

Non-Iterative PCA. In [5], An desired to develop an algorithm that is better suited to real-time image processing needs by improving computational efficiency and constraining the algorithm to a single step. An used a combination of principal components of the eigenspectra of the pseudo-inverse reconstruction combined with either a masked threshold inverse where the value of $\Sigma_{(k,k)u,v}^{-1}$ was replaced with a zero depending on a chosen threshold (1, 2 or 3) or an annular region of 20 pixels in width centered at zero spatial frequency where the unknown information is constrained to be. The algorithm is based on Mooney's work and shares ideas from SVD-POCS. The problem of finding the right degree of hyperspectral model reduction, L , is encountered here as in SVD-POCS. An did in fact show that his algorithm improves upon the pseudo-inverse solution for the imager at Hanscom AFB.

III. Implementation and Analysis

Understanding the fundamentals of hyperspectral technology and chromotomography now allows for implementation and analysis of the pseudo-inverse and iterative improvement solutions. This chapter explains work that was necessary in order to prepare for algorithm modeling and testing, discusses how data was sourced and synthetically generated, how radiometric units are handled, how noise was modeled, how both the pseudo-inverse and iterative algorithms are implemented in MATLAB, and how they compare. The SVD-POCS algorithm can be improved by finding additional constraints to maximize known information and by finding ways to aid in reconstruction performance monitoring. Improvements may result from adding additional known constraints which decrease error, automatically selecting variables used in reconstruction, or automatically monitoring reconstruction error improvement. Changes made to the SVD-POCS algorithm are tested against Brodzik's version from [9] in Section 3.5.

3.1 Data Preparation

Temperature Map. In order to gauge the effectiveness of reconstruction models, some original reference object cube is needed. A two dimensional temperature map is used to create the needed reference. The mean temperature of the temperature map is 295.3K. The blackbody curves, corresponding to each spatial temperature, are input into each spatial location creating a three dimensional hyperspectral data cube. Figure 7 shows the map and Figure 8 shows the expected blackbody curve between 2 and 5 μm for a temperature of 295 K assuming zero atmosphere absorption.

A set of atmospheric absorption coefficients obtained from the Gemini Observatory [12] website are used to simulate atmospheric absorption. Figure 9 shows the



Figure 7 A matrix of temperature values is used to create a known hyperspectral data cube by finding the blackbody curves for each spatial location of the image. The temperature map appears to have a river, fields, and hills as terrain features.

attenuation assuming the observer is at the top of the atmosphere (20km) looking straight down.

Heavy atmospheric attenuation from $2.5\text{-}3.0\text{ }\mu\text{m}$ combined with a very small signal in the blackbody radiance around 295 K from $2.0\text{-}3.0\text{ }\mu\text{m}$ makes analysis of the sub $3.0\text{ }\mu\text{m}$ region difficult, especially when also looking at the $3.0\text{-}5.0\text{ }\mu\text{m}$ region. Noise in the $2.0\text{-}3.0\text{ }\mu\text{m}$ region is more significant compared to the signal, requiring a very different optimal inversion threshold than the $3.0\text{-}5.0\text{ }\mu\text{m}$ region. This study will focus on the $3.0\text{-}5.0\text{ }\mu\text{m}$ spectral region considering attenuation and noise factors. Table 1 shows the center wavelength for each spectral band for the simulations to follow.

AVIRIS Data. AVIRIS hyperspectral data was obtained from NASA's Jet Propulsion Laboratory web site to analyze characteristics of actual hyperspectral data and to use as a baseline for unit values and conversion from spectral radiance to photons. Each AVIRIS scene consists of radiance data with 224 spectral bands

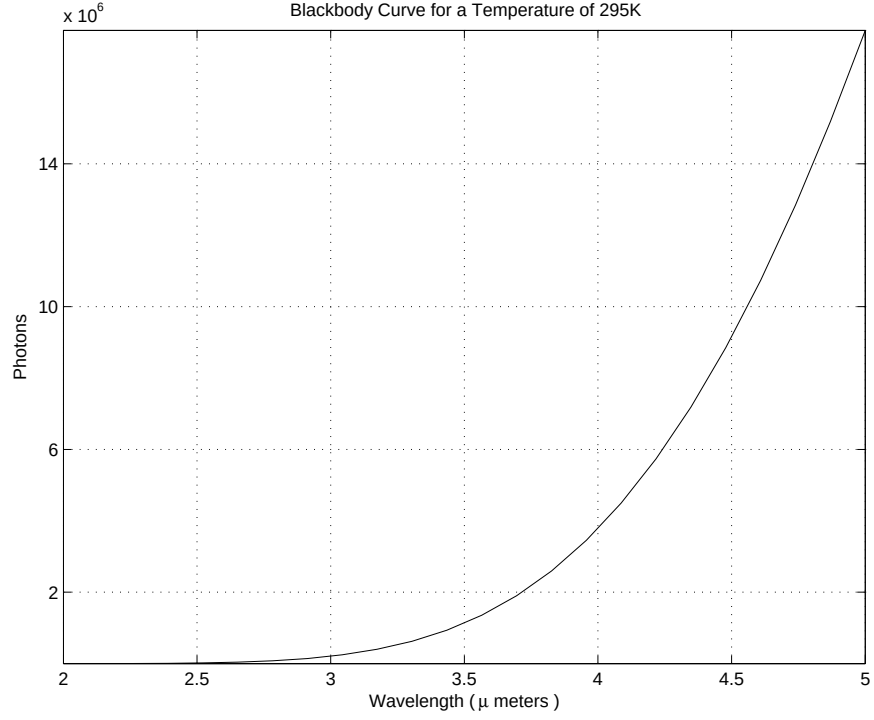


Figure 8 The blackbody response increases rapidly as wavelength increases in the region shown. The dominance of the larger wavelengths will result in better reconstruction for those bands. The blackbody curve intensity shown here is in number of photons.

Table 1 Each wavelength bin has a width of 83.33nm and is centered at a particular wavelength corresponding to the wavelength bin number.

Bin	Center λ (μm)	Bin	Center λ (μm)	Bin	Center λ (μm)
1	3.04	10	3.76	19	4.48
2	3.12	11	3.84	20	4.56
3	3.20	12	3.92	21	4.64
4	3.28	13	4.00	22	4.72
5	3.36	14	4.08	23	4.80
6	3.44	15	4.16	24	4.88
7	3.52	16	4.24	25	4.96
8	3.60	17	4.32		
9	3.68	18	4.40		

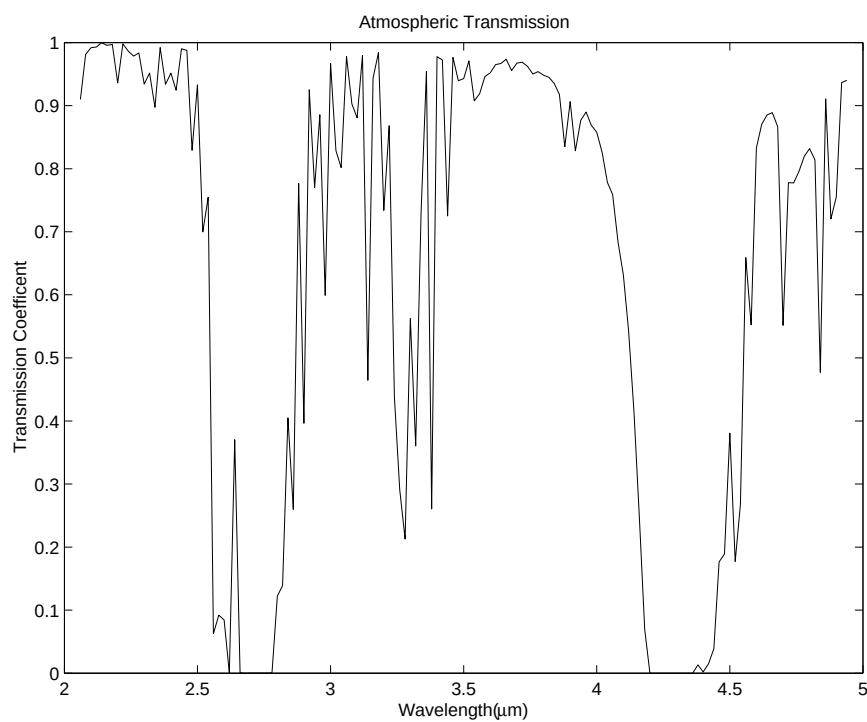


Figure 9 There are two major attenuation regions within the 2-5 μm band. The first, occurring between 2.6-3.3 μm , is mostly caused by water vapor and ozone. Carbon Dioxide is responsible for heavy absorption in the second region centered at 4.3 μm .

of 512x614 images. AVIRIS is in units of $[Gain * \text{mW} * \text{cm}^{-2} * \text{nm}^{-1} * \text{sr}^{-1}]$. The drawback of using AVIRIS data as a known hyperspectral scene is the data set is mostly for smaller wavelengths than the 2.0-5.0 μm band of interest. AVIRIS wavelengths range from .37 μm to 2.5 μm . There is a contribution from the sun in the 2.0-2.5 μm band that greatly amplifies the intensity making the blackbody generator output insignificant in comparison.

Simulated Cold Field Stop. Setting the object image border region of $\frac{K}{2}$ to zero simulates the presence of a cold field stop assuming the field stop emits zero intensity where K is the number of spectral bands being reconstructed and assuming the prism has perfectly symmetrical dispersion for all wavelengths passed into the imager. Thus, the number of possible reconstruction bands is twice the width of the cold field stop assuming uniform prism dispersion. The cold field stop constraint is:

$$\hat{o}(x, y, \lambda) = P_{cfs} o(x, y, \lambda) = \begin{cases} 0, & (x, y) \in \mathfrak{R}_{cfs} \quad \forall k, \\ o_k(x, y), & \text{otherwise} \end{cases} \quad (23)$$

where $o_k(x, y)$ is each spectral image of the source object cube $o(x, y, \lambda)$, $\mathfrak{R}_{cfs}$ is the region the cold field stop occupies, and k is the spectral band index out of K spectral images.

Conversion to Photons. The output of the blackbody generator, O_{bb} , is in units of spectral radiance ($Gain * \text{mW} * \text{cm}^{-2} * \text{nm}^{-1} * \text{sr}^{-1}$) and is converted to units of photons in order to model shot noise on the CCD. This conversion is done to the initial object cube but could be made to the synthetic data at the focal plane. The first step of the conversion process involves conversion from spectral radiance to radiance, L . The conversion to radiance is:

$$L = \int \frac{O_{bb}}{1000 * Gain} * d\lambda, \quad (24)$$

in units of ($Watts * cm^{-2} * sr^{-1}$) where the integration of λ occurs over the width in nm of the spectral band of interest. The radiance theorem states that radiance is equal to power divided by the area of the projection divided by the solid angle or $L = \Phi / (A_{proj} \Omega)$ where $\Omega = A_{aperature} / Range^2$ and Φ is the power over A_{proj} . By the radiance theorem, the conversion to power ($Watts$) is:

$$\Phi_{bb} = \frac{L}{1000 * Gain} * \frac{A_{aperature}}{Range^2} * A_{pixel}, \quad (25)$$

where $A_{aperature}$ is the area of the aperture in cm^2 , A_{pixel} is the area of the ground footprint imaged by one pixel in cm^2 and is equal A_{proj} since the imager will have a negligible offset angle from the source being imaged. $Range$ is distance to target in cm , and $Gain$ is the AVIRIS gain value. The AVIRIS website states that the angle created by one CCD pixel width relative to the aperture is approximately 1 mrad. Since $A_{pixel} / Range^2 = A_{pixelCCD} / Focallength^2$ in any optical system, Equation 25 can be simplified to:

$$\Phi_{bb} = \frac{L}{1000 * Gain} * A_{aperature} * (1 \text{ mrad})^2. \quad (26)$$

According to the AVIRIS website, the size of the aperture of the imager, $A_{aperature}$, is 18x13 cm.

To find the number of photons from watts:

$$\#P = \frac{\Phi_{bb}}{h * v} * time, \quad (27)$$

where h is Planck's constant, v is frequency in Hertz, and $time$ is the pixel exposure time in seconds.

Noise Model. The number of photons incident onto the CCD, which is needed to estimate the level of shot noise of the CCD, was found in Equation 27. Shot noise results from the uncertainty in arrival of photons at a detector element

in any light detection system [13]. The arrival of photons at the detector can be modeled as a Poisson statistic with a standard deviation:

$$\sigma_{shot_{m,n}} = \sqrt{\#P}, \quad (28)$$

where $\sigma_{shot_{m,n}}$ is expressed as a number of photons that fall within pixel (m, n) .

Even though the distribution of photon arrival is Poisson, it can be modeled as Gaussian as a consequence of the central limit theorem [14] since $\#P \gg 1$. Thus, the additive noise at each pixel (m, n) is a random variable with the distribution:

$$noise_{m,n} = \left(\sqrt{2\pi} \sigma_{shot_{m,n}} \right)^{-1} \exp\left(-\frac{1}{2} \left(\frac{t}{\sigma_{shot_{m,n}}} \right)^2\right). \quad (29)$$

The noise matrix for each rotation angle, i , is simply each value of $noise_{m,n}$ for all (m, n) ordered in an $M \times N$ matrix. The noise vector $N_{u,v}$, first presented in Equation 15, is a vector of spatial frequencies resulting from the 2D DFT of each additive noise matrix.

Other noise sources, such as read and dark noise, are assumed to have negligible effects and are not considered in noise modeling. Dark noise can be reduced by using a scientific grade CCD employing multi-pinned-phase (MPP) technology and cooling the CCD to around -25°C . Read noise is negligible provided the imager is not under low-light level conditions.

3.2 Implementation of Pseudo-inverse in MATLAB

The pseudo-inverse solution is the foundation to reconstruction in chromotomography regardless of which error improvement iterative algorithm follows. This section describes the implementation of the pseudo-inverse solution with an explanation of why certain variables were chosen over others. MATLAB code is available in Appendix A.

Synthetic Data Generation. The method used in this thesis to simulate synthetic data is most similar to Mooney’s model in [7]. This study is limited to the use of synthetic data for modeling because a working instrument is not available. Existing hyperspectral data cubes are used to create synthetic data as the CCD recording estimate. The point spread function is calculated as a function of prism dispersion per wavelength bin and prism rotation angle only. A more detailed transfer function would also consider affects of the lenses or a precise prism dispersion pattern. The pixels on the CCD are assumed to be square and the dispersion displacement at one pixel on the CCD is the spectral bin width. A more precise system transfer function of the hardware cannot be obtained at this time and is unnecessary since the scope of this work is to identify trends in chromotomographic reconstruction, not design a system specific optimal reconstruction design. See Appendix A for the MATLAB code. The method used for this study to generate synthetic data involves mimicking the effects a rotating prism has on the source object cube. In [9], Brodzik did not create synthetic data to use for reconstruction but created a synthetic pseudo-inverse reconstruction by removing zero spatial frequency information of the source object cube; and then multiplying each spatial frequency column of the object cube by $W_{u,v}^H W_{u,v}$, where $W_{u,v}$ is the STF for the particular spatial frequency.

A point spread function matrix of matrices, w , of size (M, N, K, I) is created where M is the height of the CCD in pixels, N is the width, K is the number of wavelength bins, and I is the number of prism rotation angles. Mooney showed in [7] that the point spread function w can be written as a displaced delta function given by:

$$w(m, n, k, i) = \delta(m - (k - k_0)\Delta m \cos(\phi_i), n - (k - k_0)\Delta m \sin(\phi_i)), \quad (30)$$

where n and m are spatial coordinates, Δm is the CCD pixel size, k is the spectral variable, k_0 is the undeviated wavelength, and ϕ_i is the prism rotation angle at index i ($0 \leq \phi_i < 2\pi$).

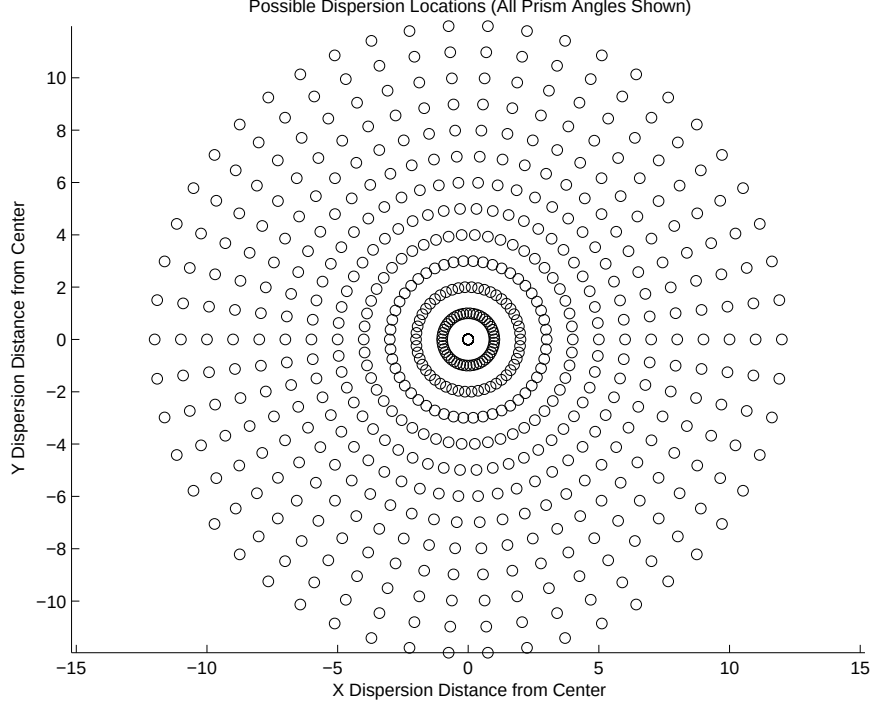


Figure 10 Any pixel location of the object cube can be dispersed at a distance from the origin represented by the circles shown here. The center of the pattern represents a shift distance and direction of zero.

If there are K wavelength bins with I rotation angles, there exist $M * N$ point spread matrices of size (K, I) or alternatively, $K * I$ matrices of size (M, N) . Taking the 2D DFT of each of these and looking at one spatial frequency gives $W_{u,v}$, the STF for each spatial frequency originally introduced in Equation 9. Each of the $K * I$ $w(m, n)$ matrices are created using an interpolation routine to estimate recorded pixel intensities based on the source object cube. An absolute dispersion distance, $(k - k_0)\Delta m$, is dependent on the wavelength bin value. Each point in the object cube is in effect mapped to a spot of the output data, d , based on the point's location in the cube and current prism rotation angle. The resulting dispersion pattern possible for each pixel, shown in Figure 10, is circular in appearance.

The discrete nature of the output data requires an interpolation routine to represent intensities that fall on non-integer pixel shift distances. To do this, the

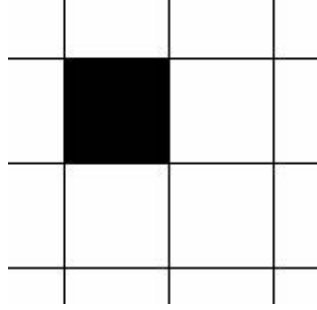


Figure 11 The object cube pixel intensity falls within the bounds of one "pixel" location of the synthetic data array. When this occurs, no interpolation is needed.

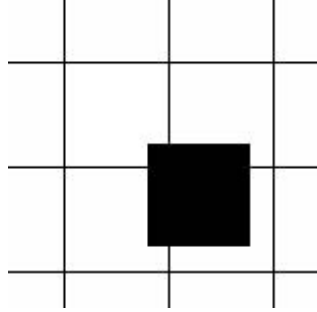


Figure 12 Here the object cube pixel intensity falls over multiple pixels. One pixel in the object cube can fall across up to four synthetic data "pixels".

magnitudes of the shift distances in both the X and Y directions is rounded down to the closest integer. The rounded number is subtracted from the magnitude distance of the corresponding direction resulting in a leftover fraction. This fraction is used to determine the percentage of the intensity of the object cube value to the pixel location of the recorded data.

Synthetic data is created in the transform domain by multiplying each spectral frequency column of the data cube $O(u, v)$ by the corresponding $W_{u,v}$ which gives $D(u, v)$ as shown in Equation 9. The spatial domain data, which is the estimate the CCD would record in an actual chromotomographic imager disregarding noise, is easily obtained by finding the inverse two-dimensional DFT of $D_i \forall i$, giving d . Figure 14 shows synthetic data representations for six different values of i , spaced evenly at an angle of 57.6° . Noise was added to each pixel in the spatial-chromatic

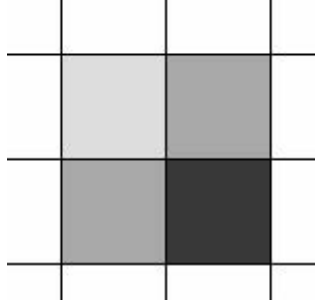


Figure 13 Intensities are assigned to the synthetic data pixels based on the percentage of overlap the object pixel has with the four data pixels. The sum of the intensities of the four pixels adds to the object cube intensity value. This method of interpolation was required to overcome the obstacle of sub-pixel shifts.

cube d by finding:

$$\acute{d}_i(m, n) = d_i(m, n) + noise_{m,n}, \quad (31)$$

where $noise_{m,n}$ comes from Equation 29. A description of how noise is accounted for is found in Section 3.1.

Pseudo-inverse Reconstruction. The pseudo-inverse reconstruction step is the first image processing step of a fielded chromotomographic imager. After imaging the scene through the direct vision prism, data recorded at the focal plane is transformed to the frequency domain one image at a time via the 2D DFT (giving $\acute{D}$). The data cube $\acute{D}$ has two spatial frequency components u and v , and a rotation index dimension i . To form the pseudo-inverse solution $\acute{O}_{u,v}^+$, the KxI spatial frequency specific $W_{u,v}$ matrix must first be decomposed by SVD, inverted by either Equation 18 or Equation 19, and subjected to the operation in Equation 20 for each spatial frequency (u, v) . The spatial cube solution is:

$$o^+(x, y, \lambda) = IFFT2(\acute{O}_k^+(u, v)) \quad \forall k, \quad (32)$$

where $\acute{O}_k^+(u, v)$ is the k th spatial frequency matrix and IFFT2 is the inverse 2D Fourier transform. Figure 15 shows an example of pseudo-inverse reconstruction for

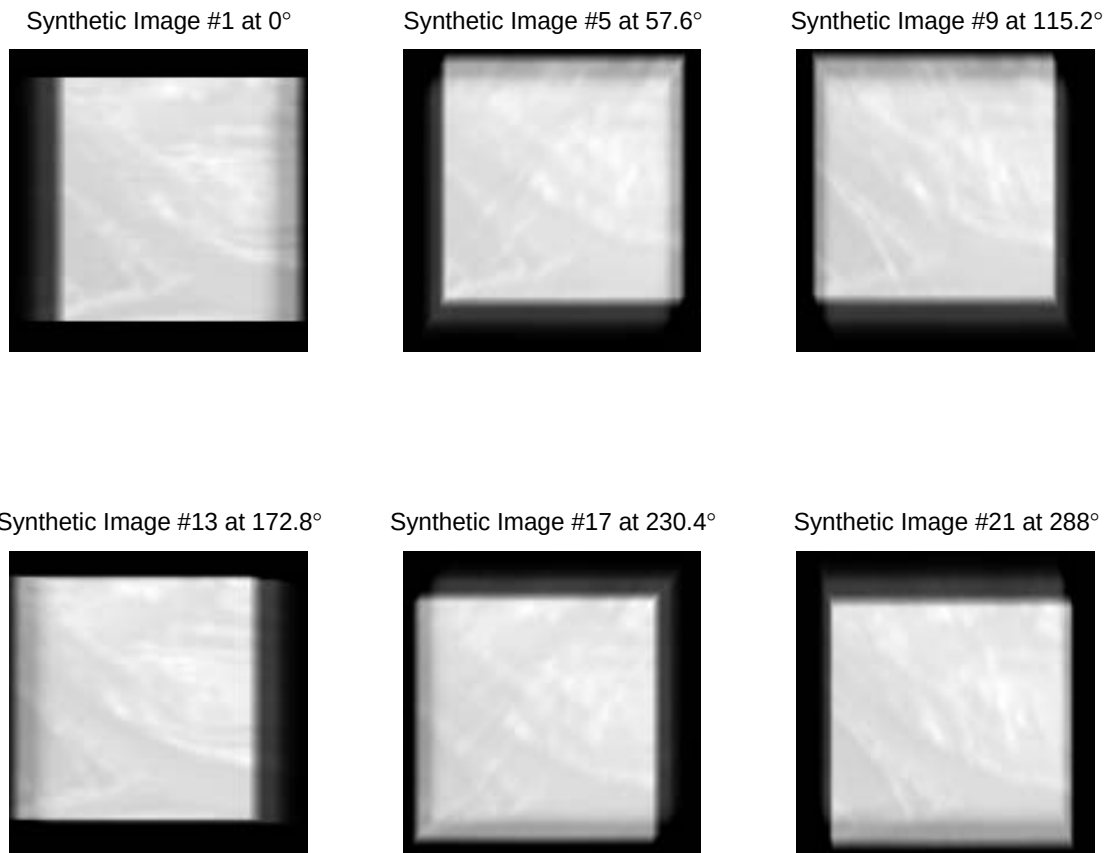


Figure 14 Data at the focal plane recorded by the CCD appears blurred in the direction of prism dispersion. In this example, the prism rotates counter-clockwise starting in a horizontal disposition in the upper left image. The dark border corresponds to the cold field stop. Noise is present in each image.

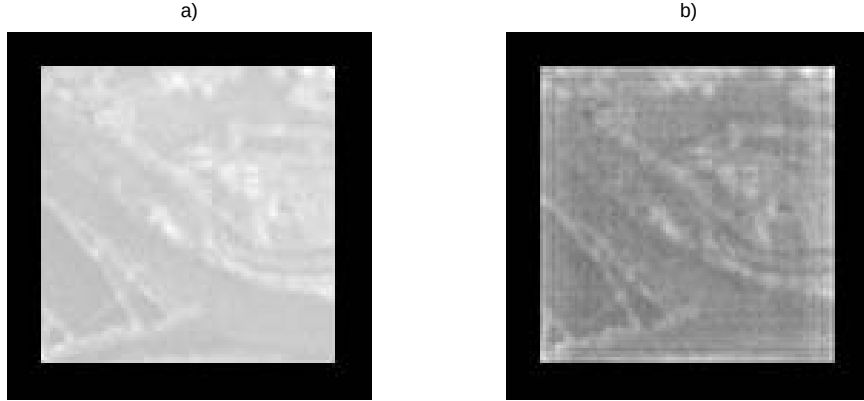


Figure 15 The pseudoinverse solution, b), to the original object, a), is a good estimate of the original scene. Noise on the CCD lowers reconstruction performance. All error calculations compare a reconstructed scene to a the original scene after atmospheric absorbtion has taken place if absorption was taken into consideration.

wavelength bin 20 with atmosphere attenuation and additive shot noise on the CCD. The reconstruction error increases with the addition of atmosphere attenuation and noise.

Pseudo-inverse Solution with the Wiener Inverse. Since the system transfer function is not directly invertible, an estimate of the inversion must be found. As mentioned earlier, there are two methods discussed previously in literature to do this ([2], [7]). The question, not answered previously, is which method allows for the smallest reconstruction error in general. To find a solution to the problem, reconstruction was performed several times while only changing the inversion parameter, ε , for each method. For the Wiener inverse, each inverse of $\Sigma_{u,v}$ is altered by finding:

$$\dot{\Sigma}_{(k,k)u,v}^{-1} = \frac{\Sigma_{(k,k)u,v}}{(\Sigma_{(k,k)u,v}^2 + \varepsilon^2)}, \quad (33)$$

where ε is chosen in a manner to balance reconstruction error from added noise with noise amplification. Table 2 shows the effect of different values of ε under both a noise free and noisy environment for both atmospheric pass bands within the 2.8-

Table 2 NRMSE average for the pseudoinverse reconstruction using the Wiener method. The best value of depends on the SNR over the band of interest.

ε	2.8-4.1 μm	2.8-4.1 μm (with noise)	4.5-5.0 μm	4.5-5.0 μm (with noise)	3.0-5.0 μm (with noise)
5e-9	53.24	5044	20.21	2520	5911.4
5e-6	53.23	637	20.23	339	398
5e-5	53.26	168	20.39	96.31	117
5e-4	53.41	72.52	20.91	34.56	72.28
1e-4	53.29	126	20.45	69.59	94.50
5e-3	53.68	55.35	21.37	22.85	69.53
0.01	53.76	54.51	21.50	22.17	69.49
0.1	54.38	54.41	22.48	22.50	71.08
1.5	56.89	56.90	23.10	23.10	71.55
100	92.57	92.52	90.48	90.48	95.88

5.0 μm band and across the entire 3.0-5.0 μm band. Mean square error was chosen to determine which inversion method was best, since the goal is to perform absolute radiometry.

Results are shown for an imager with the standard configuration shown in Figure 3, however, error trends are the same for the advanced imager configuration shown in Figure 50 which includes a warm field stop. The value of ε corresponding to the smallest error is different for each band due to the increased influence of noise for the smaller wavelengths. The congrate best value found experimentally for the 3.0-5.0 μm band is 0.01 resulting in an NRMSE of 69.49. The simulation was run 10 times for the 3.0-5.0 μm band to account for variance caused by random noise. When no noise is present, the best value of ε is the same for all bands, the smallest value tested of 5e-9.

Pseudo-inverse Solution with the Threshold Inverse. In the threshold inversion method, singular values resulting from SVD of $W_{u,v}$ less than some ε have the corresponding inverse set to zero; defined previously as:

Table 3 NRMSE average of pseudoinverse reconstruction cube using the Threshold method. The best value of depends on the SNR over the band of interest.

ε	2.8-4.1 μm	2.8-4.1 μm (with noise)	4.5-5.0 μm	4.5-5.0 μm (with noise)	3.0-5.0 μm (with noise)
5e-9	53.24	4737	20.21	4179	6531
1e-6	53.24	1343	20.21	1132	1252
1e-5	53.24	424	20.28	265	245
1e-4	53.27	136	20.42	78.52	102
1e-3	53.43	64.64	21.16	29.16	71.35
0.01	53.70	54.57	21.43	22.24	69.84
0.1	54.38	54.41	22.67	22.69	71.34
1	56.24	56.24	22.83	22.83	76.27
10	74.82	74.82	30.99	30.99	91.30
100	97.10	97.10	95.53	95.53	100

$$\hat{\Sigma}_{(k,k)u,v}^{-1} = \begin{cases} \Sigma_{(k,k)u,v}^{-1}, & \Sigma_{(k,k)u,v} > \varepsilon \\ 0, & otherwise \end{cases}. \quad (34)$$

Table 3 shows the NRMSE for several values of ε looking at different reconstruction bands under a noise free and noisy condition. All of the results include atmosphere absorption. The two bands of severe atmospheric absorption are avoided at first, but could not be avoided when looking across the entire 3-5 μm band, demonstrating the increased influence of noise in those regions. The best value of ε for 3-5 μm with noise is 0.01 with an NRMSE of 69.84, which is larger than the NRMSE when looking at each atmospheric passband individually. The increase in error is caused by atmospheric attenuation in the 4.1-4.5 μm region. As expected, the optimal value of ε depends on the SNR of system, as it did for the Wiener inverse. With no noise, the best choice of ε found experimentally for the entire 2-5 μm band is the smallest value tested, 5e-9. The 2.8-4.1 μm band has the smallest SNR, therefore requiring a larger ε of 0.1 to minimize error.

The Best Inversion Method. As expected, reconstruction results for both inversion methods are signal dependent. Both methods work about the same, mean-

Table 4 Performance of Pseudoinverse Methods.

	NRMSE	NVE	NMRE
All Bands-Wiener	69.68	2.0730e+005	8.83
All Bands-Threshold	69.84	1.7379e+005	8.82
Select Bands-Wiener	78.36	216.03	9.38
Select Bands-Threshold	78.50	209.94	9.47

ing it makes little difference which is used. The NVE error metric will be used only to gain a idea of which images visually look most similar to the original. As mentioned before, both NMRE and NVE do not include the mean, but NMRE is on the same normalization scale as NRMSE. Thus the error introduced by the mean is the difference between NRMSE and NMRE. The scale of NVE is harder to comprehend, since each variance error is divided by a mean cube variance.

Reconstruction was performed 10 times, to offset the influence of random noise, for each method to obtain the average NRMSE, NMRE and NVE. Certain bands which are considered to have no information, are not used in the average NVE and NMRE calculation. Information content was determined by calculating the SNR of each band. Only source bands with SNR greater than 177 are used, where SNR is calculated as the mean intensity in a band divided by the square root of the mean noise value for the entire cube. On average, the Threshold method has a slightly lower NVE while having a nearly equivalent NRMSE and NMRE as shown in Table 4. NVE values are much larger in regions of low signal presence. The smaller the NVE, the closer the reconstruction looks like the original object. Figure 17 shows how the NRMSE results are visually indistinguishable between the two methods.

Figure 18 shows that the bands which look the best are those which contain the most energy. Reconstruction between 4.5-5.0 μm visually looks the best. Areas of low signal from a weak plank curve or an atmospheric stop band do not look anything like the source.

Figure 19 shows that with the mean removed, error looks greater in the bands which look better according to NVE and visual observation. The bands from 4.5-

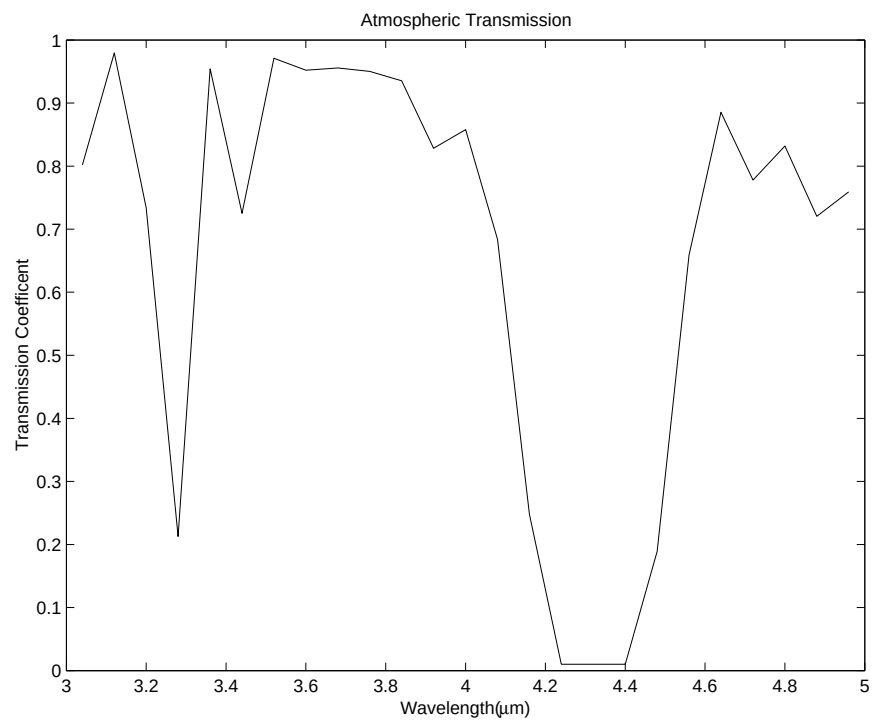


Figure 16 The atmospheric transmission curve is repeated several times through the text to provide a constant reference.

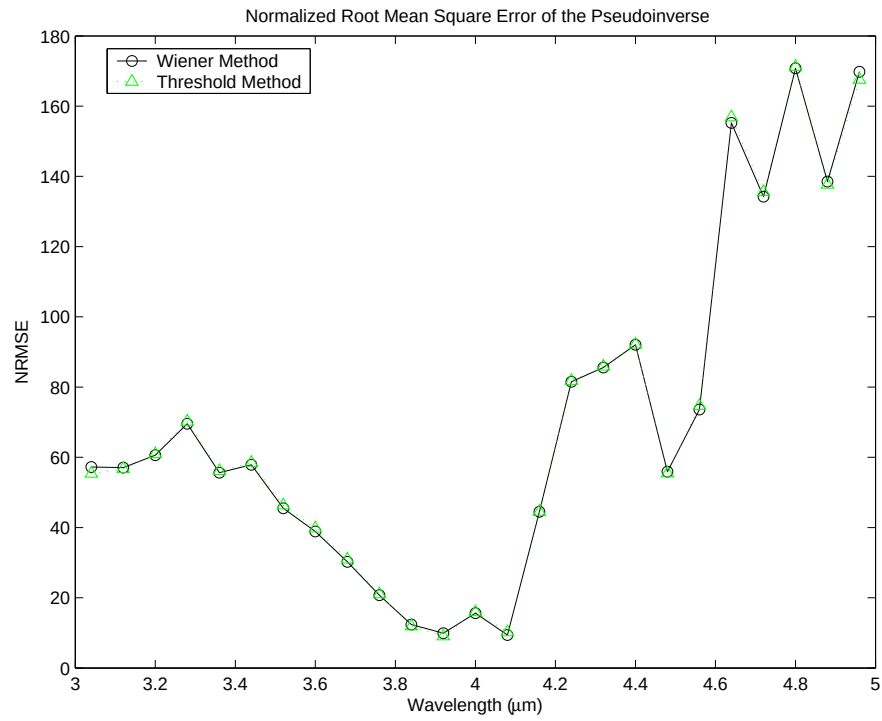


Figure 17 Error versus frequency curves of the pseudoinverse for the Wiener and Threshold methods ($\varepsilon = .01$) are visually indistinguishable from each other.

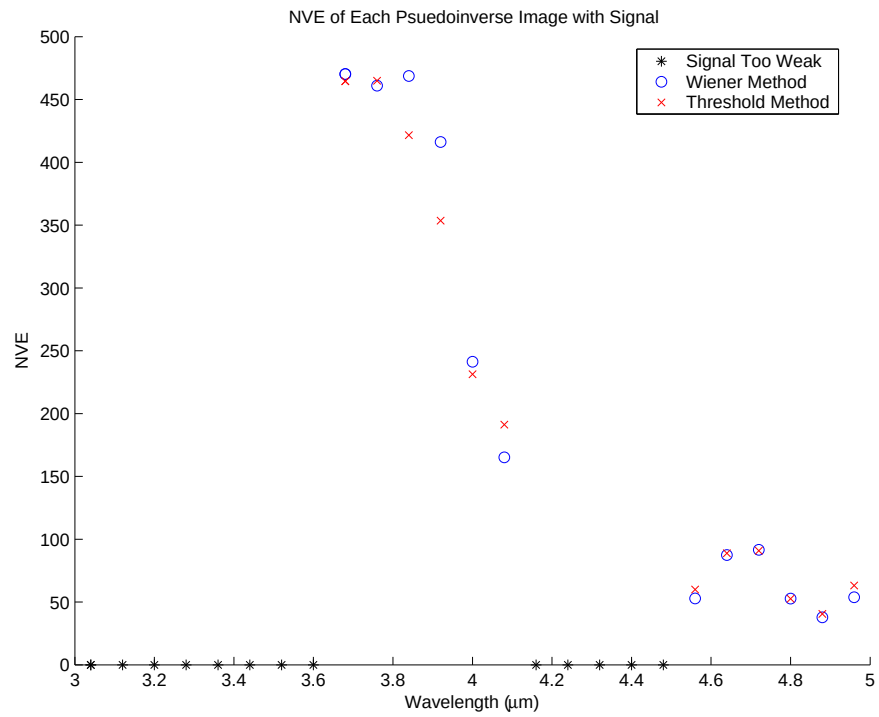


Figure 18 The NVE values were lower overall for the Wiener method than the Threshold method, meaning reconstructions for the Wiener method more closely resemble the original than the Threshold method. The results here are from the temperature map source with noise and atmospheric absorption.

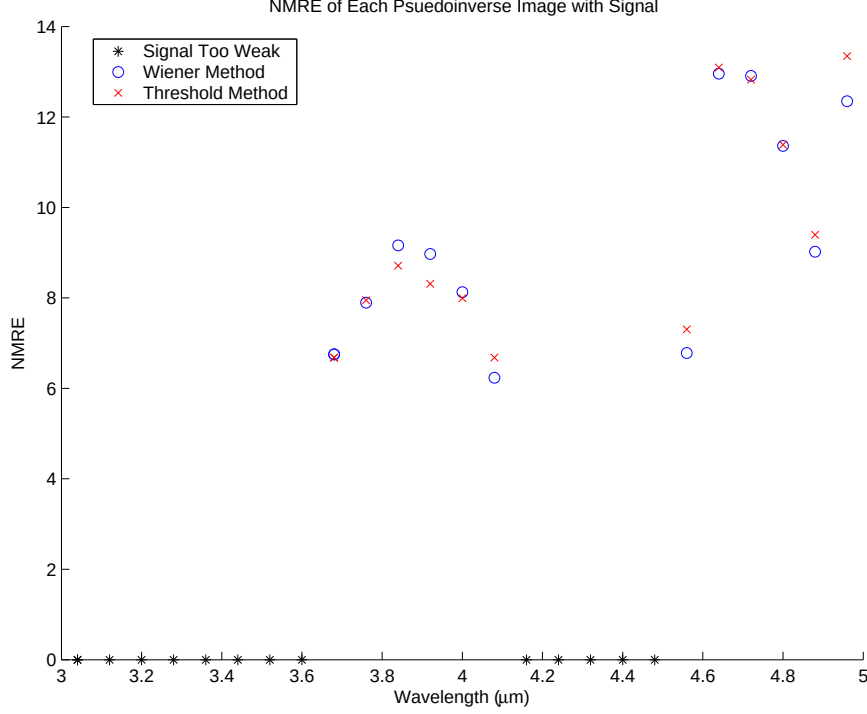


Figure 19 The NMRE looks similar to NVE, but the points occur at different magnitudes. Larger signal no longer means less error.

$5.0\text{ }\mu\text{m}$ have more energy, which leads to a larger error magnitude, yet each band error is divided by the same value, the mean of the original scene. If normalization was done on a band to band basis, the plot would look more like Figure 18. Figure 20 demonstrates the NMRE error trend versus wavelength does not look like NRMSE. A majority of error occurs due to not matching the mean of the original scene.

Impact of Signal Non-Uniformity. Over the course of performing reconstructions for several scenes, it became obvious the intensity of the spectral content relative to the scene as a whole greatly affected reconstruction potential. When dealing with blackbody responses of the temperature map scene between $3.0\text{--}5.0\text{ }\mu\text{m}$, adequate reconstruction for chromatic information between $3.0\text{--}3.42\text{ }\mu\text{m}$ was not possible due to small blackbody signal presence, nor was reconstruction possible between $4.18\text{--}4.5\text{ }\mu\text{m}$ due to atmospheric absorption. Adequate reconstruction across the en-

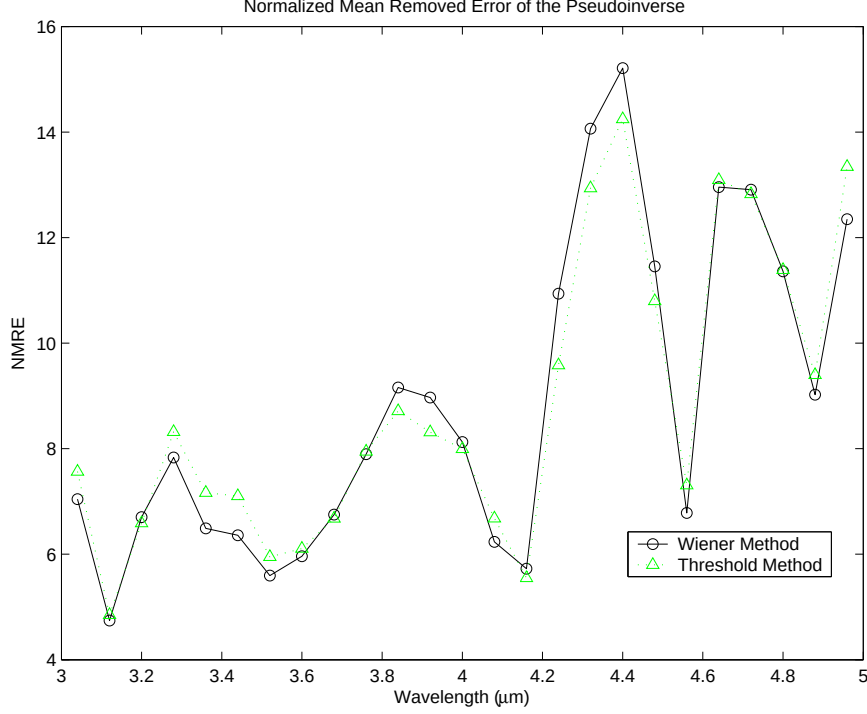


Figure 20 The NMRE curves follow each other closely for both inversion methods.

tire spectral bandwidth, not including atmospheric stopbands, is only possible if the average intensity of each source chromatic slice is relatively uniform across the spectral bandwidth.

3.3 Implementation of the Iterative Improvement Algorithm

The next step is to implement the iterative algorithm SVD-POCS in order to understand the trade-offs made in choosing parameters and showing how reconstruction error decreases compared to the pseudo-inverse.

The SVD-POCS Algorithm. Figure 22 shows a flow chart for the SVD-POCS method as described in [9]. The first step is to prepare the pseudo-inverse data set for the iterative procedure. After performing the original reconstruction, the mean of each spectral image in o^+ is removed, creating the data set g . The 2D DFT of each of the mean removed spectral images of g is reorganized lexicographically into a spatial

frequency-chromatic matrix F_0 where each row corresponds to one of the K spectral bands and each column corresponds to a spatial frequency. For this paper, the function $Lxshape(\cdot)$ represents lexicographical ordering as previously defined. The operator to reshape the information into the normal (x, y, λ) hyperspectral cube is defined as $Cushape(\cdot)$. The zero subscript shows this value of F is for the zeroth iteration, i.e. pseudo-inverse solution. F_0 is then subjected to two projections based on an object domain constraint and a transform domain constraint. Additionally, g can also be reorganized lexicographically into a spatial-chromatic matrix, f_0 , where each row corresponds to one the K spectral bands and each column corresponds to a pixel location.

Object Domain Constraint. The object domain constraint takes advantage of the similarities between spectral images by shaping unknown regions of data to be consistent with known similarities. The SVD of F is:

$$SVD(F) = U_F \Sigma_F V_F^H, \quad (35)$$

where U_F is the eigenchroma matrix consisting of chromatic singular vectors, Σ_F contains the singular values of F along its diagonal, and V_F is the eigenimage matrix containing spatial singular vectors. The values of Σ_F are arranged in a descending order where the largest values contains the majority of the information about the scene. Figure 21 shows the values of Σ_F for the temperature map source. Large singular values typically correspond to true object information, while smaller values correspond to noise and artifacts. The image may be better represented by using the dominate singular values only, which is the foundation to the object domain constraint.

Finding the covariance matrix $R_{F_0 F_0} = F_0 (F_0)^T$ provides a measurement of the similarity between the rows in F_0 . Finding $SVD(R_{F_0 F_0}) = A X A^T$ gives a matrix of eigenvectors A and a diagonal matrix of eigenvalues X since $R_{F_0 F_0}$ is symmetric.

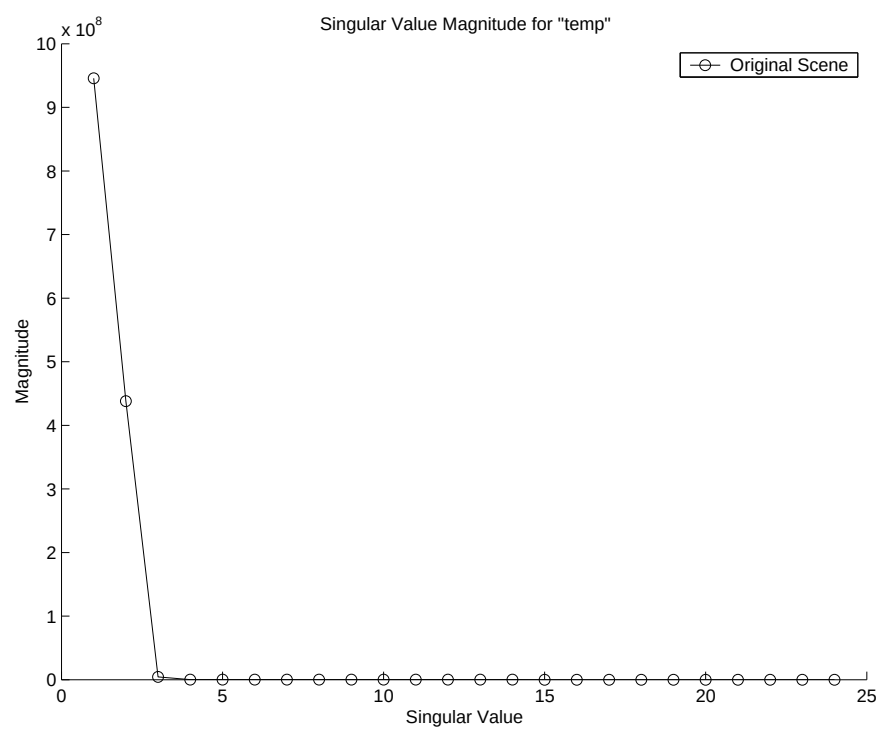


Figure 21 The first and second singular values dominate for the temperature map infrared hyperspectral data cube.

Brodzik explains that by using the first L columns of A (or A_L) the dimensionality of the object data can be reduced by finding the projection:

$$Q_m = A_L(A_L)^T F_{m-1}, \quad (36)$$

where L is the model dimension. A_L has dimension $L \times K$ and can be used instead of using A since $AI_L A^T = A_L(A_L)^T$, thereby reducing computational complexity. The matrix $A_L(A_L)^T$ is a projection matrix which projects the current solution onto the L largest principal components of the reconstruction. This operation can be carried out in the transform domain or the spatial domain since the eigenchroma matrix A is the same for the object domain matrix $f_0(f_0)^T$ as it is for $F_0(F_0)^T$ where f_0 is the lexicographically arranged spectral image with mean removed. The optimal value of L , which leads to best reconstruction, is difficult to determine a priori and has been studied by others seeking to solve similar dimension reduction problems. For testing purposes, each value of L will be tried in a brute force approach to find the best L for reconstruction.

Transform Domain Constraint. Given that the pseudo-inverse is a unique solution created by disregarding the null-space of the STF W , the solution itself can be considered to be known data and therefore used as a constraint set. Most non-zero singular values of W are considered to be correct information in the reconstruction, while zero or near zero singular values lead to incorrect data. By projecting onto the column space of W , the null space of W will not be considered. This limits the solution to the "known" spectral information. The transform domain projection is:

$$V_m = (I - W^+ W) Q_m, \quad (37)$$

meaning Q_m is projected onto the null-space of W . Only the region of Q_m that corresponds to the null-space of W will be used in the iteration. For computational

purposes, Brodzik simplifies this expression to:

$$V_m = B_{K-r}(B_{K-r})^H Q_m, \quad (38)$$

where $W^+W = BI_rB^H$ and therefore $(I - W^+W) = B_{K-r}(B_{K-r})^H$. By setting the first $K - r$ columns of B to zero column vectors to get B_{K-r} with r being the rank of W at a particular spatial frequency. Adding the projection of Q_m onto the null-space of W with the initial reconstruction completes the iteration as shown in Figure 22. The reconstruction performance for SVD-POCS is discussed in Section 3.3. The mean is removed for each iteration which is shown to improve variance image reconstruction performance while degrading mean square error performance..

Iteration Solution. The result of the m' th iteration, V_m , is the new estimate for the missing information. The iteration solution is found by adding V_m to C , which is a spatial-chromatic matrix of the pseudo-inverse arranged lexicographically where each row corresponds to a spectral band and each column corresponds to a spatial frequency. The operation is:

$$G_m = C + V_m, \quad (39)$$

where G_m is a 2D spatial-chromatic matrix arranged identically to the matrix C . If the iterative process is to continue, the current solution has the mean removed from each band by setting the zero spatial frequency column to zeros, which gives F_m . If iterations stop, the current solution, G_m , must be rearranged back to the hyperspectral data format and transformed to the spatial domain which yields the new hyperspectral reconstruction cube, o_{sp} . The subscript " sp " designates the output as SVD-POCS. Figure 22 shows a flowchart of the SVD-POCS process. The iterative process is repeated until reconstruction error is reduced to a satisfactory level. Performance of SVD-POCS is discussed in Sections 3.3 and 3.5.

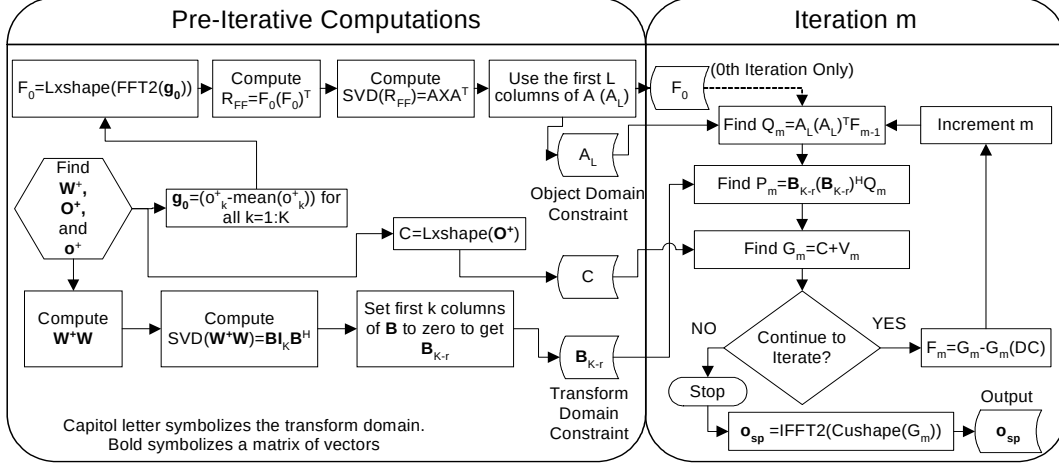


Figure 22 In SVD-POCS, the initial pseudoinverse reconstruction is projected onto the L principal components of the correlation matrix which is projected onto nullspace of the system transfer function W and finally added to the original pseudoinverse solution. The process is repeated using the new solution until error is satisfactorily reduced.

Performance. The results found in this section are what one can expect when trying to analyze algorithm reconstruction performance. There is not a single error metric which fully describes reconstruction performance. If the goal is to perform absolute radiometry by recovering the mean of the signal, then NRMSE is the metric of choice. But, if one intends to do relative radiometry, then NVE or NMRE are better.

Figure 23 shows how SVD-POCS, using the constraints of [9], is incapable of reconstructing the mean of the signal. Instead, the reconstruction of each band has an identical mean count; making absolute radiometric analysis impossible. The mean photon count per pixel for each reconstructed band is equal to the mean photon count per pixel of the entire original scene. Looking at a sample spatial frequency location, corresponding to a temperature of 294.54K, the SVD-POCS reconstruction does not match the original curve, as shown in Figure 24.

The performance of SVD-POCS depends on the selection of the pseudo-inverse inversion method, the value of ε , the model dimension L , and the number of iter-

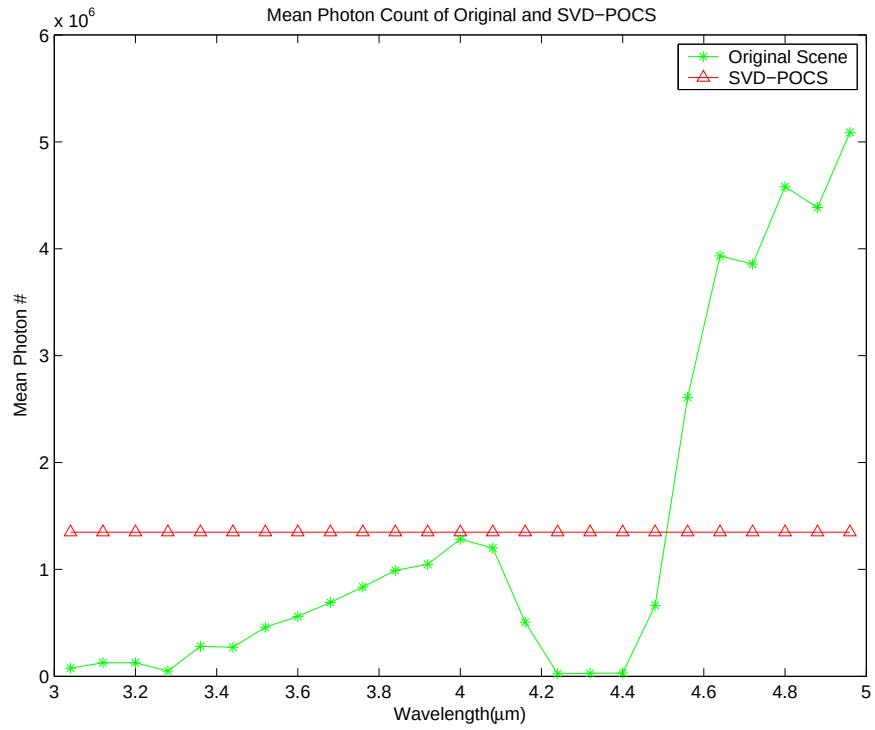


Figure 23 SVD-POCS cannot recover the mean of the original scene.

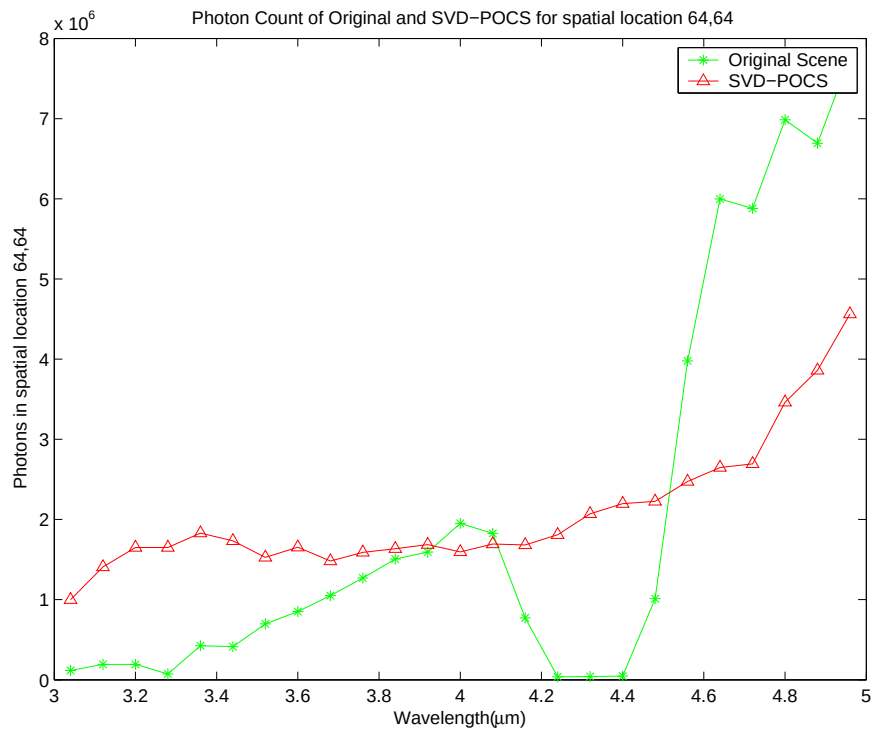


Figure 24 The spectral curve is not reconstructed in this example.

ations. The best choice of these variables depends on the scene content. The best overall method was found to be the Threshold inverse in Section 3.2 with $\varepsilon = 0.01$ for the $3\text{-}5\,\mu\text{m}$ band under a noisy environment. The optimal value of L is difficult to determine before running the algorithm, but, under a non-real time setting, can easily be found by running the reconstruction algorithm for each possible value. Of course that only works if you know the scene you are trying to view, or you have a tool in the reconstruction routine to check error against. (See Chapter IV)

The problem of automatically selecting an optimal model dimension L has been addressed previously by Shannon [15] and others who tried to establish a mathematical relation between the source and the dimensionality reduced result. The task of automatically identify the best value of L is left for future research.

The pseudo-inverse solution in the example of Figure 25 has an NRMSE of 69.84. The lowest value after SVD-POCS is 69.31 with $L=1$ and the highest is 68.85 with $L=22$. Thus, the NRMSE metric tells us mean square error is not being reduced. The mean of the reconstruction is uniform for the entire spectrum range and the NRMSE does not change much versus iteration or versus L . As a consequence, use of a mean square error metric requires an alteration of the standard SVD-POCS implementation.

Even a suboptimal selection of L does not increase the NRMSE or NMRE when comparing to the pseudo-inverse solution, but instead usually results in lower NMRE and an equivalent NRMSE. Figure 25 and 26 demonstrate that smaller NMRE or NVE values do not suggest smaller NRMSE values. Also demonstrated is how the mean NMRE and NVE of bands with signal follow nearly identical trends. This occurs versus dimension and iteration, but not versus wavelength.

The majority of error improving power of SVD-POCS occurs within the first 10 iterations, as demonstrated in Figure 28. The solution converges to a value of approximately 7.87 after just 10 iterations. The mean NMRE of bands which are deemed to contain information (see Section 3.2) is shown when plotting NMRE

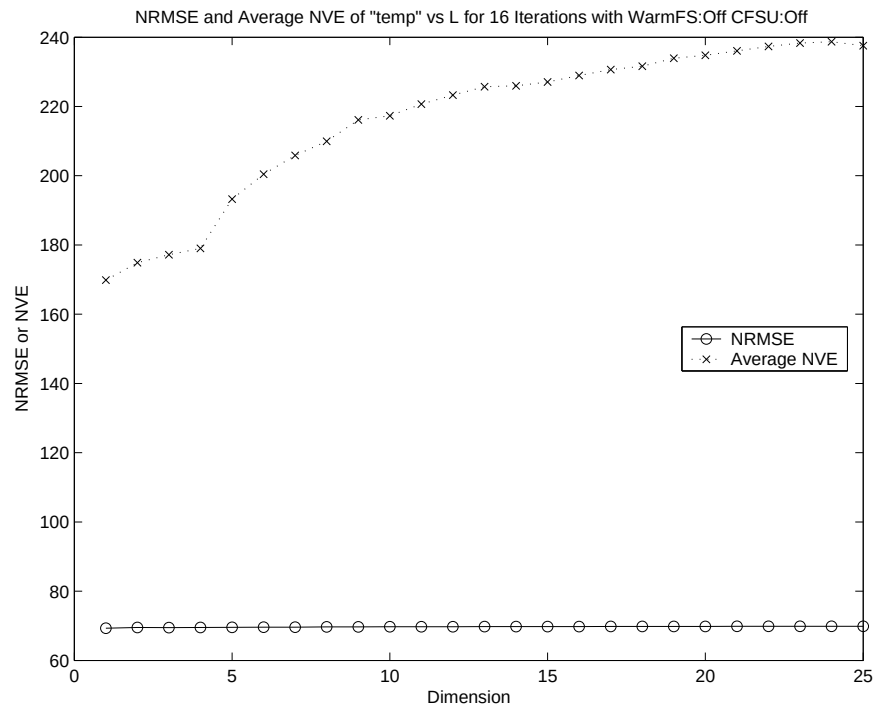


Figure 25 NVE increases as L increases, but NRMSE stays near the same value.

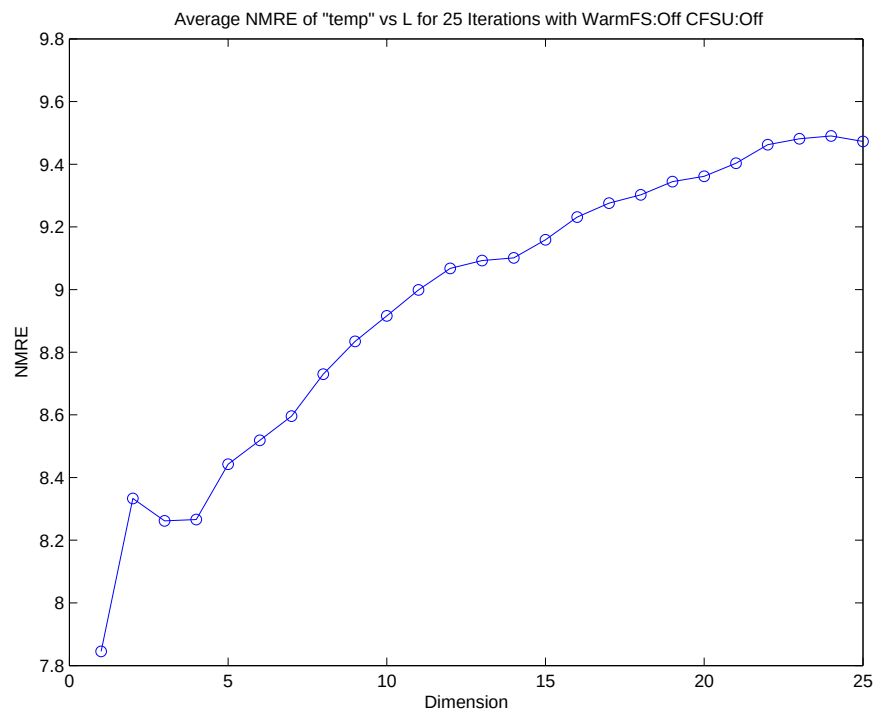


Figure 26 Smaller NMRE values do not imply smaller NRMSE values.

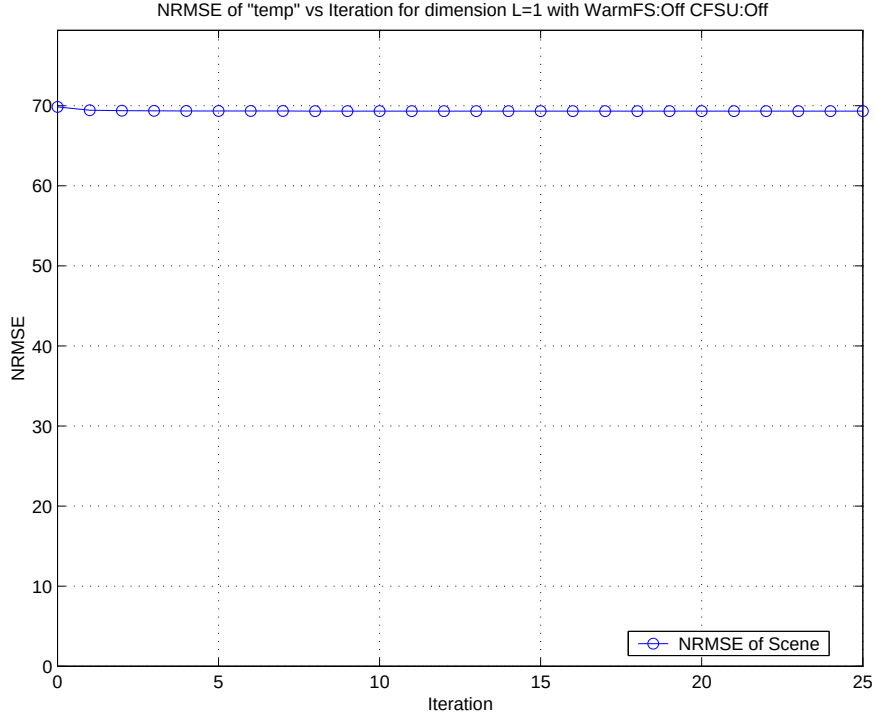


Figure 27 The NRMSE does not change as a function of iteration.

versus iteration at all times in this thesis. The particular result shown here does not diverge within 10000 iterations (results not shown), but yet may diverge at some point.

Figure 29 shows the reconstructed image for band 24 which corresponds to 4.84-4.92 μm after 25 iterations of SVD-POCS. This band had the least reconstruction variance error and should look the best, according to NVE calculations. The small error is due to strong signal presence in the band. Only the NVE values of bands which have adequate signal are shown in Figure 32. The NVE was particularly bad in areas of heavy atmospheric absorption due to the very small signal component. NVE results do a better job than NRMSE of identifying which bands look most like the original. NRMSE does a great job at measuring the error in the intensity of the scene, which NVE cannot do. The SVD-POCS algorithm successfully lowered the NVE for most chromatic reconstruction bands while maintaining the same NRMSE.

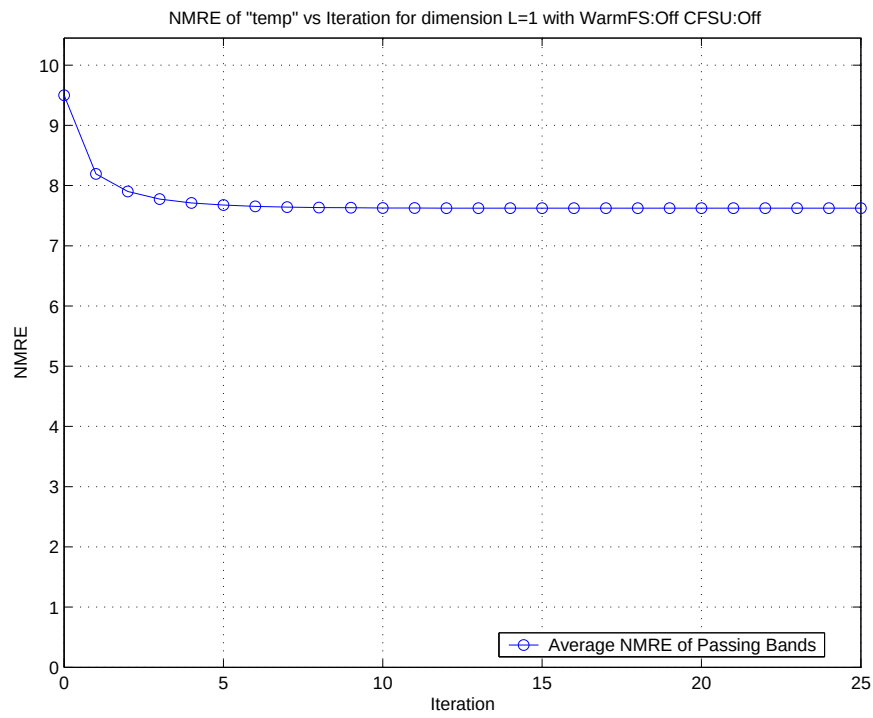


Figure 28 The mean NMRE of values satisfying a the SNR criteria decrease rapidly at first with most improvement occuring within the first 10 iterations.

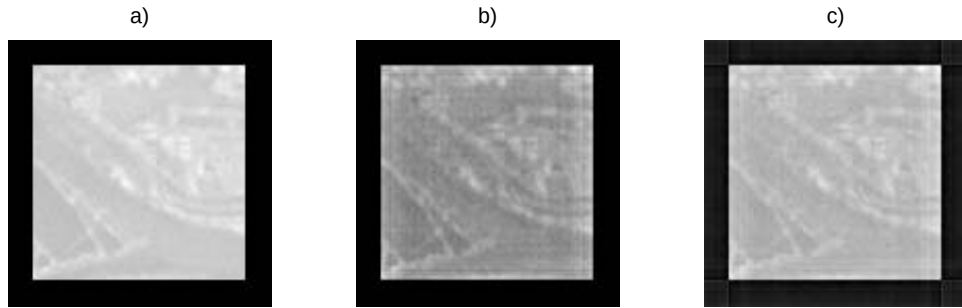


Figure 29 Source chromatic band #24, a), the pseudoinverse reconstruction using the Weiner inversion method, b), and the SVD-POCS reconstruction with $L=1$, c).

The average NVE of the pseudo-inverse accepted values dropped to 159.05 from 209.94. Figure 31 shows that SVD-POCS does indeed reduce the NMRE.

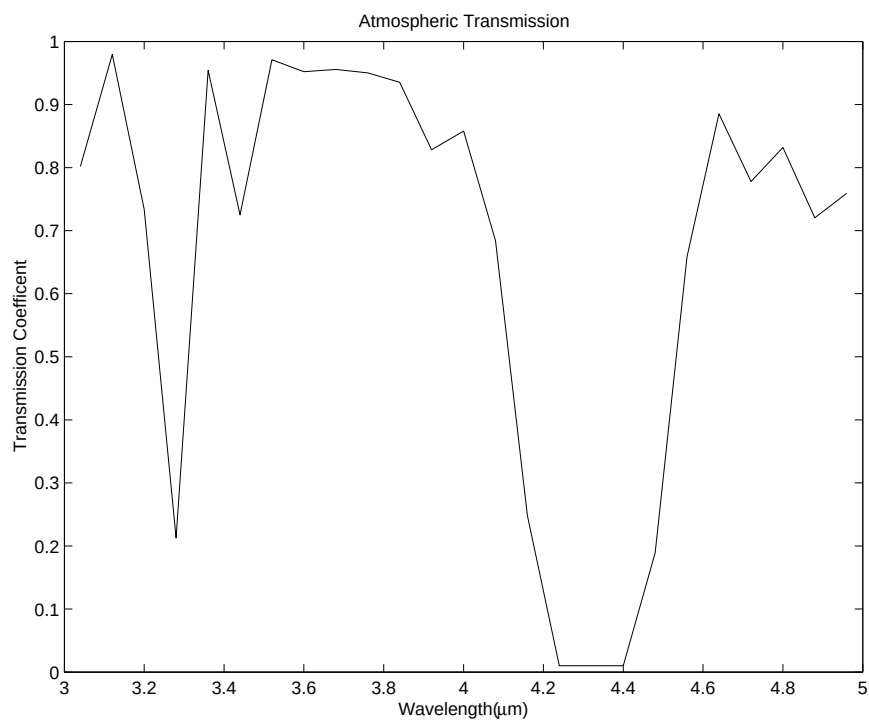


Figure 30 Atmospheric Transmission Curve

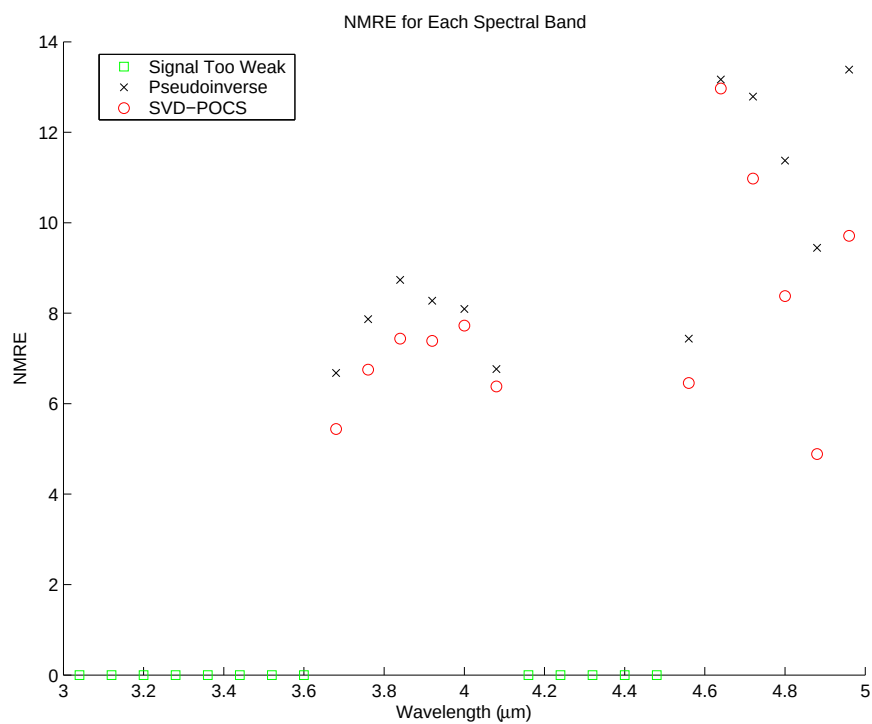


Figure 31 SVD-POCS improves upon the NMRE of the pseudoinverse.

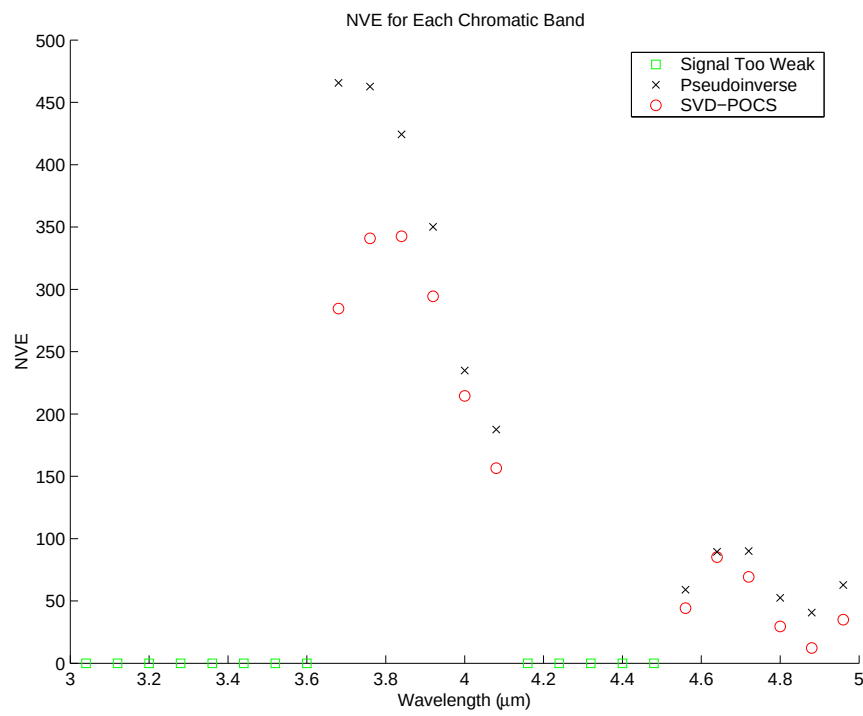


Figure 32 SVD-POCS reduces the NVE for most chromatic bands.

3.4 Modified SVD-POCS (MSP)

Additional constraints are needed to make absolute radiometry analysis practical. Constraints considered here are from information we already know from the scene and have been suggested previously by Brodzik and Mooney in [4]. The major disadvantage of the first three constraints is the addition of an inverse Fourier transform for each spectral image followed by a Fourier transform to change the data back into the transform domain. The three constraints can only be applied in the spatial domain. The advantage is improved radiometry can be performed which is validated in NRMSE calculations. Figure 33 show the modified SVD-POCS algorithm.

Negative Intensity Check. It is possible for the pseudo-inverse solution to contain negative intensities which never occur in reality since negative radiance (or photon levels) never occurs. Negative values occur in the model as a consequence of noise at the CCD or from noise introduced in the pseudo-inverse computation attributed to an ill conditioned system transfer function. The negative values can easily be set to zero by performing:

$$\dot{c}(x, y, \lambda) = P_n c(x, y, \lambda) = \begin{cases} 0, & c_k(x, y) < 0 \\ c_k(x, y), & otherwise \end{cases} \quad \forall k, \quad (40)$$

where $c_k(x, y)$ is the k 'th spectral band spatial solution to the pseudo-inverse. The negative check is applied to the initial pseudo-inverse solution and to the reconstruction after each iteration.

Cold Field Stop Update. Another simple change made to SVD-POCS was the inclusion of a cold field stop constraint for each iteration of the process, including an update before the first iteration. This update takes advantage of information we already know, the spatial extent of the cold field stop. The cold field stop update is

simply:

$$\ddot{c}(x, y, \lambda) = P_{cfs} \dot{c}(x, y, \lambda) = \begin{cases} 0, & (x, y) \in \mathfrak{R}_{cfs} \\ \dot{c}_k(x, y), & otherwise \end{cases} \quad \forall k, \quad (41)$$

where $\dot{c}_k(x, y)$ is the k 'th spectral reconstruction band after the negative intensity check and $\mathfrak{R}_{cfs}$ is the known cold field stop region of the image.

Force Sum. Assuming the only portion of light which enters the aperture of the instrument is that of the desired spectrum and prism dispersion does cause information to fall off the focal plane array, the sum of intensities for each data set for one prism angle is equal to the sum of the intensity of the entire scene. Thus the cumulative intensity of the scene is known. After finding the pseudo-inverse solution and subjecting the negative intensity and cold field stop constraints, the sum of the solution should be equal to the sum of each data set captured at any single prism angle. If it is not, the current solution magnitude is adjusted by dividing by the sum of the current solution and then multiplying by the sum of a data set. It does not matter which data set sum is used, since they are all equal. Mathematically, the operation is simply:

$$\acute{c}(x, y, \lambda) = P_s \ddot{c}(x, y, \lambda) = \ddot{c}(x, y, \lambda) * \frac{\sum \sum d_i(m, n)}{\sum \sum \sum \ddot{c}(x, y, \lambda)}, \quad (42)$$

where $\acute{c}(x, y, \lambda)$ is either the pseudo-inverse solution or the current iteration solution of the modified SVD-POCS and $d_i(m, n)$ is the i 'th recorded data set. The additional constraints added prior to the force sum prevent energy from being placed into the cold field stop or being put into negative intensities.

Include the Mean in the ODC. In SVD-POCS, the mean of each spectral solution is removed before the object domain constraint is performed. Thus, the mean removed image data is projected onto the L largest eigenspectra and no information about the mean enters the iteration. The consequence of the mean removal

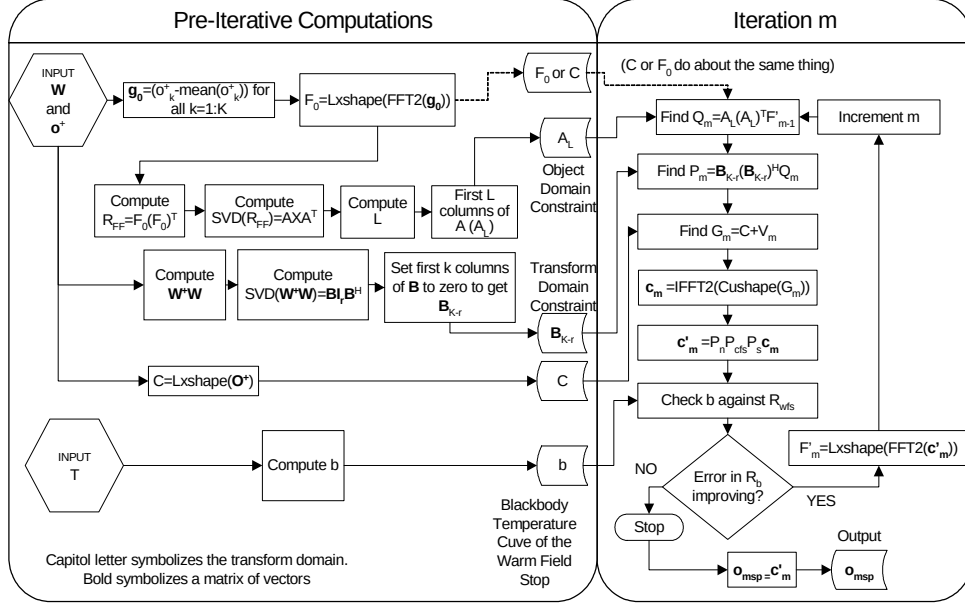


Figure 33 MSP is SVD-POCS with additional constraints. The area corresponding to the warm field stop, R_{wfs} , is compared to the blackbody curve, b , of the temperature of the warm field stop. The atmospheric absorption factor is not needed since the warm field stop has negligible attenuation.

is the SVD-POCS solution cannot change the mean of the pseudo-inverse solution. In order to better estimate the mean of the scene, the mean of each iteration solution is left in the object domain projection. The projection is once again:

$$\mathbf{Q}_m = \mathbf{A}_L(\mathbf{A}_L)^T \mathbf{F}'_{m-1}, \quad (43)$$

where $\mathbf{F}'_{m-1}$ includes projections $\mathbf{P}_{cls}$, $\mathbf{P}_n$, and $\mathbf{P}_s$ and the mean of the current solution. The mean may be removed before the first iteration, but the outcome will be the same, just one iteration later. The solution starts to track the mean of the source after it is first included and converges over the iterative sequence provided the mean is included each iteration.

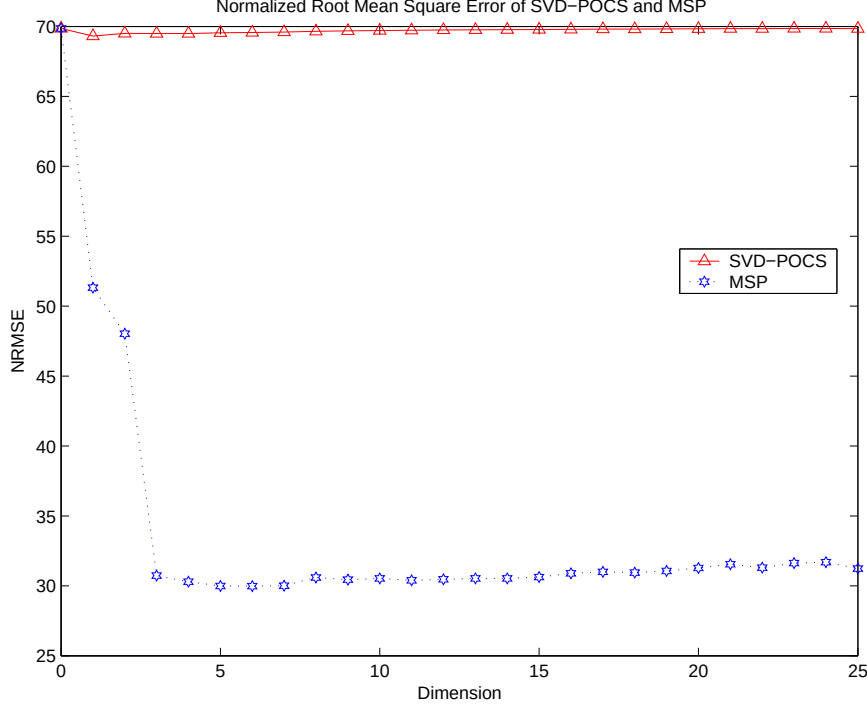


Figure 34 The NRMSE of SVD-POCS changes very little as L increases. The NRMSE decreases significantly for $L=3$ and higher.

3.5 Performance Comparison

In Section 3.3, the configuration of SVD-POCS in the Brodzik paper [9] was shown to not be a good match for a mean square error metric. Fortunately, adding extra constraints enables the algorithm to track the mean of the scene at a small cost of variance error, which also shows in the mean removed error results. Figure 34 shows how the NRMSE changes as a function of the dimension L . Once again, SVD-POCS is nearly invariant, while MSP drops significantly at $L=3$, and stays much lower than the pseudo-inverse NRMSE. Results displayed in Figures 34 and 35 show that $L=3$ is the best dimension for MSP when the source is the temperature map scene. The NMRE of MSP, seen in Figure 35 approaches that of SVD-POCS at $L=3$, but yet remains larger.

The NVE of SVD-POCS is smaller on average than MSP as shown in Figure 39. A bias variance trade-off is apparent. SVD-POCS does better at minimizing NVE

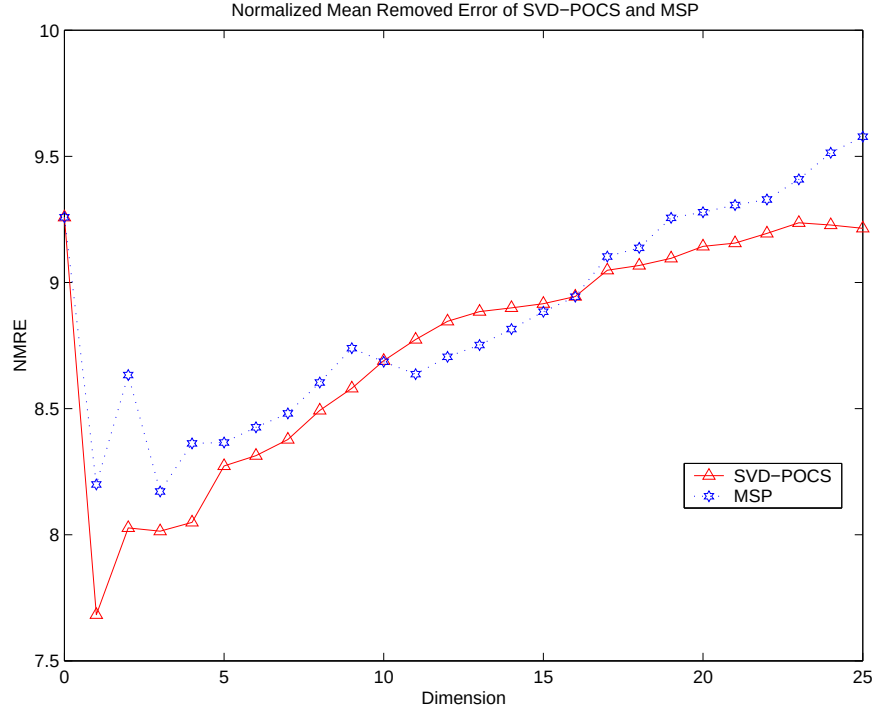


Figure 35 The NMRE gets worse as dimension increases for both SVD-POCS and MSP.

Table 5 Performance of Iterative Methods.

	NRMSE	NVE	NMRE
Pseudo-inverse	69.84	211.04	9.49
SVD-POCS	69.30	156.25	7.87
MSP	30.89	175.92	8.32

and NMRE while MSP does a better job at minimizing NRMSE. Both algorithms did equally poor in at reconstructing bands with little or no signal energy. The final NRMSE and mean of the points shown in Figure 39 are shown in Table 5. SVD-POCS had the lowest NMRE while MSP had the lowest NRMSE.

The trend continues for error as a function of iteration as shown in Figures 40 and 41. For the particular trial of MSP shown here, the mean is removed in the first iteration only, which is why the NRMSE does not drop until after the 2nd iteration. MSP converges within about 15 iterations for the temperature map scene of Figure 7 for both NMRE and NRMSE metrics, while SVD-POCS takes about 10 iterations,

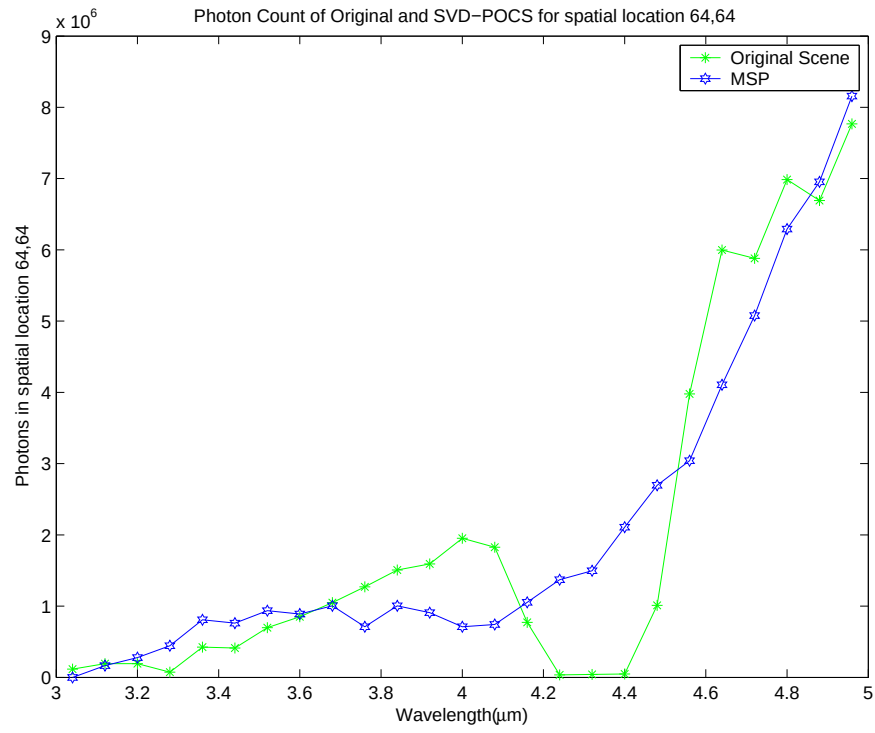


Figure 36 The reconstruction at spatial location 64,64 is closer to the original for MSP than SVD-POCS.

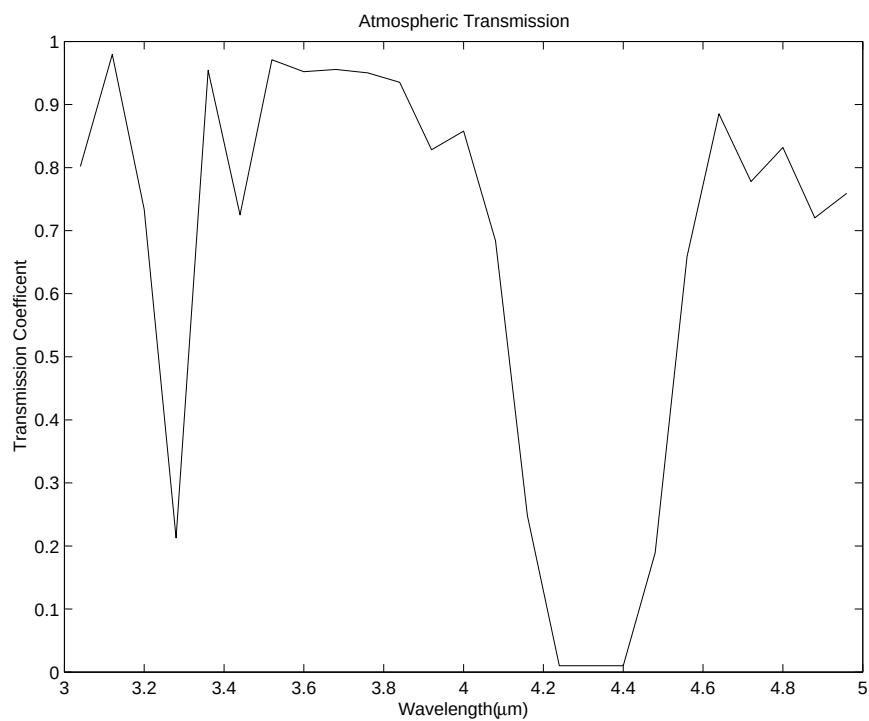


Figure 37 Atmospheric Transmission Curve

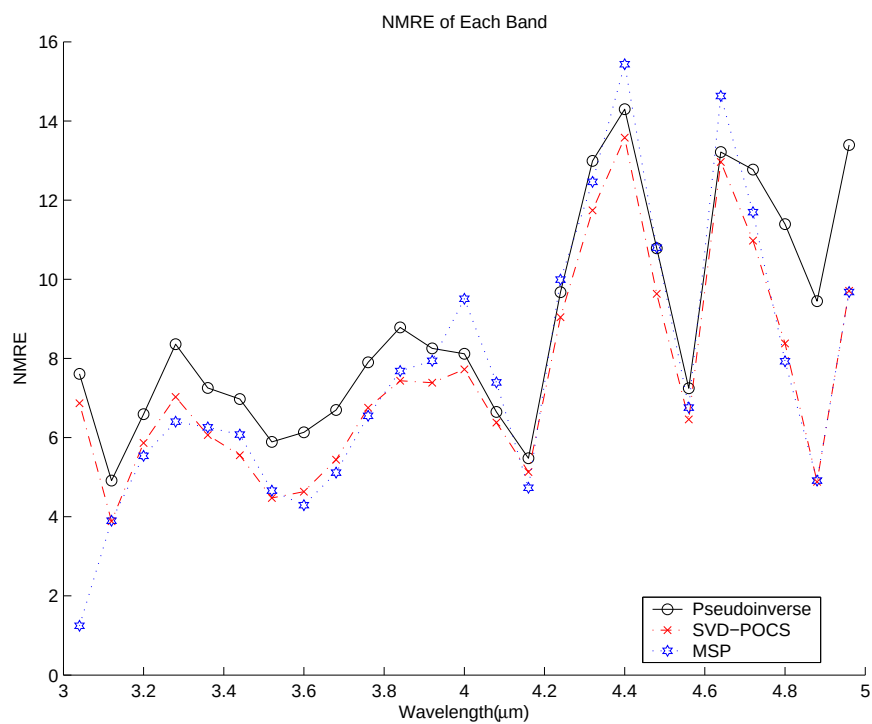


Figure 38 The NMRE of SVD-POCS is lower on average than that of MSP.

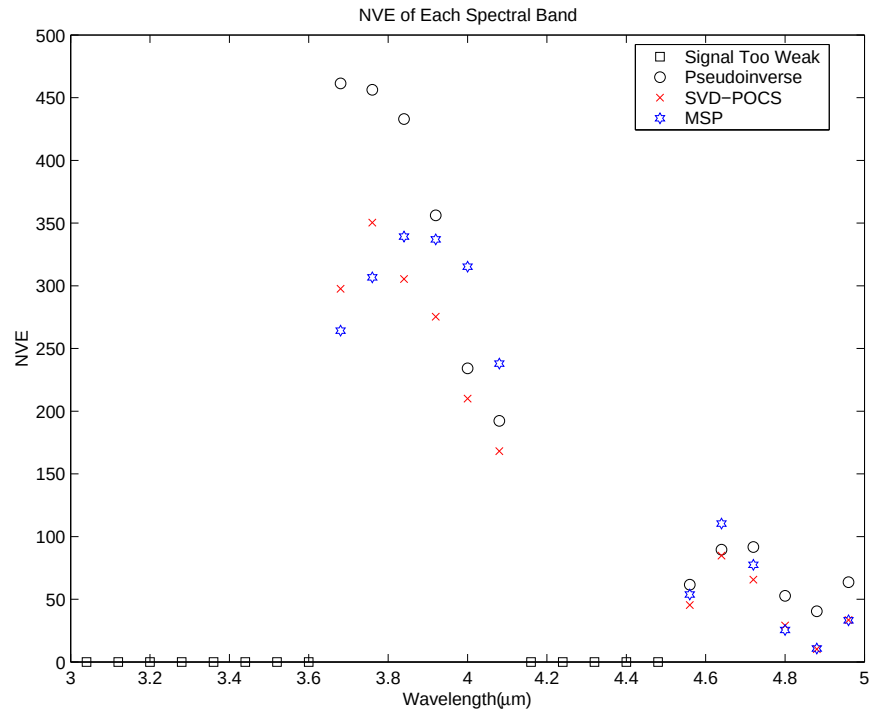


Figure 39 NVE for each spectral reconstruction. The NVE of SVD-POCS is lower on average than that of MSP.

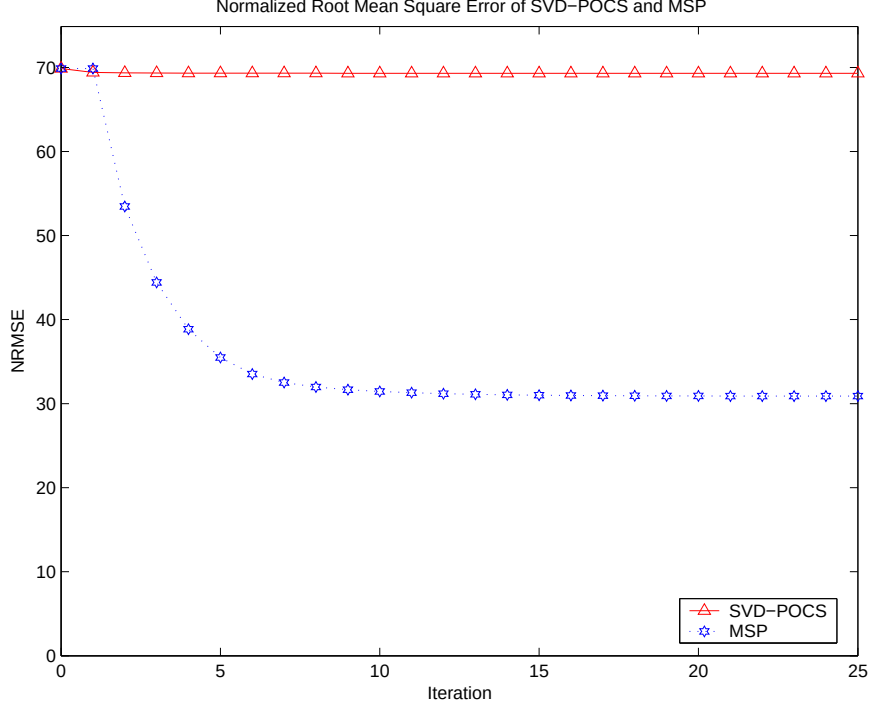


Figure 40 The NRMSE of SVD-POCS does not decrease while MSP reduces the NRMSE, converging in about 15 iterations.

with progress only seen in NMRE reduction. Figure 42 shows the NRMSE of each band for the entire $3.0\text{-}5.0\text{ }\mu\text{m}$ spectral frequency range. The NRMSE curve of MSP has a similar trend as the curve of SVD-POCS, but occurs at a lower value. The smallest NRMSE values of SVD-POCS occur in the middle of the spectral frequency range which is where the mean of the source crosses the mean of the reconstruction. Thus, SVD-POCS simply gets lucky at that point in NRMSE calculations. MSP actually tracks the mean, and provides a useful NRMSE result.

MSP can track the mean of the source while SVD-POCS cannot. The mean square error metric is now useful under MSP and can be used for iteration error monitoring. SVD-POCS has lower NVE than MSP, but provides no information on NRMSE. The mean tracking and variance tracking trade-off once again is obvious. The variance error between spectral bands doesn't tell us how intensities relate from band to band, but how close the pixels vary compared to the original.

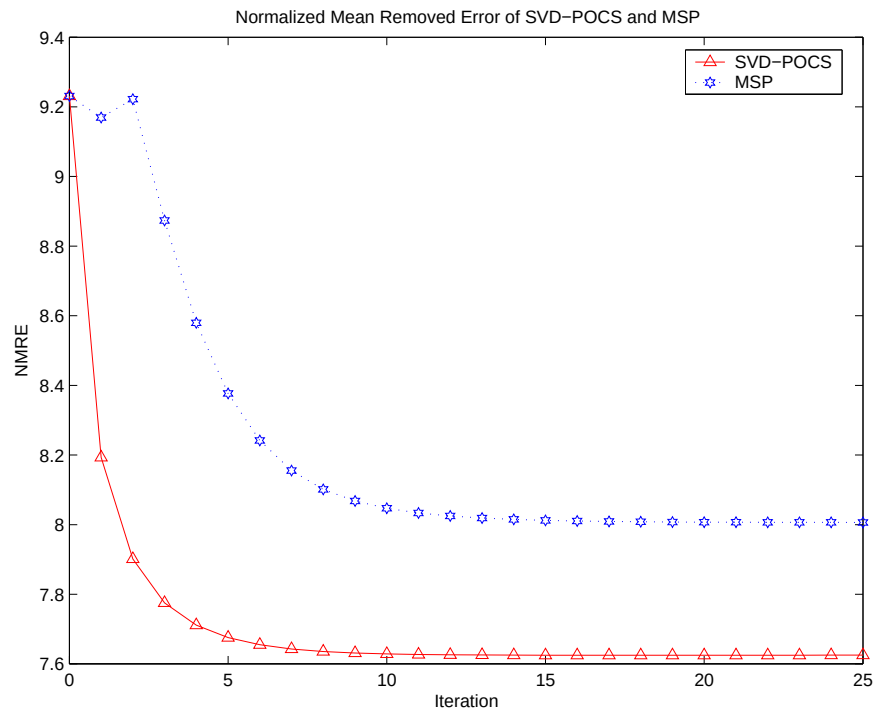


Figure 41 The NMRE decreases each iteration for both SVD-POCS and MSP, settling after 10 and 15 iterations respectively.

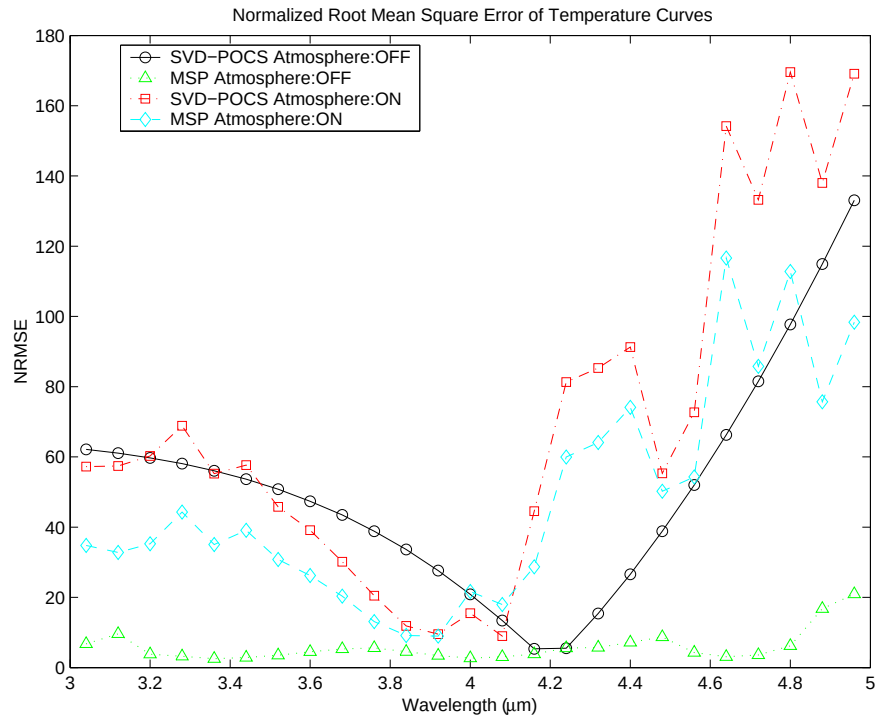


Figure 42 The NRMSE curve of MSP has a similar trend as the curve of SVD-POCS, but is lower overall. Atmospheric attenuation causes NRMSE to rise.

Error vs Temperature. The temperature map scene, shown in Figure 7, was used to test reconstruction error versus temperature by finding the error for each spatial location and grouping the spatial locations by temperature to find the mean error per temperature. The range of temperatures in the scene is limited from 290.5-300.7K. Despite a limited temperature range, results do show a trend. Figure 43 shows a flat error where atmospheric absorption was not used and a slightly increasing response with error variance mostly towards the upper and lower bounds with atmospheric absorption. The error variance is likely caused by the reduced number of spatial temperatures having the extreme temperatures of the end. Results in Figure 43 are for all pixels of the scene, for one noise trial. To get a broader idea of error versus temperature, the values of the map were scaled to range from 291-338K. The value of each spatial temperature of the 2D map was rounded to the nearest integer and once again all values spatial locations were used to obtain an average error for each temperature. Results are shown in Figure 44. The error curves for SVD-POCS and MSP look very similar in both plots, but occur at a different mean. MSP has a much lower NRMSE, reflected by its ability to track the mean of the scene while SVD-POCS cannot. The increase in error as temperature increases is caused by the relation of the plank curve to the atmosphere attenuation curve.

The temperature error curves appear at an overall higher value than those for the mean NRMSE of all bands when atmospheric absorption is on, and at a lower value without atmospheric absorption. This means that atmospheric absorption induces additional error that impacts temperature reconstruction more than it does the reconstruction mean of each band. Absolute radiometry, although improved with MSP, may still be impractical due to increased temperature reconstruction error. The atmospheric attenuation appears to add additional spectral frequency which the algorithms are have difficulty reconstructing.

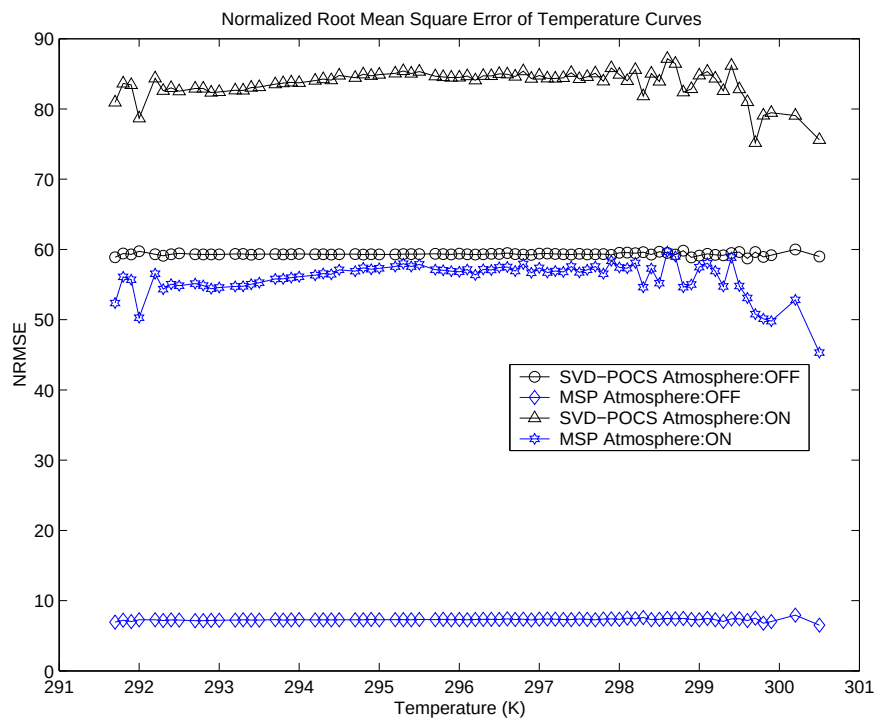


Figure 43 NRMSE Versus Temperature for 290.5-300.7K.

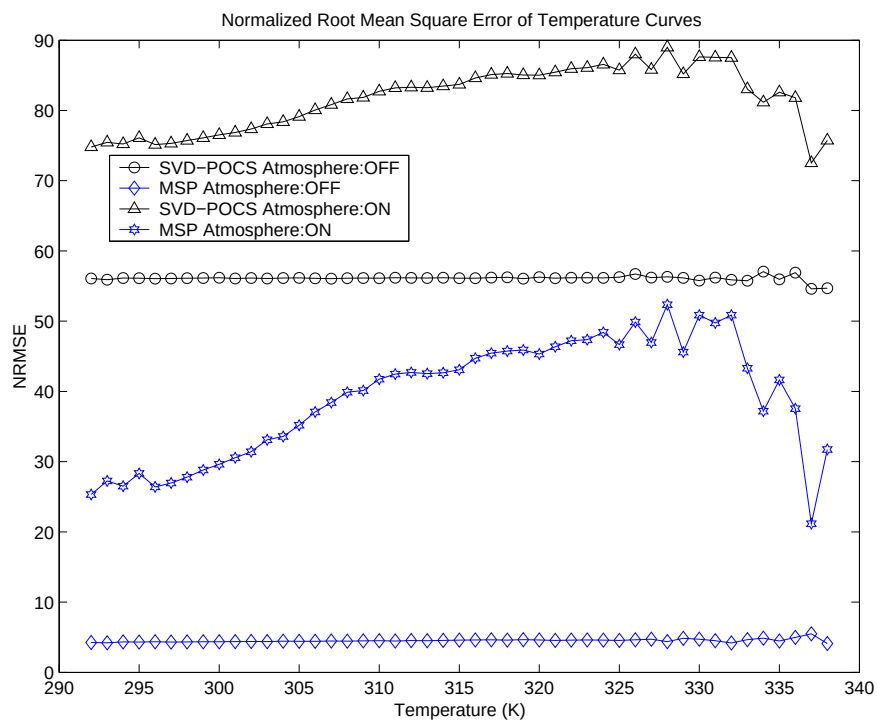


Figure 44 NRMSE Versus Temperature for 291-338K.

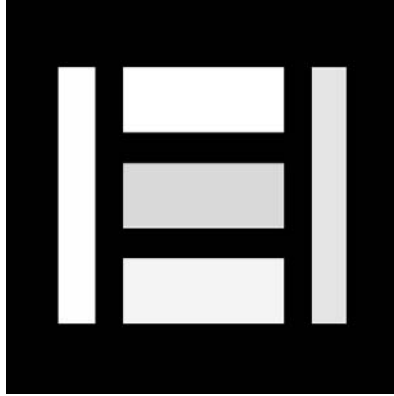


Figure 45 A broadband view the the five bars.

3.6 Monochromatic Source Test

The objective of this test is to see if the algorithm can identify which spectral band each source belongs to. Each source has a spectral response in only one spectral band. The remaining scene has no spectral response for any band in the region. Figure 45 shows a broadband representation of the object cube looking down the spectral frequency axis. You can clearly see the five bars, each at an equal intensity. Table 6 shows the band and frequency location of each bar. Each bar is located in an atmospheric passband. Figure 46 shows the NVE plot of the reconstructed scene. Notice the low NVE values passing the threshold criteria the correct corresponding bands. Figure 47 shows the reconstruction image of bands five, six, and seven using identical image display scaling. Ghost bars do appear in adjacent spectral bands where no information should be present. Thus, there is spectral bleeding between spectra that will effect spectral resolution. The bleeding gets stronger as the image gets closer to that particular source spectral location.

The five monochromatic test shows that spectral interference exists between neighboring bands in the reconstruction. The amount of intensity leaked into immediate adjacent bands is on the order of 50% of the reconstruction in the correct band. The spectral lineshape for a 25 band reconstruction is shown in Figure 48. The cross band intensity declines significantly; becoming insignificant after six adjacent bands.

Table 6 Location in Frequency of Each Bar.

Bar	Band	Frequency (μm)
Center	2	3.08-3.16
Right	5	3.32-3.40
Bottom Center	9	3.64-3.72
Left	13	3.96-4.04
Top Center	21	4.60-4.68

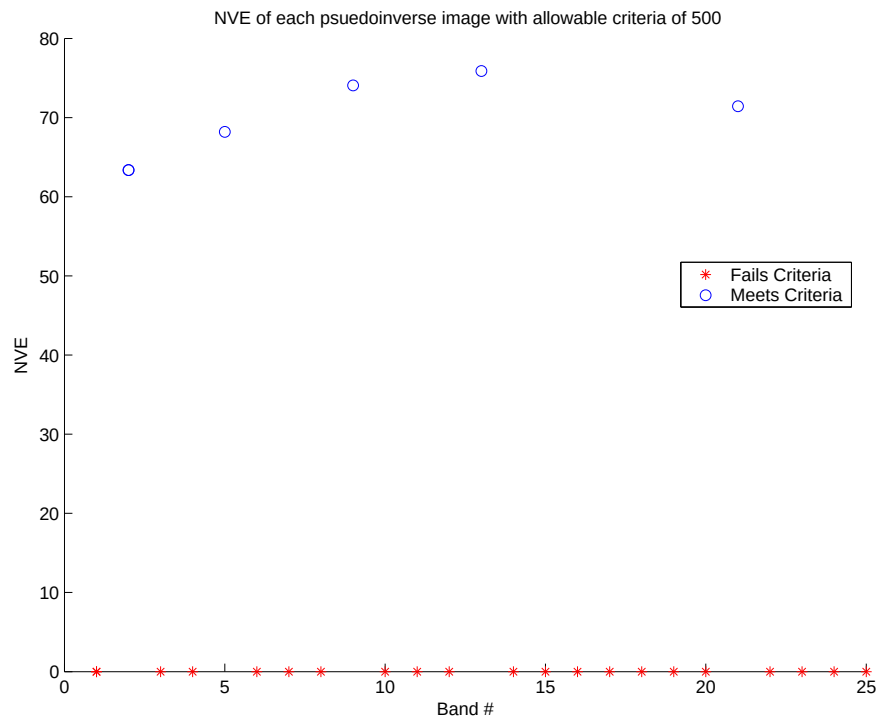


Figure 46 The bars show up in the appropriate both visually and according to NVE as shown here.



Figure 47 Bands 5, 6, and 7 are shown here from left to right. Bleeding from sources located in nearby frequency bands is shown in each image. The bleeding gets stronger as the image gets closer to that particular source location.

Spectral lineshape width is a function of the number of reconstructed bands and not a function of wavelength. The spectral bandwidth of each band establishes the bandwidth range of the bleeding. The lineshape of a reconstruction with 25 bins of 80 nm in width will be twice the size as a reconstruction with bins of 40 nm in width because of the greater bin width. Figure 49 shows how wide the lines are with 43 wavelength bins (representing a bandwidth of 46.5 nm). The lineshape continues to span six adjacent bins on each side, but the contribution is much less severe due to the increased number of bins, making the spectral interference range much less and the lineshape more narrow.

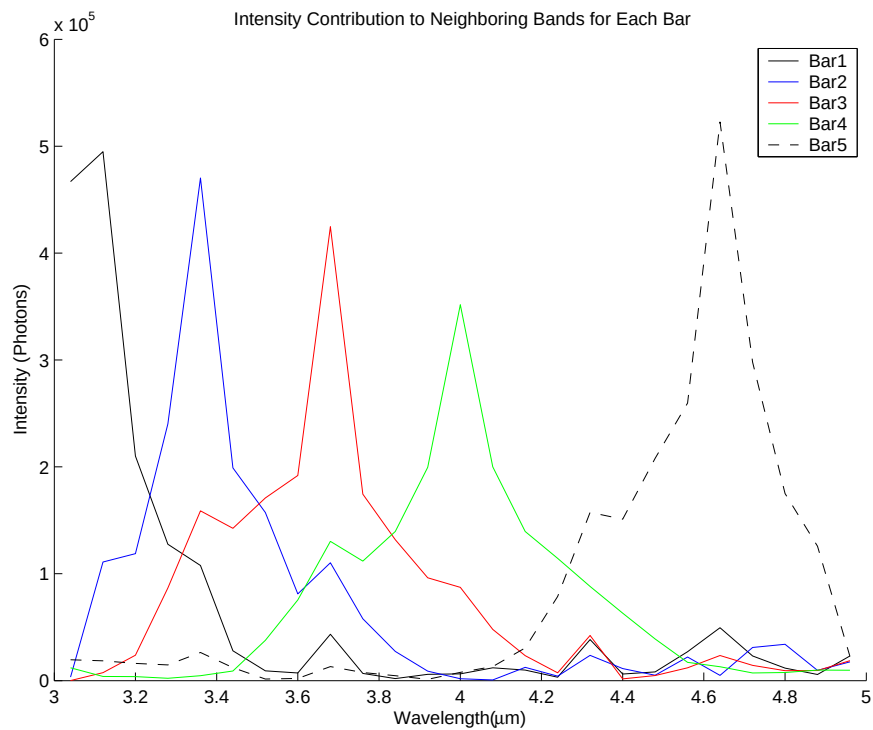


Figure 48 Spectral Lineshape for 25 Bands.

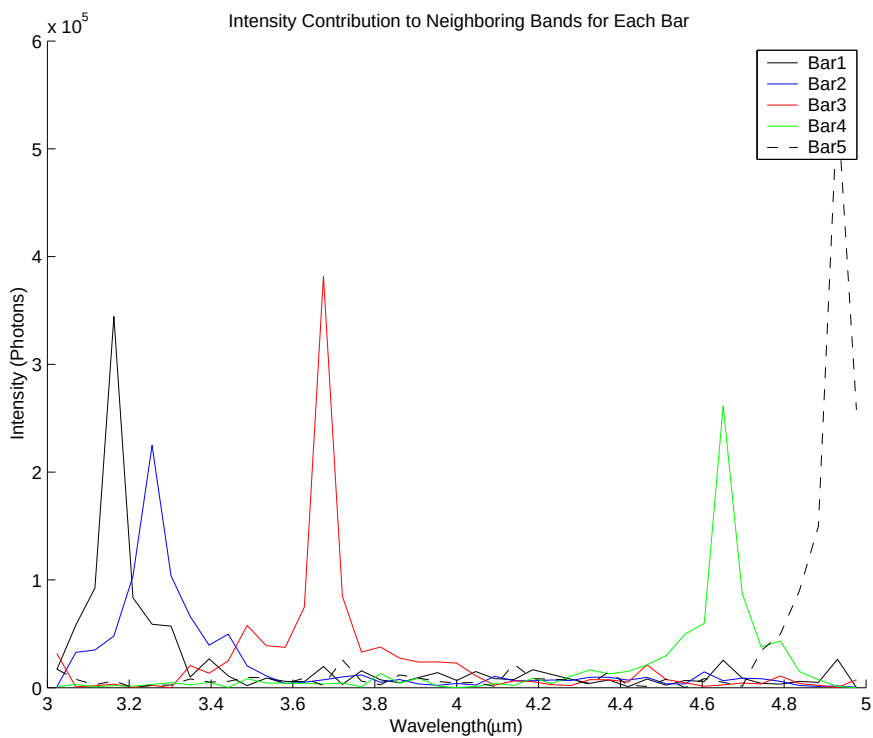


Figure 49 Spectral Lineshape with 43 Bands.

IV. Iteration Monitoring

4.1 The Warm Field Stop

In this chapter, a design change of the hardware configuration is introduced to allow for a built in iteration performance monitor. The improvement idea requires a change in the design of the imager, as shown in Figure 50. The chromotomographic imager of Mooney has a cold field stop in the focusing optics. The purpose of the stop is to limit stray photons from outside the field of view, induce high spatial frequency in the scene, and to provide a dark area on the CCD for image edge dispersion. The stop is cooled to a value where it has almost no spectral response and can therefore be simulated in MATLAB as intensity zero in modeling of the pseudo-inverse and SVD-POCS algorithms. The addition is a new set of optics in front of the original design which includes an uncooled field stop. It is easy to obtain the temperature at the “warm” field stop and, since we are working in the infrared, calculate the blackbody curve the field stop should have. A small section of the warm field stop is left unblocked by the cold field stop and therefore registers on the CCD as part of the scene. The warm field stop will have a known spectrum if the temperature is known which can be used to check error in reconstruction. The idea is to use this additional known information to stop the iterations of MSP once a certain criteria has been accomplished. Three possible stopping criteria exist 1) the error is worse than the pseudo-inverse reconstruction, 2) the error improvement has stopped, and 3) error starts to diverge.

Error Calculation in the Warm Field Stop. We have found that the NRMSE can be misleading but when combined with NVE and NMRE, a better estimate of reconstruction performance can be found. The known warm field stop is the same value for each spatial location and therefore has a variance of zero. Therefore, NVE as we have come to know it thus far cannot be obtained. However, NMRE can be

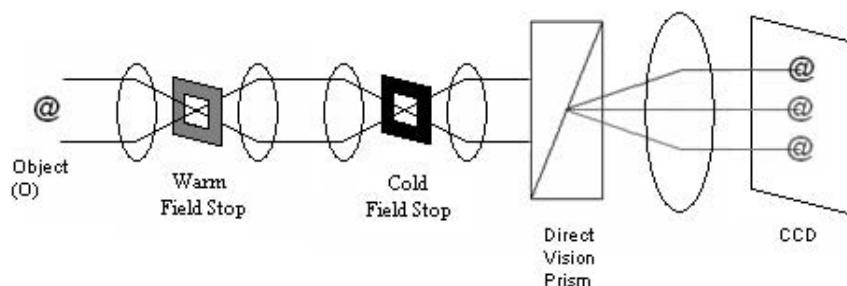


Figure 50 The new imager has a warm field stop located forward of the cold field stop. The warm field stop has a smaller opening than the cold fieldstop.

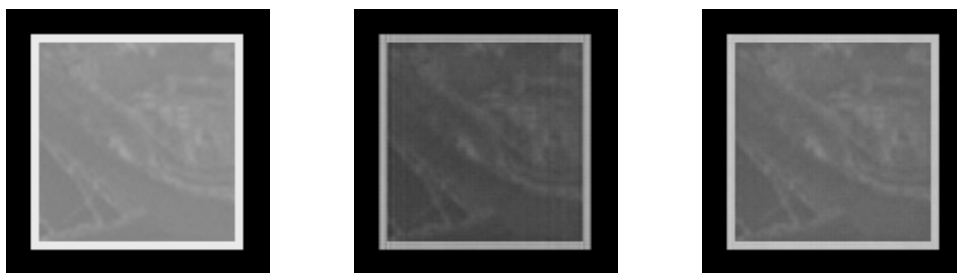


Figure 51 The pseudoinverse solution, b), to the original object, a), is a good estimate of the original scene. The MSP solution, c), looks like the scene, but yet has a smaller mean causing it to appear darker. The bright border is due to the warm field stop which is located inside the instrument and has negligible atmosphere absorption. This field stop is at 300K.

Table 7 Performance of Iterative Methods Including the Warm Field Stop

	NRMSE	NMRE
Pseudo-inverse	69.84	9.49
SVD-POCS	69.32	7.87
MSP	30.95	8.16
MSP-WarmFS	37.12	7.50

obtained and to show trends where the mean is ignored; providing a result which is very close to the variance error measurement.

4.2 Performance with the Warm Field Stop

For the warm field stop addition to be useful, it must not cause any problems with the MSP algorithm. To determine the effects caused by the warm field stop, five identical trials were conducted for each reconstruction type; pseudo-inverse, SVD-POCS, MSP, and MSP with the warm field stop for a temperature of 300K. NRMSE performance is shown for each in Figure 53. The pseudo-inverse and SVD-POCS NRMSE curves are close, confirming SVD-POCS cannot reduce mean square error. MSP reduces the NRMSE with and without the presence of the warm field stop, although not identically. Figure 54 shows NMRE performance for all bands. The NMRE curves are very close, but SVD-POCS has a smaller average NMRE. Surprisingly, MSP with the warm field stop had an even smaller mean error which is probably due to the small scene extent reduction from introducing the warm field stop. Figure 55 shows that NVE is smallest in SVD-POCS and the MSP values with and without the warm field stop are close. The warm field stop does not induce odd reconstruction behavior which would limit the effectiveness of use as an error monitor during the iterative reconstruction process.

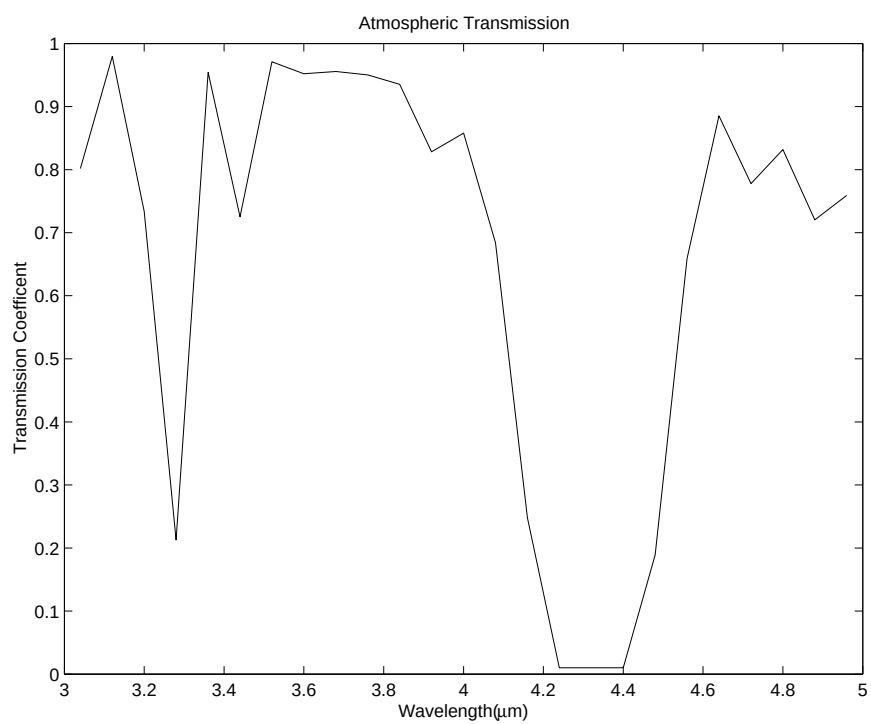


Figure 52 Atmospheric Transmission Curve

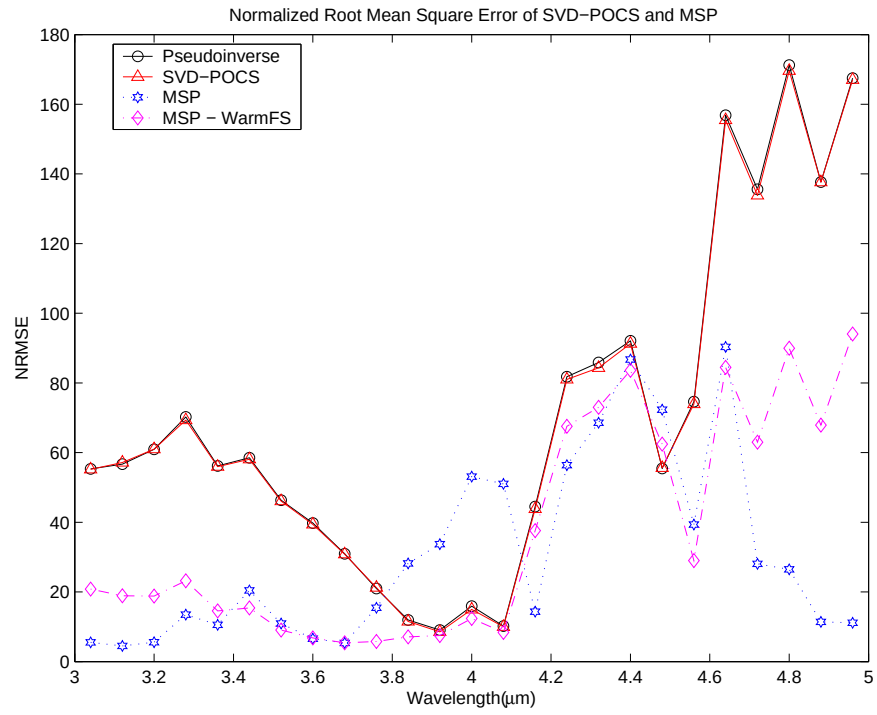


Figure 53 The NRMSE trends with and without the warm field stop are generally the same.

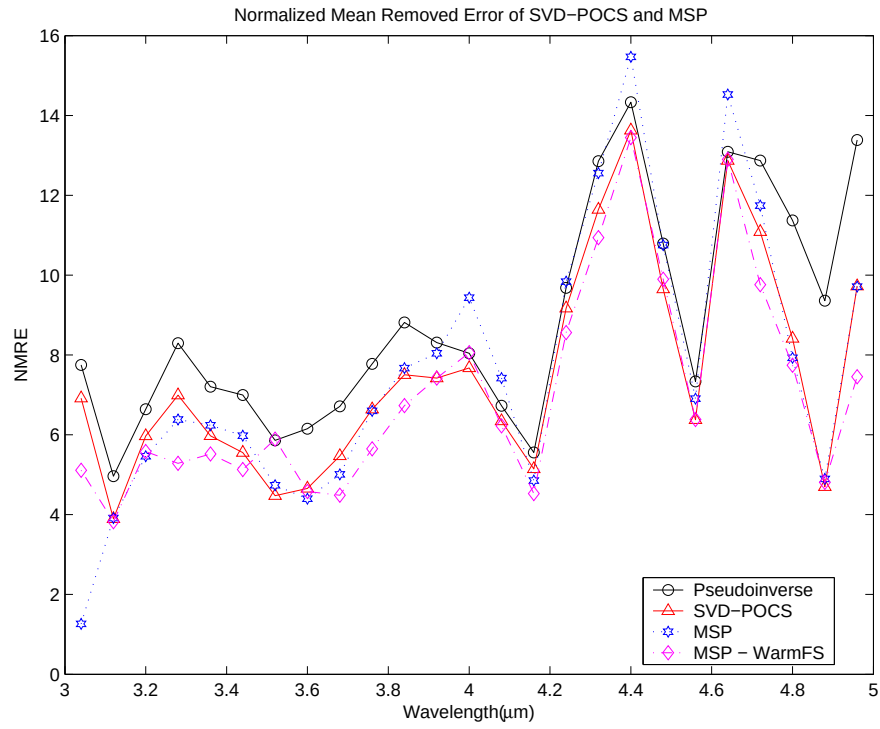


Figure 54 The NMRE of SVD-POCS is smaller on average than MSP. The warm fieldstop reduces the NMRE for a temperature of 300K

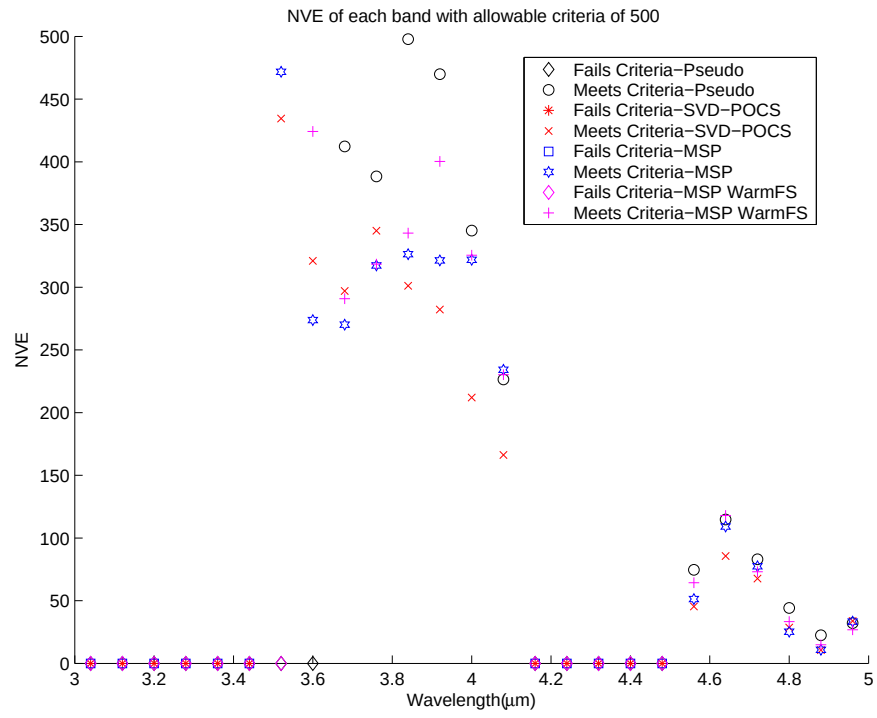


Figure 55 The NVE of SVD-POCS is smaller on average than MSP. The warm fieldstop does not change MSP reconstruction performance.

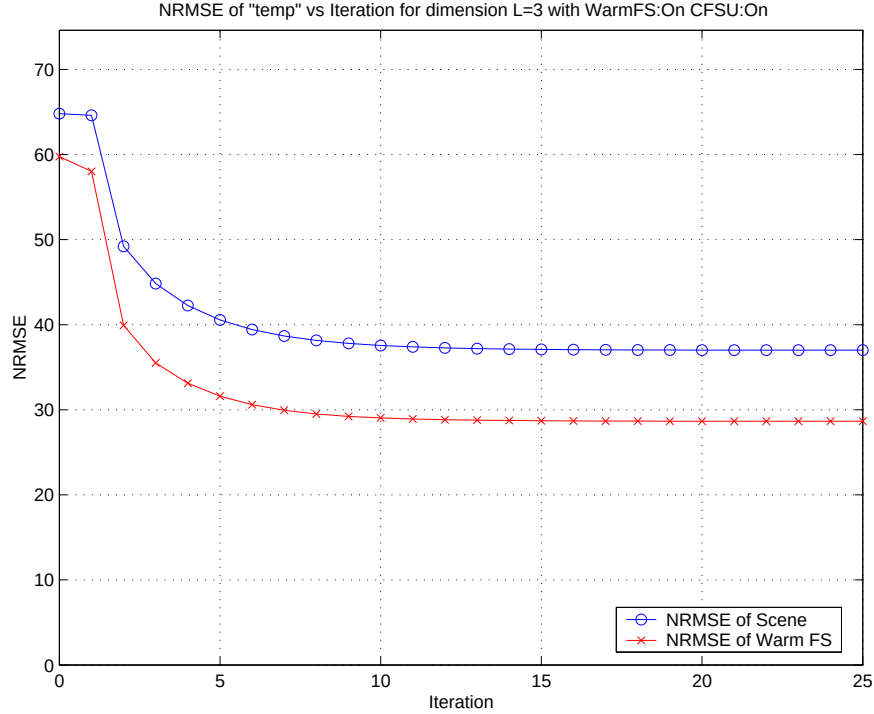


Figure 56 The NRMSE of the warm field stop follows the same trend as the NRMSE of the image as iterations occur for MSP for a warm field stop temperature of 300K.

4.3 Performance in the Warm Field Stop

The error in the warm field stop needs to follow the same trend as error in the entire scene for the warm field stop to be useful as an iteration monitoring tool. The NRMSE of the warm field stop was calculated as explained in Section 4.1 and displayed on the same plot as the NRMSE of the entire scene. Figure 56 shows that the error in the warm field stop indeed follows the same improvement relationship for each iteration of MSP. Both curves drop significantly after the first iteration, and are nearly converged by the 13th iteration. In an actual system, you would only have access to the warm field stop curve which here tell us to stop iterating after the 13th iteration. Stopping the iterative sequence at iteration 14 would lead to a solution with nearly all error reduction possible for MSP.

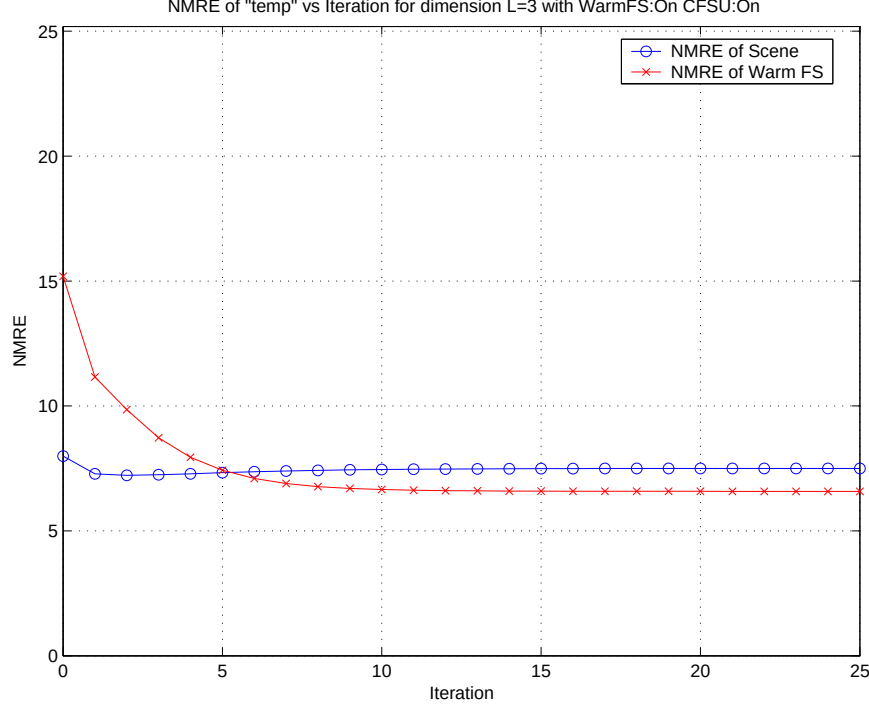


Figure 57 The NMRE of the warm field stop also decreases, but does not follow the exact curve for MSP for a warm field temperature of 300K.

The mean removed error relationship between the warm field stop region and the rest of the image does not match as does the NRMSE. The error reduction trend of the warm field stop is shown in Figure 57 along with the NMRE of the scene as a function of iteration. The warm field stop curve drops for each iteration and converges at nearly the same point, but not at the minimum of the scene NMRE curve. The convergence properties are different for mean removed error than they are for mean square error. NMRE would not be a good metric to monitor error reduction.

Figure 58 shows the NRMSE relationship when the temperature in the warm field stop is much hotter than the scene average. Here the warm field stop temperature is 500K while the mean of the scene is about 295K. The MSP algorithm converges after about 15 iterations for the warm field stop curve. The NRMSE curve of the non-field stop scene is smaller than the warm field stop curve due to the non-

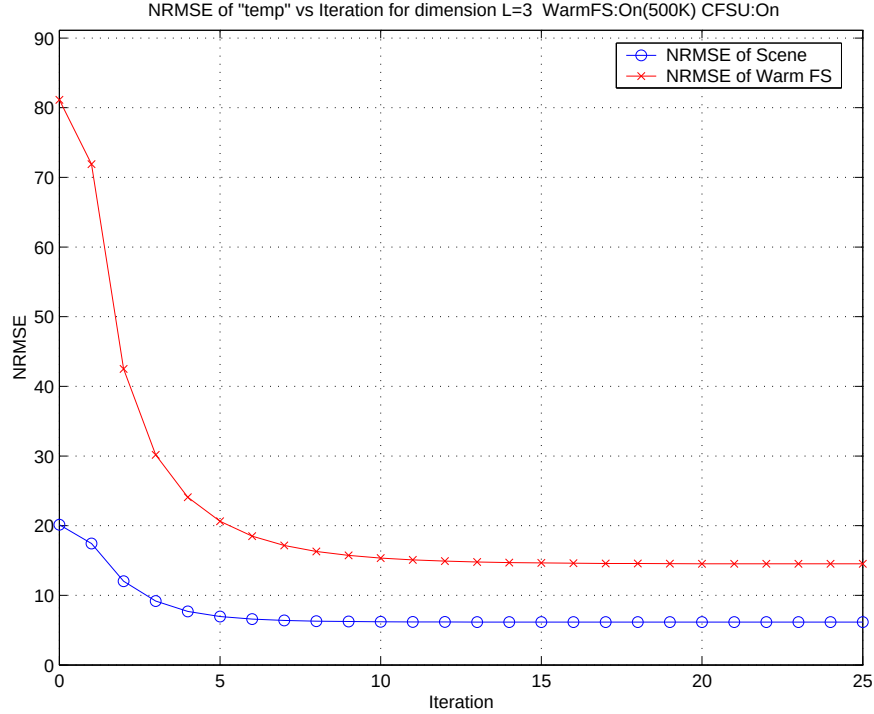


Figure 58 Even with a temperature of 500K, the warm fieldstop can still be used to find where the mean square error is minimized. The NRMSE curve of the non field stop scene is smaller than the warm field stop curve due to the normalization of the entire object cube, which has a much larger mean than the scene itself due to the 500K warm field stop.

malization of the entire object cube, which has a much larger mean than the scene itself due to the 500K warm field stop. Figure 59 shows the curves for a field stop temperature of 100K. Once again, the error in the warm field stop can be used to estimate where MSP is no longer improving reconstruction.

From the three different warm field stop temperature trials, we can see that the temperature of the warm field stop does not have to be at the mean temperature of the scene or close to it. It just have to provide enough signal to do mean square error analysis. The curves for all three temperatures would be adequate in finding when convergence has been reached, or whether the scene was not reconstructing. The objective of absolute radiometry is to measure how the spectra of the reconstruction lines up with the original source. This measurement fits a mean square error metric,

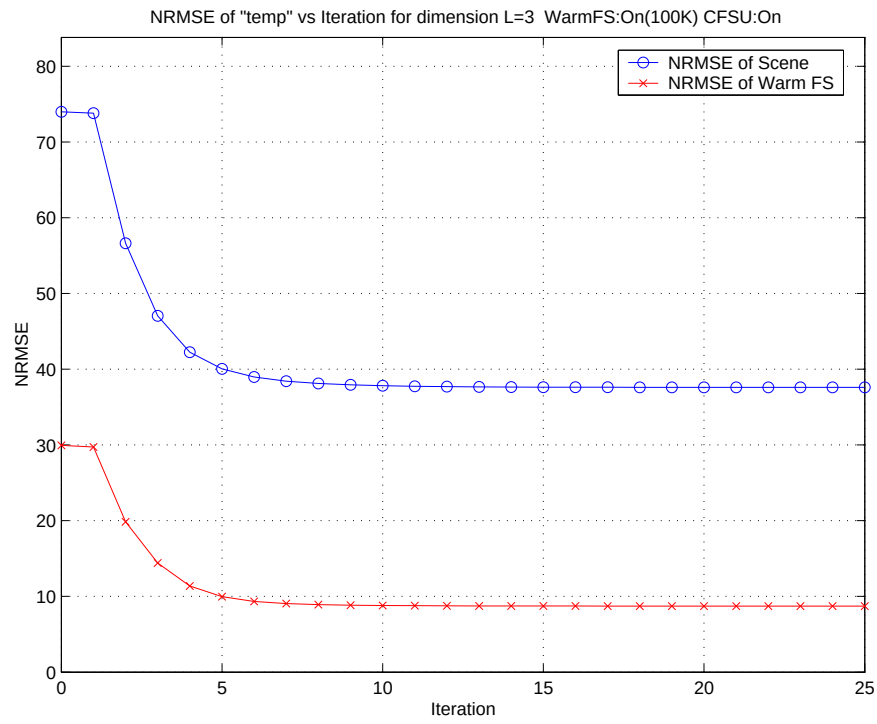


Figure 59 At 100K, the warm fieldstop continues to be useful in finding where the improvement from MSP has ended.

not a normalized variance metric or mean removed metric. Thus NRMSE is the metric to use for warm field stop monitoring. The warm field stop can be used as a tool to know when to stop the iterative sequence of MSP using a mean square error metric such as NRMSE without having a stringent requirement on warm field stop temperature.

Table 8 Performance of Iterative Methods With a Fireball

	NRMSE	NMRE
Pseudo-inverse	62.80	42.62
SVD-POCS	61.50	41.38
MSP	58.52	46.43
MSP-WarmFS	63.39	48.31

4.4 Fireball in the Field of View

Additional scenes test reconstruction performance when a fireball appears in the field of view. The scenes contain a 1000K fireball at the center of the standard temperature map scene. Reconstruction results for a scene with a 67 pixel diameter fireball of a 103x103 non-cold field stop scene was first used to compare NRMSE, NMRE, and NVE. NRMSE versus wavelength trends are very similar for each reconstruction technique. Mean error values are shown in Table 8 where the NRMSE value is the average for each band, and the NMRE value is the average bands with adequate signal..

Figure 60 shows NRMSE for SVD-POCS and MSP is about the same. The mean values in Table 8 also show the same relationship. The mean removed error has a larger percent difference, with MSP doing the worst. Less error for this scene is due to the mean, meaning more is due to the variation between pixels. The large area of the fireball results in an extensive portion of the scene having a very large magnitude. Thus difference between pixels will be larger. The differences are caused by error introduced by the CCD and by the singular matrix inversion of the pseudo-inverse solution. NMRE versus wavelength is shown in Figure 61. Each NMRE curve increased compared to the non-fireball NMRE result shown in Figure 54. Both NVE and NMRE follow the same trends versus wavelength, but NVE values are still relatively low compared to NMRE. Different normalization factors for each metric cause this relationship. Relative variance error is not so bad, but the error in magnitude is, when a large fireball of 1000K is in the scene. Figure 62 shows the variance error for passing bands. The average variance must be quite

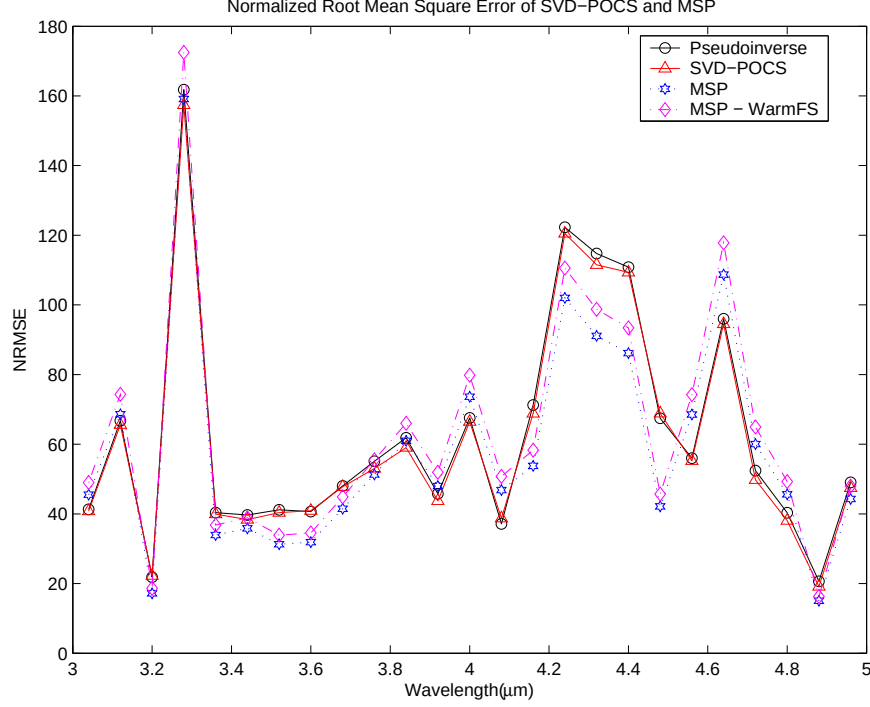


Figure 60 NRMSE for a scene with a 1000K fireball on the temperature map.

large for the entire cube compared to the average variance error, while the variation error causes large error in photon levels, but is normalized by a relatively less intense mean photon amount.

The warm field stop at 300K is able to track mean square error reduction in the scene as shown in Figure 63. However, Figure 64 shows the NMRE of the scene is not directly related to NMRE of the warm field stop. NMRE does not provide enough information for halting the iterative sequence of MSP.

The fireball size was varied to establish trends in reconstruction performance versus fireball size. Three fireball diameters are used; 6 pixels, 32 pixels, and 67 pixels. The atmospheric absorption curve is shown again in Figure 65 to provide a quick reference. The 6 pixel fireball has the least magnitude. The MSP reconstruction curve, shown in Figure 66, is not capable of matching the mean of the scene. Peaks and valleys of the source mean are cutoff. The intensity of the construc-

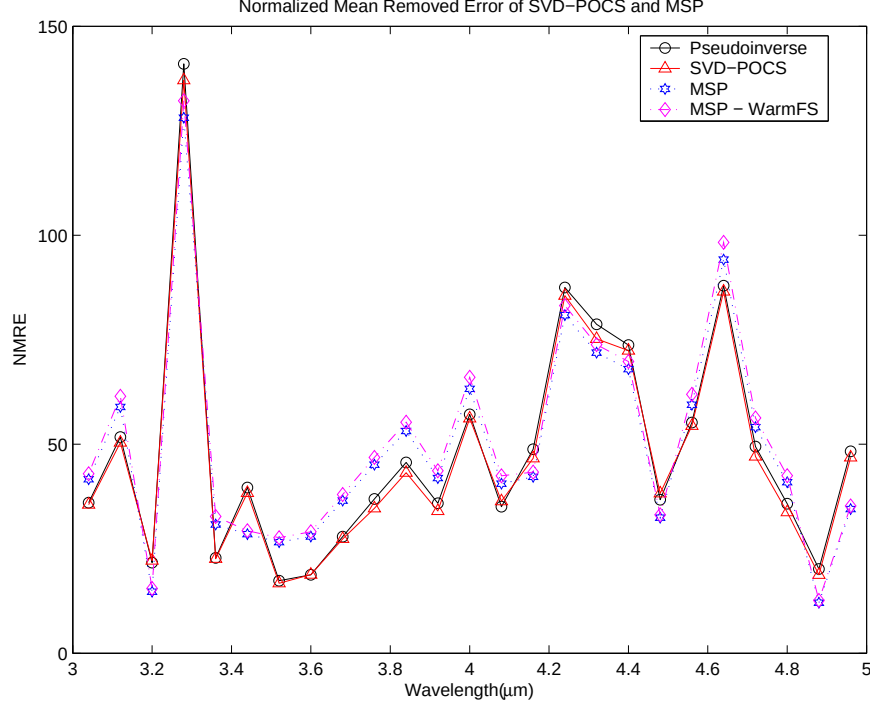


Figure 61 NMRE for a scene with a 1000K fireball on the temperature map.

tion does not have a dynamic photon range as does the source. The mean of the reconstructed scene follows the atmospheric absorption curve.

The same trends apply for the larger fireballs. The source mean intensity curve matches the atmospheric absorption curve better as fireball size increases. The reconstruction mean photon amount curve becomes smoother, appearing as a sinusoidal looking curve in the 67 pixel fireball scene. Mean reconstruction performance for the 32 pixel fireball is shown in Figure 67 and in Figure 68 for 67 pixels.

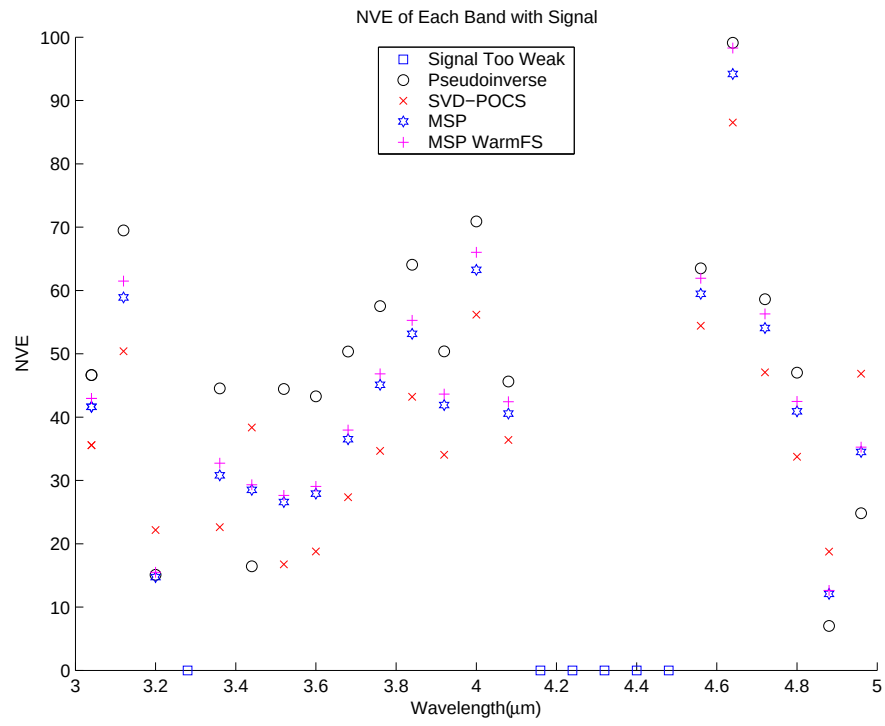


Figure 62 NVE of the fireball scene. SVD-POCS has a lower overall NVE. The NVE of both MSP trials looks to be nearly identical.

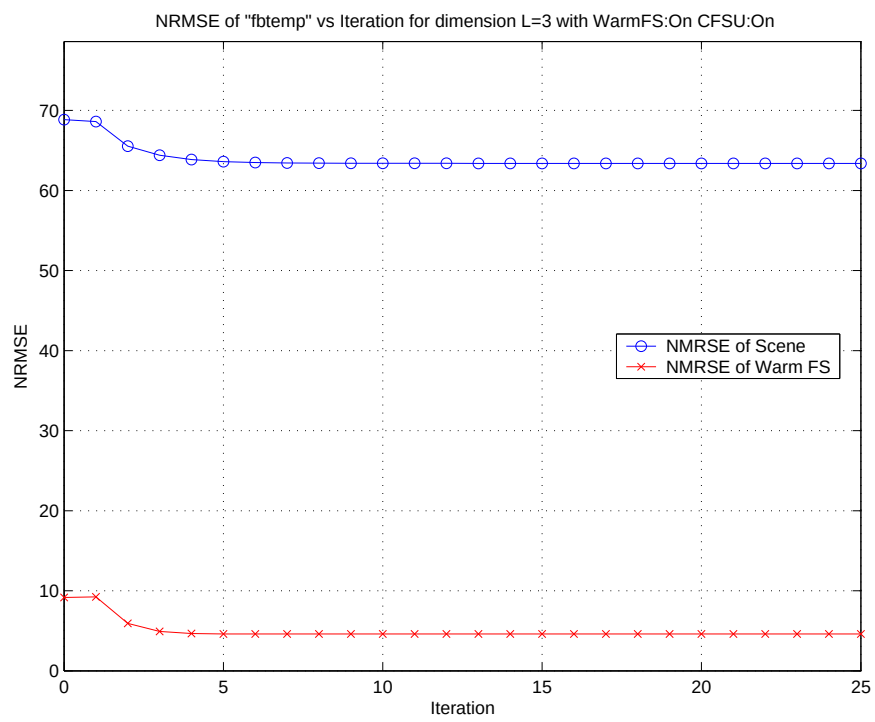


Figure 63 NRMSE Versus Iteration for the Fireball Scene

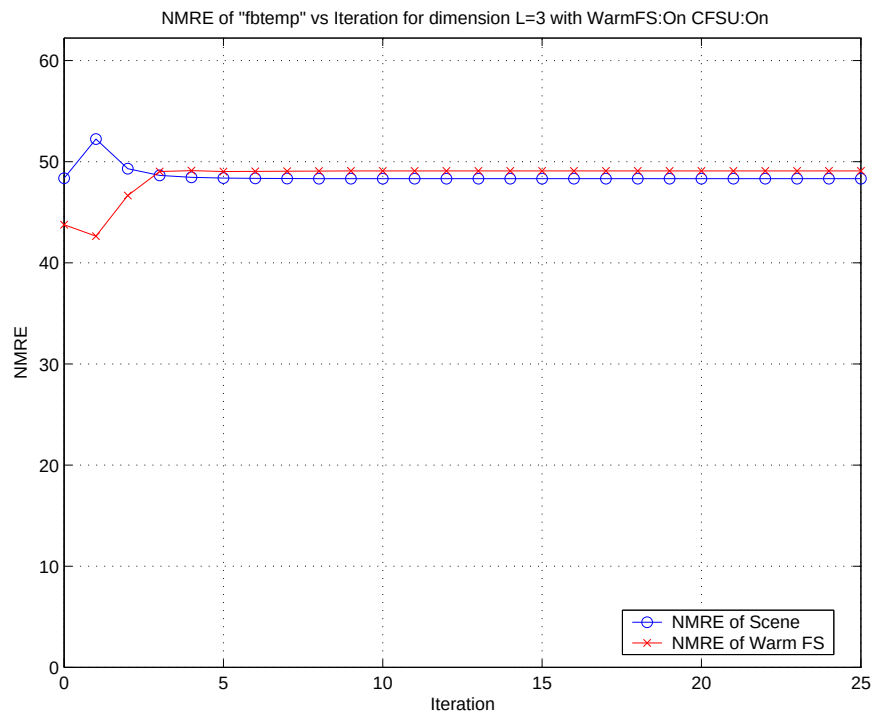


Figure 64 The NMRE curve of the warm field stop does not follow the same trend as the NMRE of the scene. Both do converge at the 4th iteration.

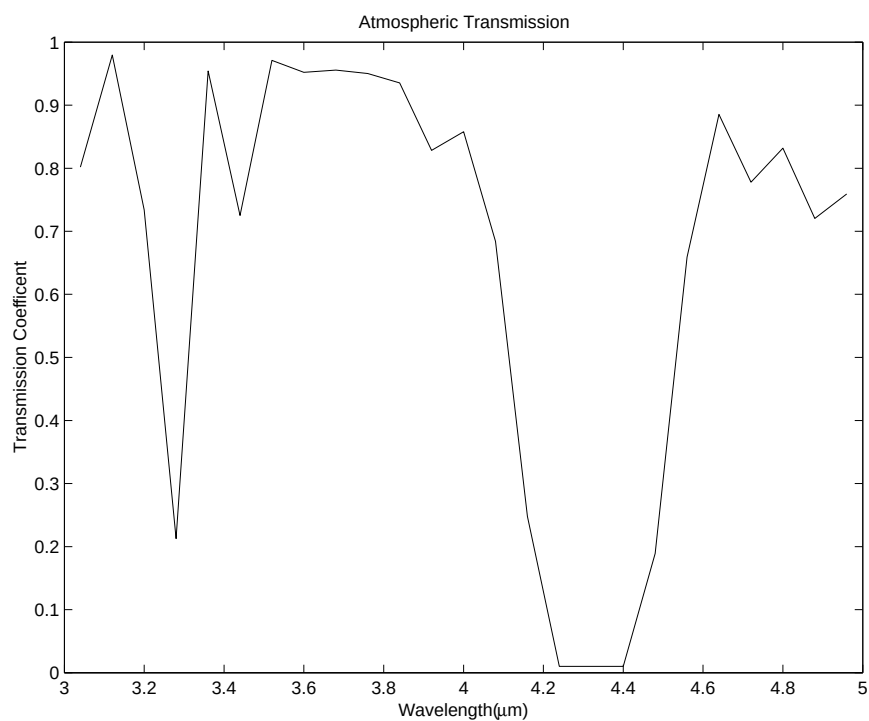


Figure 65 Atmospheric Transmission Curve

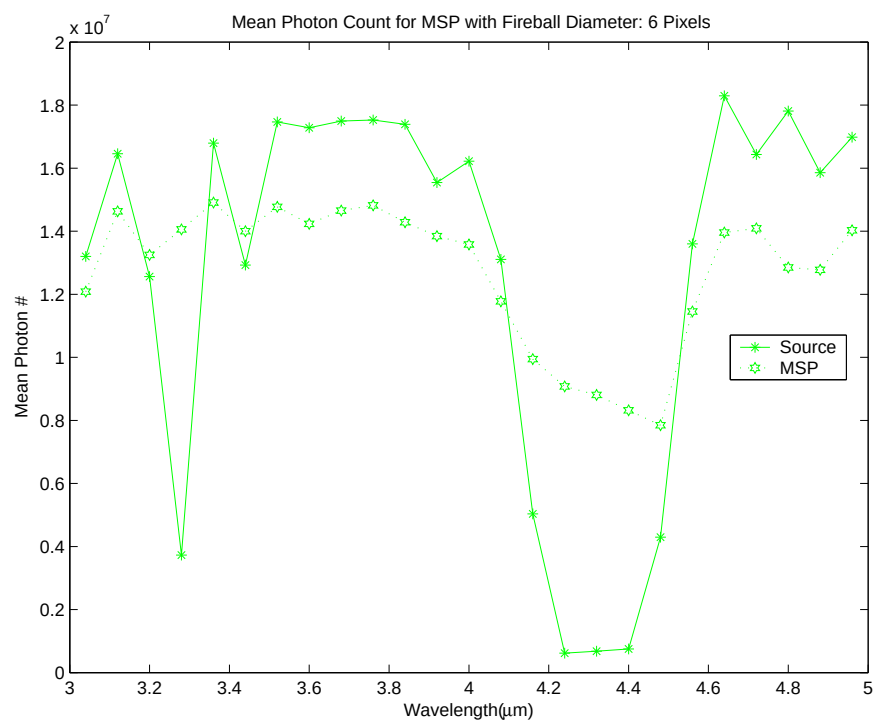


Figure 66 Mean Photon Count for MSP with a Fireball Diameter of 6 Pixels.

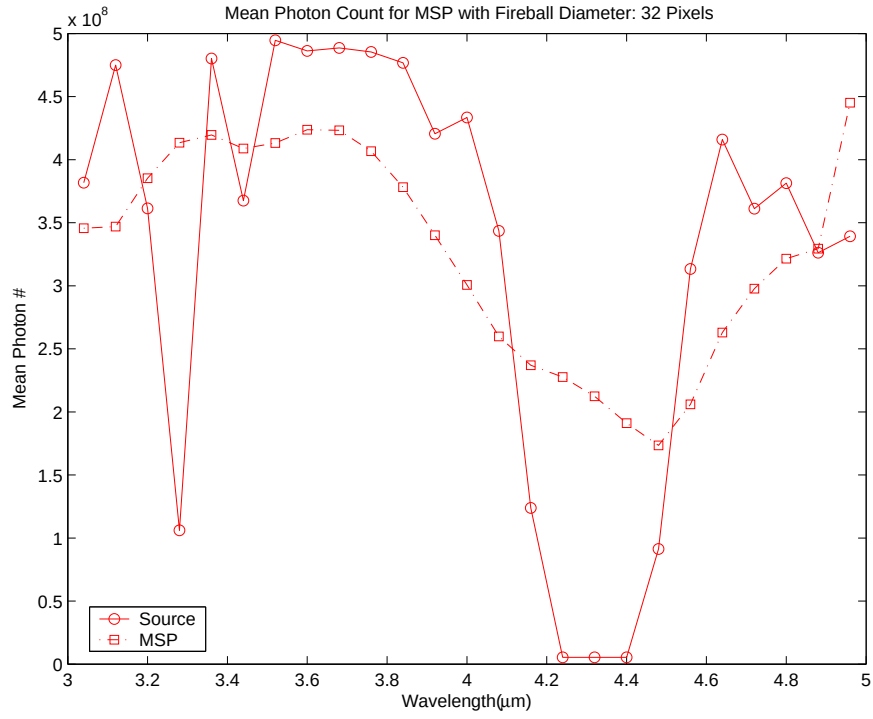


Figure 67 Mean Photon Count for MSP with Fireball Diameter of 32 Pixels.

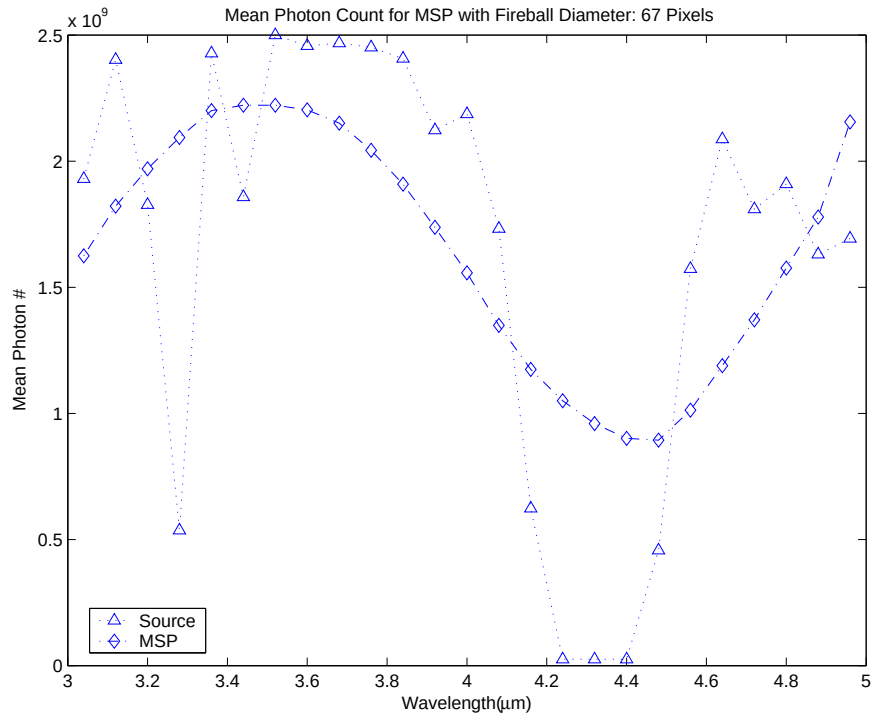


Figure 68 Mean Photon Count for MSP with a Fireball Diameter of 67 Pixels.

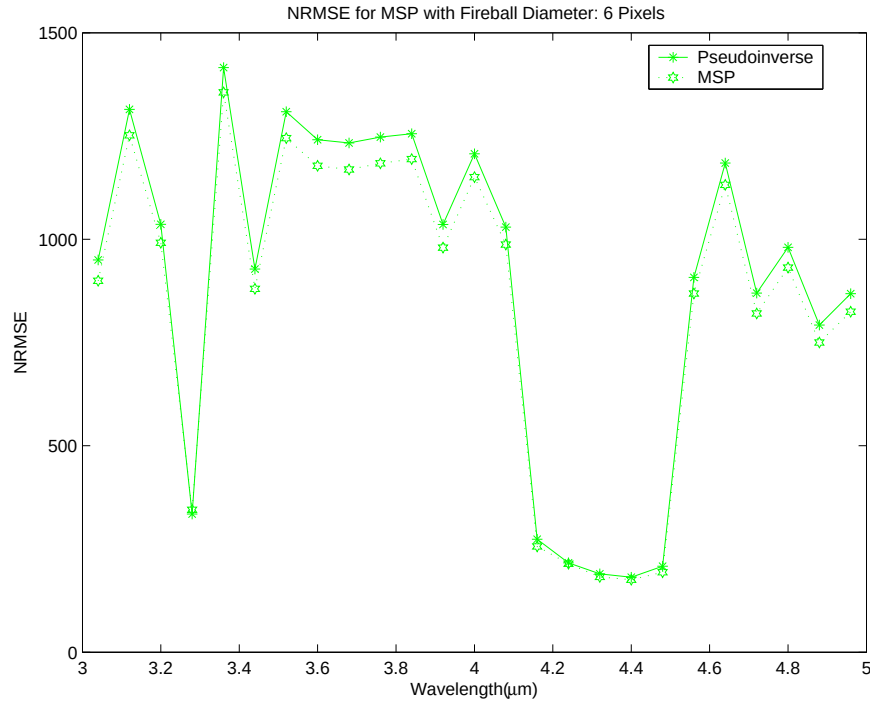


Figure 69 NRMSE for MSP with Fireball Diameter of 6 Pixels.

NRMSE results for each fireball size are shown in Figures 69-71. Content not in the fireball spatial extent is non-reconstructable and therefore unrecoverable. This phenomena is reflected in the poor NRMSE reconstruction of the 6 pixel fireball. The 67 pixel fireball has the least NRMSE, attributed to having the least non-reconstructable field of view. The NRMSE error metric does not know the fireball size and therefore uses the entire scene for error calculation. The 6 pixel fireball scene has the worst mean square error reconstruction despite having the smallest photon levels. The spatial extent of the fireball does show in the image reconstruction and spectral analysis of only the fireball portion of the image will lead to much improved error results. The reconstructed image scaled to the maximum of the original scene only shows the fireball with dark content elsewhere in the image.

The warm field stop can still provide a means to track mean square error reduction. Figures 72-74 show the convergence of both the warm field stop and

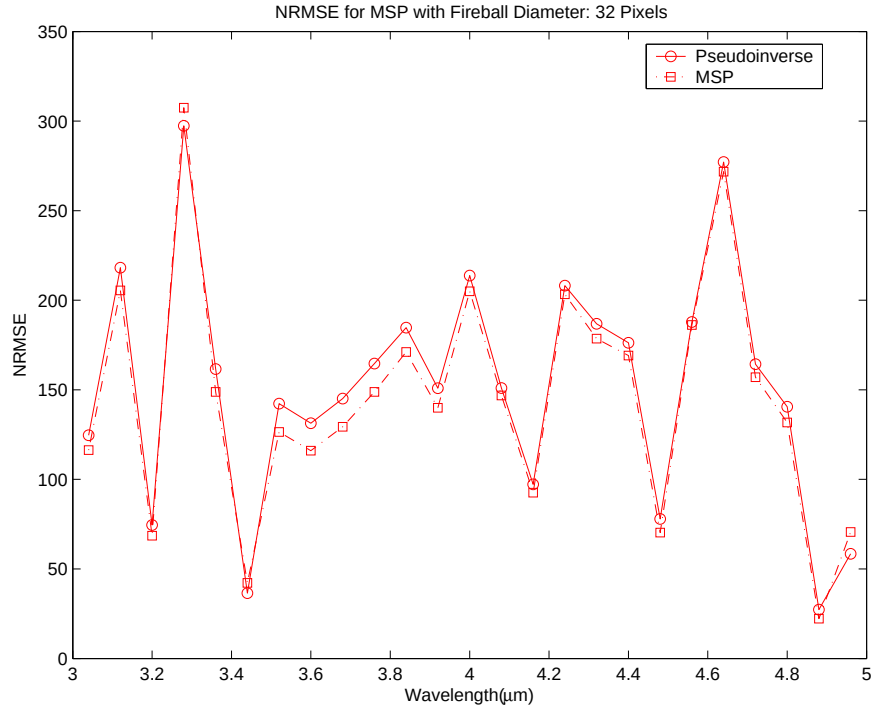


Figure 70 NRMSE for MSP with Fireball Diameter of 32 Pixels.

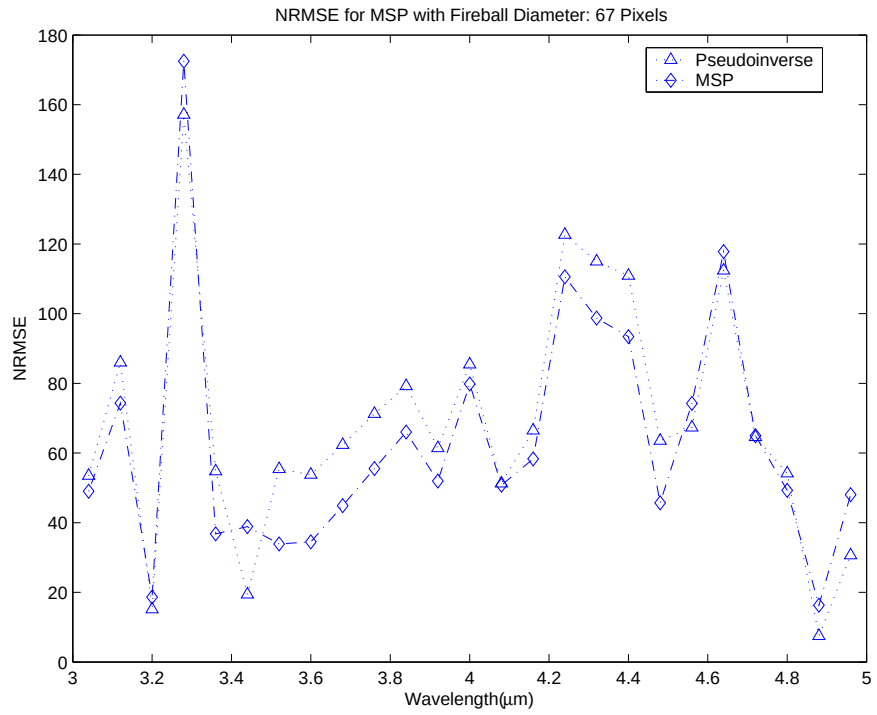


Figure 71 NRMSE for MSP with Fireball Diameter of 67 Pixels.

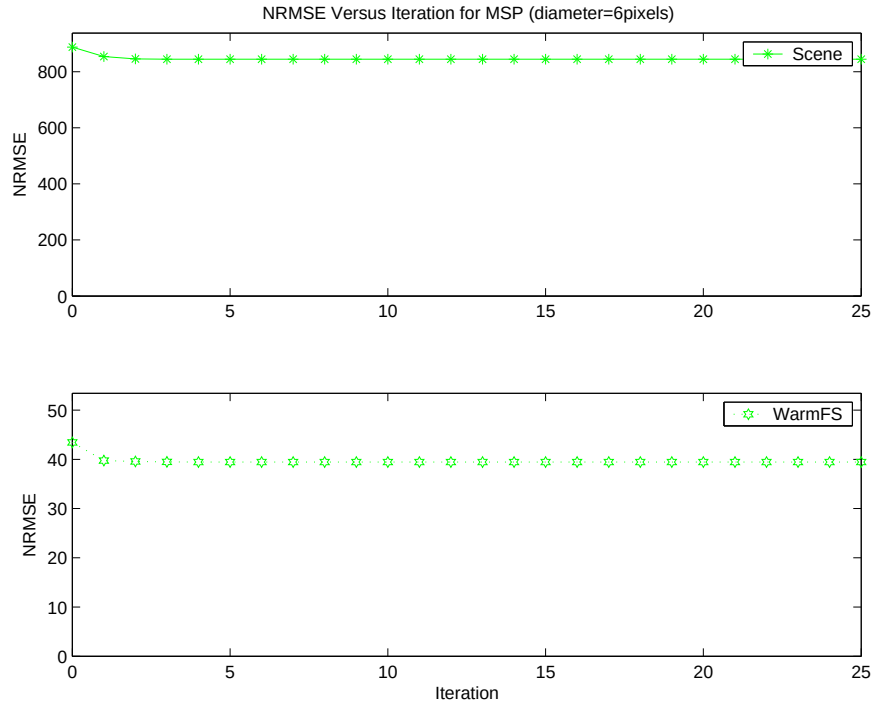


Figure 72 NRMSE Versus Iteration for MSP with Fireball Diameter of 6 Pixels.

scene occur at the same iteration, regardless of fireball size when using a field stop temperature of 300K.

The mean removed error is largest in the 6 pixel fireball scene. Figures 75-77 show NMRE results for each fireball scene. Error in mean recovery is not causing a significant portion of reconstruction error for the fireball scenes. Most mean square error results from not being able to reconstruct the non-fireball portion of the scene. Not being able to reconstruct the rest of the scene is likely not a problem, since the fireball contains all the information desirable to the user.

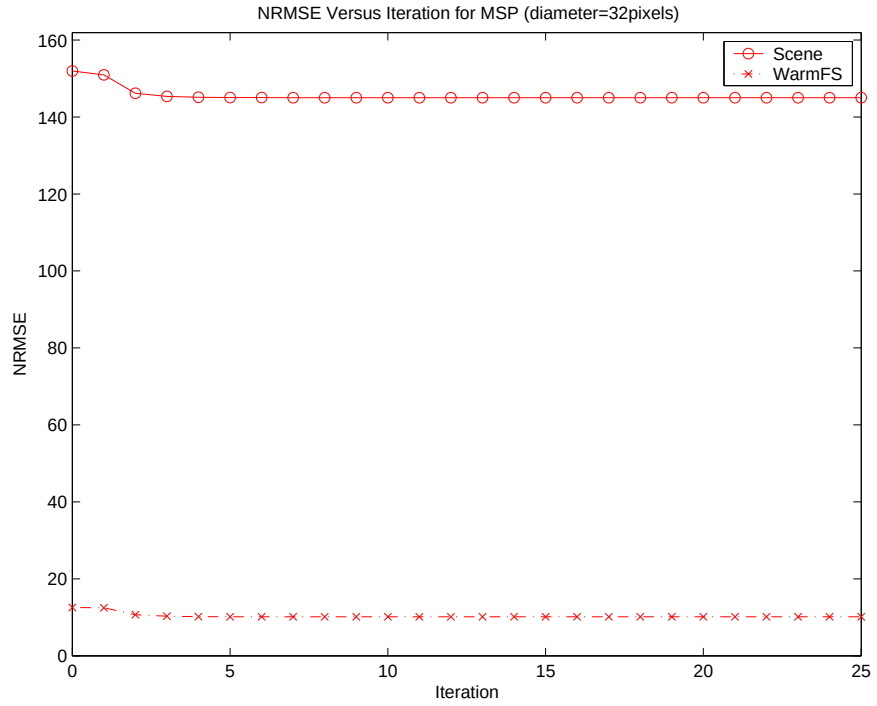


Figure 73 NRMSE Versus Iteration for MSP with Fireball Diameter of 32 Pixels.

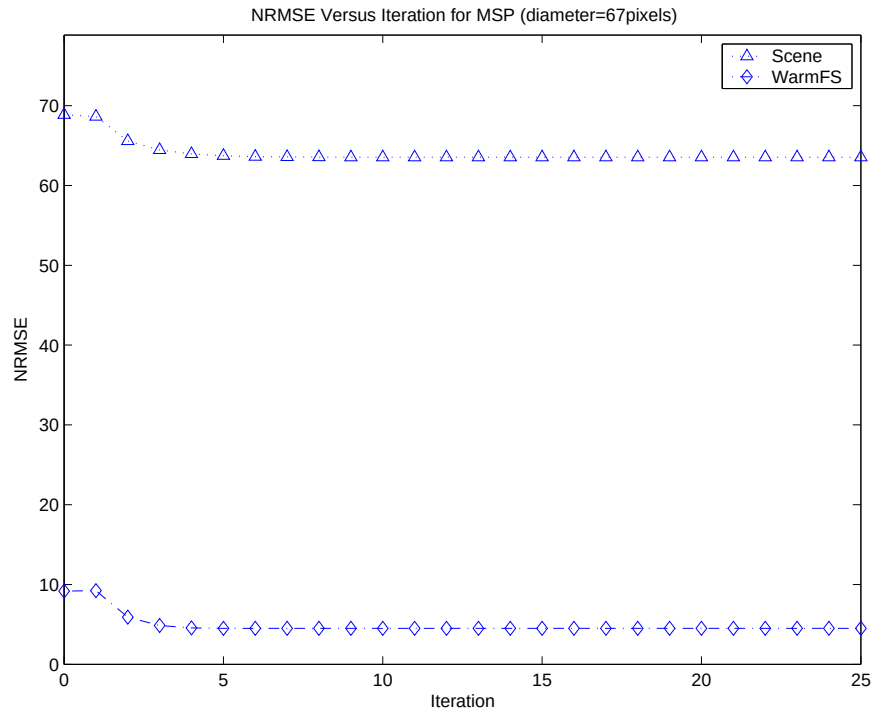


Figure 74 NRMSE Versus Iteration for MSP with Fireball Diameter of 67 Pixels.

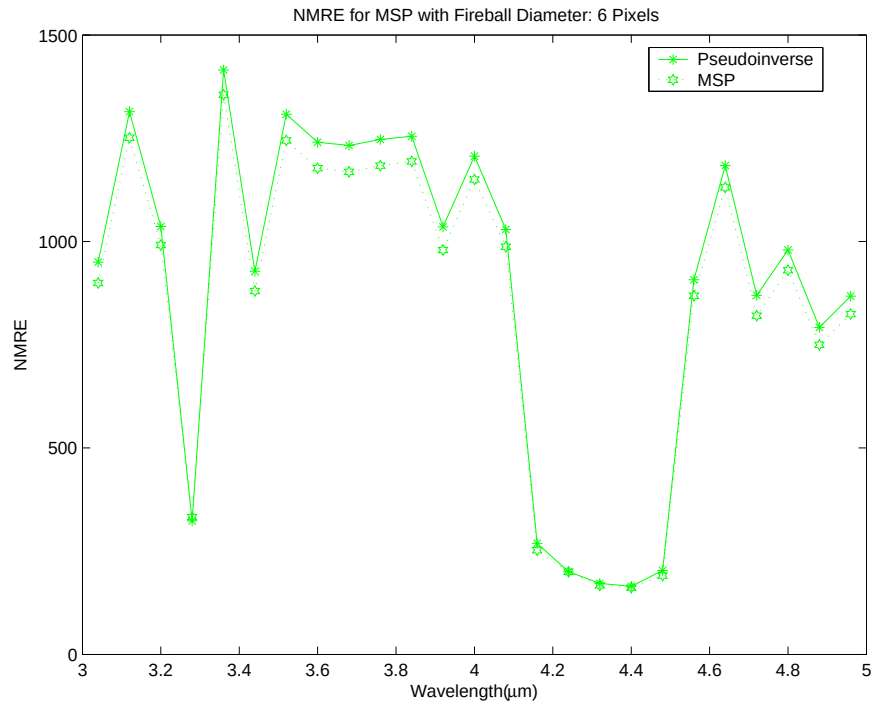


Figure 75 NMRE for MSP with Fireball Diameter of 6 Pixels.

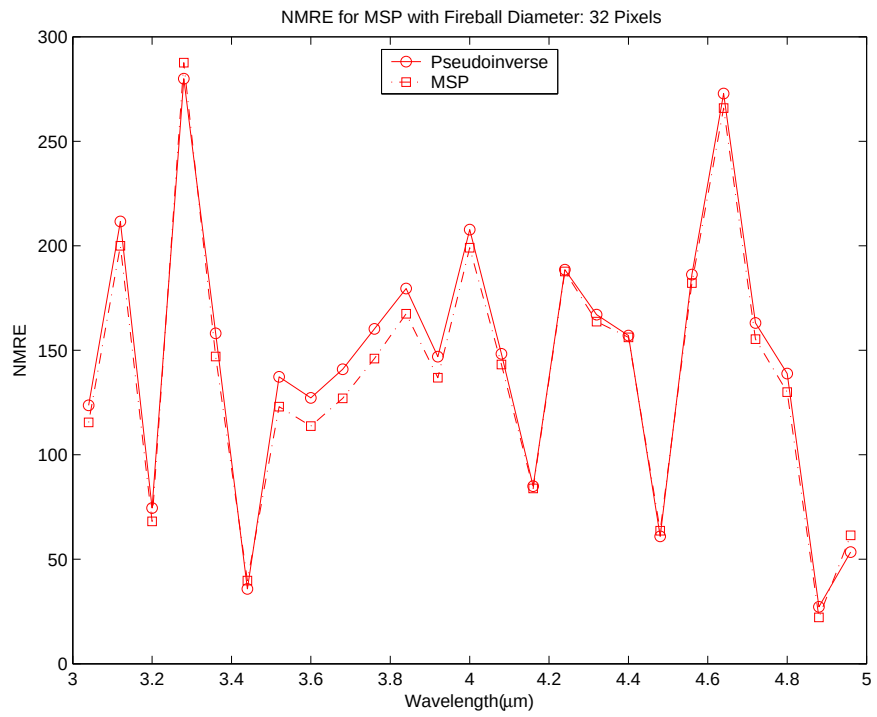


Figure 76 NMRE for MSP with Fireball Diameter of 32 Pixels.

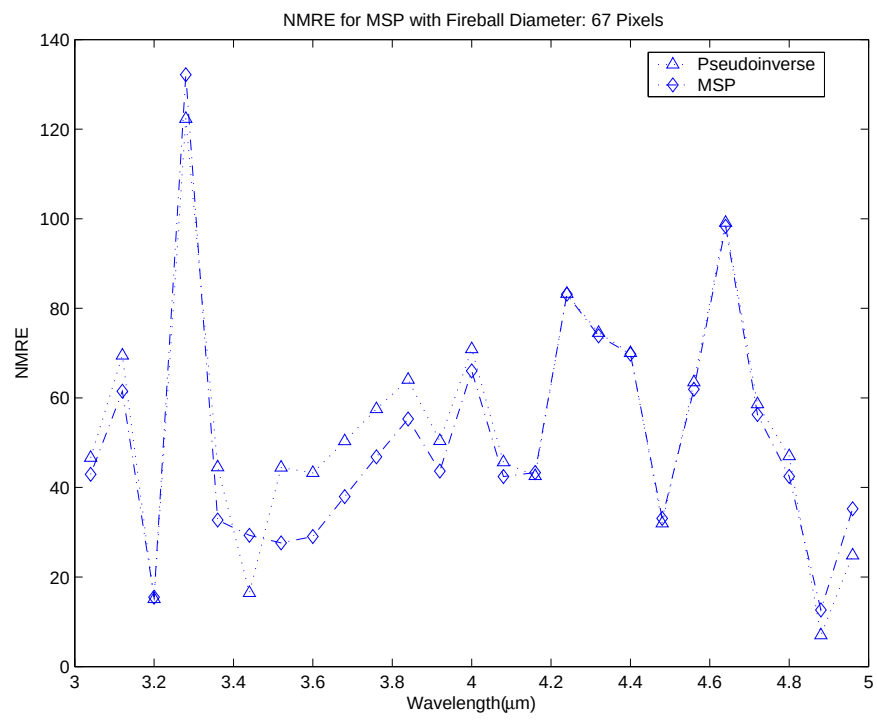


Figure 77 NMRE for MSP with Fireball Diameter of 67 Pixels.

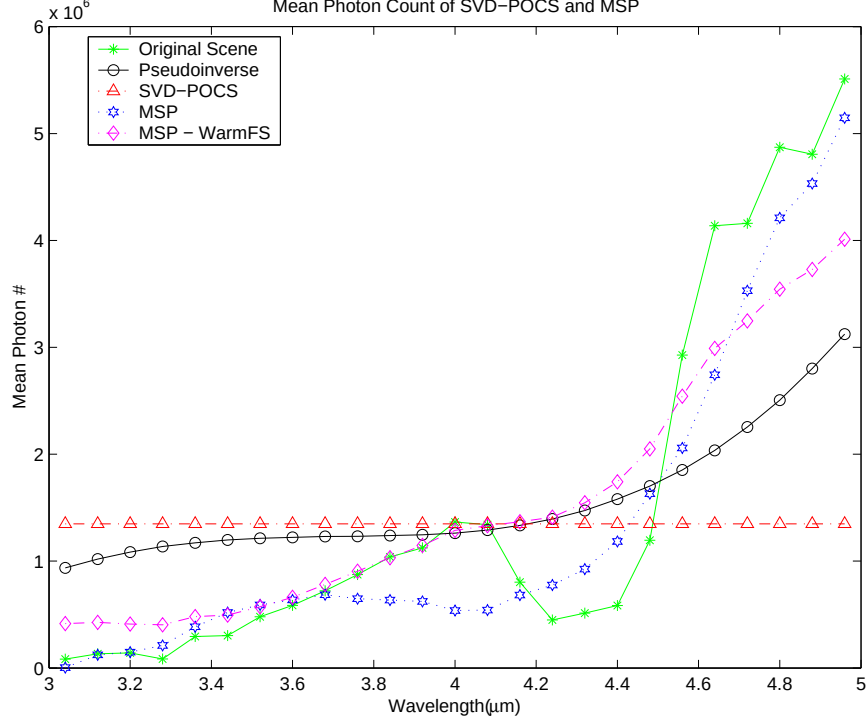


Figure 78 The mean proves hard to recover. Only MSP has any mean recovery capability, yet it does not track the mean exactly.

4.5 Absolute Radiometry in Photons

In order to perform radiometric analysis on the reconstruction, the reconstructed photon information needs to be similar in magnitude to the source information. The reconstruction must be a good approximation of the original signal to estimate temperatures or spectral intensities. A comparison of the pseudo-inverse, SVD-POCS, and MSP are shown in Figures 78 and 79. For both scenes, only MSP has any mean tracking ability. The reconstruction provided by MSP with a warm field stop scene cannot be compared directly to the other curves, since the scene is slightly different due to the addition of the warm field stop. Results show that the warm field stop does not affect the performance of MSP for scenes tested in this thesis. However, MSP is only a step in the direction of performing absolute radiometry, and cannot fully enable temperature reconstruction based on reconstructed spectral solutions.

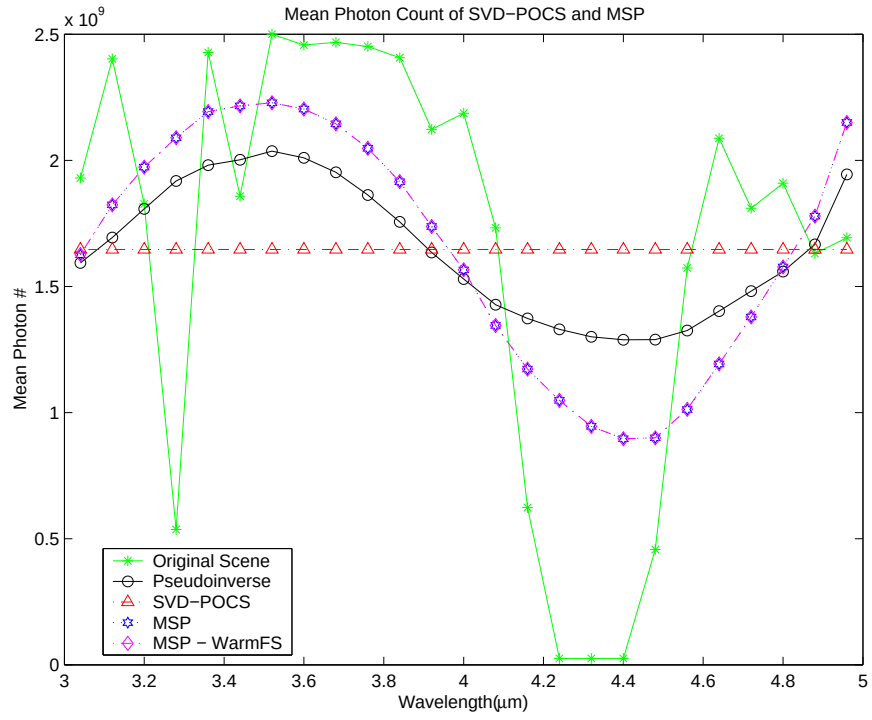


Figure 79 The mean tracking at each band is not as close when a fireball is introduced.

V. Conclusions

5.1 Summary

Hyperspectral data collection and analysis is an increasing priority with the growing need to obtain greater classification precision than offered by traditional spatial imagery. In this thesis, trends in hyperspectral reconstruction were explored where reconstruction was performed after obtaining a series of chromotomographic images. Chromotomography, developed initially by Jonathan M. Mooney formerly of AFRL/SNHI, involves capturing a series of two-dimensional images where spectral information has been dispersed on each in a unique manner by a prism rotating in front of the focal plane array.

Before the reconstruction could be tested, synthetic data was produced, approximating what would be produced from prism dispersion on the focal plane array. The pseudo-inverse singular matrix problem was addressed where two methods are compared to find which produces minimal error.

The standard iterative error reduction algorithm, SVD-POCS, was shown to not be capable of reconstructing the mean of the source scene, making absolute radiometry analysis impractical. However, SVD-POCS was shown to provide the good reconstruction if the goal is to perform relative radiometry analysis. Additional constraints are needed to make absolute radiometry analysis possible. The added constraints of non-negativity, spatial extent of the cold field stop, forcing the sum, and keeping the mean for each iteration improves radiometric performance and begins to make absolute radiometry possible.

The added constraints also make possible the use of a warm field stop to monitor reconstruction error for both the pseudo-inverse and iterative improvement algorithm. Error can be calculated each iteration to determine when a minimum has been reached in a mean square error sense. Thus, minimum mean square error of

the reconstruction can be obtained with confidence. Spatial and spectral resolution trade-offs are discussed in the next section.

5.2 *Lessons Learned*

The iterative improvement method must have constraints which change the mean of the solution to allow absolute radiometry measurements. SVD-POCS does better at shaping the reconstruction back to the original scene if the mean of the scene does not need to be recovered. Otherwise, constraints like those found in MSP are needed to recover the mean, but at a cost to variance reconstruction.

Temperature reconstruction error curves appear to have larger error than the overall scene when atmospheric absorption is on, and at a lower value without atmospheric absorption; meaning atmospheric absorption induces additional spectral frequency which the algorithms are have difficulty reconstructing. Absolute radiometry, although improved with MSP, may still be impractical due to increased temperature reconstruction error.

Several limitations to spectral resolution were discovered. The number of images recorded during prism rotation has to be at least the number of spectral bands desired. Additionally, spectral resolution is limited by the number of pixels the spectral dispersion of the prism falls over on the focal plane array. The spatial and spectral resolution are limited by the width of the cold field stop, which is limited by the size of the focal plane array. Spectral resolution can be increased if a smaller portion of the scene is desired for reconstruction. Likewise, spatial resolution can be increased if the number of wavelength bins is reduced. The cold field stop size is increased by either decreasing the field of view or forcing the original field to fall on fewer pixels of the focal plane. Spatial resolution is determined by the resolution of the scene entering the optics, pixel size on the CCD, and by the extent of the non-cold field stop image. The extent of reconstructed spectral crossband interference is

band dependent, not wavelength dependent. Increasing the number of reconstructed bands decreases the bandwidth of the spectral interference.

If you are only interested in obtaining the spectrum of a very bright object with a relatively cool background, you can allow the rest of the non-fireball portion of the scene to fall off the cold field stop while keeping the fireball content within the extent of the CCD. This will not work if the spatial extent of the fireball is not known prior to chromotomographic imaging, since spectral resolution is also determined by the prism dispersion onto a number of pixel elements of the CCD. When the warm field stop is used, the spatial extent of the scene is fixed and no scene content can fall off the cold field stop. Computer processing limitations also exist. The size of an 512x161x224 hyperspectral datacube is on the order of hundreds of megabytes.

5.3 Future Work

Automated Model Dimension Selection. Work in this thesis demonstrated the warm field stop is capable of finding the model dimension which results in the least mean square error, but this can only be done if reconstruction is performed for each model dimension size. The reconstruction process would be less expensive computationally and more effective if the dimension value could be automatically selected.

Warm Field Stop. Several warm field stop parameters come to mind which were not explored in this thesis. The question of the minimum warm field stop size which provides adequate performance monitoring has yet to be answered. Only three warm field stop temperatures were tested in this thesis, meaning additional temperature performance tests are needed to get higher degree of certainty in warm field stop performance versus temperature.

Improving Radiometry. The additional constraints used for MSP do improve absolute radiometry performance, but allow for improvement to reduce spectral re-

covery error. Additional constraints or numerical techniques may be used to reduce absolute radiometric error even further. If fireball spectral recovery is the goal of reconstruction, improvements can be made to automatically calculate spatial constraints of the fireball and then perform absolute radiometry to that spatial area. Additional constraints or numeric techniques may lead to practical absolute radiometric capabilities. An investigation of relative radiometric performance can also be done to take advantage of the strength of SVD-POCS and compare reconstruction performance to absolute radiometry results.

Speed Enhancement. The speed at which spectral cubes can be measured and processed provides a severe performance limitation for chromotomographic imagers. One possibility for improving the speed of these systems involves collecting spatial projections of the frames from the CCD used to gather the images. These spatial projections would represent vectors that are the result of row or column summing the image on the CCD array itself. The improvement in throughput would allow spectral cubes to be collected as much as one thousand times faster and would reduce the processing burden by the same factor. Spatial information would be sacrificed in an efficient way in order to improve the temporal resolution of the sensor.

Time Analysis. The work in this thesis assumed the scene was not changing with time. Additional work should be done to explore the consequences resulting from imaging a transient event.

A method for improving the temporal resolution of the sensor without sacrificing spatial information would possibly involve deconvolving the temporal window of the sensor from a sequence of temporally overlapping image reconstructions. If M image frames are required to form a spectral cube, then every time a new frame is captured the M frame window would be shifted by one frame and a new spectral cube would be reconstructed using $M-1$ old frames as well as the new frame. The

goal of this project would be to remove the temporal redundancy via a deconvolution along the temporal axis.

5.4 *Conclusion*

This study shows trends associated with chromotomographic reconstruction and how a design of the instrument allows the ability to stop iterative reconstruction when the solution has reached minimum mean square error. Additional information is needed to enable SVD-POCS reconstruction for absolute radiometry. Even with the additional constraints, the modified algorithm only improves absolute radiometry performance and does not provide a radiometrically accurate reconstruction. If you try to do absolute radiometry by adding constraints, you make relative radiometry results worse. The mean intensity of the reconstruction changes much slower between spectral bands than in the original scene, thereby limiting chromatic frequency reconstruction. High chromatic frequencies introduced by atmospheric attenuation cannot be reconstructed, causing temperature estimation to be difficult. The warm field stop does not affect performance of MSP and can be used to monitor reconstruction error during the iterative process; allowing acquisition of the minimum mean square error solution with certainty.

Appendix A. MATLAB Code

A.1 Master Routine Code

The following code comprises the “master.m” script file:

```
%Master Routine
clear all
close all
%This script executes the following subroutines
% 1 Get Source Object
% 2 Synthesize Data
% 3 Pseudoinverse Reconstruction
% 4 SVD-POCS or MSP
%Parameters
%*****
brod=0;
gus=1;
runpocs=1; % Run SVD-POCS
epstest=0; %Run the epsilon tester
warmcheck=0; %Use the warm field stop to halt the iterative process (1=Yes, 0=No)
unitchange=1; %Use the photon model instead of spectral radiance.
if brod==1
    cfsu=0; %Add the cold field stop update to SVD-POCS (1=Yes, 0=No)
    zerop=0; %Force negative values of c to be zero. Turn this off along with cfsu to get Brodzik's
pocs.
    rmeanpocs=1; %Remove the mean of c2 for each iteration. (pat rec def)
    forcesum=0; %Force the sum of the reconstructed cube to be that of the first data set.
    L=1; %Model Dimension. Set to 'step' to test each possible.
elseif gus==1
    cfsu=1; %Add the cold field stop update to SVD-POCS (1=Yes, 0=No)
    zerop=1; %Force negative values of c to be zero. Turn this off along with cfsu to get Brodzik's
pocs.
    rmeanpocs=0; %Remove the mean of c2 for each iteration. (pat rec def)
    forcesum=1; %Force the sum of the reconstructed cube to be that of the first data set.
    L=3; %Model Dimension. Set to 'step' to test each possible.
else
    %MANUAL ENTRY OF POCS PARAMETERS
    cfsu=1; %Add the cold field stop update to SVD-POCS (1=Yes, 0=No)
    zerop=1; %Force negative values of c to be zero. Turn this off along with cfsu to get Brodzik's
pocs.
    rmeanpocs=0; %Remove the mean of c2 for each iteration. (pat rec def)
    forcesum=1; %Force the sum of the reconstructed cube to be that of the first data set.
    L='step'; %Model Dimension. Set to 'step' to test each possible.
end
source='temp'; % 'temp' for the temperature map
numlam=25; % Number of Wavelength Bins
angles=25; % Number of Prism Rotation Angles
noise=1; % Noise (On:1 or Off:0)
atmosphere=1; % Run with estimated atmosphere attenuation (1=Yes, 0=No)
warm=1; % Use the Warm Field Stop Model (1=Yes, 0=No)
warmwidth=4; % Width of the Warm Field Stop
start=3000; % Starting wavelength in nm
stop=5000; % Maximum wavelength in nm
T=300; % Temperature of warm field stop
iterations=25; % Number of Iterations for SVD-POCS
method=1; % Method to find Hinv (1=threshold inverse, 0=Wiener)
%Plots
%*****
imageo=0; % Image the Object Cube
imaged=0; % Image the Synthetic Images
```

```

imagec=0; % Image the Pseudoinverse Cube
imagec2=0; % Image the POCS Reconstruction Cube
nrmseit=1; % Get a plot of NRMSE vs Iteration for SVD-POCS
plotwarm=1; % Plot the warm fieldstop error vs L or vs iterations
%Settings based on the input parameters
%*****
if noise==0
    eps1=5e-9; %Best for no noise
    eps0=5e-9; %Also best for no noise
else
    eps1=0.01; %0.01 Best for noise from 4-5um. 1 is best for 2.1-2.5um
    eps0=0.01; %Best with noise from 3-5um
end
if method==1
    leps=length(eps1);
    epss=eps1;
else
    leps=length(eps0);
    epss=eps0;
end
if warm==0
    warmwidth=0;
end
lam=1:numlam; % Wavelength Bins
phi=0:2*pi/angles:2*pi-2*pi/angles; % Rotation Angles
numphi=length(phi); % Number of Angles
bw=abs(stop-start)/numlam; % Spectral Bandwidth
wavenm=start+bw/2:bw:stop-bw/2; % Spectral Band of Interest
coldfswidth=floor(numlam/2); % Width of the cold field stop (assume uniform prism disp)
if L=='step' % Needed for Dimensionality Tests
    dim=1:numlam;
else
    dim=L;
end
NMREforL=zeros(numlam,length(dim));
if warm==1
    warmoffon='On';
else
    warmoffon='Off';
end
if cfsu==1
    cfsuoffon='On';
else
    cfsuoffon='Off';
end
%*****
%*****
%Call for the object
[o,wavenm]=getobject(source,numlam,warm,warmwidth,T,wavenm,atmosphere,unitchange);
%Create synthetic data d, output the DFT D
[D, W, sumofd]=synth(lam,phi,o,imaged,noise);
%*****
mino=min(min(min(o))); % Used for output display scaling
peako=max(max(max(o)));
[oheight,owidth,olam]=size(o);
lexlength=oheight*owidth; % Length of lexicographical rows
%*****
%Plotting the Object
%*****
if imageo==1
    for p=1:numlam
        figure(p)
        imagesc(o(:, :, p), [mino peako])
        colormap(gray(256))
        axis image
    end
end

```

```

        title(['Original Band ' num2str(p) ''])
    end
end
%*****
%PSEUDOINVERSE RECONSTRUCTION
%*****
[c, C, H, Hinv]=reco_inline(D,W,method,epss(id),epstest,zerop,cfsu,forcesum,sumofd);
if epstest==0
    clear D; clear W;
else
    clear H; clear Hinv; clear C;
end
%*****
minc=min(min(min(c)));
peakc=max(max(max(c)));
%Plotting the reconstruction
%*****
if imagec==1
    for p=numlam+numphi+1:2*numlam+numphi
        figure(p)
        imagesc(c(:, :, p-numlam-numphi), [minc peakc])
        colormap(gray(256))
        axis image
        title(['Reconstructed Band #' num2str(p-numlam-numphi) ''])
    end
end
%*****
% Plot results
%*****
intensityplot3
%*****
%Find the initial NMRE of the Warm FS
%*****
%ind=[8 9 10 11 12 13 14 20 21 22 23 24 25]; %change this to the same "index" in plotspseudo.m
ind=find(mean(mean(o))>=.5*mean(mean(mean(o)))); %change this to the same "index" in plotspseudo.m
load bbgo
if plotwarm==1 & warm==1
    for k=1:numlam
        rangev1_1=coldfswidth+1:coldfswidth+warmwidth;
        rangev1_2=coldfswidth+1+warmwidth:owidth-coldfswidth-1-warmwidth;
        v1=c(rangev1_1,rangev1_2,k)-bb(k);
        %The mean blackbody value is equal to the value, so they cancel out
        v1mr=c(rangev1_1,rangev1_2,k)-mean(mean(c(rangev1_1,rangev1_2,k)));
        rangev2_1=height-coldfswidth-warmwidth:height-coldfswidth-1;
        rangev2_2=coldfswidth+1+warmwidth:owidth-coldfswidth-1-warmwidth;
        v2=c(rangev2_1,rangev2_2,k)-bb(k);
        v2mr=c(rangev2_1,rangev2_2,k)-mean(mean(c(rangev2_1,rangev2_2,k)));
        rangev3_1=coldfswidth+1:height-coldfswidth-1;
        rangev3_2=coldfswidth+1:coldfswidth+warmwidth;
        v3=c(rangev3_1,rangev3_2,k)-bb(k);
        v3mr=c(rangev3_1,rangev3_2,k)-mean(mean(c(rangev3_1,rangev3_2,k)));
        rangev4_1=owidth-coldfswidth-warmwidth:owidth-coldfswidth-1;
        rangev4_2=coldfswidth+1,rangev4_2,k)-bb(k);
        v4=c(rangev4_1,rangev4_2,k)-bb(k);
        v4mr=c(rangev4_1,rangev4_2,k)-mean(mean(c(rangev4_1,rangev4_2,k)));
        value=[v1(:).' v2(:).' v3(:).' v4(:).'];
        valuemr=[v1mr(:).' v2mr(:).' v3mr(:).' v4mr(:).'];
        if max(ind==k)==1 %To only use spectra used elsewhere in NVE calculation
            VARinFS(k)=var(value);
        end
    end
    RMSEinFS(k)=sqrt((sum(value.^2))/length(value));
    MREinFS(k)=sqrt((sum(valuemr.^2))/length(valuemr));
end
[m,n] = size(c(coldfswidth+1+warmwidth:height-coldfswidth-1-warmwidth,coldfswidth...
+1+warmwidth:owidth-coldfswidth-1-warmwidth,1)); %Size of the non-field stop
opseudomean=sum(sum(sum(o)))/((m+2*warmwidth)*(n+2*warmwidth)*numlam);

```

```

NRMSEinFS = RMSEinFS./opseudomean.*100;
NMREinFS = MREinFS./opseudomean.*100;
NRMSEFSmean_0=mean(NRMSEinFS);
NMREmean_0=mean(NMREinFS);
VARinFS_0=mean(VARinFS);
end
%*****
%SVD-POCS RECONSTRUCTION
%*****
if runpocs==1
    for Ldim=min(dim):max(dim)
        imgnum=1;
        [c2, dL, dLratio,NRMSEmean_it,NRMSEinFSmean_it,NMREit,NMREFSmean_it]=pocsfun(c, C, o,...
            numphi, iterations, Ldim, method, imgnum, warmwidth, warm, warmcheck, H, Hinv,epss,...
            cfsu,zerop,rmeanpocs,sumofd,forcesum);
        if warm==0
            NRMSEinFSmean_it=0; %mean(NRMSEinFS); %Junk value for a filler.
            NMREFSmean_it=0;
        end
        data(Ldim, 2:7)=[Ldim, dL(Ldim), dLratio(Ldim), NRMSEmean_it(end),...
            NRMSEinFSmean_it(end), NMREFSmean_it(end)];

        %If the NRMSEvsIterations is desired... plot the results.
        %*****
        if nrmseit==1
            close(ffigure(3*numlam+numphi+12))
            figure(3*numlam+numphi+12)
            plot(0:iterations,[mean_NRMSE NRMSEmean_it],'o-')
            hold on
            if plotwarm==1 & warm==1
                plot(0:iterations,[NRMSEFSmean_0 NRMSEinFSmean_it],'xr-')
                legend(['NRMSE of Scene ','NRMSE of Warm FS'],0);
                grid on; axis([0 iterations 0 max(max([NRMSEmean_it NRMSEFSmean_0...
                    NRMSEinFSmean_it]))+10])
                title(['NRMSE of " ' source '" vs Iteration for dimension L=' num2str(Ldim) ' WarmFS:'...
                    char(warmoffon) ' (' num2str(T) 'K) CFSU:' char(cfsuoffon) "])
            else
                legend(['NRMSE of Scene '],0);
                grid on; axis([0 iterations 0 max(max(NRMSEmean_it)+10)])
                title(['NRMSE of " ' source '" vs Iteration for dimension L=' num2str(Ldim) ' with WarmFS:'...
                    char(warmoffon) ' CFSU:' char(cfsuoffon) "])
            end
            xlabel(['Iteration'])
            ylabel(['NRMSE'])
            set(3*numlam+numphi+12, 'color', [1 1 1]);
            close(figure(3*numlam+numphi+14))
            figure(3*numlam+numphi+14)
            plot(0:iterations,[mean(NMRE(ind)) NMREit],'o-')
            hold on
            if plotwarm==1 & warm==1
                scale=mean([NMREmean_0 NMREFSmean_it])/mean(NMREit);
                plot(0:iterations,[NMREmean_0 NMREFSmean_it]./scale,'xr-')
                legend(['NMRE of Scene ','NMRE of Warm FS'],0);
                grid on; axis([0 iterations 0 max(max([mean(NMRE(ind)) NMREit...
                    [NMREmean_0 NMREFSmean_it]./scale]))+10])
            else
                legend(['Average NMRE of Passing Bands'],0);
                grid on; axis([0 iterations 0 max(max([mean(NMRE(ind)) NMREit]))+...
                    .1*max(max([mean(NMRE(ind)) NMREit])))]
            end
            title(['NMRE of " ' source '" vs Iteration for dimension L=' num2str(Ldim) ' with WarmFS:'...
                char(warmoffon) ' CFSU:' char(cfsuoffon) "])
            xlabel(['Iteration'])
            ylabel(['NMRE'])
            set(3*numlam+numphi+14, 'color', [1 1 1]);

```

```

end
%*****
%Get the NMRE The NRMSE was already found and so the NRMSE value is thrown away as 'temp'
for k=1:numlam
    [temp NMRE_L(k)]=getnrmse(c2(coldfswidth+1+warmwidth:oheight-coldfswidth-1-warmwidth,...
        coldfswidth+1+warmwidth:owidth-coldfswidth-1-warmwidth,k),o(coldfswidth+1+...
        warmwidth:oheight-coldfswidth-1-warmwidth,coldfswidth+1+warmwidth:owidth-...
        coldfswidth-1-warmwidth,k),o,warmwidth);
end
clear temp
NMREforL(:,Ldim)=NMRE_L(:);

end
%*****
%If L was stepped through... plot the results.
%*****
if L=='step'
    close(ffigure(3*numlam+numphi+8))
    figure(3*numlam+numphi+8)
    plot(data(:,2),data(:,5),'o-')
    hold on
    if plotwarm==1 & warm==1
        plot(data(:,2),data(:,6),'-r')
        legend(['NRMSE of Scene '];['NRMSE of Warm FS'],0);
        grid on; axis([1 numlam 0 max(max([data(:,5) data(:,6)]))])
        I2=find(data(:,6)==min(data(:,6)))
        plot(I2, min(data(:,6)),'r*')
    else
        legend(['NRMSE of Scene '],0);
        grid on; axis([1 numlam 0 max(max([data(:,5)+10 ]))])
    end
    title(['NRMSE of "' source '" vs L for ' num2str(iterations) ' Iterations with WarmFS:'...
        char(warmoffon) ' CFSU:' char(cfsuoffon) "])
    xlabel(['Dimension'])
    ylabel(['NRMSE'])
    set(101, 'color', [1 1 1]);
    I=find(data(:,5)==min(data(:,5)))
    % plot(I, min(data(:,5)),'*')
    %NMRE for each L
    %ind=[9 11 12 13 14 19 20 21 22 23]; % Can use ind, or use index2 from intensityplot2.m
    aveNMRE_orig=mean(NMRE(ind));
    for Ldim=min(dim):max(dim)
        temp=NMREforL(:,Ldim);
        avgNMRE_indpts(Ldim)=mean(temp(ind));
    end
    close(ffigure(3*numlam+numphi+10))
    figure(3*numlam+numphi+10)
    plot(1:numlam, avgNMRE_indpts,'o-')
    title(['Average NMRE of "' source '" vs L for ' num2str(iterations) ' Iterations with WarmFS:'...
        char(warmoffon) ' CFSU:' char(cfsuoffon) "])
    xlabel(['Dimension'])
    ylabel(['NMRE'])

    close(ffigure(3*numlam+numphi+11))
    figure(3*numlam+numphi+11)
    plot(data(:,2),data(:,5),'o-')
    hold on
    plot(1:numlam, avgNMRE_indpts,':xr')
    title(['NRMSE and Average NMRE of "' source '" vs L for ' num2str(iterations) ' Iterations...
        with WarmFS:' char(warmoffon) ' CFSU:' char(cfsuoffon) "])
    xlabel(['Dimension'])
    ylabel(['NRMSE or NMRE'])
    legend(['NRMSE '; 'Average NMRE'],0);
    set(3*numlam+numphi+11, 'color', [1 1 1])
end

```

```

        save(num2str(datafile),'data')
%*****
%Plotting the final reconstruction
%*****
    if imagec2==1
        minc2=min(min(min(c2)));
        peakc2=max(max(max(c2)));
        for p=2*numlam+numphi+1:3*numlam+numphi
            figure(p)
            imagesc(c2(:,p-2*numlam-numphi),[minc2 peakc2])
            colormap(gray(256))
            axis image
            title(['Reconstructed Band After POCS #' num2str(p-2*numlam-numphi) ''])
        end
    end
%*****
    intensityplot_pocs4
    plotNMRE
end

```

A.2 Get Object Code

The following code comprises the “getobject.m” script file:

```

function [oat,wavenm]=getobject(source,numlam,warm,warmwidth,T,wavenm,atmosphere,unitchange)
%function [oat,wavenm]=getobject(source,numlam,warm,warmwidth,T,wavenm,atmosphere,unitchange)
%This function gets the object cube and inserts the warm and cold field stop.
%The Source can be a temperature map, AVIRIS data, or test sources such as
%a point.
%Setup to determine the source input. There is a better way i'm sure.
len=length(source);
var=source;
for k=len+1:6;
var=[var num2str(ones(1,1))];
end
a1='zoom11';
a2='point1';
a3='line11';
a4='temptn'; %Normal Temperature Test
a5='fbt_s1';
a6='5freqs';
a7='fbt_m1';
a8='fbtemp';
a9='tempt2'; %Higher Temperature Test
a10='temp11';
a11='smallc';
a12='tempte'; %Smallcirc Temperature Test
%Atmosphere attenuation preparation
%*****
curve=load('trans_10_10.dat'); %Atmosphere absorption curve for 1.0mm water column from .9-5.6 microns
% wavelength spacing is 5.0100e-004 microns
for k=1:numlam
    spot=find(min(abs(wavenm(k)/1000-curve(:,1)))==(abs(wavenm(k)/1000-curve(:,1))));
    index(k)=spot(1); %spot is used to pick an index value when it so happens that a wavenm value falls
    exactly between values in curve
end
mycurve=curve(index,2); %This is my atmosphere absorption coefficient
index=find(mycurve<0.01); %A mycurve value equal to zero will cause a ZERO intensity/variance image. That
causes error metrics to misbehave.
mycurve(index)=0.01;
%*****

```

```

%GET THE OBJECT
%*****
fprintf(['Retrieving Object "' source '" ...\\n']);
if var==a1 %AVIRIS 64x64x16
    load cube6416;
    o=newcube;
    [oheight,owidth,numlam]=size(newcube);
    type='aviris';
elseif var==a2 %Point Source
    o=zeros(64,64,numlam);
    intensity = blackbodyg(wavenm,T);
    gain=1000;
    intensity=1000*gain*intensity;
    extragain=1;
    [oheight,owidth,numlam]=size(o);
    for k=1:numlam
        o(oheight/2+1,owidth/2+1,k)=intensity(k);
    end
    type='temp11';
elseif var==a3 %A Line
    o=zeros(64,64,numlam);
    intensity = blackbodyg(median(wavenm),T);
    gain=1000;
    intensity=1000*gain;
    extragain=1;
    [oheight,owidth,numlam]=size(o);
    for k=1:2
        o(:,2:2:64,k+4)=intensity;
    end
    type='temp11';
elseif var==a4 % Temptest where regular temperatures used
    load temp;
    temp=round(temp*10)/10; %Rounded to have less temperatures
    temptestfile2=temp;
    tt=2;
    opt_flg=0;
    [cube]=makecube(temp,wavenm,opt_flg,tt);
    gain=1000;
    cube=cube.*gain*1000;
    extragain=1;
    o=cube;
    [oheight,owidth,numlam]=size(o);
    type='temp11';
elseif var==a5 % Small Fireball
    load temp;
    fid=fopen('fireballs.bmp','r');
    fclose(fid);
    temporary = imread('fireballs.bmp','bmp');
    temporary=double(real(temporary));
    infire=find(temporary==0);
    diameter=find(temporary(63,:)==0)
    temp(infire)=1000;
    tt=2;
    opt_flg=0;
    [cube]=makecube(temp,wavenm,opt_flg,tt);
    gain=1000;
    o=cube.*gain*1000;
    extragain=1;
    type='temp11';
    [oheight,owidth,numlam]=size(o);
elseif var==a6 %5Freqs
    I=find(mycurve>=.85)
    if length(I)<5
        error('Chose a larger waveband to have enough passbands for the blocks');
    end
end

```

```

o=zeros(128,128,numlam);
intensity = blackbodyg(median(wavenm),T);
gain=1000;
intensity=intensity*1000*gain;
o(54:74,39:90,I(1))=intensity; %Freq 1
o(23:105,100:110,I(2))=intensity; %Freq 2
o(85:105,39:90,median(I))=intensity; %Freq 3
o(23:105,18:29,I(end-1))=intensity; %Freq 4
o(23:43,39:90,I(end))=intensity; %Freq 5
extragain=1;
type='temp11';
[oheight,owidth,numlam]=size(o);
elseif var==a7 %Fireball
load temp;
fid=fopen('fireballm.bmp','r');
fclose(fid);
temporary = imread('fireballm.bmp','bmp');
temporary=double(real(temporary));
diameter=find(temporary(63,:)==0);
infire=find(temporary==0);
temp(infire)=1000;
tt=2;
opt_flx=0;
[cube]=makecube(temp,wavenm,opt_flx,tt);
gain=1000;
o=cube.*gain*1000;
extragain=1;
type='temp11';
[oheight,owidth,numlam]=size(o);
elseif var==a8 %Fireball on temp
load temp;
fid=fopen('fireball.bmp','r');
fclose(fid);
temporary = imread('fireball.bmp','bmp');
temporary=double(real(temporary));
infire=find(temporary==0);
diameter=find(temporary(63,:)==0);
temp(infire)=1000;
tt=2;
opt_flx=0;
[cube]=makecube(temp,wavenm,opt_flx,tt);
gain=1000;
o=cube.*gain*1000;
extragain=1;
type='temp11';
[oheight,owidth,numlam]=size(o);
elseif var==a9 %For Temptest_temp Where temp map is used
load temp;
temp=temp-min(min(temp));
temp=temp-mean(mean(temp));
maxtemp=max(max(temp));
mintemp=min(min(temp));
temp=temp.*30/maxtemp;
temp=310+temp;
temp=round(temp);
temp(14479)=337; %So I have one temperature at 337
temptestfile=temp;
tt=2;
opt_flx=0;
[cube]=makecube(temp,wavenm,opt_flx,tt);
gain=1000;
cube=cube.*gain*1000;
extragain=1;
o=cube;
[oheight,owidth,numlam]=size(o);

```

```

        type='temp11';
elseif var==a10 %Temperaturemap data
    load temp;
    tt=2;
    opt_flg=0;
    [cube]=makecube(temp,wavenm,opt_flg,tt);
    gain=1000;
    cube=cube.*gain*1000;
    extragain=1;
    o=cube;
    [oheight,owidth,numlam]=size(o);
    type='temp11';
elseif var==a11
    fid=fopen('smallcirc.bmp','r');
    fclose(fid);
    temporary = imread('smallcirc.bmp','bmp');
    intensity = blackbodyg(median(wavenm),T);
    temporary=temporary(19:82,19:82);
    temporary=double(real(temporary));
    in=find(temporary==1);
    temporary(in)=intensity;
    for k=1:numlam
        cube(:,:,k)=temporary;
    end
    gain=1000;
    o=cube*gain*1000;
    extragain=1;
    type='temp11';
    [oheight,owidth,numlam]=size(o);
elseif var==a12 %Error vs Temperature
    fid=fopen('smallcirc.bmp','r');
    fclose(fid);
    temporary = imread('smallcirc.bmp','bmp');
    intensity = blackbodyg(wavenm,T);
    temporary=double(real(temporary));
    temporary=temporary(19:82,19:82);
    in=find(temporary==1);
    cube=zeros([size(temporary) numlam]);
    for k=1:numlam
        temporary(in)=intensity(k);
        cube(:,:,k)=temporary;
    end
    gain=1000;
    o=cube*gain*1000;
    extragain=1;
    [oheight,owidth,numlam]=size(o);
    type='temp11';
else %Invalid
    error('Invalid object handle!')
end
%*****
%%BLACKBODY GENERATION FOR THE WARM FIELD STOP
%*****
coldfswidth=floor(numlam/2);
%%
if type=='temp11'
    Celsius=T-273.3;
    Fahrenheit=Celsius/5*9+32;
    bb2= blackbodyg(wavenm,T); %units of [watt cm94-2 nm94-1 sr94-1]
    gain=1000;
    bb=bb2.*gain*1000; % Units [gain*miliwatt/cm942 nm94-1 sr]
    bb=extragain.*bb;
elseif type=='aviris' %I tried this since aviris data is too intense compared to the bb curve
    T=400
    Celsius=T-273.3

```

```

        Fahrenheit=Celsius/5*9+32
        bb2= blackbodyg(wavenm,T); %units of [watt cm94-2 nm94-1 sr94-1]
        gain=1000;
        bb=bb2.*gain*1000; % Units [gain*miliwatt/cm942 nm94-1 sr]
    else % This path is followed when temp or aviris is not used
        for k=1:coldfswidth
            bb(k)=k*255/numlam;
            bb(numlam-k+1)=k*256/numlam;
            bborig=bb;
        end
    end
end
%*****
%INSERT WARM FIELD STOP
%*****
if warm==1
    fprintf('Inserting Warm FieldStop to the Object...\n');
    for k=1:numlam
        o(coldfswidth+1:coldfswidth+warmwidth, :,k)=bb(k);
        o(oheight-coldfswidth-warmwidth:oheight-coldfswidth-1,:,k)=bb(k);
        o(:,coldfswidth+1:coldfswidth+warmwidth,k)=bb(k);
        o(:,owidth-coldfswidth-warmwidth:owidth-coldfswidth-1,k)=bb(k);
    end
end
%*****
%INSERT THE COLD FIELD STOP
%*****
fprintf(['Inserting Cold Stop of width ' num2str(coldfswidth) ' to the Object...\n']);
o(1:coldfswidth, :,)=0;
o(oheight-coldfswidth:oheight,:,)=0;
o(:,1:coldfswidth,:)=0;
o(:,owidth-coldfswidth:owidth,:)=0;
%*****
%Atmosphere Absorbsion
%*****
if atmosphere==1;
    fprintf(['Applying Atmospheric Absorption Estimation to the Object...\n']);
    curve=load('trans_10_10.dat'); %Atmosphere absorbtion curve for 1.0mm water column from .9-5.6 microns
    for k=1:numlam
        spot=find(min(abs(wavenm(k)/1000-curve(:,1)))==(abs(wavenm(k)/1000-curve(:,1))));
        index(k)=spot(1);
    end
    mycurve=curve(index,2); %This is my atmosphere absorbtion coefficient
    index=find(mycurve<0.01); %A mycurve value equal to zero will cause a ZERO intensity/variance image.
    mycurve(index)=0.01;
    oat=o; %o with atmosphere
    for k=1:numlam
        oat(coldfswidth+1+warmwidth:oheight-coldfswidth-1-warmwidth,coldfswidth...
            +1+warmwidth:owidth-coldfswidth-1-warmwidth,k)...
            =o(coldfswidth+1+warmwidth:oheight-coldfswidth-1-warmwidth,...
                coldfswidth+1+warmwidth:owidth-coldfswidth-1-warmwidth,k)*mycurve(k);
    end
else
    oat=o;
end
end
%%%Assume the warm blackbody bb is not affected by atmosphere. Thus is why
%%%the index above is set as it is.
%*****
%UNIT CHANGE TO PHOTONS
%*****
% # of Photons Calculation
% %Photons(NP)=R./h./v.*Area.*deltnm.*pi.*time./gain
% %12 Hz Scan Rate = 12 lines/sec
% %614 pixels/line... so 7368 pixels/sec.. thus time is
if unitchange==1
%Constants:

```

```

con = 299792458; %Speed of light [m/s]
planck = 6.6260755e-34; %[J*s]
bw=wavenm(2)-wavenm(1);
wavenmspan=wavenm(1)-1/2*bw:bw:wavenm(end)+1/2*bw+bw;
wavem=wavenm.*1e-9;% [m]
v=con./wavem; %[Hertz]
time=1.357e-4; % [s]
gain=1000;
Area=10*18; % [cm942]
sfsqrd=(1e-3)942;
deltnm=wavenm(2)-wavenm(1); %[nm]
for k=1:numlam
    oat(:,k)=oat(:,k)./planck./v(k).*Area.*deltnm.*sfsqrd.*time./gain./1000;
end
%o is now in photons
bb=bb./planck./v.*Area.*deltnm.*sfsqrd.*time./gain./1000; %if bb is in [gain*miliwatt/cm942 nm sr]
%bb is now in photons
end
%*****
save bbgo bb %for use in pocs

```

A.3 Data Synthesis Code

The following code comprises the “synth.m” script file:

```

function [D, W, sumofd]=synth(lam,phi,o,imaged,noise)
%function [D, W, sumofd]=synth(lam,phi,o,imaged,noise)
%Synthetic Data Generation using the "circletrace method"
%lam and phi are vectors of equal length.
%The point spread function is wavelength dependent and will have an x and y
%component.
numlam=length(lam); % Number of lambdas
numphi=length(phi); % Number of This
[oheight,owidth,numlam]=size(o);
lexlength=oheight*owidth;
%Defind the point spread function w
%phi is the current "prism" rotation angle
%Plan of attack
%1. Create 'w' for each lambda
% 2. Take the FT of W
% 3. Take the FT of the object at that lambda
% 4 Mult in Freq W*F
% 5. Sum to get D(m,n,phi)
% 6. Repeat for all Phis
lam0=round(numlam/2);
disp=(lam-lam0);
shiftx=cos(phi).'*disp; %Shift values for the x direction.
shifty=sin(phi).'*disp; %Shift values for the y direction.
if size(o)==[128 128 24] & numphi==numlam
    fprintf('Loading W... \n');
    load W
else
    w=zeros(oheight,owidth,numlam,numphi);
    svalue=1; %Scaling value
    fprintf('Calculating Point Spread Function W...\n');
    %Phi Loop
    for p=1:numphi
        for k=1:numlam %Lambda loop
            tbyt=zeros(3);
            %%%%%%%%%%%%%%subpixel shift routine %%%%%%%%%%%%%%5
            xdis=shiftx(p,k); %absolute shift distance from center xdirection

```

```

ydis=shifty(p,k); %absolute shift distance from center ydirection
if ydis < 0 & xdis < 0
    rxdis=ceil(xdis); %rounded distance to get operating point
    rydis=ceil(ydis); %rounded distance to get operating point
elseif ydis < 0 & xdis > 0
    rxdis=floor(xdis); %rounded distance to get operating point
    rydis=ceil(ydis); %rounded distance to get operating point
elseif ydis > 0 & xdis < 0
    rxdis=ceil(xdis); %rounded distance to get operating point
    rydis=floor(ydis); %rounded distance to get operating point
else
    rxdis=floor(xdis); %rounded distance to get operating point
    rydis=floor(ydis); %rounded distance to get operating point
end
difx=xdis-rxdis; dify=ydis-rydis;
difya=abs(dify); difxa=abs(difx);

if difya==0 & difxa126=0
    tbyt(2,2)=(1-difxa)*1*svalue;
    tbyt(2,2+sign(difx))=difxa*1*svalue;
elseif difxa==0 & difya126=0
    tbyt(2,2)=(1)*(1-difya)*svalue;
    tbyt(2-sign(dify),2)=(1)*difya*svalue; %always in column 2, dependent on y shift
elseif difxa==0 & difya==0
    tbyt(2,2)=(1)*(1)*svalue;
else
    tbyt(2,2+sign(difx))=difxa*(1-difya)*svalue; %always in row 2, dependent on x shift
    tbyt(2-sign(dify),2)=(1-difxa)*difya*svalue; %always in column 2, dependent on y shift
    tbyt(2-sign(dify),2+sign(difx))=difxa*difya*svalue;
    tbyt(2,2)=(1-difxa)*(1-difya)*svalue; %2,2 for pos
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%end subpixel shift routine %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pixspace=1;
midx=owidth/2+1+pixspace*rxdis; midy=oheight/2+1-pixspace*rydis; %pixspace is the pixelspacing
between wavelength bins
w(midy-1:midy+1,midx-1:midx+1,k,p)=tbyt; %spread to neighboring pixels
end
end
%Find W, the FFT2 of w
W=zeros(oheight,owidth,numlam,numphi);
for i=1:numlam
    for j=1:numphi
        tempw=w(:,i,j);
        W(:,i,j)=fft2(fftshift(tempw));
    end
end
end
%Find O, the FFT2 of o
O=zeros(oheight,owidth,numlam);
for i=1:numlam
    O(:,i)=fft2(o(:,i));
end
%Initialize Ouput data matrices
D=zeros(oheight,owidth,numphi);
fprintf('Applying W to the Object Cube (O)...\n');
for p=1:oheight
    for k=1:owidth
        WT=reshape(W(k,p,:),numlam,numphi);
        OT=squeeze(O(k,p,:));
        D(k,p,:)=WT.'*OT;
    end
end
end
clear O
%Find the inverse FFT2 of D
d=zeros(oheight,owidth,numphi);

```

```

for k = 1:numphi
    d(:,:,k)=(ifft2(D(:,:,k)));
end
d=real(d);
%Negative Check Constraint
Index=find(d<0);
d(Index)=0;
clear Index
save dclean d
sumofd=sum(sum(d(:,:,1)));
%Noise
*****
noisevalue=zeros(1,lexlength*numphi);
if noise==1
    for k=1:lexlength*numphi
        noisevalue(k)=normrnd(0,sqrt(d(k)));
        d(k)=d(k)+noisevalue(k);
    end
    d=real(d);
    meannoise=mean(noisevalue);
    save meannoise meannoise
    clear noisevalues
    Index=find(d<0);
    d(Index)=0;
    clear Index
    save dnoise d
    for p = 1:numphi
        D(:,:,p)=fft2(d(:,:,p));
    end
end
*****
save D D
if imaged==1
    for p=numlam+1:numphi+numlam
        figure(p)
        imagesc(d(:,:,p-numlam))
        colormap(gray(256))
        axis image
        title(['Synthetic Image #' num2str(p-numlam) ''])
    end
end
save dd d

```

A.4 Pseudo-inverse Reconstruction Code

The following code comprises the “reco_inline.m” script file:

```

function [c, C, H, Hinv]=reco_inline(D,W,method,epss,epstest,zerop,cfsu,forcesum,sumofd)
%This is the initial pseudoinverse reconstruction.
%Recon4 differs from recon3 in that I changed the spatial frequency to corner, to zero middle spatial
frequency
%The method used is the matrix inversion from Mooney et al
%Dpts will be numlamx1 lexlength times containing the k,h point from all D matrices
[oheight owidth numlam numphi]=size(W);
coldfswidth=floor(numlam/2);
lexlength=oheight*owidth;
Dpts=zeros(numphi,lexlength);
for j=1:numphi
    DT=D(:,:,j);
    DT=DT(:).';
    Dpts(j,:)=DT;

```

```

end
clear D; clear DT;
%AUTOMATIC H AND HINV LOADER, MAKER, SAVER
% haveH=0;
fileH=['H_' num2str(owidth) '_' num2str(numlam) '_' num2str(numphi) "];
fileHinv=['Hinv_' num2str(owidth) '_' num2str(numlam) '_' num2str(numphi) '_' num2str(method) '_' num2str(epss)
"];
flag=[num2str(oheight) num2str(numlam) num2str(numphi)];
hmade=exist([num2str(fileH) '.mat'],'file');
hinvmade=exist([num2str(fileHinv) '.mat'],'file');
fileBkp=['Bkp_' num2str(owidth) '_' num2str(numlam) '_' num2str(numphi) '_' num2str(method) '_' num2str(epss)
"];
bkpmade=exist([num2str(fileBkp) '.mat'],'file');
if hmade==2
    clear W
    load([num2str(fileH) '.mat']);% Load Hinv off the disk.
end
if hinvmade==2
    fprintf('Loading Hinv...\n');
    load([num2str(fileHinv) '.mat']); % Load Hinv off the disk.
end
if hmade==0;
%Calculation of the big H matrix, were each H(v,u,g) corresponds to one spatial frequency g.
    fprintf(['Creating H for ' num2str(numlam) ' bands and ' num2str(numphi) ' rotations of dimension
    ',...
    num2str(oheight) ' x ' num2str(owidth) '...\n']);
    H=zeros(numphi,numlam,lexlength);
    g=1;
    for h=1:owidth;
        for k=1:oheight;
            HT=W(k,h,:,:); %W is oheight,owidth,numlam,numphi
            HT=HT(:)'; %HT=HT(:)';
            Htemp=reshape(HT,numlam,numphi);
            H(:,g)=Htemp'; %Take off the transpose ot see the affect
            g=g+1;
        end
    end
end
%The above works, dont break it!
clear W
end
%% Done making H
if hinvmade==0
    fprintf(['Calculating Hinv for ' num2str(numlam) ' bands and ' num2str(numphi) ' rotations of dimension
    ' num2str(oheight) ' x ' num2str(owidth) '...\n']);
    U=zeros(numphi,numlam);
    V=zeros(numlam,numlam);Sinv=V;
    info.clearance=0.5;
    info.msg='Processing Matrix Inversion';
    p=progbar(info);
    epssqrd=epss942;%Fudge factor
    Hinv=zeros(numlam,numphi,lexlength); %Define Hinv to speed it up. Hinv is about 16meg for 64x64
scenes
    warning off
    for k=1:lexlength
        [U,S,V] = svd(H(:,k),0);
        G=diag(S);
        %Original way to get G
        %%START ALT
        if method==1
            for h=1:numlam
                if G(h)>epss
                    G(h)=1/G(h); %Threshold Inverse Method
                else
                    G(h)=0;
                end
            end
        end
    end
end

```

```

        end
    else
        for h=1:numphi
            G(h)=G(h)/(G(h).942+epssqrd); % Wiener Inverse Method
        end
    end
end
%% END ALT
for j=1:length(G)
    Sinv(j,j)=G(j);
end
Hinv(:, :, k)=V*Sinv*U'; %this line is the speed killer
%Or do the following, does it speedup? IT DOES!
UT=U'; SU=zeros(numlam, numphi);
for j=1:numlam
    SU(j, :)=Sinv(j, j).*UT(j, :);
end
Hinv(:, :, k)=V*SU;
pro=k*100/lexlength; %Progress Bar
progbars(p, pro) %Progress Bar
end
warning on
progbars(p, -1)
if epstest==0
    save([num2str(fileHinv) '.mat'], 'Hinv')
end
if bkpmade==2
    H='erased';
% Set H and Hinv to dummy variables since they are not needed to make Bkp.
end

end
clear U; clear S; clear V; clear UT; clear SU;
C=zeros(numlam, lexlength);
info.clearance=0.5;
fprintf('Applying Hinv to the Data...\n');
info.msg='Applying Hinv to the Data';
p=progbars(info);
for g=1:lexlength
    C(:, g)=Hinv(:, :, g)*Dpts(:, g);
    pro=g*100/lexlength; %Progress Bar
    progbars(p, pro) %Progress Bar
end
progbars(p, -1)
clear Dpts
if bkpmade==2
    Hinv='erased';
end
for j=1:numlam
    Csq(:, :, j)=reshape(C(j, :), oheight, owidth);
end
for j=1:numlam
    c(:, :, j)=ifft2((Csq(:, :, j)));
end
clear Csq %To save some RAM
c=real(c);
csum=sum(sum(sum(c)));
factor_pre=sumofd/csum
%Negative Check Constraint
if zerop==1
    Index=find(c<0);
    c(Index)=0;
    clear Index
end
% %Constrain the result by the known field stop
if cfsu==1
    fprintf('Subjecting Field Stop Constraint...\n');

```

```

        c(1:coldfswidth, :, :)=0;
        c(oheight-coldfswidth:oheight, :, :)=0;
        c(:, 1:coldfswidth, :)=0;
        c(:, owidth-coldfswidth:owidth, :)=0;
    end
    if forcesum==1
        csum=sum(sum(sum(c)));
        factor=sumofd/csum
        c=c*factor;
    end
end

```

A.5 SVD-POCS Code

The following code comprises the “pocsfun.m” script file:

```

function [c2, dL, dLratio, NRMSEmean_it, NRMSEFSmean_it, NMREit, NMREFSmean_it]=...
    pocsfun(c, C, o, numphi, iterations, L, method, imgnum, warmwidth, warm, warmcheck,...
        H, Hinv, epss, cfsu, zerop, rmeanpocs, sumofd, forcesum)
%This script performs the pocs algorithm on the initial pseudoinverse reconstruction
%Initialize
clear error_sep_last
if length(H)>10
    [numphi, numlam, lexblah]=size(H); %Needed to get numphi
end
[oheight, owidth, numlam]=size(c);
lexlength=oheight*owidth;
stop=0;
coldfswidth=floor(numlam/2);
%tic
%%OBJECT DOMAIN CONSTRAINT PREP%%
%*****
%Remove the mean of c (cmeanremoved)
cmean=squeeze(mean(mean(c,1)));
cmeanremoved=zeros(size(c));
for k=1:numlam
    cmeanremoved(:,k)=c(:,k)-cmean(k);
end
%Create lexicographically orded c (clexicon)
clexicon=zeros(numlam,lexlength);
for k=1:numlam
    ctemp=cmeanremoved(:,k);
    clexicon(k,:)=ctemp(:)';
end
%Find the SVD of Rff
Rff=clexicon*clexicon'; %Hermetian here did nothing
[A,LAM2,J] = svd(Rff);
clear Rff; clear J; clear clexicon;
%%CHOOSE L%%
%Examine the eigenvalues, look at the total
dL=diag(LAM2);
dsum=sum(dL);
dLratio=dL/dsum;
fprintf(['Applying SVD-POCS for ' num2str(iterations) ' iterations with L= ' num2str(L) ' ...\n']);
Il=zeros(size(A));
for k=1:L
    Il(k,k)=1;
end
Pf=A*Il*A.';
Al=A(:,1:L);
%*****
%Transform cmeanremoved, then rearrange lexicographically to form (Cnemeanlex)

```

```

Cnmeanlex=zeros(size(C));
for k=1:numlam
    ctemp=fft2(cmeanremoved(:,:,k));
    Cnmeanlex(k,:)=ctemp(:).';
end
clear cmeanremoved
%*****
%Warm Fieldstop Prep%%
%*****
if warm==1
    load bbgo %load the known blackbody values
    for k=1:numlam
        v1=c(coldfswidth+1:coldfswidth+warmwidth, coldfswidth+1+warmwidth:...
            owidth-coldfswidth-1-warmwidth,k)-bb(k);
        v2=c(oheight-coldfswidth-warmwidth:oheight-coldfswidth-1,coldfswidth...
            +1+warmwidth:owidth-coldfswidth-1-warmwidth,k)-bb(k);
        v3=c(coldfswidth+1:oheight-coldfswidth-1,coldfswidth+1:coldfswidth...
            +warmwidth,k)-bb(k);
        v4=c(coldfswidth+1:oheight-coldfswidth-1,owidth-coldfswidth-warmwidth:...
            owidth-coldfswidth-1,k)-bb(k);
        value=[v1(:).', v2(:).', v3(:).', v4(:).'];
        RMSEinFS_initial(k)=sqrt((sum(value.^2))/length(value));
    end
    [m,n] = size(c(coldfswidth+1+warmwidth:oheight-coldfswidth-1-warmwidth,...
        coldfswidth+1+warmwidth:owidth-coldfswidth-1-warmwidth,1));
    opseudomean=sum(sum(sum(o)))/((m+2*warmwidth)*(n+2*warmwidth)*numlam);
    FSError = RMSEinFS_initial./opseudomean.*100;
end
%%%TRANSFORM DOMAIN CONSTRAINT PREP%%%
%*****
fileBkp=['Bkp_' num2str(owidth) '_' num2str(numlam) '_' num2str(numphi) '_'...
    num2str(method) '_' num2str(epss) "];
bkpmade=exist([num2str(fileBkp) '.mat'],'file');
if bkpmade==2
    clear H; clear Hinv;
    load([num2str(fileBkp) '.mat']);% Load Bkp off the disk.
else
    %%%%%%%%% Start the uncomment below if B is not available. H is used.
    fprintf('Calculating Hinv*H ...\n');
    B=zeros(numlam,numlam,lexlength);
    info.clearance=1.0;
    info.msg='Calculating Hinv*H ...';
    h=progbar(info);
    for g=1:lexlength
        T=Hinv(:,:,g)*H(:,:,g);
        [B1 I B2]=svd(T);
        clear I; clear B2;
        B(:,:,g)=B1;
        pro=g*100/lexlength; %Progress Bar
        progbar(h,pro) %Progress Bar
    end
    progbar(h,-1)
    clear Hinv;
    % The number r comes from the rank at that particular spatial frequency.
    % Get the rank
    fprintf(['Finding the rank of H...\n']);
    for k=1:lexlength
        r(k)=rank(H(:,:,k));
    end
    clear H;
    Bk=B;
    clear B
    %Now replace the first r columns of B with zeros. (Bk)
    fprintf(['Finding Bk...\n']);
    for k=1:lexlength

```

```

        for p=1:r(k)
            Bk(:,p,k)=zeros(numlam,1);
        end
    end
    %To speed up the iterations, pre-multiply Bk(:, :,g)*Bk(:, :,g) (Bkp)
    fprintf(['Pre-Multiplying Bk*Bk.' ' ' '...\n']);
    info.clearance=1.0;
    info.msg='Pre-Multiplying Bk';
    p=progbar(info);
    for g=1:lexlength
        Bkp=Bk(:, :,g)*Bk(:, :,g)';
    % *For memory, I will reuse the Bk matrix to store Bkp values
        Bk(:, :,g)=Bkp; clear Bkp;
        pro=g*100/lexlength; %Progress Bar
        progbar(p,pro) %Progress Bar
    end
    progbar(p,-1)
    Bkp=Bk; % Now the Bkp Matrix values have completely replaced the Bk values.
    clear Bk
    save([num2str(fileBkp) '.mat'],'Bkp')

end
Clex=Cnemeanlex;
clear Cnemeanlex
%%ITERATIONS%%
*****
info.clearance=1.0;
if rmeanpocs==1
    info.msg=['Applying POCS for ' num2str(iterations) ' iterations for L='...
        num2str(L) ' and image ' num2str(imgnum) "];
elseif rmeanpocs==0
    info.msg=['Applying POCS for ' num2str(iterations) ' iterations for L='...
        num2str(L) ' and image ' num2str(imgnum) "];
end
ppp=progbar(info);
%-----
for m=1:iterations
%%APPLICATION OF OBJECT DOMAIN CONSTRAIN%%
    Q=Pf*Clex;
%%APPLICATION OF TRANSFORM DOMAIN CONSTRAIN%%
    P=zeros(numlam,lexlength);
    for g=1:lexlength
        P(:,g)=Bkp(:, :,g)*Q(:,g);
    end
    clear Q
%%ADD TO THE ORIGINAL RECONSTRUCTION
%-----
    Cnew=C+P;
    clear P
%-----
%%Fieldstop Constraint%
%-----
    Cs2=zeros(oheight,owidth);
    c2=zeros(size(c));
    for k=1:numlam
        Cs2=reshape(Cnew(k,:),oheight,owidth);
        c2(:, :,k)=ifft2(Cs2);
    end
    c2=real(c2);
    clear Cnew
    if zerop==1
        Index=find(c2<0);
        c2(Index)=0;
        clear Index
    end
end

```

```

        if cfsu==1
            c2(1:coldfswidth, :, :)=0;
            c2(oheight-coldfswidth:oheight, :, :)=0;
            c2(:, 1:coldfswidth, :)=0;
            c2(:, owidth-coldfswidth:owidth, :)=0;
        end
        if forcesum==1
            c2sum=sum(sum(sum(c2)));
            factor=sumofd/c2sum;
            c2=c2*factor;
        end
        %Remove the mean, transform, and reshape to the lexicographical form
        %-----
        if rmeanpocs==1
            c2mean=squeeze(mean(mean(c2,1)));
            for k=1:numlam
                ctemp=c2(:, :, k)-c2mean(k);
            %This only affects NRMSE. if we subtract the mean, the final mean is
            %equivalent for all bands. The NMRE is unchanged.
                Ctemp=(fft2(ctemp));
                Clex(k,:)=Ctemp(:).';
            end
            clear ctemp; clear Ctemp;
        else
            %Transform and reshape to the lexicographical form
            %-----
            for k=1:numlam
                Ctemp=(fft2(c2(:, :, k)));
                Clex(k,:)=reshape(Ctemp,1,lexlength);
            end
        end
        %-----
        %Find the PSNR for each Iteration
        %-----
        for k=1:numlam
            [NRMSE_it(k) NMRE_it(k)]=getnrmse(c2(coldfswidth+1+warmwidth:oheight...
            -coldfswidth-1-warmwidth,coldfswidth+1+warmwidth:owidth-coldfswidth...
            -1-warmwidth,k),o(coldfswidth+1+warmwidth:oheight-coldfswidth-1-warmwidth...
            ,coldfswidth+1+warmwidth:owidth-coldfswidth-1-warmwidth,k),o,warmwidth);
        end
        NRMSEmean_it(m)=mean(NRMSE_it);
        ind=[8 9 10 11 12 13 14 20 21 22 23 24 25];
        NMREit(m)=mean(NMRE_it(ind)); %Special metric only good for tempmap source
        %-----
        %Find the mean for each Iteration
        %-----
        Meanit(m)=mean(mean(mean(c2)));
        %-----
        %Warm Fieldstop Monitor%
        %-----
        NRMSEFSmean_it(m)=0;
        NMREFSmean_it(m)=0;
        VARinFS_it(m)=0;
        if warm==1 %Overall reconstruciton is better, but may be due to using warm field stop and smaller
        image
            fprintf('Running Warm Fieldstop Monitor...\n');
            for k=1:numlam
                rangev1_1=coldfswidth+1:coldfswidth+warmwidth;
                rangev1_2=coldfswidth+1+warmwidth:owidth-coldfswidth-1-warmwidth;
                v1=c2(rangev1_1,rangev1_2,k)-bb(k);
                v1mr=c2(rangev1_1,rangev1_2,k)-mean(mean(c2(rangev1_1,rangev1_2,k)));
            %The mean blackbody value is equal to the value, so they cancel out
                rangev2_1=oheight-coldfswidth-warmwidth:oheight-coldfswidth-1;
                rangev2_2=coldfswidth+1+warmwidth:owidth-coldfswidth-1-warmwidth;
                v2=c2(rangev2_1,rangev2_2,k)-bb(k);
            end
        end
    end
end

```

```

v2mr=c2(rangev2_1,rangev2_2,k)-mean(mean(c2(rangev2_1,rangev2_2,k)));
rangev34_1=coldfswidth+1:height-coldfswidth-1;
rangev3_2=coldfswidth+1:coldfswidth+warmwidth;
v3=c2(rangev34_1,rangev3_2,k)-bb(k);
v3mr=c2(rangev34_1,rangev3_2,k)-mean(mean(c2(rangev34_1,rangev3_2,k)));
rangev4_2=owidth-coldfswidth-warmwidth:owidth-coldfswidth-1;
v4=c2(rangev34_1,rangev4_2,k)-bb(k);
v4mr=c2(rangev34_1,rangev4_2,k)-mean(mean(c2(rangev34_1,rangev4_2,k)));
value=[v1(:).' v2(:).' v3(:).' v4(:).'];
valuemr=[v1mr(:).' v2mr(:).' v3mr(:).' v4mr(:).'];
if max(ind==k)==1 %To only use spectra used elsewhere in NVE calculations
    VARinFS(k)=var(value);
end
RMSEinFS(k)=sqrt((sum(value.^2))/length(value));
MREinFS(k)=sqrt((sum(valuemr.^2))/length(valuemr));
end
NRMSEinFS = RMSEinFS./opseudomean.*100;
NRMSEFSmean_it(m)=mean(NRMSEinFS);
NMREinFS = MREinFS./opseudomean.*100;
NMREFSmean_it(m)=mean(NMREinFS);
VARinFS_it(m)=mean(VARinFS);
error_separation=mean(FSError-NRMSEinFS)
end
if warmcheck==1
%The following Routine watches for convergence or divergence of the error
if exist('error_sep_last') %Halter is commented out for testing L
sep=error_separation-error_sep_last;
if sep<0
    if stop==1
        fprintf(['DIVERGENCE REACHED! Halted after ' num2str(m-1) ' iterations. \n'])
        fprintf(['Output is likely best after ' num2str(m-2) ' iterations. \n'])
        c2=cout;
        break;
    end
    cout=c2;
    stop=1;
end
if sep<=1 & sep>=0
    fprintf(['CONVERGENCE REACHED! Halted after ' num2str(m) ' iterations. \n'])
    break;
end
end
error_sep_last=error_separation;
end
%-----
pro=m*100/iterations; %Progress Bar
progbar(ppp,pro) %Progress Bar
end
progbar(ppp,-1)
%*****
%%End Iterations
fprintf(['Stopped after ' num2str(m) ' iterations. \n'])

```

Appendix B. Additional Figures

B.1 Warm Field Stop NVE Versus Iteration

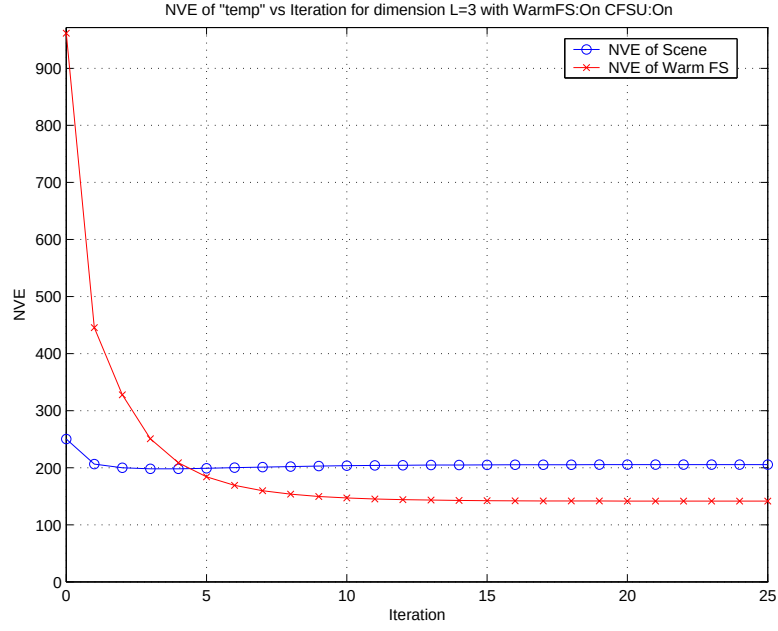


Figure 80 The NVE of the warm field stop also decreases, but does not follow the exact curve for MSP for a warm field temperature of 300K.

Bibliography

1. Murguia, James E. and Toby Reeves and Jonathan Mooney and William Ewing and Freeman Shepard and Andrzej Brodzik. *A Compact Visible/Near Infrared Hyperspectral Imager*. Technical Report, Solid State Scientific Corporation, June 2003. From Unpublished article. n. pag. <http://www.solidstatescientific.com/projects/cthis/>.
2. Mooney, Jonathan M. "Angularly Multiplexed Spectral Imager." *Imaging Spectrometry I, Proc. SPIE 2480*. 66–77. 1995.
3. Mooney, Jonathan M. and Andrzej Brodzik and Myoung An. "Principal Component Analysis in Limited Angle Chromotomography." *Imaging Spectrometry III, Proc. SPIE 3118*. 170–178. 1997.
4. Brodzik, Andrej K. and Jonathan M. Mooney and Myoung An. "Image Restoration by Convex Projections: Application to Image Spectrometry." *Imaging Spectrometry II, Proc. SPIE 2819*. 231–242. 1996.
5. An, Myoung and Andrzej Brodzik and Jonathan Mooney and Richard Tolimieri. *Data Restoration in Chromotomographic Hyperspectral Imaging*. Technical Report, Solid State Scientific Corporation, June 2003. From Unpublished article. n. pag. <http://www.solidstatescientific.com/projects/cthis/>.
6. Lim, Jae S. *Two-Dimensional Signal and Image Processing*. Upper Saddle River, New Jersey: Prentice Hall PTR, 1990.
7. Mooney, Jonathan M. and Virgil Vickers and Myoung An and Andrzej Brodzik. "High Throughput Hyperspectral Infrared Camera," *J. Opt. Soc. Am., A, Vol 14*(No. 11):2951–2961 (November 1997).
8. Strang, Gilbert. *Linear Algebra and its Applications, Third Edition*. San Diego: Harcourt Brace Jovanovich, Inc., 1988.
9. Brodzik, Andrej K. and Jonathan M. Mooney. "Convex Projections Algorithm for Restoration of Limited-Angle Chromotomographic Images," *J. Opt. Soc. Am., A, Vol 16*(No. 2):246–257 (1999).
10. Duda, Richard O. and Peter E. Hart and David G. Stork. *Pattern Classification*. New York, NY: John Wiley & Sons, 2001.
11. Combettes, Patrick L. "The Foundations of Set Theoretic Estimation." *Proc. IEEE 81 (No.2)*. 182–208. 1993.
12. Lord, Steven D. *A New Software Tool for Computing Earth's Atmospheric Transmission of Near- and Far-Infrared Radiation*. Technical Report, Gemini Observatory, 1992. NASA Tech. Memo. 103957.

13. “CCD Image Sensor Noise Sources (Rev 1),” August 2001. From Unpublished article. n. pag. <http://www.kodak.com/global/plugins/acrobat/en/digital/ccd/>.
14. Shanmugan, K. Sam and A.M. Breipohl. *Random Signals: Detection, Estimation, and Data Analysis*. New York, NY: John Wiley & Sons, 1988.
15. Shannon, Claude E. “Coding Theorems for Discrete Source with a Fidelity Criterion.” *Institute of Radio Engineers National Convention Record, Part 4*. 142–163. 1959.

Vita

Captain Kevin Carl Gustke was born in 1976 in Lubbock, Texas. He graduated from Cabot High School, Cabot, Arkansas in 1994. Gustke then matriculated at the University of Missouri-Rolla in Rolla, Missouri. Graduating with honors in 1999 and receiving his B.S. in Electrical Engineering, he accepted a commission to the United States Air Force. In June 1999, Gustke went active duty and was assigned to the National Air Intelligence Center, Wright-Patterson Air Force Base, Ohio. Upon completion of his first assignment, Gustke was assigned to the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio to pursue a Master of Science degree in Electrical Engineering. Upon graduation, Captain Gustke will transfer to Patrick Air Force Base, located in the sunshine state of Florida, and work for the Air Force Technical Applications Center.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 23-03-2004		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From – To) Jun 2003 – Mar 2004	
4. TITLE AND SUBTITLE RECONSTRUCTION ALGORITHM CHARACTERIZATION AND PERFORMANCE MONITORING IN LIMITED-ANGLE CHROMOTOMOGRAPHY				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Gustke, Kevin, C., Captain, USAF				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way Street, Building 640 WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/04-12	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) NASIC/DEMS Attn: Ms. Susan Boyd 4180 Watson Way WPAFB OH 45433-5648 DSN: 787-3615 e-mail: Susan.Boyd@wpafb.af.mil				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Hyperspectral data collection and analysis is an increasing priority with the growing need to obtain greater classification precision than offered by traditional spatial imagery. In this thesis, trends in hyperspectral chromotomographic reconstruction are explored where reconstruction is performed using a series of spatial-chromatic images. Chromotomography involves capturing a series of two-dimensional images where each image is created by placing a prism in front of the focal plane array; causing spectral dispersion corresponding to a series of prism angles over a single rotation. Before testing reconstruction, synthetic data is produced, approximating what would be produced from prism dispersion on the focal plane array. The pseudo-inverse singular matrix problem is addressed where two methods are compared to find which produces minimal error. The standard iterative error reduction algorithm, SVD-POCS, is shown to be incapable of reconstructing the mean of the source scene, making absolute radiometry analysis impractical. However, SVD-POCS is shown to provide the least error if the goal is to perform relative radiometry analysis. Additional constraints are needed to make absolute radiometry analysis possible. The added constraints of non-negativity, spatial extent of the cold field stop, forcing the sum, and keeping the mean for each iteration improves absolute radiometric performance. These additional constraints also allow use of a warm field stop to monitor reconstruction error for both the pseudo-inverse and iterative improvement algorithm. Error can be calculated each iteration to ascertain when a minimum has been reached in a mean square error sense. Thus, minimum mean square error of the reconstruction can be obtained with confidence.					
15. SUBJECT TERMS Chromotomography, Hyperspectral imagery, Convex-projections, Spectrometry, Infrared, Iterative algorithms, Tomography, Principal Component Analysis, Spectrum analysis, Image reconstruction					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)
U	U	U	UU	141	Stephen C. Cain (ENG) (937) 255-3636, ext 4625; e-mail: Stephen.Cain@afit.edu