Anomaly Detection and Compensation for Hyperspectral Imagery

by

Choongyeun Cho

B.S. Electronics Engineering, Korea University (1998)
S.M. Electrical Engineering and Computer Science,
Massachusetts Institute of Technology (2001)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2005

© Massachusetts Institute of Technology 2005. All rights reserved.

Author .....
Department of Electrical Engineering and Computer Science
September 2, 2005

Certified by .....
David H. Staelin
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Anomaly Detection and Compensation for Hyperspectral Imagery

by

Choongyeun Cho

Submitted to the Department of Electrical Engineering and Computer Science
on September 2, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Hyperspectral sensors observe hundreds or thousands of narrow contiguous spectral bands. The use of hyperspectral imagery for remote sensing applications is new and promising, yet the characterization and analysis of such data by exploiting both spectral and spatial information have not been extensively investigated thus far. A generic methodology is presented for detecting and compensating anomalies from hyperspectral imagery, taking advantage of all information available – spectral and spatial correlation and any a priori knowledge about the anomalies. An anomaly is generally defined as an undesired spatial and spectral feature statistically different from its surrounding background.

Principal component analysis (PCA) and the Iterative Order and Noise (ION) estimation algorithm provide valuable tools to characterize signals and reduce noise. Various methodologies are also addressed to cope with nonlinearities in the system without much computational burden. An anomaly compensation technique is applied to specific problems that exhibit different stochastic models for an anomaly and its performance is evaluated. Hyperspectral anomalies dealt with in this thesis are (1) cloud impact in hyperspectral radiance fields, (2) noisy channels and (3) scan-line miscalibration. Estimation of the cloud impact using the proposed algorithm is especially successful and comparable to an alternative physics-based algorithm. Noisy channels and miscalibrated scan-lines are also fairly well compensated or removed using the proposed algorithm.

Thesis Supervisor: David H. Staelin
Title: Professor of Electrical Engineering

Acknowledgments

First of all, I am very grateful to my research supervisor, Professor David Staelin, for his guidance, support, and patience. He is my role model as a teacher, a mentor, and an engineer. He showed me what it takes to be a genuine and natural engineer, being able to map almost any real-world situation to a relevant engineering problem. I also thank the thesis readers – Prof. Alan Willsky, Dr. Phil Rosenkranz, and Dr. Bill Blackwell – for their valuable feedbacks and suggestions.

I would also like to appreciate the unbounded love and cheer from my family: my wife, Sung Yun Choi, my parents, Namjo Cho, Gilja Kwon, my parents-in-law, Namsick Choi, Daisill Kim, my brother, Sangyeun, and my sister, Heunjung. Without their support and prayers I could not have come to the United States, receiving the finest education in the first place. Sung Yun has been and will continue to be the sunshine of my life. How lucky I am to be able to get both Ph.D. from MIT and the most beautiful wife!

Thanks to all the Remote Sensing and Estimation Group colleagues, staff and alumni: Phil Rosenkranz, Laura von Bosau, Scott Bressler, Pop (Chinnawat) Surussavadee, Siddhartan Govindasamy, Danielle Hinton, Keith Herring, Jack Holloway, Fred Chen, Bill Blackwell, R. Vince Leslie, Amy Mueller, Jessica Loparo, Filip Antic, Amanda Leckman, Andrew Sanchez, Carri Chan, and Seth Hall. Bill has graciously provided me helpful resources such as a huge AIRS data set which kept me busy for a while. Carri's handsome UROP work is also thankfully acknowledged, which has become part of this thesis.

Friendship with Korean friends at MIT enormously enhanced my life here. Thanks to Jihwan Choi, Seongmoo Heo, Junmo Kim, who along with me entered MIT EECS from Korea in 1998, for sharing good times and good *Tekken* fights with me. Billiard games with Seonghwan Cho were always terrific refreshment, well, unless I lost. I especially cherish the pleasant and often crazy memory with Taehong Park, Sejung Park, Songjun Park, Chanwoo Park, and Sungtae Kim. When I was a newbie at MIT, Sungtae once threateningly told me a myth: one can never graduate from MIT

until he (cumulatively) eats up 1000 bowls of dumpling soup at the Golden Gate restaurant[1] in Chinatown. I believe when the curse of the Bambino was reversed in 2004[2], so was the curse of the Golden Gate: I have had only 30 bowls so far.

I am greatly indebted to Onnuri Church *Soon* family: especially, Youngsub Yoon, Baeyoung Kim, Eunjoo Chang, Jonghyup Lee, Yoonjoo Kim, Sunyoung Lee, and Shinyoung Park, who didn't spare the moral and spiritual support for me.

I am proud to be a member of New England Korea University Alumni Association (NE-KUAA) through which I had so many enjoyable occasions. The experience with KU alumni here was so invaluable that I already expect to have the same fun with New York KUAA.

Among countless memorable moments, I am sure I will most miss the allnighter Hoola games with Kyungjin Lee, Eesook An, Won Ik Lee, Soojong Chin, Daniel An and Jiwon Hwang. Special thanks to Won Ik who ushered me to the evil world of Hoola.

I gratefully acknowledge the financial support from Korea Foundation for Advanced Studies (KFAS), National Aeronautics and Space Administration (NASA) under contracts NAS5-31376, NNG04HZ51C, and National Oceanic and Atmospheric Administration (NOAA) under contract DG133E-02-CN-0011.

Last but not least, I am most obliged to my God who has been with me all the time – good times and not-so-good times – and gave me strength to move on. Thank you and I love you, Lord!

[1] Once forced to shut down due to the violation of public sanitary code.

[2] In my first semester in 1998, the last 6.432 problem set was due "next time the Red Sox win the World Series" which happened to be October 27th 2004.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The evolution of passive remote sensing has witnessed the collection of measurements with significantly greater spectral breadth and resolution. Hyperspectral sensors collect a spectrum from each point in a scene. They differ from multispectral sensors in that the number of bands is much higher (hundreds or even thousands) and the spectral bands are generally contiguous. For remote sensing applications, they are typically deployed on either aircraft or satellites. The data product from a hyperspectral sensor is a three-dimensional array or "cube" of data with the width and length of the array corresponding to spatial dimensions and the spectrum of each point as the third dimension. Figure 1-1 shows a 224-spectral-band image cube observed from the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [46]. Hyperspectral imaging from airborne or spaceborne sensors has broad applications in resource management, agriculture, mineral exploration, environmental monitoring, and weather forecasting.

1.1 Problem statement

One of the challenges associated with the analysis and processing of a hyperspectral image is that it often contains undesired attributes. Real-world data often have unwanted artifacts of several different natures. They can be various types of noise or instrument miscalibration, or just contributions from a benign but unwanted source.

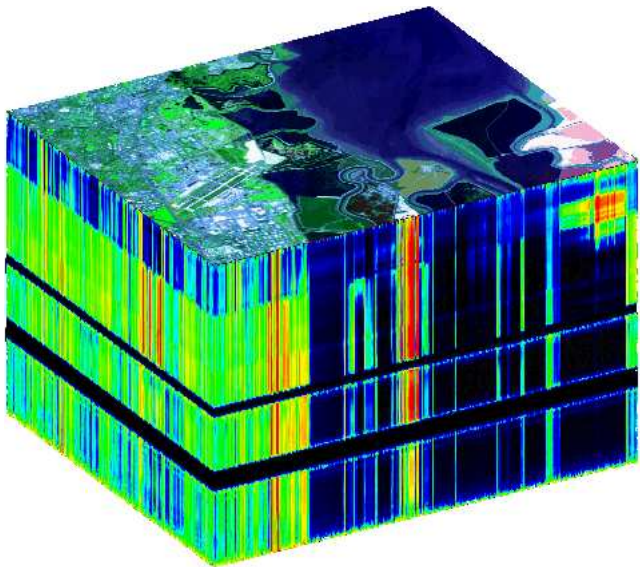


Figure 1-1: Hyperspectral image cube from Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [46]

For example, if it is only atmosphere that we are investigating, then the surface effect needs to be eliminated, or vice versa. In the thesis, I will establish an efficient method to detect and compensate for the unwanted contributions in hyperspectral images.

An anomaly is defined as an unwanted spatial and spectral signature that is statistically distinct from its surrounding. The statistics and a priori knowledge about an anomaly and its surrounding background is specific to an application at hand.

A hyperspectral data set, $\mathbf{X}$, in "radiance" or reversibly "brightness temperature" as will be discussed in the subsequent chapter, can be represented as a $I \times J$ matrix, with I being the number of channels, and J the number of pixels. Each element of data matrix $\mathbf{X}$ has an additive anomaly component:

$$x_{i,j} = \tilde{x}_{i,j} + \delta_{i,j} \tag{1.1}$$

where $\tilde{x}_{i,j}$ is original anomaly-free data at i-th pixel and j-th channel and $\delta_{i,j}$ represents an anomaly, an element of the anomaly matrix, Δ. Given $\mathbf{X}$ and a priori information about Δ, what is the best estimate of Δ, or equivalently $\tilde{\mathbf{X}}$, and how is it obtained? A priori information about an anomaly can have different forms:

- Statistical description, e.g. mean, covariance matrix, n-dimensional PDF of an anomaly pixel, and covariance with respect to neighboring anomalies. This statistical information is often unavailable in practice, thus needs to be estimated with ensemble of observations.
- Spatial structure or texture, e.g. having mostly high-frequency components.
- Joint spatial/spectral statistical description.

A (near) optimum solution may be obtained by cooperatively combining spatial and spectral processing so that the signal-to-noise ratio (SNR) of an anomaly of interest is successively enhanced, thereby making it increasingly prominent, permitting its removal. The preliminary results of spectral processing based on the ONA (Order, Noise and mixing matrix A) algorithm [33], for example, show that noise can be effectively suppressed solely using spectral information of a data set without using any prior knowledge about an underlying physical model that represents the data. Spatial structures of the data can, in turn, be employed to further increase the SNR [23].

1.1.1 Examples of anomalies

A generic anomaly detection technique will be applied to specific hyperspectral problems that exhibit different stochastic models for an anomaly:

- Cloud perturbation in a hyperspectral radiance field.
- Noisy channels.
- Scan-line miscalibration.

Each of these applications will be dealt with in depth in subsequent chapters.

1.2 Prior art

Since hyperspectral sensor technology is emerging and new, only in the past decade has there been much active research on hyperspectral data analysis in the remote-sensing communities [16, 21, 22, 36, 38]. Most of this work, however, has focused on

the reflective regime of the radiance data (not the emissive one) and on the problem of classifying objects, so-called spectral unmixing, located on a surface (not in the atmosphere). Moreover, it has emphasized applying algorithms intended for multispectral data to hyperspectral data by reducing the number of spectral bands used for processing. These algorithms fall into three basic categories: spatial-only, spectral-only, and spatial-spectral algorithms. The spectral-only algorithms almost all rely on a known spectral signature for the target or targets of interest. Basically, they are classification rather than detection algorithms. Algorithms that fall into this category are the spectral matched filter [16], spectral angle mapper [22], and linear mixture models [21]. The main limitation of these spectral-only algorithms is that, in addition to ignoring the available spatial information, they require a known target signature. Reliable target signatures are difficult to ascertain due to variations in the target signature that result from atmospheric and illumination effects.

Principal component analysis (PCA) is most often used prior to another detection or classification algorithm for purposes of reducing the dimensionality of the hyperspectral data sets, thus making the applied detection and classification algorithms more efficient computationally.

In general, spatial-spectral algorithms can be further divided into local anomaly and global anomaly detectors.

Local anomaly detectors process small spatial windows of the hyperspectral image in order to compare the spatial and spectral properties of the centrally located pixels in the window (target region) with the properties of the perimeter pixels (clutter region). Those pixels that are spatially-spectrally different from their surrounding backgrounds are considered detections. Reed and Yu [35] derived an algorithm commonly referred to as the RX-algorithm. This is the benchmark anomaly detection algorithm for multispectral data, which, in contrast to hyperspectral data, is characterized by less than about 20 spectral bands. The RX-algorithm is a maximum likelihood (ML) anomaly detection procedure that spatially whitens the clutter. The RX-algorithm uses a binary hypothesis approach to detection, and implements a generalized likelihood ratio test (GLRT). Evaluation of the ML-detection statistic re-

quires full spectral sample covariance matrices to be estimated and then inverted, or the evaluation of their determinants. Suppose that J is the number of spectral bands and $\mathbf{x}$ is an $J \times 1$-column pixel vector in a multispectral image. Then the RX-detector implements a filter specified by:

$$\mathrm{RX}(\mathbf{x}) = (\mathbf{x} - \mu)^T \mathbf{C}_{\mathbf{xx}}^{-\mathbf{1}} (\mathbf{x} - \mu) \tag{1.2}$$

where μ is the global sample mean and $\mathbf{C}_{\mathbf{xx}}$ is the sample covariance matrix of the images.

Extending application of the RX-algorithm from multispectral to hyperspectral imagery suffers from two major limitations. First, the clutter model implemented in the RX-algorithm is restricted to being spatially uncorrelated, or spatially white. This model neglects the potentially valuable spatial correlation information of the clutter. Second, its computational cost arises from the expensive inversion of the covariance matrix of the data under each of the hypotheses. The computational complexity of this detector increases as J^3 where J is the number of spectral bands, making it difficult for hyperspectral images.

Ferrara [17] presented an adaptive spatial-spectral detection method in which it is originally assumed that the clutter is fully spatially and spectrally correlated. However, in evaluating the spatial-spectral covariance matrix the cross-covariance terms are neglected in order to reduce the computational cost of the algorithm. This algorithm suffers from the same computational complexity as the RX-algorithm when applied to hyperspectral imagery: It requires taking the inverse of a spectral covariance matrix that has dimensions equal to the number of spectral bands used for processing.

In global anomaly detection, the image scene is first segmented into its constituent classes, then detection is achieved by determining the outliers of these classes. In general, the algorithms vary in the method of segmentation, but tend to use ML-detection once the classes are determined. One approach to global anomaly detection is stochastic expectation maximization (SEM) coupled with ML-detection [39]. This

algorithm uses the stochastic expectation maximization clustering algorithm as a preprocessing stage to the detector. The number of classes is assumed to be known a priori. There is need for computationally efficient detectors for hyperspectral sensors that can jointly process all the available spectral bands, and that can exploit simultaneously the spatial and spectral correlation properties of the anomaly.

The majority of global anomaly detectors employ two-dimensional Markov random field (MRF) modeling in order to incorporate spatial features into the segmentation process, since MRFs have been proven to be quite powerful models in the classification of 2-D images [36].

In [44], surface prior information reflectance estimation (SPIRE) algorithms estimate additive and multiplicative illumination noise vectors by exploiting both spatial and spectral information. The spatial SPIRE algorithm spatially filters out the additive noise, moves to log space, and then spatially filters out the multiplicative noise. The spectral SPIRE algorithm uses variant forms of the PC transform to reduce noise. The combined spatial-spectral algorithm takes advantage of both algorithms in that only a selection of PC images are spatially filtered. For all cases, prior information is employed to compensate for the signal constituent that may have been filtered out as noise.

In the context of the compression of hyperspectral data, [7] also takes advantage of spatial and spectral information. By treating data compression as a noisy source coding problem, the noise source is modeled by an additive mixture of Gaussian noise and impulsive noise, and is estimated by using both spectral and spatial knowledge of the data.

1.3 Thesis outline

Chapter 2 is reserved to briefly explain the satellite instruments, the Atmospheric Infrared Sounder and the Advanced Microwave Sounding Unit (AIRS/AMSU), that provide the brightness temperature data in the infrared and microwave range, respectively, and the radiative transfer equation which links the atmospheric and ter-

restrial parameters to hyperspectral data. In Chapter 3, we shall explore useful signal-processing tools as to how to characterize hyperspectral data, and reduce noise and artifacts. They are applied to hyperspectral data set from AIRS to demonstrate their capability to cope with various types of artifact in hyperspectral data. The physics governing the radiative transfer is nonlinear; thus, the estimation of physical parameters must deal with the nonlinear physics. We shall review the mathematical foundation of nonlinear estimation and powerful techniques to handle nonlinearities in Chapter 4. Chapter 5 describes a stochastic approach to estimate cloud perturbations in the hyperspectral infrared brightness temperatures, or equivalently, to infer the brightness temperatures that would be observed in the absence of clouds. The evaluation and validation of this stochastic cloud-clearing algorithm using four different validation schemes are covered in Chapter 6. In Chapter 7, the stability of the infrared sensor, AIRS, relative to the microwave sensor, AMSU, is analyzed. Chapter 8 provides the summary and contributions of this study, and the suggestions for future work.

Chapter 2

Background

The framework of the anomaly detection and compensation proposed in this study can be applied to any multivariate data. Examples of such data include remote sensing data, manufacturing data, multiple-antenna wireless commutation data, financial data, and so forth. Although only hyperspectral satellite data will be covered in this thesis, one can develop a specific methodology to characterize and compensate for a multivariate anomaly model at hand. In this chapter, the satellite instruments that produce the data set used in this thesis will be introduced. Also, the physics of radiative transfer that links environmental parameters to the hyperspectral data will be explained.

2.1 Instruments

2.1.1 AIRS/AMSU/HSB sounding suite

The Atmospheric Infrared Sounder (AIRS), the Advanced Microwave Sounding Unit (AMSU), and the Humidity Sounder for Brazil (HSB) form an integrated cross-track scanning temperature and humidity sounding system on the Aqua satellite of the Earth Observing System (EOS) [2]. AIRS is an infrared spectrometer/radiometer that covers 2378 spectral channels in the 3.74–4.61-μm, 6.20–8.22-μm, 8.8–15.4-μm infrared wavebands at a nominal spectral resolution of $\lambda/\Delta\lambda = 1200$. The AIRS

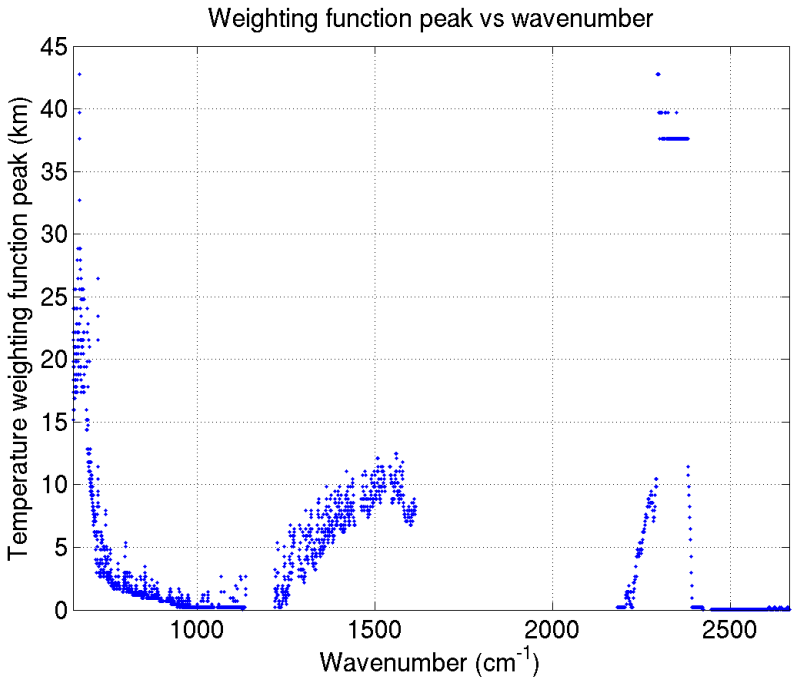


Figure 2-1: Weighting function peak versus wavenumber for 2378 AIRS channels for US standard atmosphere

instrument also includes four visible/near-IR (Vis/NIR) channels between 0.40 and 0.94 μm, with a 2.3-km field-of-view (FOV). Figure 2-1 shows the temperature weighting function peaks versus wavenumber for 2378 AIRS channels. The temperature weighting function roughly represents the contribution of pressure level (or height) to an observed radiance, and will be defined later in this chapter. AMSU on Aqua is composed of two separate sensor units, AMSU-A1 and AMSU-A2 with co-aligned, synchronized, and equal-size FOV, and covers between 23 and 89 GHz microwave spectrum with 15 channels. See Table 2.1 for the AMSU channel specification. HSB[1] is a four-channel microwave radiometer that makes measurements between 150 and 190 GHz. Figure 2-2 illustrates the AIRS/AMSU/HSB scan geometry and pattern [2]. In one 3.3° AMSU footprint, there are 3-by-3 1.1° AIRS and HSB footprints. Table 2.2 and Table 2.3 list the high-level parameters for AIRS and AMSU/HSB instruments [2].

In addition to supporting the National Aeronautics and Space Administration's

[1]The HSB instrument ceased operation on February 5, 2003 due to a mirror scan motor failure.

Channel	Channel frequencies (MHz)	Bandwidth (MHz)	Weighting function peak height (km)
1	23,800±72.5	2×125	0
2	31,400±50	2×80	0
3	50,300±50	2×80	0
4	52,800±105	2×190	0
5	53,596±115	2×168	4
6	54,400±105	2×190	7
7	54,940±105	2×190	10
8	55,500±87.5	2×155	12
9	57,290.344±87.5	2×155	17
10	57,290.344±217	2×77	21
11	57,290.344±322.2±48	4×35	24
12	57,290.344±322.2±22	4×15	28
13	57,290.344±322.2±10	4×8	35
14	57,290.344±322.2±4.5	4×3	40
15	89,000±1000	2×1000	0

Table 2.1: AMSU-channel specification

Size	116.5 × 158.7 × 95.3 cm (when deployed)
Mass	177 kg
Power	220 Watt
Data rate	1.27 Mbits/sec
Spectral range	IR: 3.74–4.61-μm, 6.20–8.22-μm, 8.8–15.4-μm 2378 channels with $\lambda/\Delta\lambda = 1200$ resolutions VIS/NIR: 0.4–0.94 μm with 4 channels
Instrumental FOV	IR: 1.1° (13.5 km at nadir from 705 km altitude) VIS/NIR: 0.2° (2.3 km at nadir from 705 km altitude)
Swath width	99° (1650 km from 705 km altitude)
Scan sampling	IR: 90$samplesat$1.1 degree VIS/NIR: 720$samplesat$0.2 degree

Table 2.2: High-level AIRS parameters

Instrument	AMSU-A1	AMSU-A2	HSB
Size	72 × 34 × 59 cm	73 × 61 × 86 cm	70 × 65 × 46 cm
Mass	49 kg	42 kg	51 kg
Power	77 Watt	24 Watt	56 Watt
Data rate	1.5 Kbits/sec	0.5 Kbits/sec	4.2 Kbits/sec
Spectral range	50–90 GHz	23–32 GHz	150–190 GHz
Number of channels	13	2	4
Instrumental FOV	3.3° (40.5km at nadir from 705 km)	3.3° (40.5km at nadir from 705 km)	1.1° (13.5km at nadir from 705 km)
Swath width	100° (1690 km from 705 km)	99° (1650 km from 705 km)	99° (1650 km from 705 km)
Scan sampling	30 × 3.33 degree	30 × 3.33 degree	90 × 1.1 degree

Table 2.3: High-level microwave instrument (AMSU-A1/AMSU-A2/HSB) parameters

interest in process study and climate research, AIRS is the first hyperspectral infrared radiometer designed to support the operational requirements for medium-range weather forecasting of the National Ocean and Atmospheric Administration's National Centers for Environmental Prediction (NCEP) and other numerical weather forecasting centers. AIRS, together with the AMSU and HSB microwave radiometers, will achieve global retrieval accuracy of atmospheric temperature profiles to better than 1 K, with 1-km vertical layers, and humidity soundings with 10% accuracy in 2-km layers in the troposphere. The limiting effects of cloud contamination in the FOV of the AIRS were quickly recognized. Cloudy infrared radiances are difficult to use in retrieval or radiance assimilation schemes since they do not generally provide accurate information on the thermodynamic state of the atmosphere below the cloud. Thus, detection and compensation for possible cloud contamination within a FOV is critical. There are several ways to deal with the cloud effects, including carefully detecting and rejecting FOVs from cloud-contaminated footprints or eliminating the effects of clouds from the data.

Figure 2-3 shows typical AIRS brightness temperature spectra of adjacent 3-by-3 FOVs (called a *golfball*), overlaid on top of each other. It is seen that some FOVs are warmer than others because of the true temperature variation, or, more probably because each FOV has different cloud properties such as cloud fraction or cloud

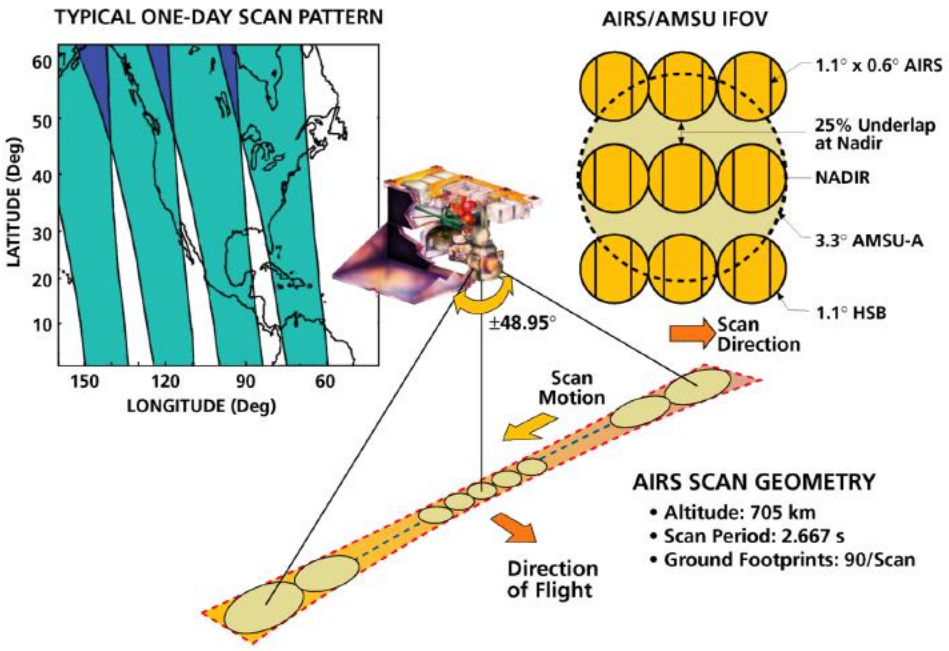


Figure 2-2: AIRS/AMSU/HSB scan geometry and pattern

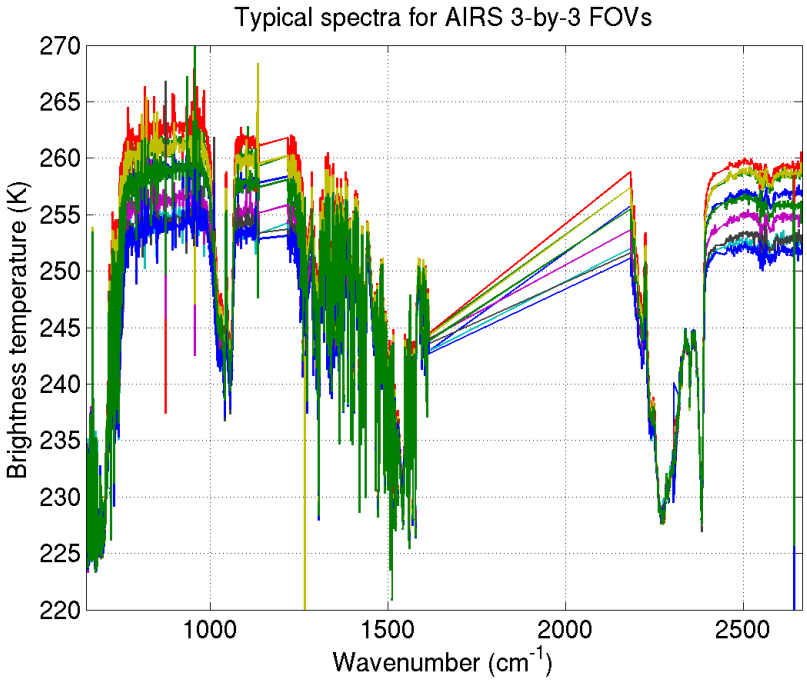


Figure 2-3: AIRS spectra for adjacent 3-by-3 FOVs

altitude.

2.2 Physics of radiative transfer

A sensor high above the Earth's surface receives emission from the Earth and its atmosphere, along with any reflected components of solar and cosmic background radiation. Measurements of this emission allow the retrieval of many atmospheric parameters, including the temperature and water vapor profiles, the amount of cloud liquid water, rain rates, and sea surface temperatures. In this section, the radiative transfer equation that links environmental parameters to the hyperspectral data, will be briefly explained.

2.2.1 Blackbody radiation

The physics of atmospheric temperature sounding begins with the Planck radiation law which relates the intensity of the radiation emitted by a blackbody to the physical temperature of the blackbody. A blackbody is defined to be a perfectly absorbing body that emits radiation with an intensity that is solely dependent on the temperature of the body and not on any other physical properties. Spectral brightness is a measure of how much energy a body radiates at a specified frequency per unit receiving area, per transmitting solid angle, per unit frequency. The spectral brightness of a blackbody is a function of its physical temperature T (K) and frequency f (Hz) and is given as:

$$I(f,T) = \frac{2hf^3}{c^2(e^{hf/kT}-1)} \quad \mathrm{Wm^{-2}ster^{-1}Hz^{-1}} \tag{2.1}$$

where h is Planck's constant, k is Boltzmann's constant, and c is the speed of light. The Planck equation exhibits a nonlinear relationship between intensity and temperature. The degree of the nonlinearity is dependent on frequency, shown in Figure 2-4. The nonlinearity is most severe at the higher frequencies and almost nonexistent at the microwave frequencies.

The preceding discussion dealt with radiation from ideal blackbodies. However,

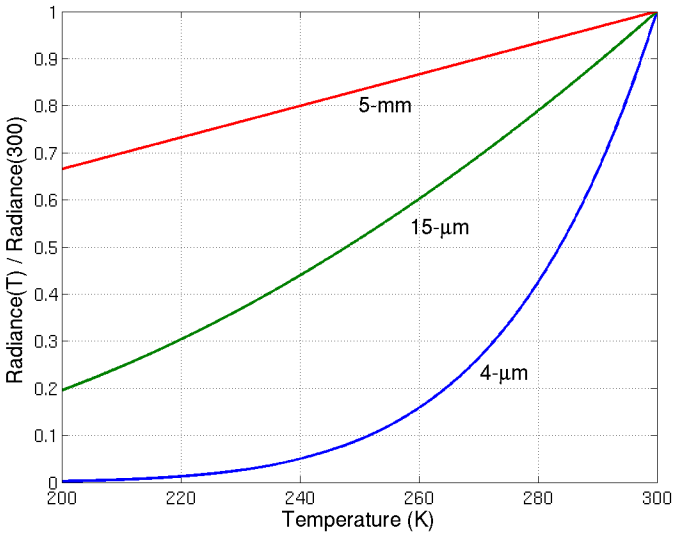


Figure 2-4: Nonlinearity of the Planck function as a function of wavelength.

a real object at temperature T can emit radiation that differs significantly from the corresponding blackbody at temperature T. Specifically, the non-ideal nature of the surface of the earth must be taken into account when performing radiative transfer calculations. To correct for the non-ideal nature of the earth's surface, a scaling factor is introduced. The reflectivity, ρ, of an object is defined to be the fraction of incident power reflected from the object. By using conservation of energy principles, a dual quantity, named the emissivity, ϵ, can also be defined such that

$$\rho + \epsilon = 1. \tag{2.2}$$

For a blackbody, $\rho = 0$ and $\epsilon = 1$ by definition.

2.2.2 Equation of radiative transfer

The electromagnetic radiation that arrives at a sensor at the top of the atmosphere will have been emitted by and transmitted through the atmosphere. As the radiation travels through the atmosphere, it gets absorbed by interactions with the matter. Emission from the matter, assumed in local thermodynamic equilibrium at temper-

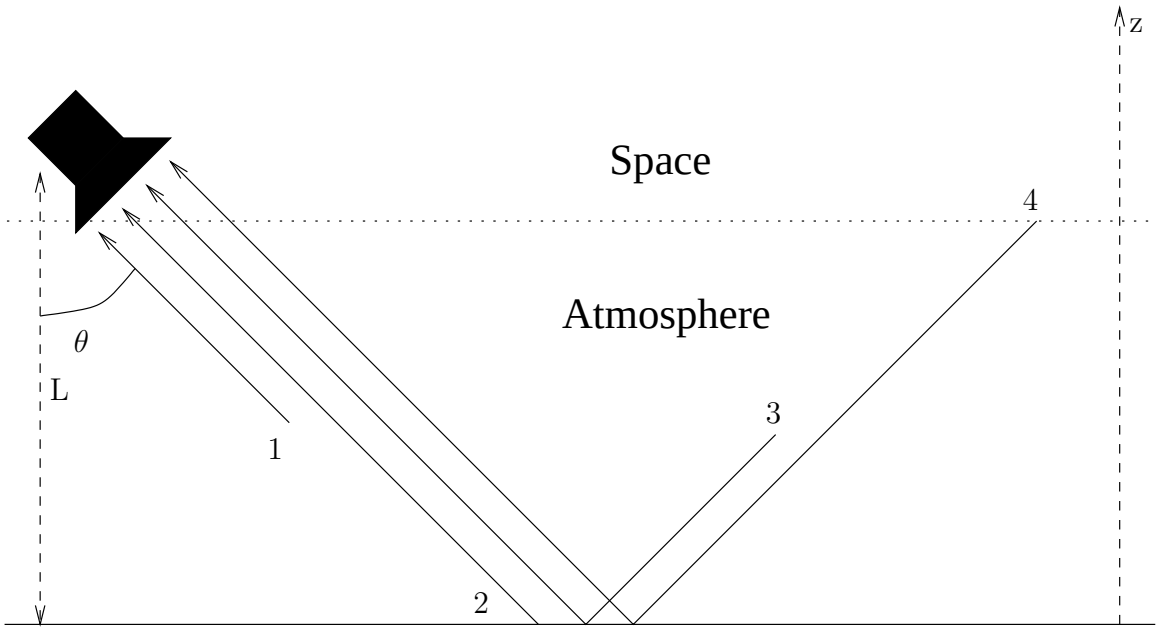


Figure 2-5: Four components of the radiative transfer equation

ature T, also takes place. The emission and absorption of electromagnetic radiation in the atmosphere is described by the equation of radiative transfer. The upwelling electromagnetic radiation at the top of the atmosphere is the sum of the four contributions shown in Figure 2-5. The four terms correspond to:

1. The upwelling atmospheric radiation.

2. The upwelling radiation emitted by the surface.

3. The downwelling atmospheric radiation which is reflected upward by the surface.

4. The downwelling cosmic background radiation which is reflected upward by the surface.

The equation of radiative transfer in a non-scattering, plane parallel, vertically stratified atmosphere is given by [12]. A sensor at height L above the earth's surface

looking down at the atmosphere will observe the following radiation intensity:

$$\begin{aligned} I(f,L) = \sec(\theta) \int_0^L I(f,z)\alpha(f,z)e^{-\int_z^L \alpha(f,z)\sec(\theta)dz}dz \\ + (1-\rho(f))I_{\text{surface}}e^{-\tau_0 \sec(\theta)} \\ + \rho(f)\sec(\theta)e^{-\tau_0 \sec(\theta)} \int_0^L I(f,z)\alpha(f,z)e^{-\int_0^z \alpha(f,z)\sec(\theta)dz}dz \\ + \rho(f)I_{\text{cosmic}}e^{-2\tau_0 \sec(\theta)} \end{aligned} \tag{2.3}$$

where θ is a scan angle, $I(f,z)$ is the intensity of atmospheric radiation at frequency f and height z, $\alpha(f,z)$ is the energy absorption coefficient at frequency f and height z, $\rho(f)$ is the reflectivity of the surface at frequency f, I_{surface} is the radiation emitted by the surface at frequency f, τ_0 is the opacity through the entire atmosphere, i.e. $\tau_0 = \int_0^L \alpha(f,z)dz$, and I_{cosmic} is the cosmic background radiation. The four terms in Equation 2.3 correspond to the radiation paths shown in Figure 2-5 in the absence of any scattering within the atmosphere by clouds, etc.

Note that the relationship between the intensity of radiation and blackbody temperature in microwave regime is almost perfectly linear. This can be verified analytically by approximating Equation 2.1 with its first order Taylor expansion, yielding:

$$I(f,T) \cong \frac{2kf^2}{c^2}T. \tag{2.4}$$

This approximation is valid when $hf \ll kT$ which is true for microwave frequencies and typical atmospheric or terrestrial temperatures. This linear approximation is not applicable with infrared frequencies where $hf > kT$.

With this simplification, the intensity of the radiation emitted by a blackbody is proportional to its physical temperature. For this reason, it is common to characterize the microwave radiation intensity of real objects (i.e. non-blackbodies) by the temperature of an equivalent blackbody. This Raleigh-Jeans approximation can be used to write Equation 2.3 in terms of physical temperatures and brightness temperatures

instead of radiation intensities:

$$\begin{aligned} T(f,L) = \sec(\theta) \int_0^L T(f,z)\alpha(f,z)e^{-\int_z^L \alpha(f,z)\sec(\theta)dz}dz \\ + (1-\rho(f))T_{\text{surface}}e^{-\tau_0 \sec(\theta)} \\ + \rho(f)\sec(\theta)e^{-\tau_0 \sec(\theta)} \int_0^L T(f,z)\alpha(f,z)e^{-\int_0^z \alpha(f,z)\sec(\theta)dz}dz \\ + \rho(f)T_{\text{cosmic}}e^{-2\tau_0 \sec(\theta)} \end{aligned} \tag{2.5}$$

Thus, in the microwave case, the temperature profile of the atmosphere, $T(f,z)$, appears explicitly in the radiative transfer equation.

2.2.3 Weighting function

The first term in Equation 2.3 can be recast in terms of the transmittance function $\mathcal{T}(f,z)$ [5]:

$$\begin{aligned} I(f,L) &= \int_0^L I(f,z)\left(\alpha(f,z)\sec(\theta)e^{-\int_z^L \alpha(f,z)\sec(\theta)dz}\right)dz \\ &= \int_0^L I(f,z)\left(\frac{d\mathcal{T}(f,z)}{dz}\right)dz. \end{aligned} \tag{2.6}$$

The derivative of the transmittance function with respect to altitude is called the *Planck* weighting function:

$$W(f,z) = \frac{d\mathcal{T}(f,z)}{dz}. \tag{2.7}$$

and gives the relative contribution of the radiance emanating from each altitude. It is useful to define a *temperature* weighting function ($W_T(f,z)$), where the temperature profile is weighted directly. One approach is to express the radiance intensity in terms of a blackbody-equivalent brightness temperature and linearize about a nominal temperature profile ($T_0(z)$) and corresponding radiance:

$$W_T(f,z) = \frac{dI(f,z)^{-1}}{dW(f,z)}\frac{dW(f,z)}{dI(f,z)}\frac{dI(f,z)}{dT(z)}\bigg|_{T_0(z)}. \tag{2.8}$$

The difference between the Planck weighting function and the temperature weight-

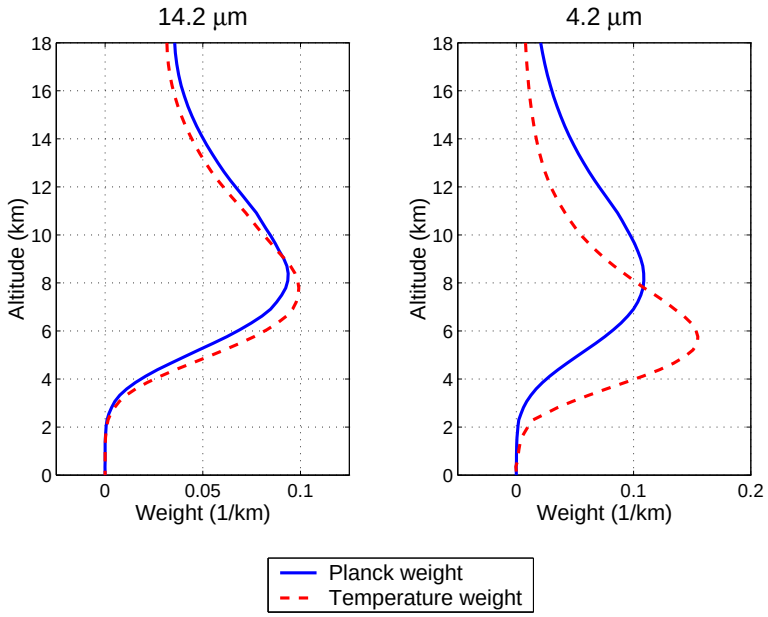


Figure 2-6: The Planck radiance weighting function and the temperature weighting function for two infrared channels [5].

ing function can be significant for short-wavelength channels, as shown in Figure 2-6 [5]. The temperature weighting functions are sharper and peak lower in the atmosphere. Hereafter a weighting function (or weighing function peak) will refer to a temperature weighing function (or temperature weighting function peak).

2.2.4 The atmospheric absorption spectra

Microwave frequency

Figure 2-7 shows the atmospheric absorption spectrum for microwave frequencies [5]. Notable features include the water vapor absorption lines centered at 22.235, 183.31, and 325.15 GHz (lines at 380.20 and 448.00 GHz are difficult to identify on the plot) and oxygen absorption lines near 60, 118.75, 368.50, 424.76, and 487.25 GHz.

Infrared frequency

The atmospheric absorption spectrum for infrared wavelengths between 3.6 and 16.1 μm is shown in Figure 2-8 [5]. It is noted that water vapor affects most of the infrared

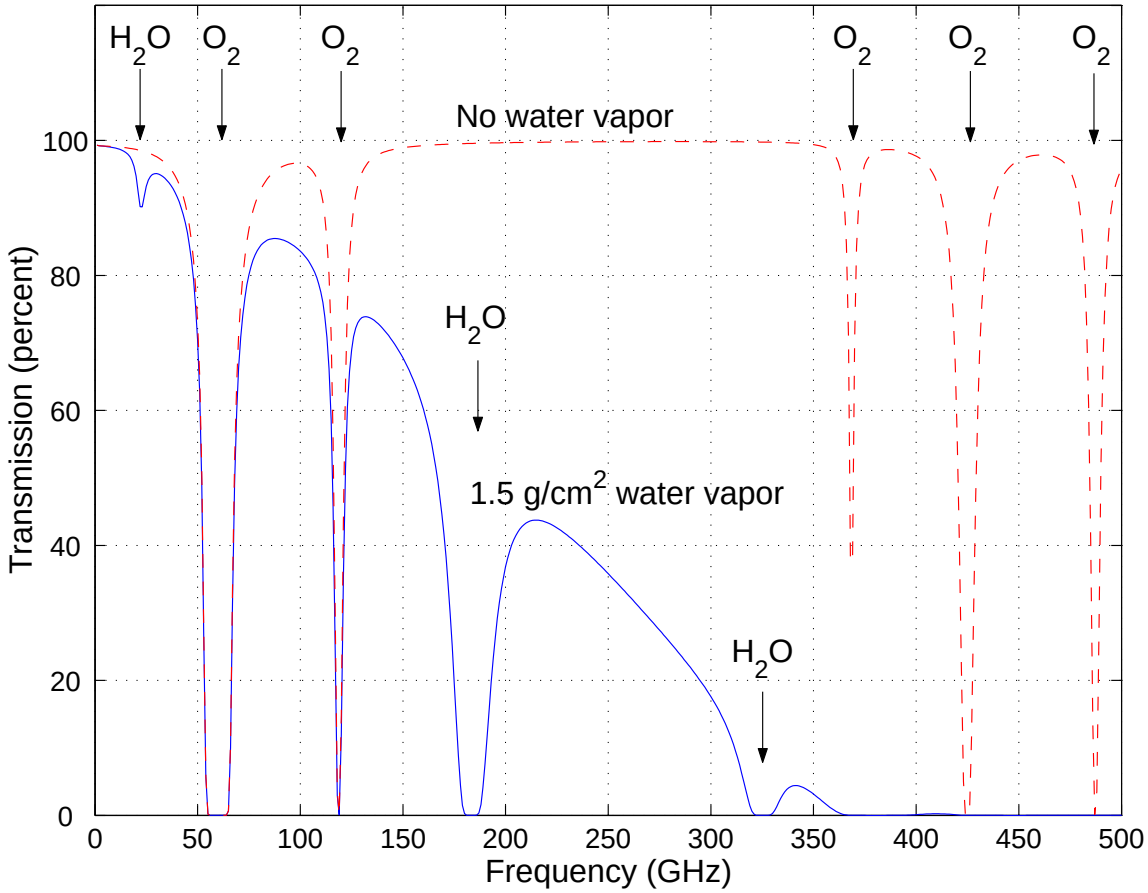


Figure 2-7: The microwave absorption spectrum. Two calculations for the percent transmission (nadir view) using the 1976 Standard Atmosphere are shown, one assuming no water vapor and one assuming 1.5 g/cm^2 [5].

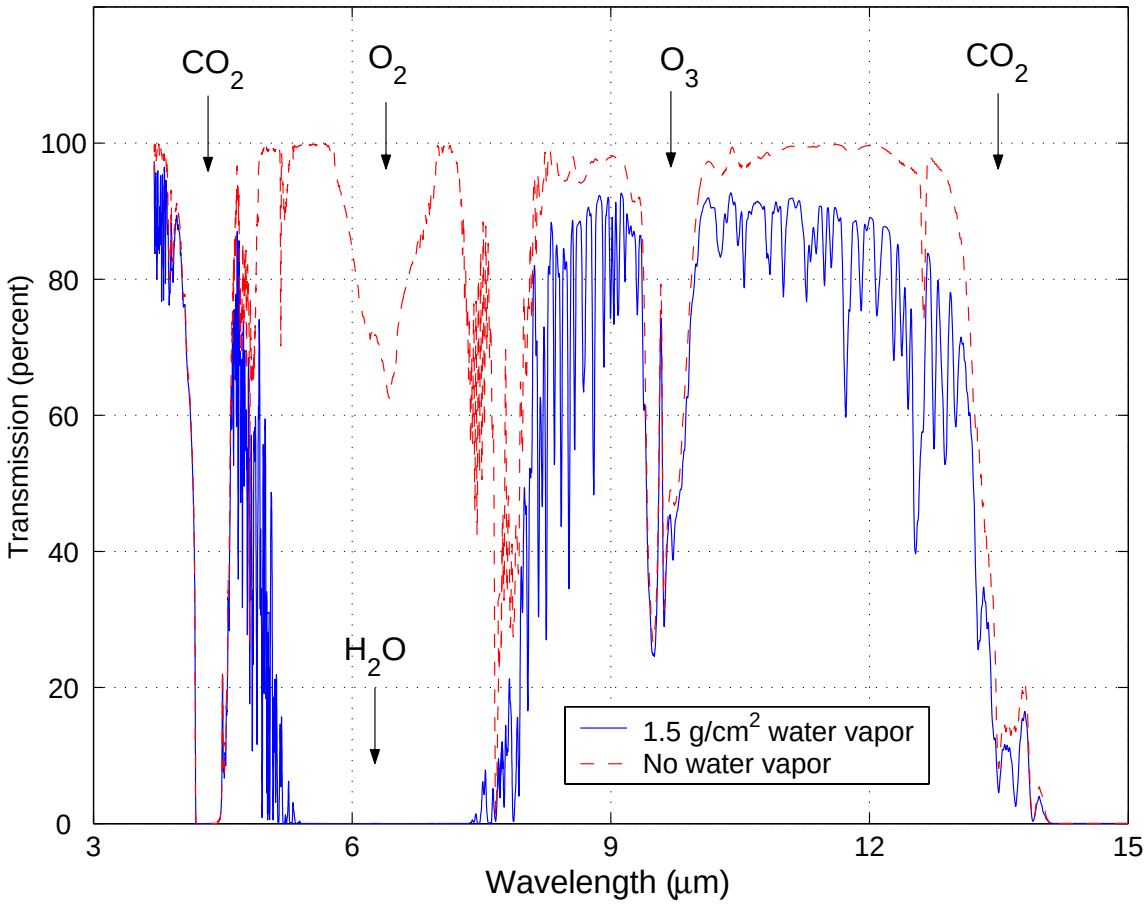


Figure 2-8: The thermal infrared absorption spectrum. Two calculations for the percent transmission (nadir view) using the 1976 Standard Atmosphere are shown, one assuming no water vapor and one assuming 1.5 g/cm^2 [5].

spectrum, especially 6–7 μm.

There are four regions over which water vapor, ozone, and carbon dioxide exhibit a significant absorption spectrum. CO_2 absorbs IR radiation in the 15 μm band from about 12.5 μm to 16.7 μm (600 to 800 cm^{-1}). In addition, CO_2 also absorbs radiation in the 4.3 μm region that overlaps with solar radiation. Absorption due to ozone is primarily confined to the 9.6 μm band. Water vapor exhibits absorption lines over the entire infrared spectrum. The most pronounced absorption occurs in the 6.3 μm vibrational-rotational band and in the pure rotational band with wavenumbers less than about 20 μm (500 cm^{-1}). From about 8.3 μm to 12.5 μm (800 to 1200 cm^{-1}), referred to as the atmospheric window, absorption due to atmospheric gases shows

a minimum, except in the 9.6 μm ozone band. There are also absorption bands for various greenhouse gases that can be used for their determination by remote sensing: the CH_4 7.6 μm band, the N_2O 7.9 μm band, and some CFC lines in the window.

Chapter 3

Signal Characterization and Reduction of Noise and Artifacts

3.1 Signal characterization

Many multivariate signals in the real world have correlated components. A hyperspectral image usually has highly correlated channels such that only a small fraction of components can explain the most of the variability. The principal component analysis (PCA) or Karhunen-Loève expansion has been widely used in the enhancement and characterization of multispectral remote-sensing data [29]. It has the significant property that it concentrates the data variability to the maximum extent possible with any number of components. Several variants of PCA will also be discussed in the following subsections.

3.1.1 Principal component analysis

Let us assume that a multivariate signal of interest, $\mathbf{x}$, is a vector of dimension m. The basic idea of PCA is to find the components, $y_1, y_2, ..., y_n$, so that they explain the maximum amount of variance of $\mathbf{x}$ possible by n linearly transformed components. Since PCA is often used to reduce the dimension of the data, one usually chooses $n \ll m$. PCA can be defined in an intuitive way using a recursive formulation.

Define the direction of the first principal component, say $\mathbf{w}_1$, by

$$\mathbf{w}_1 = \arg \max_{||\mathbf{w}||=1} E(\mathbf{w}^T\mathbf{x})^2 \tag{3.1}$$

where $\mathbf{w}_1$ is of the same dimension m as the random data vector $\mathbf{x}$. Thus the first principal component is the projection on the direction in which the variance of the projection is maximized. Having determined the first $k-1$ principal components, the k-th principal component is determined as the principal component of the residual:

$$\mathbf{w}_k = \arg \max_{||\mathbf{w}||=1} E[\mathbf{w}^T(\mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w}_i\mathbf{w}_i^T\mathbf{x})]^2 \tag{3.2}$$

The principal components (or principal component scores) are then given by $y_i = \mathbf{w}_i^T\mathbf{x}$. In practice, computation of the $\mathbf{w}_i$ can be simply accomplished using the covariance matrix $E[\mathbf{x}\mathbf{x}^T] = \mathbf{C_{xx}}$. The $\mathbf{w}_i$ are the eigenvectors of $\mathbf{C_{xx}}$ that correspond to the n largest eigenvalues of $\mathbf{C_{xx}}$. For convenience we define a m-by-n matrix $\mathbf{W} = [\mathbf{w}_1|\mathbf{w}_2|\ldots|\mathbf{w}_n]$, thus yielding:

$$\mathbf{y} = \mathbf{W}^T\mathbf{x}. \tag{3.3}$$

The major application of PCA is the reduction of the dimensionality. Such a reduction in dimension has important benefits. First, the computational overhead of the subsequent processing stages is reduced. Second, noise may be reduced, as the data not contained in the first n components may be mostly due to noise. Third, a projection into a subspace of a very low dimension, for example, two, is useful for characterizing and visualizing the data.

It can be proven that PCA guarantees to minimize the expected value of the squared error of the original data vector with respect to its reconstruction, i.e.

$$C(\cdot) = E[(\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})], \tag{3.4}$$

where the reconstructed estimate of $\mathbf{x}$, $\hat{\mathbf{x}} = \mathbf{W}\mathbf{W}^T\mathbf{x}$ [29]. Note that if $n = m$, then

$\mathbf{W}\mathbf{W}^T = \mathbf{I}$ and $\hat{\mathbf{x}} = \mathbf{x}$.

3.1.2 Noise-adjusted principal components

Because the variability of the data is scale-dependent, PCA is sensitive to the scaling of the data to which it is applied. For example, if one of the multispectral bands is arbitrarily doubled in amplitude, its contribution to the variance of the data set will be increased fourfold, and it will therefore be found to contribute more to the earlier eigenvalues and eigenvectors. To some extent the scaling of the multispectral bands is arbitrary, or even the physical unit of each element can differ. The noise-adjusted principal component (NAPC) transform is equivalent to a transformation of the data to a coordinate system in which the noise covariance matrix is the identity matrix, followed by a principal components transformation. Thus, it provides a normalization where the noise is each band is equal in magnitude and uncorrelated with the noise in any other. The cost function which is minimized by employing NAPC can be proven to be the mean squared error of the original noiseless data vector, $\mathbf{z} = \mathbf{x} - \mathbf{n}$ with respect to its reconstruction [29]:

$$C(\cdot) = E[(\mathbf{z} - \hat{\mathbf{z}})^T(\mathbf{z} - \hat{\mathbf{z}})]. \tag{3.5}$$

Figure 3-1 plots variance of each PC and NAPC in a decreasing order with respect to eigenvalue index of a typical AIRS image[1]. Such a graph is called a scree plot. Note that only first 400 (out of 2378) eigenvalues are shown to highlight the difference of scree plots for PC and NAPC in the knee region. NAPC shows a sharper knee and flatter noise plateau than PC, which demonstrates its better performance to separate the signal component and the noise component. Figure 3-2 shows the explained variance of the data with respect to the number of eigenvectors, for the same data set. The explained variance is simply the accumulation of the eigenvalues normalized by the sum of all eigenvalues. Figure 3-2 shows that the first 6 principal components

[1]The data set used for the analysis is 12,150 AIRS observed brightness temperature spectra, observed on August 21, 2003.

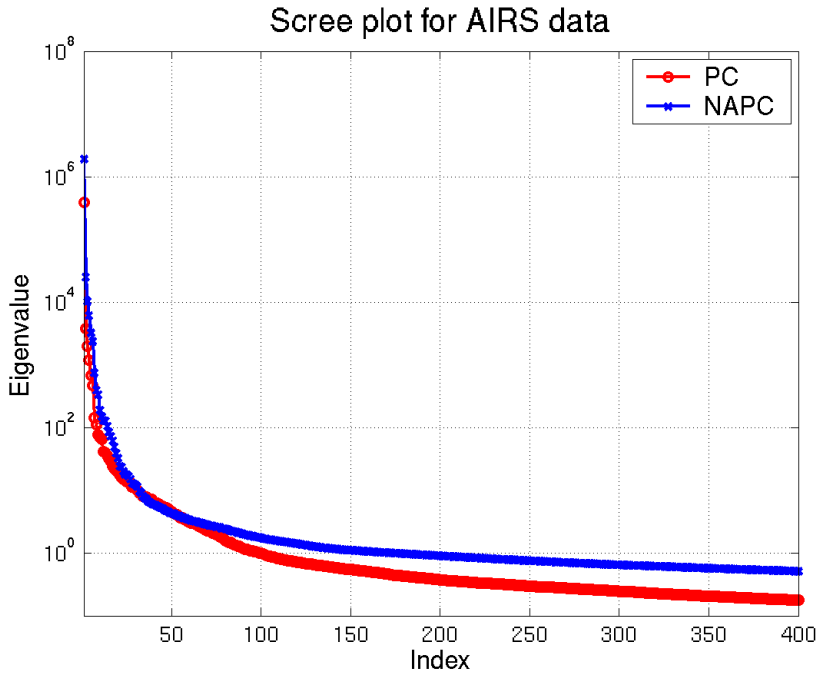


Figure 3-1: Typical scree plot for AIRS data

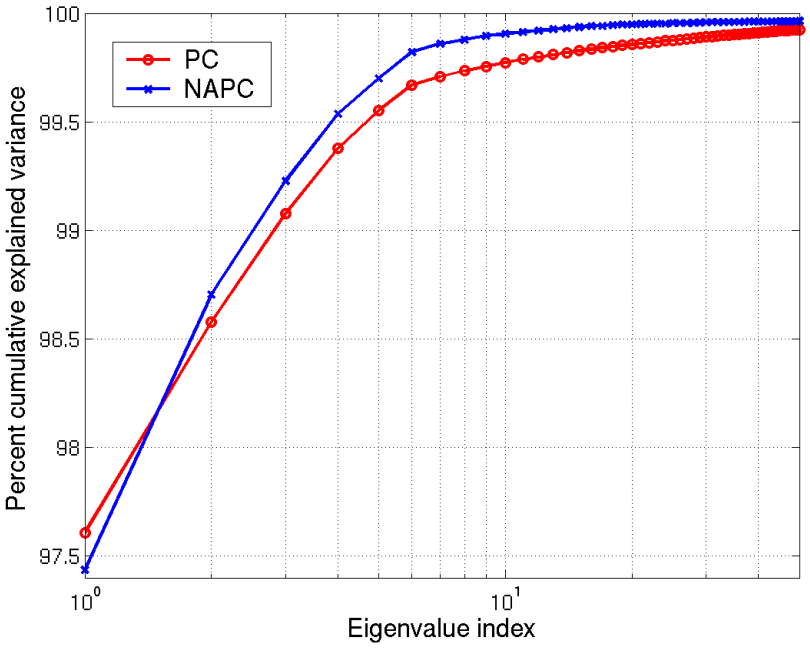


Figure 3-2: Percent cumulative explained variance

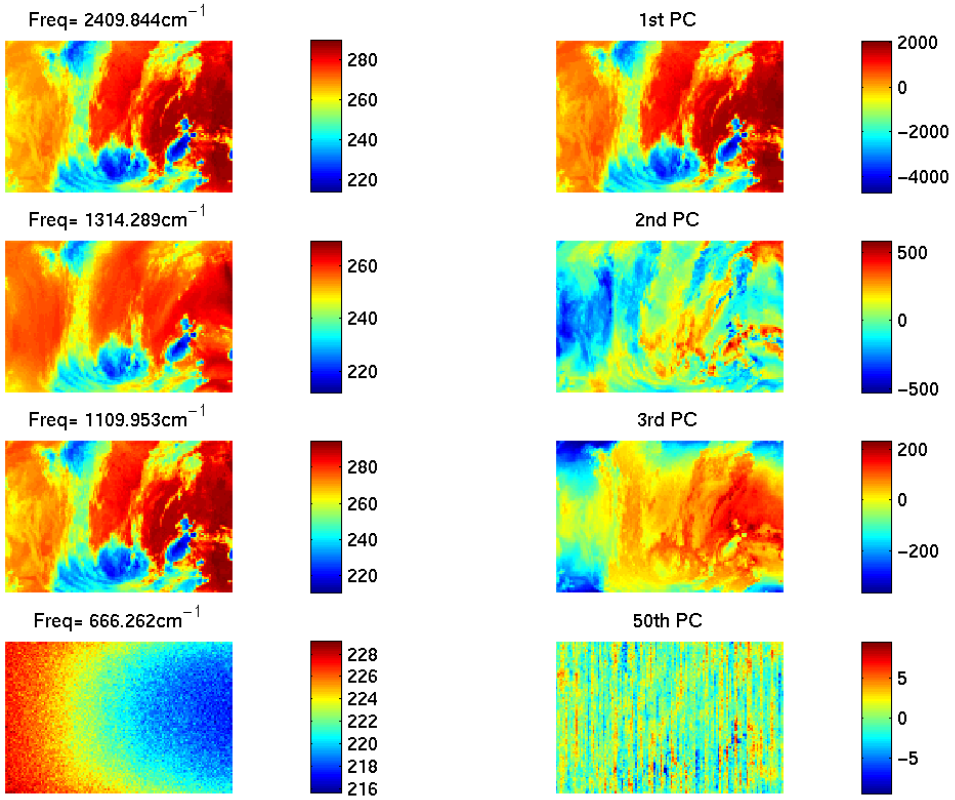


Figure 3-3: Sample AIRS observation images and NAPC images

account for 99.82% and 99.67% of the total variance for NAPC and PC, respectively. Thus, NAPC is expected to capture more energy in the signal than PC, with the same degrees of freedom. Figure 3-3 shows sample AIRS brightness temperature images and noise-adjusted principal component (NAPC) images. Images on the left show AIRS observed images for four different channels. Right images correspond to the first to third, and 50th NAPC images. Note that the first NAPC image has much higher dynamic range than the rest.

3.1.3 Projected principal component analysis

The cost function to be minimized for PCA is the mean-squared error of the original vector and its reconstruction. Cost functions other than Equation 3.4 are sometimes

used in practice. Suppose we want to find the n-by-m transform matrix $\mathbf{W}$ which gives components that, when regressed against a geophysical state vector $\mathbf{s}$, minimize the resulting sum-squared error for any ordered subset of components $\mathbf{y}$ given by

$$\mathbf{y} = \mathbf{W}^T\mathbf{x}. \tag{3.6}$$

The estimate of $\mathbf{s}$ in terms of $\mathbf{y}$ is

$$\begin{aligned} \hat{\mathbf{s}} &= \mathbf{C_{sy}}\mathbf{C_{yy}^{-1}}\mathbf{y} \\ &= \mathbf{C_{sx}}\mathbf{W}[\mathbf{W}^T\mathbf{C_{xx}}\mathbf{W}]^{-1}\mathbf{W}^T\mathbf{x} \end{aligned} \tag{3.7}$$

and the cost function to be minimized is given by

$$C(\cdot) = E[(\mathbf{s} - \hat{\mathbf{s}})^T(\mathbf{s} - \hat{\mathbf{s}})], \tag{3.8}$$

where $\mathbf{C_{sx}}$ is the cross-covariance of $\mathbf{s}$ and $\mathbf{x}$. It can be shown [5] that the $\mathbf{W}$'s that minimize Equation 3.8 are the n right eigenvectors with highest singular values of the reduced-rank regression matrix $\mathbf{M}$:

$$\mathbf{M} = \mathbf{K}\mathbf{K}^T\mathbf{C_{sx}}\mathbf{C_{xx}^{-1}}, \tag{3.9}$$

where $\mathbf{K} = [k_1|k_2|\ldots|k_n]$ are the n most significant eigenvectors of $\mathbf{C_{sy}}\mathbf{C_{yy}^{-1}}\mathbf{C_{sy}^{T}}$. This transform is called the Projected Principal Components (PPC) transform because the measurements $\mathbf{x}$ are projected into an n-dimensional subspace of $\mathbf{s}$ spanned by $\mathbf{K}$.

3.2 Blind signal separation

NAPC analysis requires that the noise variances are known. If noise variances are unknown, which is often the case, they need to be estimated to apply NAPC analysis. Iterative Order and Noise (ION) estimation algorithm [27] is an efficient technique for blind signal separation and for the estimation of noise variances when the signal order is also unknown. ION blindly separates signal and noise components of noisy

jointly Gaussian multivariate signal $\mathbf{x} = \mathbf{Ap} + \mathbf{Gn}$ where $\mathbf{A}$ is the mixing matrix, $\mathbf{p}$ is the signal of unknown dimension k, $\mathbf{G}$ is the diagonal noise covariance matrix, and $\mathbf{n}$ is the white Gaussian noise with unit variance. It is difficult to accurately estimate the signal order, k, without prior knowledge of the noise variances, $trace(\mathbf{G})$, or to estimate noise variances without knowledge of the signal order. Thus, ION is an iterative algorithm which takes advantage of progressively improved estimates of both signal order and noise variances to further improve the estimates of both parameters. The plot of eigenvalues with respect to eigenvector index, or scree plot, reveals the noise plateau in most practical cases, thus allowing estimation of the signal order. In the previous scree plot (Figure 3-1), for example, noise plateau is seen to be beyond $\sim$ 30-th eigenvalues for NAPC, and $\sim$ 80-th eigenvalues for PC.

The expectation-maximization (EM) algorithm is the most successful known separation technique for Gaussian signals, so it is used to estimate the noise variances based on the previous estimate of the signal order.

3.3 Artifacts in AIRS data

Real-world data often have several different sources of artifacts, either expectedly or unexpectedly. Observations from AIRS sensor have unwanted artifacts of several different natures. Among many possible sources of artifacts, three different types will be discussed in this chapter:

1. Instrumental white noise.

2. Consistently noisy channels.

3. Scan-line miscalibration, resulting in striping patterns along scan-lines.

3.3.1 Instrument white noise

Instrument white noise is unavoidable due to the combination of the thermal noise and shot noise in a spectrometer. In remotely sensed imagery, the most important

information is often concentrated in the spatially low-frequency region. In this case, a useful approach to suppress white noise is to use some type of low-pass filter. PC or NAPC filtering is an effective and intuitive method to implement a low-pass filter.

Principal component analysis is analogous to Fourier transform in that the data is described in terms of the coefficients of a predetermined orthogonal set. Rather than using complex exponentials, the orthogonal set in PCA is determined *adaptively* based on the analyzed data set. In particular, PCA derives the directions of a set of orthogonal vectors that point into the direction of the highest variance of the data, which often corresponds to the direction of spatially slowly varying components in remote sensing applications. The corollary is that a first few PC's (or NAPC's) capture most of the variance of the data, and the last few PC's (or NAPC's) generally contain the noisiest component of the data. It is logical to filter out the PC's associated with the least eigenvalues and subsequently transform back to the original coordinate system. The resulting PC filtered version of a given multivariate signal ($\mathbf{x}$) is:

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{W}^T\mathbf{x} \tag{3.10}$$

where $\mathbf{W}$ is an m-by-n matrix discussed in Section 3.1.1. This is called PC filtering as seen in the previous section, or NAPC filtering when proper normalization is done before PC filtering. The threshold for how many PC's to retain should be judicially chosen. One effective method is to scrutinize a scree plot to locate the noise plateau as shown in Section 3.2.

NAPC filtering is performed in many different stages in the proposed cloud-clearing algorithm as will be discussed later in the context of more explicit applications.

Figure 3-4 shows a typical noise pattern in AIRS data. Each black dot represents more than 0.2 K absolute error using NAPC filtering with PC cutoff threshold of $n = 50$, i.e. $|x_{i,j} - \hat{x}_{i,j}| > 0.2$ where $x_{i,j}$ is AIRS observation at channel i for j-th pixel, and $\hat{x}_{i,j}$ is the corresponding NAPC-filtered AIRS observation using $n = 50$. (Note that m, the number of channels is 2378.) In this plot, y-axis is channel number

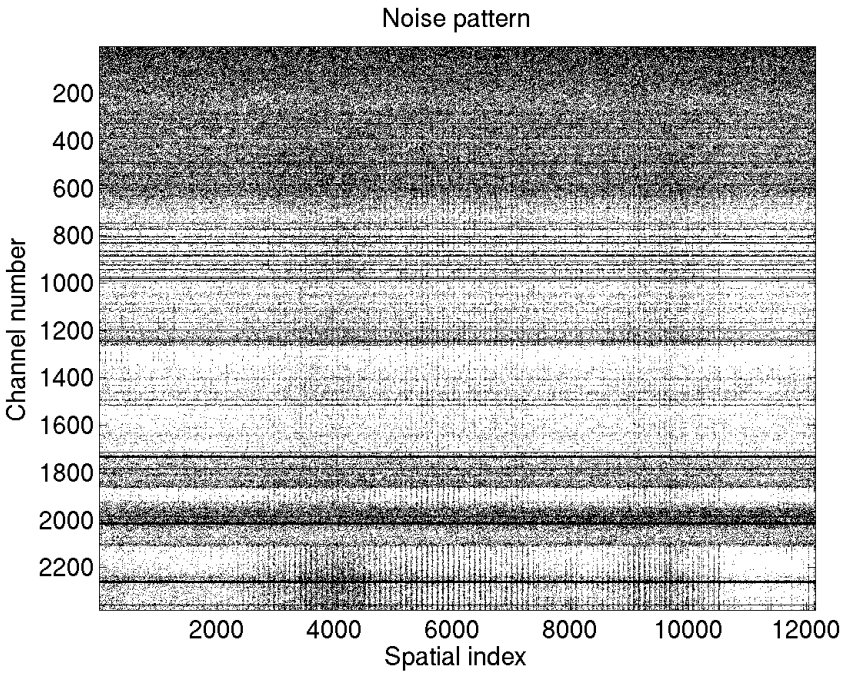


Figure 3-4: Typical noise pattern for AIRS brightness temperature data

(1–2378), and x-axis is spatial index (1–12,150). The RMS error can be expressed as:

$$\varepsilon_{\mathrm{RMS}} = \sqrt{\frac{\sum_{i,j}(x_{i,j} - \hat{x}_{i,j})^2}{IJ}}, \tag{3.11}$$

where $I = 2378$ (number of channels), and $J = 12,150$ (number of pixels). The RMS (or the standard deviation) error for this AIRS data is 0.4302 K. In this figure, the channels that are consistently noisy over most of pixels are evident. Also, the periodic patterns in spatial dimension suggest there may exist scan-line artifacts.

3.3.2 Noisy channels

In Figure 3-4, we have seen that some channels are consistently noisy. Figure 3-5 shows the counts of pixels having more than 2.151 K error (5 times RMS error), for all AIRS channels. The y-axis (noisy pixel counts) is truncated at 100 in this plot because most of the channels having more than 100 counts are noisy for all 12,150 pixels. In this analysis, 545 channels (or 23% of all channels) have at least one

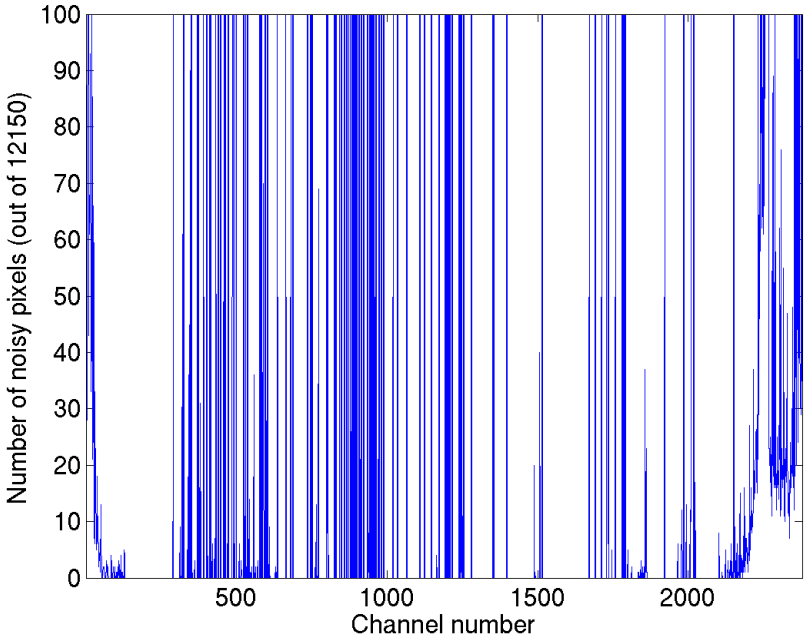


Figure 3-5: Count of pixels having more than 5 times RMS error, for all AIRS channels

noisy pixel having more than error of 2.151 K. Figure 3-6 shows the histogram of the NAPC filtering error (in K) over all pixels and all channels, in order to illustrate the consistency of the bad channels in a different way. This histogram is intended for the contrast with a Gaussian distribution. As many as $\sim 0.55\%$ of pixels fall outside of 5σ-tails whereas in a Gaussian world, only $5.73 \times 10^{-5}\%$ would have.

The consistently noisy channels will deteriorate any type of estimation of physical parameters. Thus, these channels need to be excluded before a retrieval stage to ensure the accurate estimation. The detection of noisy channels can be implemented in a number of ways, depending on the definition of a noisy pixel and the procedure to flag a noisy channel given a number of samples. Figure 3-7 illustrates the block diagram of a proposed noisy channel detector based on NAPC/ION filtering. NAPC/ION filtering provides a simple and computationally efficient method to detect noisy channels [7, 11]. Let us denote $\mathbf{X}$ an AIRS observation matrix where its element, $x_{i,j}$, is an AIRS observation at channel i for j-th pixel. The ION algorithm estimates the noise variances for all channels, or trace($\mathbf{G}$). Then, the noise-adjusted

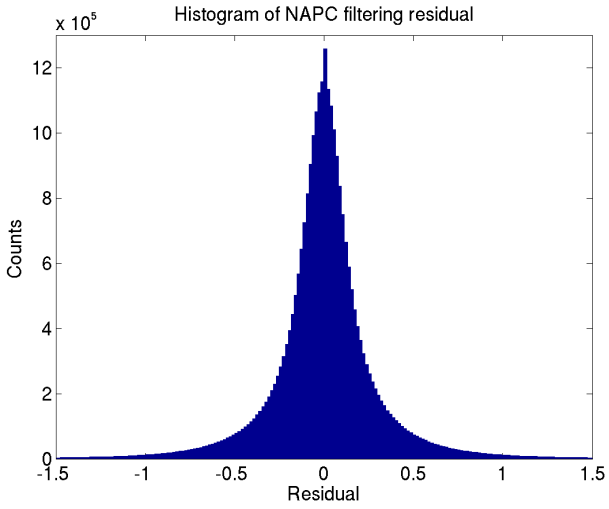


Figure 3-6: Histogram of NAPC filtering residual (original TB - NAPC filtered TB) for all 12,150 pixels and all 2378 channels, observed on August 21, 2003 (Granule # 76). The horizontal axis is the residual in degrees K, and the vertical axis is the counts.

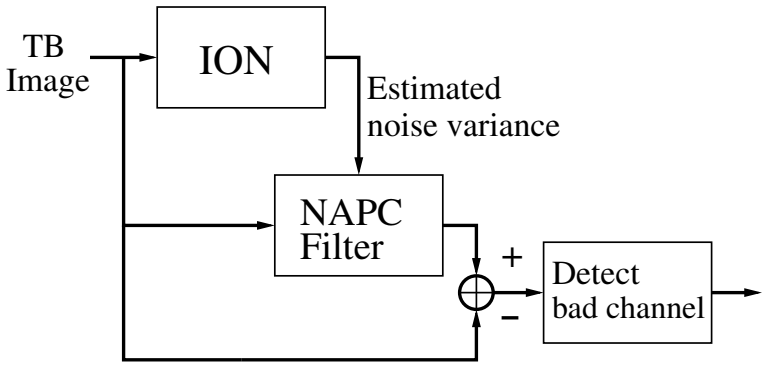


Figure 3-7: Block diagram of NAPC/ION filtering

data is found as:

$$\mathbf{X}_{\mathrm{NA}} = \mathbf{G}^{-1/2}\mathbf{X} \tag{3.12}$$

where the subscript "NA" represents a noise-adjusted version. The NAPC filtered version of $\mathbf{X}_{\mathrm{NA}}$ is:

$$\hat{\mathbf{X}}_{\mathrm{NA}} = \mathbf{W}\mathbf{W}^T\mathbf{X}_{\mathrm{NA}} \tag{3.13}$$

Thus, the error matrix is given by:

$$\begin{aligned} \mathbf{E} &= \mathbf{G}^{1/2}(\mathbf{X}_{\mathrm{NA}} - \hat{\mathbf{X}}_{\mathrm{NA}}) \\ &= \mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X} \\ &= (\mathbf{I} - \mathbf{W}\mathbf{W}^T)\mathbf{X}. \end{aligned} \tag{3.14}$$

Pixels for which original and NAPC filtered differ more than 2.151 K (5 times standard deviation of the error), are flagged bad, and any channels exceeding the this threshold even once were defined as bad. For the test and evaluation of the bad-channel detector, 320 channels in 4-μm and 15-μm spectrum, are arbitrarily selected. More (or all) channels can be used in this analysis; however, the selected subset of channels are sufficient for the sake of the validation purpose. The data set used for this analysis is 900 FOVs in Northern America observed on September 6, 2002 (AIRS L1B version 2.9.6). Within the test set of 320 frequencies, the NAPC/ION algorithm detected 80 noisy channels using 2 K threshold.

The AIRS science team has compiled their own list of channel properties based on the pre-launch and in-flight calibration of AIRS measurements, available for download at [47]. Out of 320 selected channels used in this analysis, the AIRS team flagged 21 channels as problematic for various physical reasons. Among these 21 channels, 15 channels are also detected in the proposed algorithm. Table 3.1 summarizes the detection rates for different types of bad channels. The second column shows the number of channels AIRS team declares bad, out of the 320 channels used in this analysis. The third column lists the number of channels the proposed NAPC/ION algorithm flags as bad, out of 320 channels. Note that the spectral response function

Type of bad channel	Number of NASA bad channels	Number detected by NAPC/ION algorithm	% detected	Description
Noise	5	5	100%	Detector has high noise
Bad SRF	1	0	0%	Detector spectral response functions (SRF) centroid does not match model
Poor SRF	13	8	61.5%	Detector SRF shape does not match model
Popping	2	2	100%	Detector response has shown unexpected steps
Total	21	15	71.4%	

Table 3.1: Detection rates for each type of noisy channel

(SRF) flaws are generally not a source of serious noise, unlike bad channels and popping channels, all of which were detected.

3.3.3 Striping patterns

AIRS/AMSU sensor suite operates in a scanning mode. Hence, miscalibration of a detector in one scan-line relative to another will cause striping artifacts in an observed image. Left three images in Figure 3-8 show observed AIRS brightness temperature (TB) images at 14.12 μm, 14.35 μm, and 14.78 μm, respectively. The stripy artifacts are more or less evident for all three channels. Figure 3-9 illustrates the block diagram of a stripe-removing algorithm. TB image cube is fed into a 1-dimensional high-pass spatial filter[2] with respect to the along-track dimension. The resulting high-pass filtered image cube is transformed to NAPC images in order to capture the striping features in a small number of high-order NAPC images. An experiment showed that only 20 significant NAPC's contain the striping artifacts. Lower-order NAPC images which may have some stripes are considered negligible also because their contributions to the variance in the original image domain is very small — for example, 0.2% in Figure 3-2. The 20 highest-order NAPC images are processed with the low-pass

[2]The high-pass filter is implemented with Parks-McClellan equiripple FIR filter design, with order 10 and cutoff frequency of 0.15 times the Nyquist frequency.

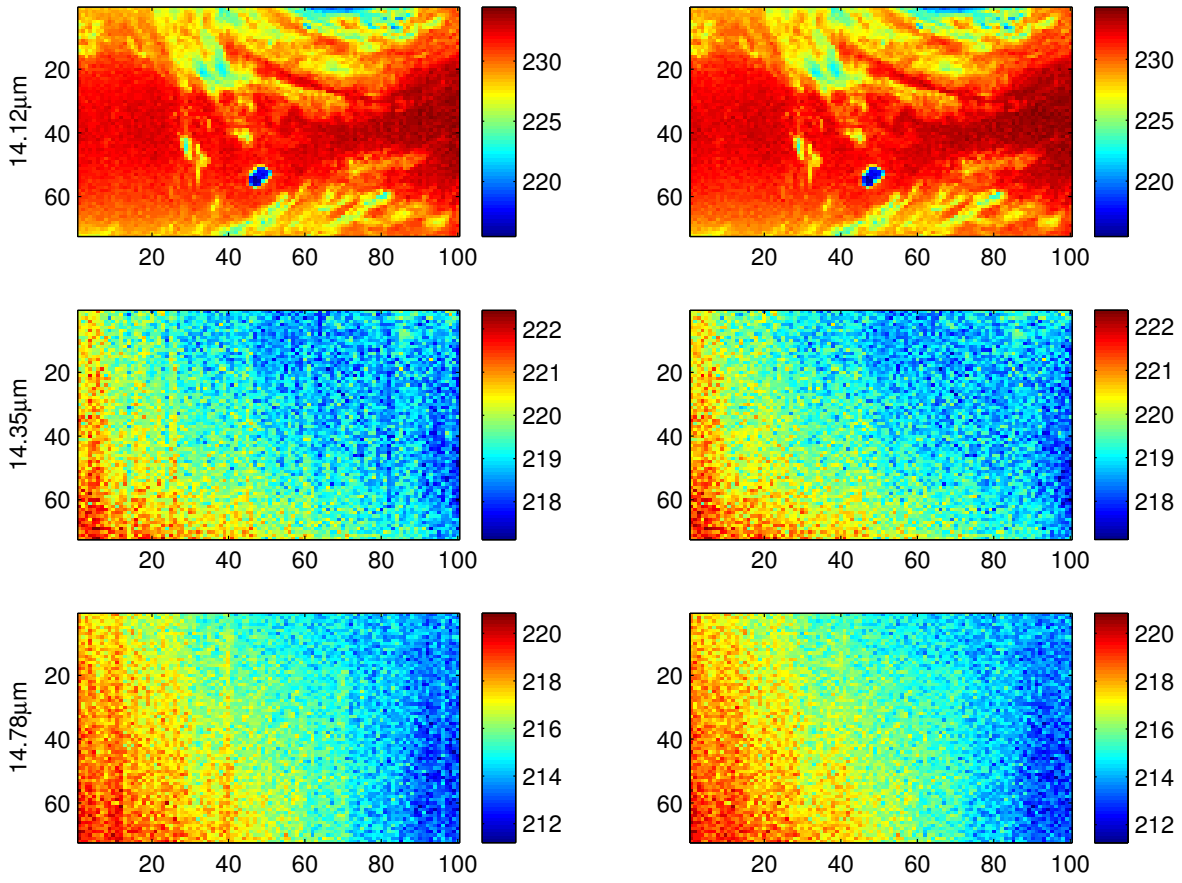


Figure 3-8: TB images before and after removing striping patterns for three different channels

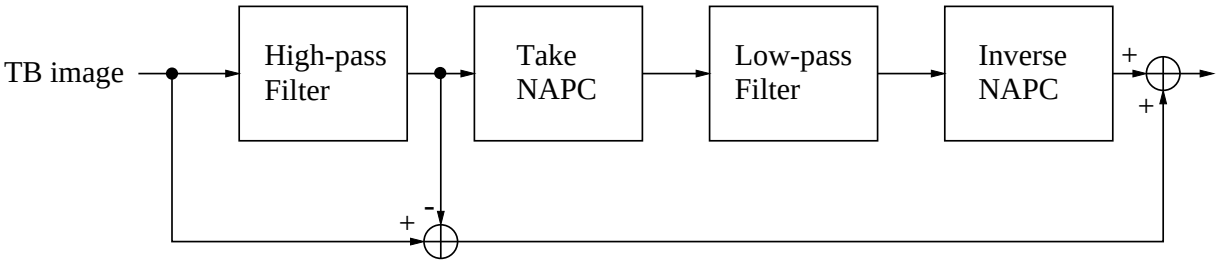


Figure 3-9: Block diagram of a stripe-removing algorithm

filter[3]. Then, the resulting images in NAPC domain are then transformed back into brightness temperature images in the original domain. The low-pass component of the original TB image will be added back. Right images in Figure 3-8 illustrate the post-processed AIRS TB images at 14.12 μm, 14.35 μm, and 14.78 μm, respectively, using the proposed stripe-removing algorithm. It is observed that most of the striping patterns are effectively removed by the algorithm. The RMS stripe error for each channel can be defined as:

$$\Delta_{\mathrm{RMS}}(i) = \sqrt{\frac{\sum_j (x_{i,j} - x_{i,j}^{\text{stripe removed}})^2}{J}}, \tag{3.15}$$

where J is a number of pixels. Figure 3-10 shows Δ_{RMS} as a blue curve and the noise-equivalent delta temperature (NeDT) at a scene temperature of 250 K as a red curve, with respect to the wavelength. NeDT on the measurements was independently obtained during the preflight radiometric assessment [34]. It is noted that the stripes are present for all 15 μm channels, and that Δ_{RMS} closely follows the shape of NeDT. Roughly half of NeDT can be compensated by using the algorithm.

[3]This low-pass filter is complementary to the high-pass filter[2], i.e. with order 10 and cutoff frequency of 0.15 times the Nyquist frequency.

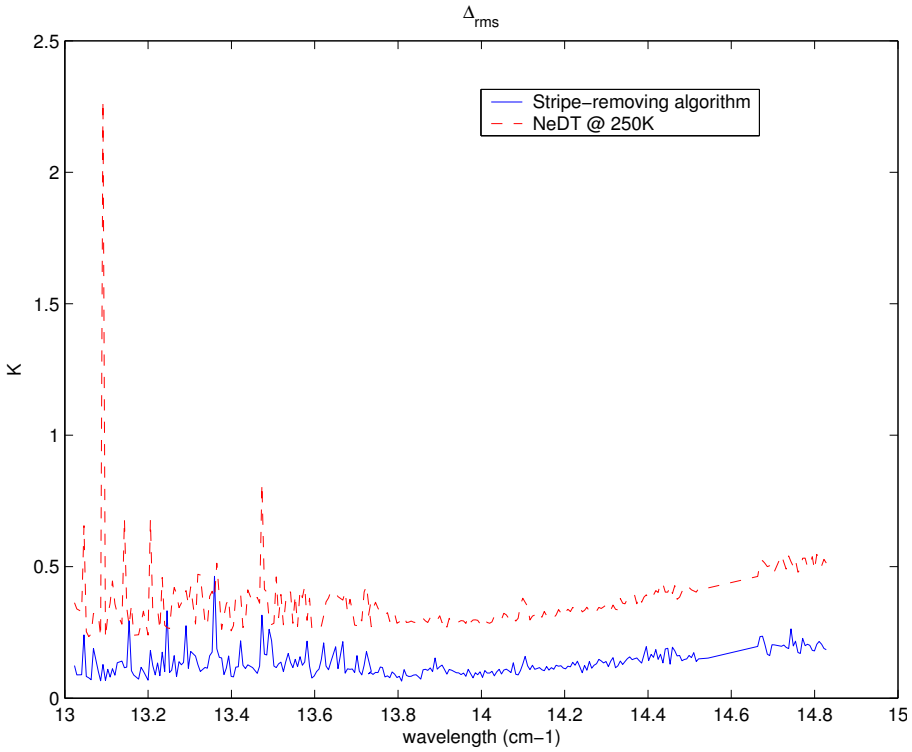


Figure 3-10: Δ_{RMS} for 15 μm channels

Chapter 4

Nonlinear Estimation

Retrieval of a geophysical parameter often involves nonlinear estimation when the underlying physics is nonlinear and the statistics are not jointly Gaussian. Underlying physical model for cloud contamination in the infrared radiance is, for example, complex and nonlinear. Scatter plots in Figure 4-1 illustrates typical relationships between various pairs of principal components for AIRS brightness temperature data[1]. The first PC and the second PC, for example, are by construction uncorrelated, but have a strong non-Gaussian relationship.

Nonlinear estimation techniques are generally superior to linear methods when the relationship between the observed and desired parameters is nonlinear, or when the statistics characterizing the problem are non-jointly-Gaussian. A simple illustration of the superiority of nonlinear estimator is provided in Figure 4-2, which characterizes the nonlinear physical relationship between the parameter to be estimated, x, and the observed data, y, in terms of a scatter plot representing the outcomes of multiple observations. However, nonlinear estimation is generally challenging because Bayesian least-square estimator requires a complete statistical representation of the relationship between the observed and desired parameters which is challenging except for the jointly-Gaussian problem, and also because the computation load is much higher than linear estimation method, for example, linear regression. In this chapter, various

[1]The data set used for the analysis is 12,150 AIRS observed brightness temperature spectra, observed on August 21, 2003.

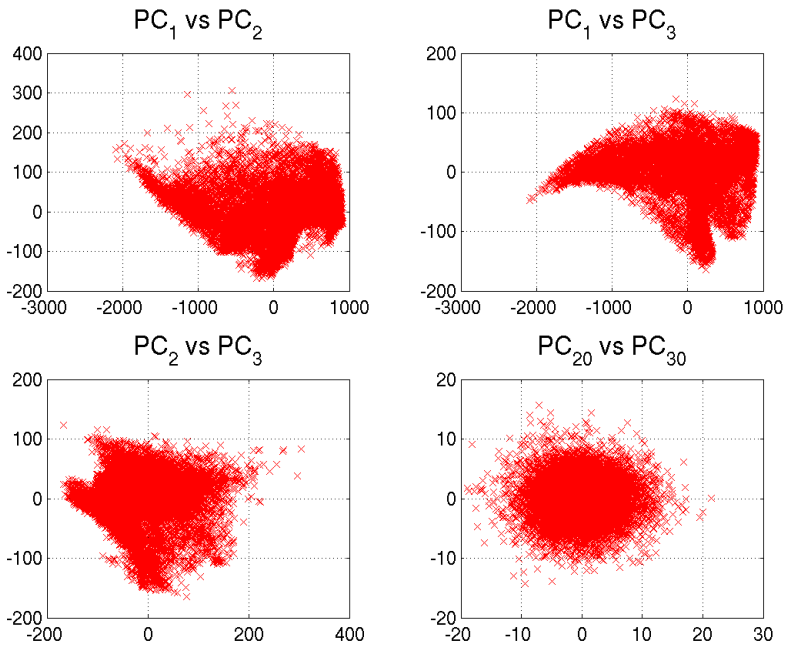


Figure 4-1: Typical scatter plots of different principal component coefficients

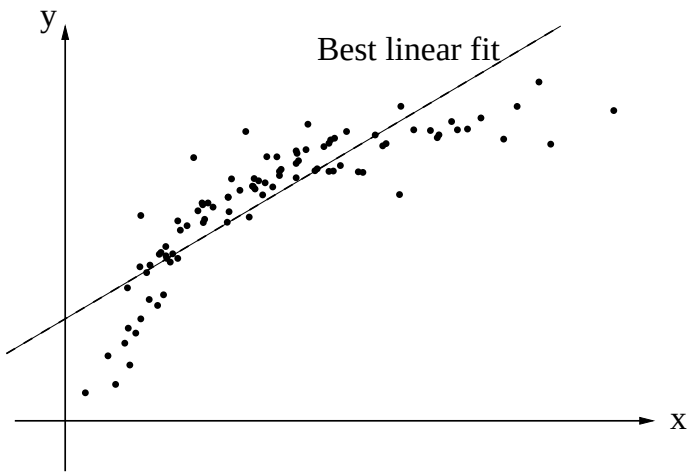


Figure 4-2: Best-fit linear regression line for a finite set of training data characterizing a nonlinear physical relationship between the parameter to be estimated, x and the observed data y

techniques of nonlinear estimation are explored. In the following section we shall review the linear least-square estimator, which can be a basis of nonlinear estimator.

4.1 Maximum a posteriori estimator

Estimation problems generally involve the formulation of an estimate of a random vector $\mathbf{x}$ given an observation of another random vector $\mathbf{y}$ [43]. In estimation theory, the estimator is chosen to minimize a suitable cost function. One well-known cost function is:

$$C(\mathbf{x}, \hat{\mathbf{x}}) = \begin{cases} 1 & \text{if } |\mathbf{x} - \hat{\mathbf{x}}(\mathbf{y})| > \epsilon \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

which uniformly penalizes all estimation errors with magnitude bigger than ϵ. The solution for this particular cost function for $\epsilon \to 0$ leads to a maximum a posteriori (MAP) estimator [43]:

$$\hat{\mathbf{x}}_{\text{MAP}}(\mathbf{y}) = \arg\max_{\mathbf{x}} p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) \tag{4.2}$$

Hence, the MAP estimate of $\mathbf{x}$ based on observation $\mathbf{y}$ is the mode of the posterior density $p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})$. This is closely related to maximum likelihood (ML) estimation. We can use the fact that log is strictly monotonic to obtain the following:

$$\begin{aligned} \hat{\mathbf{x}}_{\text{MAP}}(\mathbf{y}) &= \arg\max_{\mathbf{x}} \log p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) \\ &= \arg\max_{\mathbf{x}} \Big(\log p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) + \log p_{\mathbf{x}}(\mathbf{x}) - \log p_{\mathbf{y}}(\mathbf{y}) \Big) \\ &= \arg\max_{\mathbf{x}} \Big(\log p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) + \log p_{\mathbf{x}}(\mathbf{x}) \Big). \end{aligned} \tag{4.3}$$

In dropping the last term, we use the fact that it does not depend on $\mathbf{x}$. If the prior distribution of $\mathbf{x}$ is uniform, the MAP estimate reduces to the ML estimate.

4.2 Bayesian least-square estimator

In Bayesian estimation theory, another widely used cost function is least square error:

$$C(\mathbf{x}, \hat{\mathbf{x}}) = E[(\mathbf{x} - \hat{\mathbf{x}}(\mathbf{y}))^T(\mathbf{x} - \hat{\mathbf{x}}(\mathbf{y}))]. \tag{4.4}$$

The optimum Bayes' estimator for this cost criterion is:

$$\hat{\mathbf{x}}_{\text{BLS}}(\mathbf{y}) = E[\mathbf{x}|\mathbf{y}], \tag{4.5}$$

which is simply the mean of the posterior density $p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})$. Hence, it is trivial to check that a Bayesian least-square (BLS) estimator is the MAP estimator for a unimodal and symmetric probability density such as a Gaussian. The calculation of the Bayesian least-squares estimator requires knowledge of the complete statistical characterization of the relationship between $\mathbf{x}$ and $\mathbf{y}$. Often this characterization is not available, and suboptimal estimators must be used. One suboptimal estimator which requires only a partial characterization of the relationship between $\mathbf{x}$ and $\mathbf{y}$ can be obtained by constraining the estimator to be a linear function of the observed data, i.e.,

$$\hat{\mathbf{x}}_{\text{LLS}}(\mathbf{y}) = \mathbf{A}\mathbf{y} + \mathbf{b}. \tag{4.6}$$

This estimator is referred to as the linear least-squares (LLS) estimator. $\mathbf{A}$ and $\mathbf{b}$ can be determined by plugging Equation 4.6 into the cost function, Equation 4.4, and minimizing it. The LLS estimator is then given by:

$$\hat{\mathbf{x}}_{\text{LLS}}(\mathbf{y}) = \mathbf{C}_{\mathbf{xy}}\mathbf{C}_{\mathbf{xx}}^{-\mathbf{1}}(\mathbf{y} - E[\mathbf{y}]) + E[\mathbf{x}]. \tag{4.7}$$

From Equation 4.7, it can be seen that only first- and second-order statistics are needed to calculate the LLS estimate. Usually these statistics must be estimated from an ensemble of observations, i.e. training set of $\mathbf{x}$-$\mathbf{y}$ pairs; the mean vector and the covariance matrix need to be replaced by the sample mean and the sample covariance matrix. In this context, a linear least-squares estimator becomes a linear regression.

A special property of jointly Gaussian case [43] is:

$$\hat{\mathbf{x}}_{\mathrm{MAP}}(\mathbf{y}) = \hat{\mathbf{x}}_{\mathrm{BLS}}(\mathbf{y}) = \hat{\mathbf{x}}_{\mathrm{LLS}}(\mathbf{y}). \tag{4.8}$$

In other words, the linear least-squares estimate for a Gaussian case is as good as the Bayesian least-squares estimate.

4.3 Nonlinear estimation

The underlying physics for geophysical parameters is often nonlinear and the statistics are seldom jointly Gaussian. One simple way to observe and characterize nonlinearity is to inspect a scatter plot in an observation and parameter space. However, this method may not be suitable for high dimensional data such as hyperspectral data unless only a handful of representations such as PC scores can capture the most variability of the data. We shall discuss a number of general nonlinear estimation techniques.

Before we explore the various techniques to deal with a nonlinear relationship between the unknown parameter and the observed data points, it is important to note that there can be a case that a linear estimator yields a perfect answer even though the observation-parameter relationship is nonlinear. Consider the case where a single parameter x is to be estimated based on two observed pieces of data, y_1 and y_2, where

$$y_1 = a_0 + a_1 x + a_2 x^2 \tag{4.9}$$

$$y_2 = b_0 + b_1 x + b_2 x^2. \tag{4.10}$$

Assuming the data is noiseless, the relationship among x, y_1, and y_2 is linear:

$$\begin{aligned} y_1 &= (a_0 - \frac{a_2 b_0}{b_2}) + (a_1 - \frac{a_2 b_1}{b_2})x + \frac{a_2}{b_2} y_2 \\ &= c_0 + c_1 x + c_2 y_2. \end{aligned} \tag{4.11}$$

This example involving two observations, y_1 and y_2, and second-order polynomials in x can be generalized to k-th order nonlinearities where x can be expressed as a linear function of $y_1, \ldots y_k$. The cloud clearing algorithm proposed in Chapter 5 is based solely on linear regression estimators, yet validated to perform very successfully, partly because the nonlinear-term cancellation like an example above may take place, also because the nonlinearities are well treated using the combinations of nonlinear estimation techniques described in the subsequent sections.

4.3.1 Linear regression using augmented nonlinear term

Mildly nonlinear estimators can be implemented by a linear estimator using a data vector augmented with simple polynomials, trigonometric functions, or other nonlinear elements which efficiently represent the kind of nonlinearity at hand.

4.3.2 Iterative linear estimation

Iterative linear algorithm is best understood by referring to Figure 4-2. It is clear that a single linear estimator will be non-optimum if we know that the desired parameter is in a region where the linear estimator is biased. If, however, the first linear estimate of the desired parameter y is followed by a second linear estimator which is conditioned on the revised probability distribution, which is much more narrowly focused on a limited range of x, the second estimate should be much better. This process can be iterated more than once, particularly if the random noise is small compared to the bias introduced by the problem nonlinearities.

4.3.3 Stratified estimation

Figure 4-3 illustrates the stratified estimation. Based on an observation input, two different linear estimators are trained using different training sets. Once trained on local data samples, an estimation of x given y first entails data segmentation based on y (in this one-dimensional case, the test: $y \gtrless y_0$). A local linear regression estimator will then be used to provide an estimate of x. However, in generally there

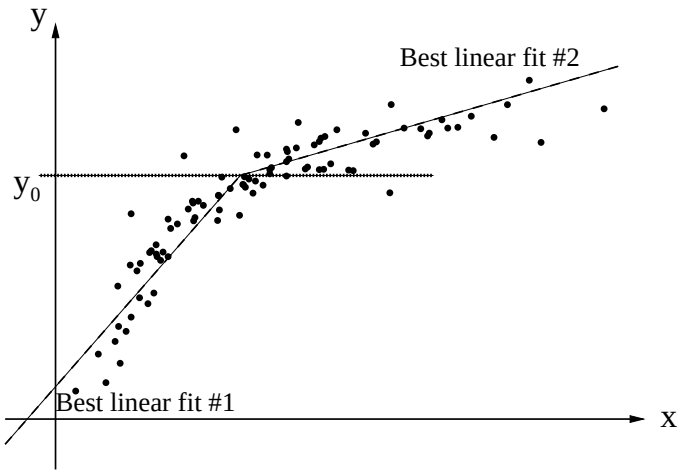


Figure 4-3: Stratified estimation using two local linear regression estimators

is discontinuity at the threshold, y_0, i.e.:

$$\hat{x}(y_0 - \epsilon) \neq \hat{x}(y_0 + \epsilon). \tag{4.12}$$

There are several methods to reduce or remove this discontinuity, including: (1) by allowing some overlapping region for training of linear fits, and (2) by the constraint that the two linear regression lines must meet. This technique is a straightforward extension of linear regression, and is useful when the dimensionality of the data or its representation is relatively small. For hyperspectral data it is generally a demanding task to segment a data set in a physically meaningful way.

4.3.4 Neural networks

Arithmetic neural networks, modeled in part after biological neural networks, compute complex polynomials with great efficiency and simplicity, and provide a means for matching the polynomials to given training ensembles so as to minimized mean-square estimation error. Neural nets can be used to learn and compute functions for which the relationships between inputs and outputs are unknown or computationally complex. A multilayer neural network is shown in Figure 4-4 [13]. In Figure 4-4, y_i is the ith input, n is the number of inputs, w_{ij} is the weight associated with the

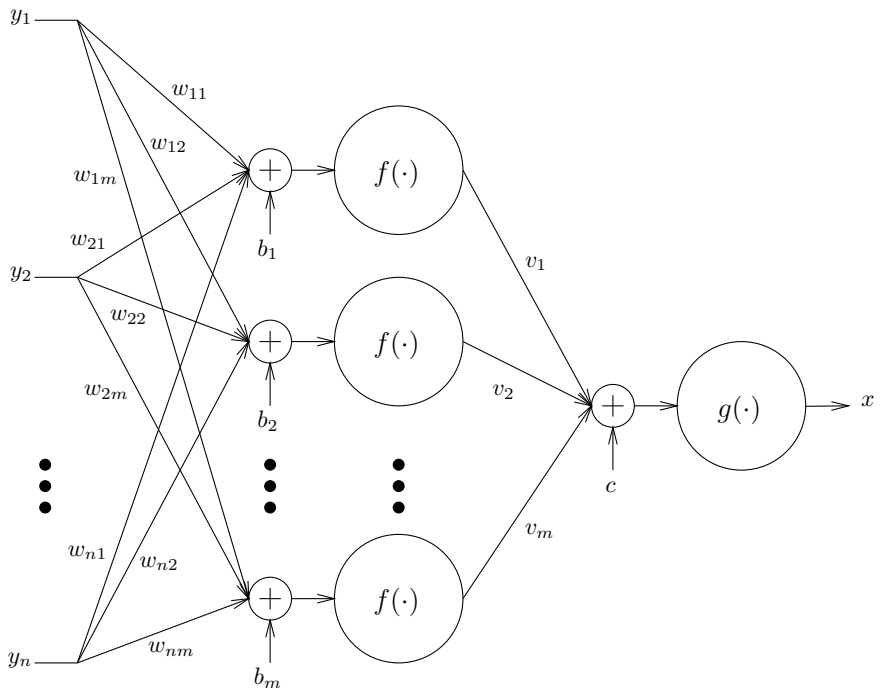


Figure 4-4: A 2-layer feedforward neural net with one output node [13]

connection from the ith input to the jth node in the hidden layer, b_i is the bias of the ith node, m is the number of nodes in the hidden layer, f is the transfer function in the hidden layer, v_i is the weight between the ith node and the output node, c is the bias of the output node, g is the transfer function of the output node, and x is the output. Then, x can be expressed as:

$$x = g\left(\sum_{j=1}^{m} v_j f\left(\sum_{i=1}^{n} w_{ij} y_i + b_j\right) + c\right). \tag{4.13}$$

A sigmoidal function is often used for f.

4.4 Characterization of nonlinearity

4.4.1 Physics-based characterization

In some cases where the underlying physics is known or predictable, one might be able to characterize nonlinearity by examining the physical relationship between the

observed and the to-be-estimated parameter. An example of physics-based characterization of nonlinearity is a scan angle effect on cloud perturbation. We saw in Chapter 2 the sounding equation, Equation 2.6, where the weighting function, $W(f,z) = \sec(\theta)\alpha(f,z)e^{-\int_z^L \alpha(f,z)\sec(\theta)dz}$. Although the weighting function has the complicated dependence on the secant of scan angle, $\sec(\theta)$, a scan angle has some multiplicative dependence on the cloud impact on the observed brightness temperature; this fact will be exploited in the cloud clearing algorithm later in this thesis.

4.4.2 Data-based characterization

Often the underlying physics is too complex or unyielding to model, in which case a data-driven approach may be available to characterize the nonlinear relationship. One simple example of such a characterization of nonlinearity is the use of a scatter plot of an observation vs a parameter. In case of hyperspectral image, their principal component scores can be used instead.

4.5 Summary

We have seen that the Bayesian least-square estimator is in general a nonlinear function of the observed data, and computing this estimator requires a complete statistical representation of the relationship between the observed and desired parameters. Thus, the BLS estimator is not practical either because implementing the nonlinear estimator is computationally exorbitant, or because a complete statistical characterization of the relationship between the observed and desired parameters (specifically, the posterior density for the parameters given observations) is unavailable from which to compute the estimator.

One alternative suboptimal estimator is linear least-square (LLS) estimator, or linear regression. In a jointly-Gaussian case, an LLS estimator turns out to be an optimum BLS estimator. A number of extensions of linear regression were addressed to capture the nonlinearities of the input/output relationship within the simplicity and practicality of the linear regression framework: Namely, the stratification, the

augmented nonlinear terms, and the iterative estimation. In this thesis, we use a linear regression and its variants for the linear and mildly nonlinear estimation of parameters.

Chapter 5

Stochastic Cloud-Clearing Algorithm

5.1 Background

In Section 2.2.4, it was shown that infrared radiance is severely affected by water vapor or clouds. In order to retrieve accurate atmospheric temperature or water vapor profiles from infrared data obtained from satellites, detection and compensation of possible cloud contamination within a field of view (FOV) is critical. Cloudy infrared radiances (or, reversibly, brightness temperatures) are difficult to use in retrieval or radiance assimilation schemes since they do not generally provide accurate information on the thermodynamic state of the atmosphere below the cloud. Therefore if clouds cover a portion of the FOV of an infrared radiometer, an accurate treatment of the effects of clouds on the observed IR radiances is imperative for obtaining accurate soundings.

5.2 Prior work on cloud-clearing

There are several approaches for treating cloud effects on the infrared observations:

- Look for only clear spots and therefore avoid the problem.

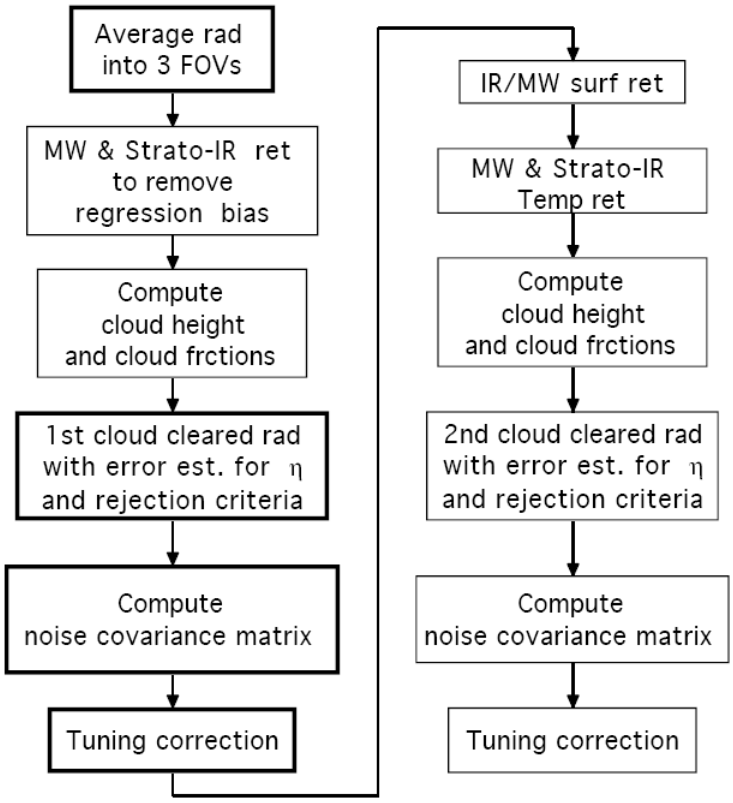


Figure 5-1: Physical cloud clearing flow diagram [10]

- A physics-based attempt to solve for the radiative effects of clouds directly in the inversion process.
- Attempt to infer what the radiances in the clear portions of the scene would be, called clear-column (or cloud-cleared) radiances, from observations in a number of adjacent FOVs.

The second and the third approach will be described hereafter.

5.2.1 Physical cloud-clearing

A physics-based method generally is computation-intensive because it requires iteration between estimation of physical parameters (retrieval) and calculation of observed radiance (radiative transfer) for a chosen number of passes or until some convergence criterion is met. Figure 5-1 shows an example flow diagram of AIRS physical cloud-

clearing algorithm [10], to illustrate the complexity of this approach. In this example, cloud-clearing involves two-pass correction processes and the estimation of cloud height and cloud fraction.

5.2.2 Adjacent-pair cloud-clearing

The key assumption made in this approach is that while there may be many types of clouds in the different FOVs, the radiative properties of a given type of cloud are identical in adjacent FOVs regardless of channels and the FOVs differ only in the relative amounts of these cloud types. FOVs containing clouds with the same optical properties but at different heights, or clouds at the same height but with different optical properties, can be considered as having multiple cloud types. The other key assumption of the approach is that the FOVs have the same characteristics in the clear portions of their scenes, with unknown temperatures, humidities, etc. that we are trying to solve for.

One well-known example of this approach is the N^* method [8, 9]. In [8], the radiances emerging from two adjacent FOVs are considered to have different fractional cloud covers at the same height; the difference between the cloud-cleared radiance $\hat{R}_i$ and the measured radiance R_i for a pair of adjacent FOVs is:

$$\begin{aligned} \hat{R}_{i,1} - R_{i,1} &= N_1 G_{i,1} \\ \hat{R}_{i,2} - R_{i,2} &= N_2 G_{i,2} \end{aligned} \tag{5.1}$$

where $R_{i,k}$ (or $\hat{R}_{i,k}$) is for the channel i observation in FOV k, and $G_{i,k}$ contains all the radiative transfer properties of the clouds, and N_1 and N_2 are the fractional cloud covers in the first and second FOVs respectively. If the two FOVs are contiguous, we can assume

$$\hat{R}_{i,1} \approx \hat{R}_{i,2} \approx \hat{R}_i \tag{5.2}$$

where $\hat{R}_i$ is defined as a clear radiance for the two FOVs. We can then reconstruct

the clear radiance $\hat{R}_i$ by substituting Equation 5.2 into Equation 5.1 to obtain:

$$\hat{R}_i = R_{i,1} + \eta[R_{i,1} - R_{i,2}] \tag{5.3}$$

where η is $N_1/(N_2 - N_1)$ for $N_1 \neq N_2$ and is independent of channel i. In principle, η can be determined from a knowledge of the clear-column radiance at any channel, i' as

$$\eta = \frac{\hat{R}_{i'} - R_{i',1}}{R_{i',1} - R_{i',2}}. \tag{5.4}$$

Note that neither FOV must be clear. Chahine [9] has also expanded this to the case of $K - 1$ adjacent FOVs. Observations in K FOVs yield channel i clear-column radiances $\hat{R}_i$:

$$\hat{R}_i = R_{i,1} + \sum_{k=1}^{K-1} \eta_k (R_{i,1} - R_{i,K+1-k}). \tag{5.5}$$

The advantage of this approach is that it does not have the clear-sky sampling bias of the first approach, nor does it require the ability to accurately model the spectral emissive, reflective, and transmissive properties of the clouds, and their dependence on the vertical microphysics and geometry, as required by the physics-based approach.

Blackwell evaluated and tested the physical and adjacent-pair approaches, but with eight adjacent pixels [5]. The better algorithm is implemented in two stages. First, a projected principal components (PPC) transform is used to reduce the dimensionality of the 9-pixel data set while preserving the geophysical profile information from the data. Second, a multilayer feedforward neural network (NN) is used to estimate the desired cloud-free radiance. It was trained on simulated data. Based on this cloud-clearing method, cloud-clearing RMS error for altitudes from 0 to 10 km ranged from 0.1 K to 1.25 K for 15 μm channels, and from 0.05 K to 0.8 K for 4 μm channels.

5.2.3 Other methods

Another approach to cloud-clearing involves the detection of clear FOVs, and estimation of cloud-cleared FOVs using spatial filtering assuming that a clear radiance

field must have some horizontal homogeneity at a fixed altitude. The approach maps the cloud-clearing to a problem of restoration or interpolation of two-dimensional radiance field from sparse clear data. In [1], the clear-sky mean latitudinal variation is restored using two-dimensional Gaussian low-pass filter.

There is an alternative approach for cloud-clearing, in which observations from another sensor usually having finer spatial resolution is utilized to help cloud-clearing. For example, the Moderate Resolution Imaging Spectroradiometer (MODIS), also on the Aqua satellite, provides colocated clear radiances at several spectrally broad infrared bands with 1-km spatial resolution. Thus, an optimal cloud-correction algorithm extends the traditional N^* technique, based on the fact that AIRS cloudy footprints may contain clear MODIS pixels [26]. In this work, about 30% of AIRS cloudy FOVs are successfully cloud-cleared with the help of MODIS high spatial resolution data. The MODIS imager provides a cloud mask for AIRS FOVs while the multispectral MODIS IR channels yield clear radiance observations to synergistically determine N^*.

Note that a data-fusion approach can enhance the cloud-clearing performance of any of the other cloud-clearing methods described above when used together. A practical issue with using another sensor to assist cloud-clearing or retrieval of physical parameters is that the collocation and alignment of two sensors, both in time and space may be challenging.

5.3 Description of stochastic cloud-clearing algorithm

5.3.1 Overview of method and rationale

Stochastic cloud-clearing methods rely on stable multivariate statistical relationships between AIRS observations and the brightness temperatures within each 45-km AMSU FOV that would have been observed in the absence of clouds [15]. Hereafter these Stochastic Clearing methods are designated "SC" algorithms. The cloud-free

spectra are estimated using European Center for Medium range Weather Forecasting (ECMWF) weather data plus a rapid radiative transfer program. The radiative transfer program, Stand-Alone Radiative Transfer Algorithm (SARTA) v1.05 available for download at [45], is one of an evolving series tuned to AIRS channels, an early version of which was described by [41]. Inputs to the Radiative Transfer Algorithm include temperature profile, water vapor profile, representative ozone profile, scan angle, and surface level altitude. Another important input is the cloud fraction which is set to zero to simulate cloud-cleared brightness temperature. The infrared spectral emissivity is assumed to be characteristic of water and modeled as the representative IR emissivity of water. The SARTA program, then, estimates brightness temperature spectra for the 2378 AIRS channels.

The nonlinear aspects of cloud perturbation in IR brightness-temperature spectra must be accommodated. For simplicity the nonlinearities were introduced here primarily by stratifying the AIRS data into ten categories characterized by different multi-linear statistics, and use of a few ad hoc nonlinear operators and iterations. The operational cloud-clearing code described below consists of 664 lines of Matlab script that can cloud-clear an entire day of AIRS data (314 channels) within 20 minutes on an 1.7-GHz Pentium PC.

The rationale for SC algorithms relies upon the observed nearly monotonic nonlinear multivariate relationship between cloudy and cloud-cleared TB's, provided the FOV being examined is at least partly clear. Although the physical degrees of freedom include the complex three-dimensional distributions of cloud particle size, phase, and shape within each FOV, all of which must be modeled for physical retrieval methods, four degrees of freedom appear to be sufficient in stochastic models. Figure 5-2 illustrates that four degrees of freedom capture most of the IR cloud impact. Although physical methods sometimes characterize FOVs by the altitudes and fractional coverages of two cloud layers [42], stochastic models apparently fold these four degrees of freedom together with others in an effective but obscure manner. Early SC experiments assumed these four degrees of freedom were linearly related to TB corrections, but the approach presented below has achieved greater success by assuming these

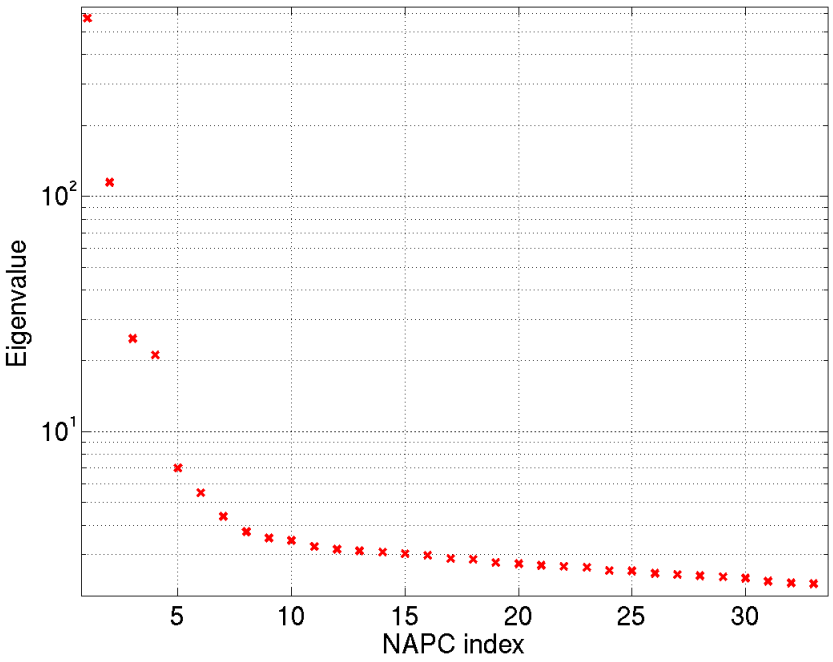


Figure 5-2: Eigenvalues (in a decreasing order) versus NAPC index for cloud perturbation

relations are mildly nonlinear.

The SC algorithm presented here accommodates nonlinearities in three ways:

1. A few simple nonlinear operators are inserted between linear matrix multiplications.

2. The data is stratified into a few sub-categories (ten here) which utilize different linear operators.

3. Nonlinear behavior is exhibited by linear combinations of TB's that are nonlinearly and differently related to the desired TB corrections.

The data is stratified into the ten categories depending on (1) land versus sea, (2) latitude within ±30 degree versus from ±30 to ±70 degree and (3) ascending orbit (daytime) versus descending orbit (nighttime); there is no stratification by scan angle. These stratification will be discussed in more detail in the following chapter.

In this study, the latitude range is arbitrarily restricted to ±70 degrees. The land/sea distinction is based on a fixed geographic database, and cloudy FOVs are assumed when the initial estimated TB corrections exceed a threshold. The inserted nonlinear operators include: 1) computation of the product of an initial TB correction (a scalar) for each FOV, and the secant of the instrument scan angle, and 2) selection of the warmest 14-km FOVs to use in estimating the cleared TB's for each set of nine such FOVs.

The agreement found here between AIRS and the corresponding time- and space-interpolated ECMWF TB's is blind to any stable biases introduced by the equations of radiative transfer or the instrument. This is because the linear estimators were trained and tested using global instrument data obtained from the same three days. The comparison is not statistically "inbred" however, because the thousands of FOVs used for training and testing are different, interspersed, and not adjacent, and the TB corrections have only four degrees of freedom across the full spectrum. The results for the selected channels are therefore indicative of what could be implemented in operational numerical weather prediction (NWP) systems assimilating satellite TB's.

The channels validated in this thesis were chosen because of their good agreement with ECMWF which will be discussed in detail in the following chapter. Extensions of these methods should be able to clear many of the remaining channels once the reasons for their divergence are better understood. Significant portions of the brightness temperature discrepancies observed on other AIRS channels are believed due to residual weaknesses in the ECMWF data involving upper tropospheric water vapor and the treatment of trace gases and CO_2 variations.

5.3.2 Initial linear estimate of TB corrections for cloud and surface effects

AIRS observes nine ∼14-km FOVs within a single AMSU-A 45-km FOV, which is called a "golfball". The stochastic cloud-clearing algorithm produces one set of cleared AIRS brightness temperatures for each golfball based on inputs that include:

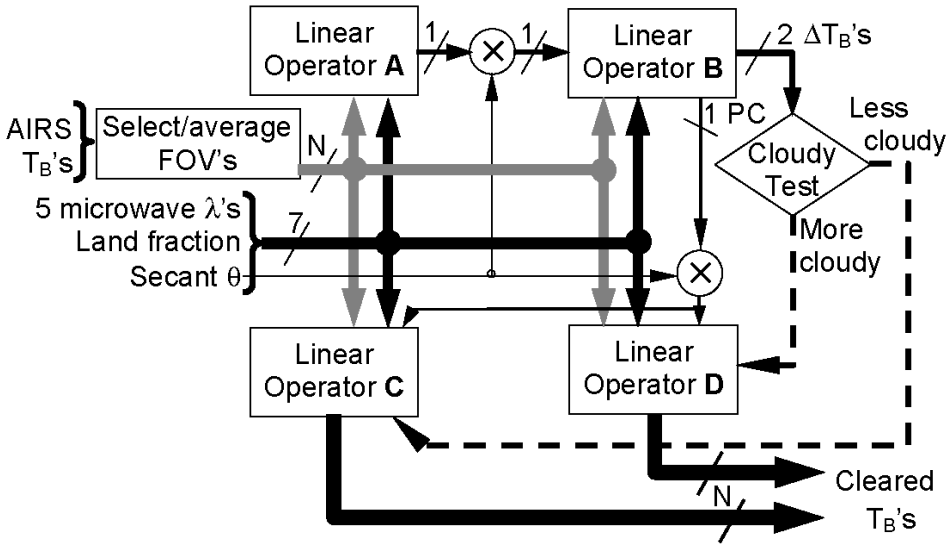


Figure 5-3: Block diagram of cloud-clearing algorithm

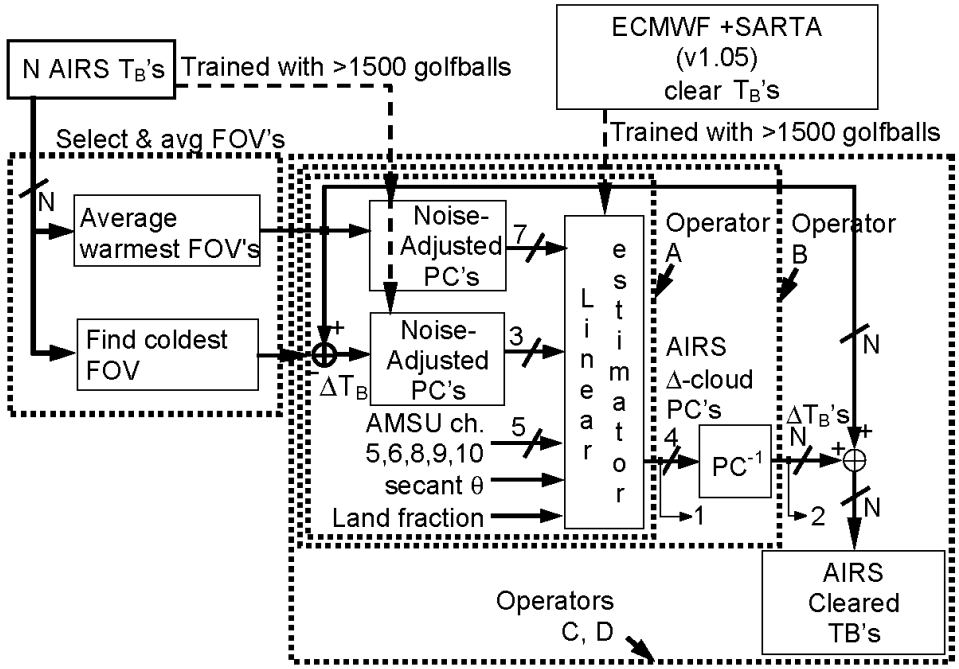


Figure 5-4: Operator for selecting and averaging FOVs for N channels, and operators A, B, C, and D

- The N AIRS Level-1B TB's of interest; the number of channels, N, is generally more than 300 for each of 9 FOVs as defined later.
- The brightness temperatures for AMSU channels 5, 6, 8, 9, and 10. See Table 2.1 for the channel specification.
- The secant of the instrument scan angle.
- The a priori fraction of land in the golfball FOV.

The SC algorithm tested here is diagrammed in Figure 5-3 and Figure 5-4, and consists of five main steps:

1. The FOVs to be used for each golfball are selected and their TB's are averaged for each of N channels.
2. An initial linear estimate of cloudiness is made (operator A). It is the first PC score, P_1.
3. A nonlinear input related to scan angle ($P_1 sec\theta$) is computed and a second linear operator (operator B) estimates two Δ brightness temperatures sounding low altitudes (927.86 cm^{-1} peaking at 0.47 km and 715.94 cm^{-1} peaking at 2.95 km) that are used to classify each golfball as either "less cloudy" or "more cloudy".
4. A final estimate of four principal components of the TB correction spectrum is made using operator C or D for the less or more cloudy golfballs, respectively.
5. This correction spectrum is added in C or D to that of the warmest FOVs for that golfball to yield the final N cloud-cleared TB's.

These steps are elaborated below.

The nine AIRS FOVs per golfball offer nine opportunities to avoid or minimize clouds per sounding. Although one FOV is generally most clear, averaging more FOVs reduces instrument noise. Brief empirical tradeoffs led to a policy of using the clearest FOV for weighting functions peaking below 5 km, an average of the

four clearest FOVs for weighting functions peaking between 5 and 10 km, and an average of all nine FOVs above that. FOV cloudiness is inferred from the average TB observed at eleven 4-μm channels having weighting function peak heights 1–3 km because the warmer FOVs are presumed to be less cloudy. To better characterize each golfball the TB's for the most cloudy FOV are also determined, although the resulting improvement is marginal. Future performance improvements should result from more elaborate FOV selection and averaging protocols.

Next the N selected warmest infrared TB's are converted to seven noise-adjusted principal component (NAPC) scores. As seen previously, NAPC's are computed for variables that have been scaled so that the variances of their additive Gaussian noises are equal; this avoids dominance of the statistics by noisy variables. These seven numbers are fed to operator A. Also fed to A are the first three NAPC's for the warmest minus coldest, together with the land fraction, the secant of the satellite scan angle, and the brightness temperatures for five AMSU channels from 53.6 to 57.5 GHz that sound tropospheric and lower stratospheric temperatures. AMSU channels 5, 6, 8, 9, and 10 were used[1]. Three NAPC's for the warmest FOV minus coldest FOV provides possible cloud information if FOVs in a golfball observe different cloud fractions. The principal components are trained on large ensembles of AIRS data as will be discussed later. These 17 numbers are fed to a linear operator A that estimates the value of the first principal component for the infrared correction spectrum. Operator A simply multiplies the 17-element input vector by the matrix A. The algorithm for this initial linear estimate of TB corrections is diagrammed in Figure 5-4.

5.3.3 Final estimate of brightness temperature corrections for cloud and surface effects

So far only a preliminary TB correction estimate exists, the scalar output of matrix A. Next a nonlinear operator computes a parameter that approximates the angular

[1]Channel 7 was too noisy.

dependence of the TB correction factor; it is the product of the output of operator A and the secant of the instrument scan angle. The motivation for taking the product is seen in Section 4.4.1: scan angle has a partly multiplicative relationship with the cloud perturbation. Although separate estimators could be constructed for each view angle and other angle dependences could be used, this estimator functions well at all angles and has the advantage of simplicity. Operator B multiplies matrix B by the same 17-element input vector plus the multiplied output of operator A. This produces estimated brightness temperature corrections for 11-μm (927.86 cm^{-1}) and 15-μm (715.94 cm^{-1}) TB's (window and temperature channels) having weighting functions peaking near 0.47 and 2.95 km for the standard atmosphere. The distributions of these corrections are indicated by the horizontal axis in Figure 5-5. The vertical axis is the final cloud-clearing error. All golfballs with brightness temperature corrections for both the 0.47- and 2.95-km channels of less than 2 and 1 K, respectively, are classified as "less cloudy"; the rest are "more cloudy". These acceptance thresholds of 2 and 1 K are set arbitrarily for illustration; they will be automatically determined for a given acceptance rate. For example, for a set of 30% clearest pixels, an acceptance threshold $T_{11\mu\mathrm{m}}$ for 11-μm channel is found via iterations such that the pixels having cloud correction less than $T_{11\mu\mathrm{m}}$ are 30% of all pixels. The other acceptance threshold $T_{15\mu\mathrm{m}}$ for 15-μm is determined in same way. Pixels which pass *both* tests are, hence, slightly less than 30%.

Initial studies show that similar performance is obtained for alternate pairs of similar channels at 4- or 15-μm wavelength. In the Figure 5-5 it is observed that many outlying pixels with large errors are effectively excluded by the dual threshold requirement.

The SC estimation process then begins anew, multiplying the same 18-element input vector by either matrix C or D, depending on whether the golfball classification is less or more cloudy, respectively. The intermediate outputs of operators C and D are the scores of the dominant four principal components of the TB correction spectrum. This correction is the estimated difference spectrum between that observed by AIRS and that computed by applying the SARTA v1.05 equation of radiative transfer to

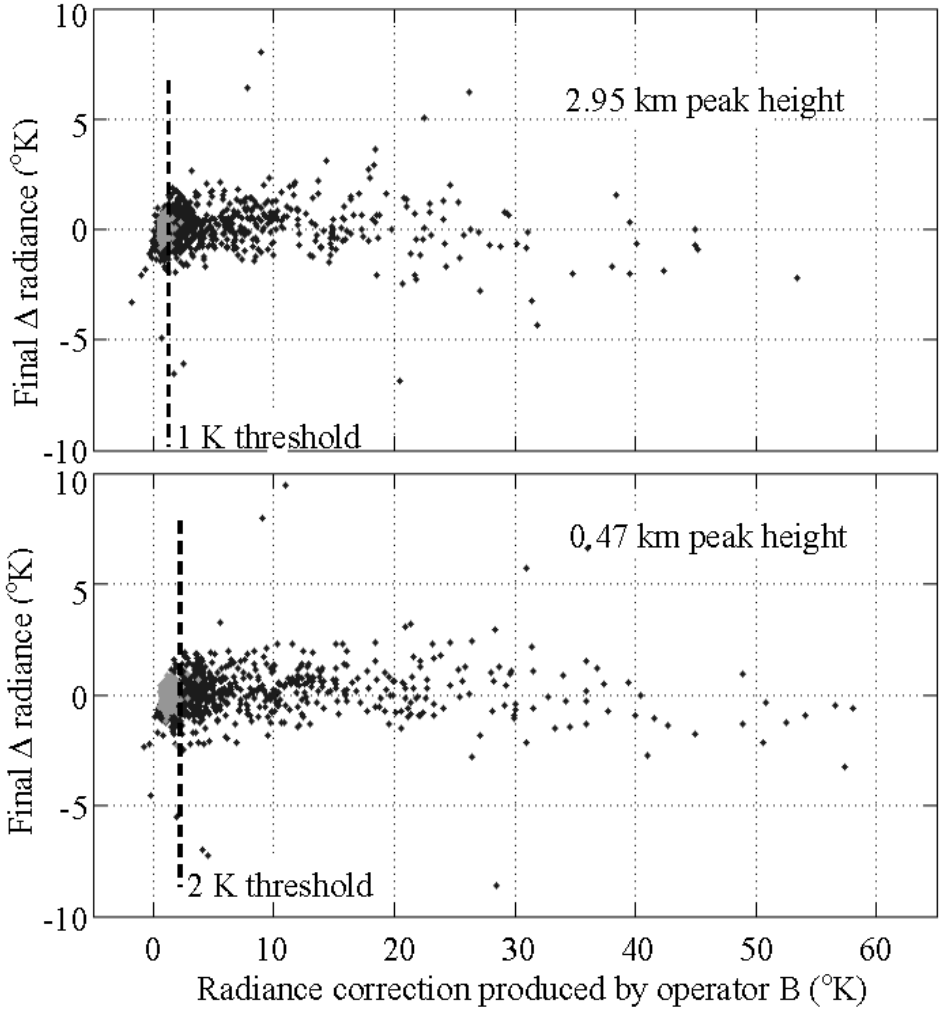


Figure 5-5: Scatter plot of cloud correction and RMS difference. Top panel represents a scatter plot of cloud-clearing error versus cloud correction for 2.95-km channel (715.94 cm^{-1}); the bottom panel for 0.47-km channel (927.86 cm^{-1})

ECMWF atmospheric fields that have been interpolated in time and space to AIRS FOV coordinates. Note that for different set of acceptance thresholds, a separate set of linear estimators C and D are used. Training and test ensembles are discussed further in the following chapter.

5.4 Summary

The SC algorithms detailed here are only simple examples of what can be implemented under the SC strategy. Alternative routines could be developed for FOV selection and averaging, for handling all scan angles, for establishing protocols and thresholds for classifying golfballs into two or more categories, for incorporating other nonlinearities, for iterating results, and for training. Neural networks can effectively combine some of these functions. The essence of SC algorithms is their substitution of stochastic models for physical ones, although physical reasoning can be incorporated in their design. The specific configuration of the current SC algorithm is somewhat ad hoc in that each parameter in the algorithm is optimized separately with no explicit consideration of possible interactions among various parameters. Number of NAPC's used for the cloud correction for example was chosen to be 3 based on the scree plot (Figure 5-2). Exploring all possible SC algorithm architectures in order to find an optimum for a specific retrieval problem is combinatorially challenging, so one future task might be to develop efficient design-of-experiment approaches for this search process.

In the context of an anomaly compensation theory, the proposed cloud-clearing algorithm relies on the combination of the following techniques:

- Spectral processing: NAPC to reduce dimensionality and computation; blind signal separation (ION algorithm) to estimate noise statistics.
- Spatial processing: selection of warmest and coldest FOVs; averaging FOVs to reduce instrument noise.
- Nonlinear estimations: correction for nonlinear scan angle effect; stratified es-

timation based on cloudiness.

The current version of the SC algorithm makes very limited use of signal processing in the spatial domain, and does not rely on any spatial filter to exploit the fact that a clear brightness temperature field must have some horizontal homogeneity at a fixed altitude. Rather the current algorithm heavily depends on spectral processing – NAPC and ION. One can set up a cloud-free temperature-field variation model using clear atmosphere statistics, and accordingly design a spatial filter which allows the clearing or reduction of cloud effect. Although this is not sought after in this study, once the spatial filter is designed, one can further iterate the current cloud-clearing algorithm and the reliable spatial filtering, or merge these two processes.

Chapter 6

Validation of Cloud-Clearing Algorithm

6.1 Classification of AIRS channel behavior

An early SC experiment involved cloud clearing 827 AIRS channels, including all 4-μm and 15-μm channels plus one-fifth of the rest[1], and excluding the noisy channels detected by the algorithm described in Section 3.3.2. Figure 6-1 shows the RMS difference between the SC AIRS brightness temperatures (TB's) and those predicted by ECMWF/SARTA; the horizontal axis indicates the altitude at which the weighting function peaks for a US standard atmosphere. The weighting function refers to the temperature weighting function of a given channel, $W(f, z)$, discussed in Section 2.2.3. This analysis was restricted to oceans at night in order to minimize surface and other effects. Three full days (August 21, September 3, and October 21, 2003) of data at all scan angles were analyzed for latitudes within 40 degrees of the equator. Only the least cloudy 22% of all golfballs were included in the statistics. In general the 4-μm band exhibits the highest accuracy in the troposphere, whereas the 15-μm band excels in the stratosphere.

[1]Specifically, AIRS channel numbers for selected 827 channels are all 101–500 (12.34–14.82 μm), every fifth from 501–1260 (8.81–12.34 μm), all 1263–1301 (8.09–8.22 μm), every fifth from 1302–1900 (4.52–8.08 μm), all 1900–1975 (4.508–4.517 μm), and every other from 1977–2300 (3.87–4.37 μm).

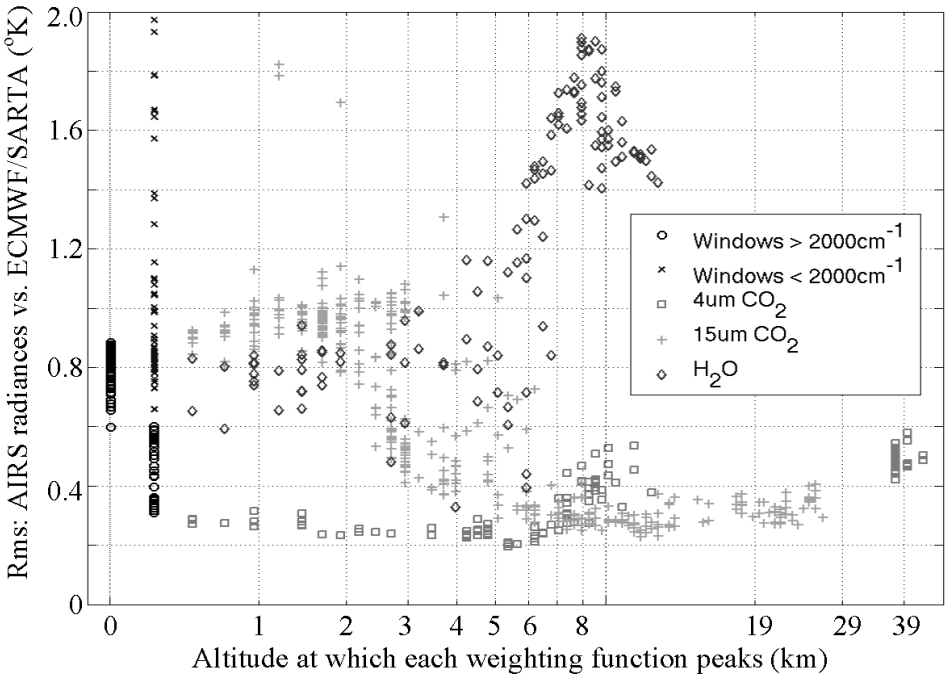


Figure 6-1: RMS difference of cloud-cleared estimate and ECMWF (K)

The accuracies for the window and water vapor channels vary considerably. The systematic peaking of RMS discrepancies for the water vapor channels near 8-km altitude could very likely be due to decreased ECMWF humidity accuracies there, suggesting that assimilation of AIRS water vapor data into operational numerical models could be quite helpful.

The 4-μm channels exhibit RMS errors generally below 0.4 K at all altitudes between $\sim$300 meters and 40 km. The larger 4-μm errors and channel absences evident in the figure at certain altitudes are largely explained by the Planck function; the weighting function widths and channel sensitivities near 4 μm become problematic at the low temperatures and reversed lapse rates found in and above the tropopause. The excellent cloud- and surface-clearing performance below 2 km altitude results largely from the strong temperature dependence of the 4-μm Planck function and the ability of multiple channels with different temperature and aerosol sensitivities to compensate for partially cloudy FOVs, even in the absence of large clear "holes" in the atmosphere.

The 15-μm channels exhibit three regimes of behavior. The first regime below ∼3-km altitude exhibits more than twice the variance of those channels in the second regime sounding higher altitudes. The third regime involves selected channels peaking near 3–8 km altitude. In general, the low-variance group above 3-km altitude is behaving as expected, whereas the group below 3 km has additional errors due to some combination of: 1) imperfections in the SC algorithm, 2) CO_2 variability, and 3) instrument anomalies, some of which have since been compensated algorithmically. The third group of channels appears to be affected in part by one or more variable trace gases not present in the ECMWF data.

Only four water vapor channels exhibit agreement with ECMWF below 0.5 K, all in the 3–6 km altitude range (only one-fifth of the water vapor channels appear in the figure). Most water vapor channels at comparable altitudes exhibit larger RMS discrepancies, however, due to an unknown combination of trace gas effects, instrument anomalies, and errors in ECMWF fields and spectroscopy. In contrast, those water vapor channels below 3 km all behave quite similarly, exhibiting RMS discrepancies of ∼0.7 K. The group above ∼6 km exhibits RMS errors that exhibit an arc-like distribution that peaks distinctly near 8-km altitude and 2 K. This arc-like distribution is even more unambiguous in daytime data, with RMS discrepancies peaking near 3.6 K. Because all water vapor channels in this altitude range behave similarly, one likely explanation is imperfections in the ECMWF upper tropospheric humidity fields.

6.2 Evaluation of selected channels with respect to ECMWF

The analysis here primarily explores the precision of AIRS since the SC algorithm is both trained and tested on the same type of data, resulting in cancellation of multi-day global mean errors [15]. The revealed precision does indicate, however, the utility of AIRS cloud-cleared TB's for operational numerical weather predictions.

Type	Description
Coverage	Global grid
Date	August 21, September 3, and October 12, 2003
Level 2 version	3.0.8
Radiative transfer program used	SARTA v1.05
Scan angle	All scan angle used
Latitude	Up to 70° North and South
Orbit	Both ascending and descending
Channels analyzed	314 channels including 4- and 15-μm sounding channels, 8-μm window channels, and water vapor channels (See Appendix A for the full channel listing.)
Land definition	Land is defined by land fraction $>$ 0.8 and elevation $<$ 0.5 km

Table 6.1: Data specification used for the evaluation of selected channels with respect to ECMWF

The validation standard used here is ECMWF analysis fields processed using SARTA v1.05 for radiative transfer. Three full days of global data are analyzed here: August 21, September 3, and October 12, 2003, the third day being relatively cloudy. During this period the instrument appeared to be relatively stable. Level 1B v3.08 AIRS data within 70° latitude of the equator were used in the stratified fashion: FOVs within 40° latitude and FOVs from latitude 30° to 70° are trained separately using a different set of estimators, denoted by A, B, C and D, in Figure 5-3. When tested, the overlapping regions, i.e. FOVs from latitude 30° to 40°, are calculated using the estimators trained with ±40°. Table 6.1 summarizes the data specification used for this evaluation.

For each evaluation approximately half the qualified golfballs were used for training and half for testing, both sets being arranged in superimposed non-contacting regular lattices. Since no training or testing golfball was ever adjacent, and since both land and clouds have correlation distances generally less than ∼100 km, the two sets can be regarded as largely independent for purposes of evaluating instrument precision. Systematic variable errors in spectroscopy, atmospheric modeling, and clearing algorithms are evident only as unexplained increases in variance, and mean errors are not revealed. The variances reported here therefore place only an

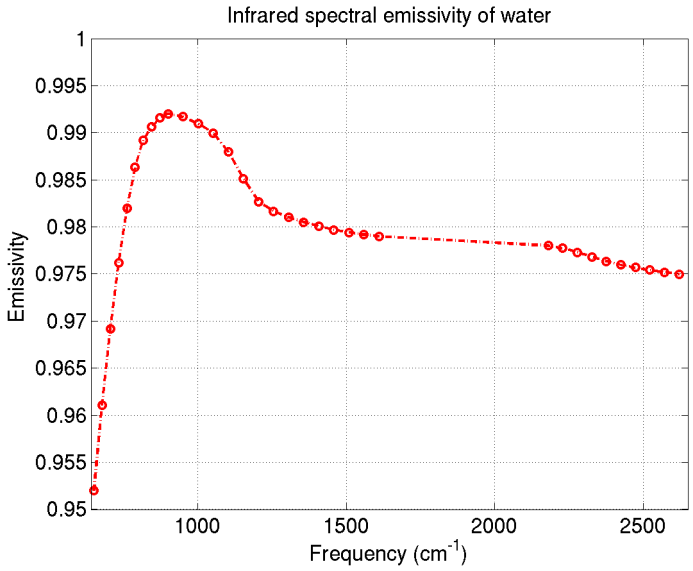


Figure 6-2: Infrared spectral emissivity

upper bound on all systematic variable errors; the instrument could in fact be better.

The ECMWF data consists of temperatures and absolute humidity at the surface and at 60 pressure levels extending to 0.1 mbar ($\sim$ 66 km). They were interpolated in time and space to the center of each AIRS golfball. The ECMWF fields utilized by SARTA did not incorporate any clouds, aerosols, or precipitation. The surface of both land and sea was assumed to be characteristic of water, varying between 0.95 and 0.99, depending on wavenumber as shown in Figure 6-2 [18]. Since the average errors in the assumed emissivity partly cancel because they occur both in training and testing, and because the AIRS observations alone can largely compensate for surface variations, the assumption of ocean emissivity characteristics everywhere should not introduce much error.

One of the single most important applications of AIRS data will involve variational assimilation of AIRS radiances by operational weather forecasting models. This section therefore focuses on those channels initially best suited to this purpose — the set of 314 exhibiting RMS TB discrepancies below 0.5 K in Figure 6-1. The selected 314 channels are listed in Appendix A. The SC algorithm of Figures 5-3,5-4 was again employed over ocean and land using only these 314 channels for all angles

and both day and night, yielding the results shown in figures in Appendix C if ∼78 and ∼28% of all golfballs are considered, depending on the acceptance threshold used in the cloudy test (see Figure 5-3). The acceptance thresholds for channels peaking near 2.7 and 0.47 km are determined based on a given acceptance rate. For example, if the acceptance rate is set at 30%, each acceptance threshold is iteratively found to yield the acceptance rate of 30%. The actual percentage of golfballs which pass both threshold tests are, therefore, less than 30%, usually ∼28% for all categories. The reason the listed RMS cloud-clearing performance actually improves near the limb is that somewhat fewer golfballs pass the cloud test there. Note that the agreement would be still better if the performance had not been averaged over all channels peaking within a given 1-km altitude block, where these channels may include both water vapor channels and 4- and 15-μm CO_2 channels. There are 10 stratifications depending on land/sea, lower or upper latitude (lower being within ±40°, upper being from ±30° to ±70°), and day/night/day+night:

- Land, |latitude| < 40, day.
- Land, |latitude| < 40, night.
- Land, |latitude| < 40, day + night.
- Land, 30 < |latitude| < 70, day.
- Land, 30 < |latitude| < 70, night.
- Land, 30 < |latitude| < 70, day + night.
- Sea, |latitude| < 40, day.
- Sea, |latitude| < 40, night.
- Sea, 30 < |latitude| < 70, day.
- Sea, 30 < |latitude| < 70, night.

A set of five figures on each page in Appendix C represents:

- 1-km-bin average RMS difference for cloud-cleared TB relative to ECMWF/SARTA plotted against weighting function peak in km.

- 1-km-bin average RMS difference for cloud-cleared TB when AMSU is eliminated in both training and testing.

- RMS cloud-clearing difference for all 314 channels using best ∼28% golfballs[2].

- RMS cloud-clearing difference for all 314 channels using best ∼28% golfballs, if AMSU is unavailable.

- 1-km-bin average RMS difference comparisons for best ∼78% and ∼28% with and without AMSU contribution.

The number of golfballs used for the training is different for each category, but is more than 1000 for all cases. For all categories, 13,960 golfballs are used for the training and 13,088 are used for the testing. Figure C-1 and Figure C-8 (best cases for land and sea, respectively) demonstrate that for the best ∼28% of all golfballs the RMS discrepancies are generally ∼0.3 K down to ∼5-km altitude, and degrade to ∼0.5 K at the surface; over land the lowest kilometer exhibits ∼0.85 K differences. These results could be improved somewhat by being still more selective with channel selection, as indicated by the discrepancies shown in Figure 6-1 that approach 0.2 K for some channels. The acceptable percentage of golfballs generally increases with altitude, particularly above 3–7 km. For example, relative to the 28% group the variance is very little worse for the best 78% of golfballs for channels peaking above ∼3-km altitude, and for all golfballs for channels peaking above ∼10 km. Over land the discrepancy in Figures C-2 for all golfballs is less than 0.5 K down to 7 km. The preliminary implication is that SC methods work reasonably well if any FOVs are partially clear, and being more selective within that group of golfballs offers little additional improvement.

[2]The "best" or "clear" X% golfballs refer to ones which pass the dual-channel threshold test, thus presumably least cloudy data. These golfballs then go to a linear operator C. For different set of acceptance thresholds, a separate set of linear estimators C and D are used.

Weighting function peak height (km)	Ocean				Land					
	$\|Lat\| < 40$		$30 < \|Lat\| < 70$		$\|Lat\| < 40$			$30 < \|Lat\| < 70$		
	day	night	day	night	day	night	all	day	night	all
0-1	0.38	0.40	0.86	0.91	1.68	0.77	1.36	1.48	0.78	1.19
1-2	0.27	0.29	0.54	0.57	0.94	0.38	0.75	0.84	0.44	0.70
4-5	0.28	0.30	0.45	0.45	0.34	0.29	0.33	0.41	0.33	0.39
6-7	0.23	0.27	0.34	0.36	0.25	0.24	0.28	0.34	0.26	0.31
10-11	0.24	0.27	0.33	0.35	0.23	0.25	0.26	0.24	0.28	0.27

Table 6.2: Cloud-clearing RMS difference with respect to ECMWF for the best 28% golfballs

Weighting function peak height (km)	Ocean				Land					
	$\|Lat\| < 40$		$30 < \|Lat\| < 70$		$\|Lat\| < 40$			$30 < \|Lat\| < 70$		
	day	night	day	night	day	night	all	day	night	all
0-1	0.70	0.74	1.49	1.47	1.62	1.11	1.53	1.49	1.21	1.39
1-2	0.50	0.53	1.02	0.98	0.90	0.68	0.91	0.92	0.77	0.87
4-5	0.42	0.42	0.65	0.62	0.43	0.44	0.48	0.51	0.46	0.49
6-7	0.36	0.36	0.54	0.49	0.37	0.36	0.42	0.42	0.35	0.40
10-11	0.28	0.30	0.38	0.37	0.29	0.28	0.32	0.25	0.31	0.30

Table 6.3: Cloud-clearing RMS difference with respect to ECMWF for the best 78% golfballs

In the figures, the improvement for multiplicative scan angle correction is small and concentrated on 5–10 km for the lower latitude cases. Interestingly, all the upper latitude results (Figures C-4, C-5, C-6, C-9, C-10) show almost no improvement. In the previous experiment in which no stratification (based on land/sea, day/night, and latitude) was used, however, there was significant improvement using multiplicative scan angle correction. By employing the stratified estimation, the effect of multiplicative scan angle correction diminished.

Tables 6.2,6.3 summarize the SC performance. For ten different stratifications, cloud-clearing RMS differences are shown in Tables 6.2, 6.3 for best 28% and best 78% golfballs, respectively.

Weighting function peak height (km)	Ocean				Land					
	|Lat| < 40		30 < |Lat| < 70		|Lat| < 40			30 < |Lat| < 70		
	day	night	day	night	day	night	all	day	night	all
0-1	0.00	-0.01	0.00	0.06	0.10	0.09	0.21	-0.05	0.21	0.06
1-2	-0.01	0.00	0.01	0.00	-0.06	0.10	0.10	-0.03	0.11	0.00
4-5	0.00	0.00	0.00	-0.02	0.00	0.03	0.03	-0.01	0.02	-0.03
6-7	0.00	0.00	0.02	-0.02	0.00	0.01	0.01	-0.03	0.01	0.00
10-11	-0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.01	-0.01	0.01

Table 6.4: Cloud-clearing penalty without using AMSU for the best 28% golfballs

Weighting function peak height (km)	Ocean				Land					
	|Lat| < 40		30 < |Lat| < 70		|Lat| < 40			30 < |Lat| < 70		
	day	night	day	night	day	night	all	day	night	all
0-1	0.01	0.06	0.19	0.35	0.11	0.47	0.13	0.18	0.40	0.34
1-2	0.01	0.04	0.14	0.24	0.10	0.37	0.11	0.15	0.29	0.24
4-5	0.01	0.02	0.08	0.08	0.04	0.15	0.07	0.04	0.07	0.09
6-7	0.01	0.01	0.05	0.03	0.01	0.08	0.06	0.01	0.01	0.04
10-11	0.00	0.00	0.00	0.01	-0.01	0.01	0.01	0.00	-0.01	0.01

Table 6.5: Cloud-clearing penalty without using AMSU for the best 78% golfballs

6.2.1 Cloud-clearing without using AMSU

Also tabulated in Tables 6.4, 6.5 are the RMS AMSU contribution to cloud-clearing, or equivalently, the cloud-clearing penalty (increase in RMS error) when AMSU is unavailable, for the best 28% and best 78% golfballs, respectively. It is evident that microwave contributions are essential, especially for ocean/upper latitude/night case, and land/night cases, for the best 78% group.

6.2.2 Cloud-cleared images

Figure 6-3[3] shows original AIRS brightness temperature at 2392.1 cm^{-1} channel, for which the weighting function peaks ∼230 meters above the nominal surface. Each vertical scan-line contains 90 FOVs. This image is intended to illustrate the cloud patterns present in the observed TB field.

[3]Be advised that, including this figure, there are several figures in this thesis which are best viewed in color to convey the maximum amount of information.

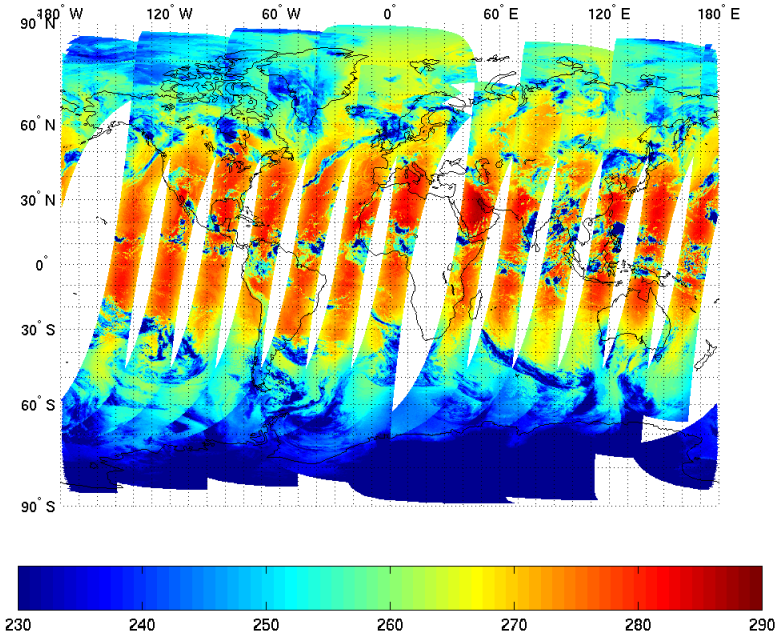


Figure 6-3: Global AIRS original brightness temperature at 2392.1 cm^{-1}

Figure 6-4 characterizes SC performance in another fashion. The SC algorithm was trained as before on the 314 best channels (see Appendix A for the full channel listing), and was then applied to typical daytime AIRS orbits obtained August 21, 2003. The top image in Figure 6-4 shows the best ∼78% cloud-cleared brightness temperatures for the same channel; ∼22% golfballs which failed the "clear" test are shown as white. Also surface elevation higher than half of kilometer is shown as white since those are not included in training. We can observe that almost all cloud patterns are either eliminated as cloudy or effectively cleared.

The discontinuities occur where the edge of one scan overlaps the interior of another due to the scan-angle dependence of the brightness temperature and the effects of sea ice, which was not included in the training. The bottom image in Figure 6-4 illustrates the spatially Gaussian-high-pass filtered version[4] of the SC image above,

[4]The original image is low-pass filtered with a 2-dimensional Gaussian-shaped impulse function. The Gaussian impulse function used here has standard deviation of 2 and 8.4 pixels in cross-track and along-track direction, respectively. The high-pass filtered image is, then, defined as the original image subtracted by the low-pass filtered image.

thereby highlighting the fine-scale variations. Figure 6-5 is an enlargement of residual image in Figure 6-4, in the Southeastern pacific. Features in the residual image appear to be of true meteorological origin.

The top image in Figure 6-6 shows the original AIRS brightness temperatures at 2390.1 cm^{-1}, for which the weighting function peaks $\sim$ 230 meters above the nominal surface. This data is observed in the Pacific Ocean near Hawaii on July 14, 2003. The baseline has been increased toward the limb by averaging the SC results for all scans and subtracting that average function of angle from both the upper and lower images. Since that cleared average is everywhere warmer than the uncorrected brightness, the original image is everywhere negative, the offset being $\sim$1 K for the 16 clearest golfballs. As seen in the figure, although some clouds exhibit 14-km holes, most clouds have no clear 14-km FOVs. The middle image in Figure 6-6 shows the angle-flattened relative SC cloud-cleared TB's, most of which fit within a 2-K dynamic range and, more locally, within a $\sim$ 0.6 K range. Each vertical scan-line contains 30 golfballs that have been bilinearly interpolated. The bottom image shows the sea surface temperature provided by NCEP. It is clear that most hole-free clouds have been cleared with reasonable accuracy, and that only the more intense events remain evident in the middle image. The cleared image has a temperature difference left-to-right of 1.36 K, whereas the corresponding difference for the NOAA/NCEP-provided sea-surface temperatures was 1.6 K. The NCEP sea surface data shown in the bottom also exhibit in the same position the sharp thermal front that is centered in the lower image, thus confirming the horizontal gradient in the cloud-cleared brightness temperature is meteorologically true. Figure 6-7 shows similar images as in Figure 6-6, but observed in the Southwestern Indian Ocean on January 1, 2003, at 2399.9 cm^{-1}, for which the weighting function peaks $\sim$ 230 meters. This set of images again verifies that the temperature gradient in the cleared image matches the corresponding sea surface image.

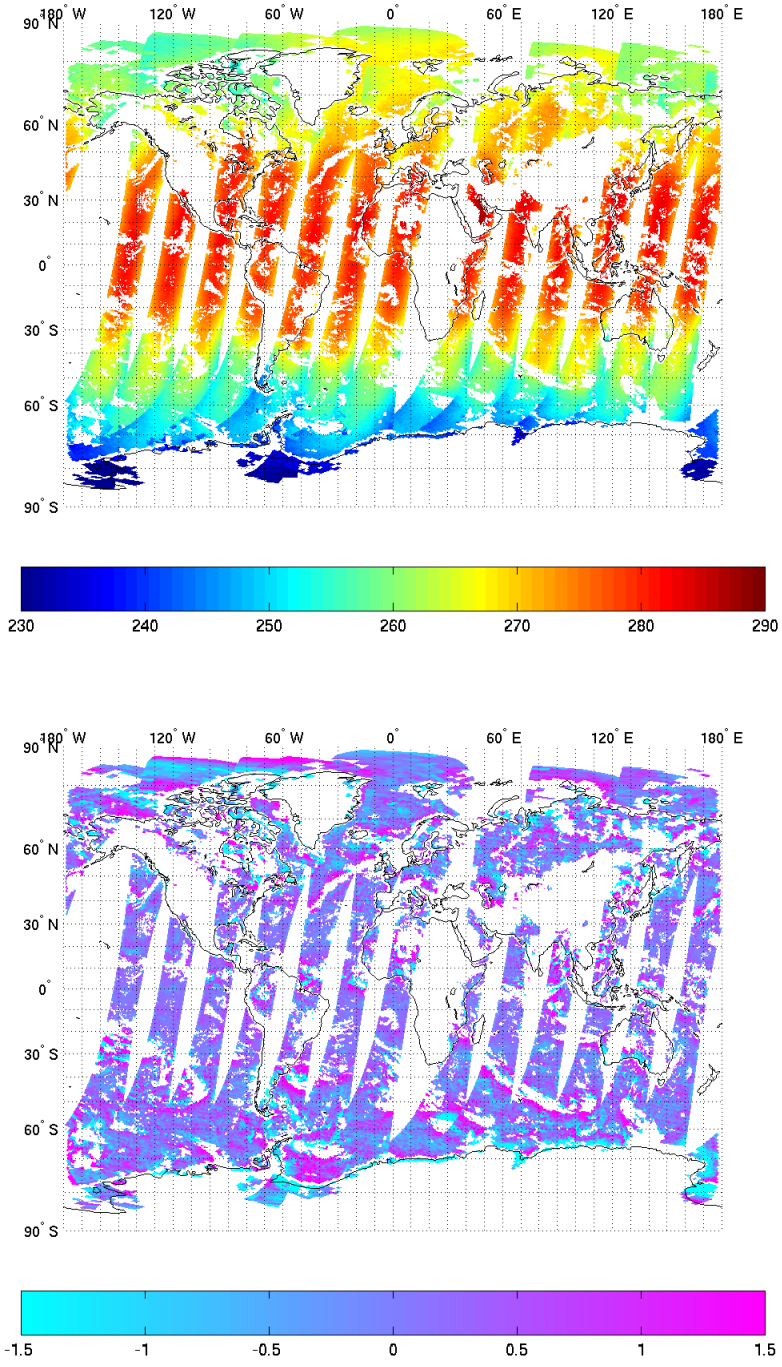


Figure 6-4: Global AIRS cloud-cleared and residual brightness temperatures

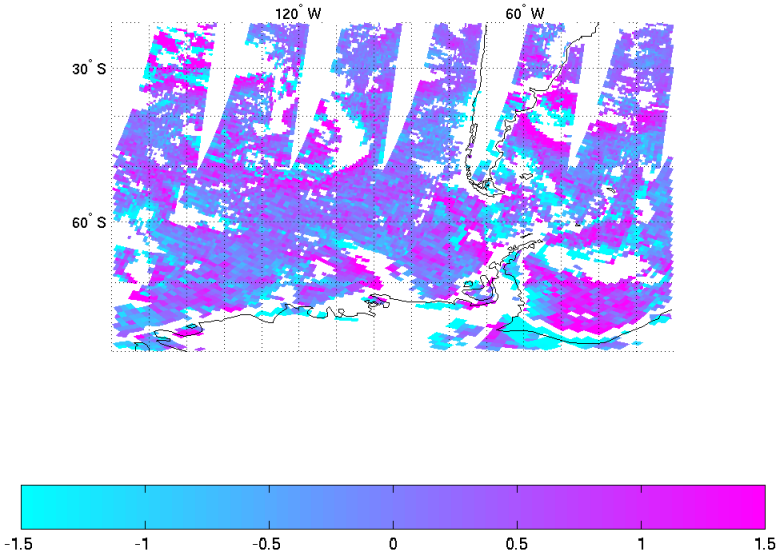


Figure 6-5: Residual of AIRS cloud-cleared TB's in Southeastern pacific

6.2.3 Angular dependence of performance

One of the surprising results from these SC experiments is the near lack of performance degradation at extreme scan angles. Table 6.6 lists the RMS differences between the SC-corrected AIRS TB's as a function of scan angle for a representative channel at 2390.1 cm^{-1} peaking near 1.9 km. All golfballs were tested for three days (August 21, September 3, and October 12, 2003), including day and night, land and sea. The percentages of golfballs that passed the threshold for each angular group are also listed. Together the slightly improved performance and reduced yield near the limb suggest that SC performance is not only nearly independent of viewing angle, but also largely independent of spatial resolution, for the FOV area increases more than a factor of three at the extreme viewing angle. This result is expected, however, if the SC algorithm can indeed successfully use FOVs that are each only partly cloud-free. Thus this result for the clouds of Figures 6-6,6-7 reinforces the observation earlier that stochastic cloud clearing appears successful even when golfballs have no totally

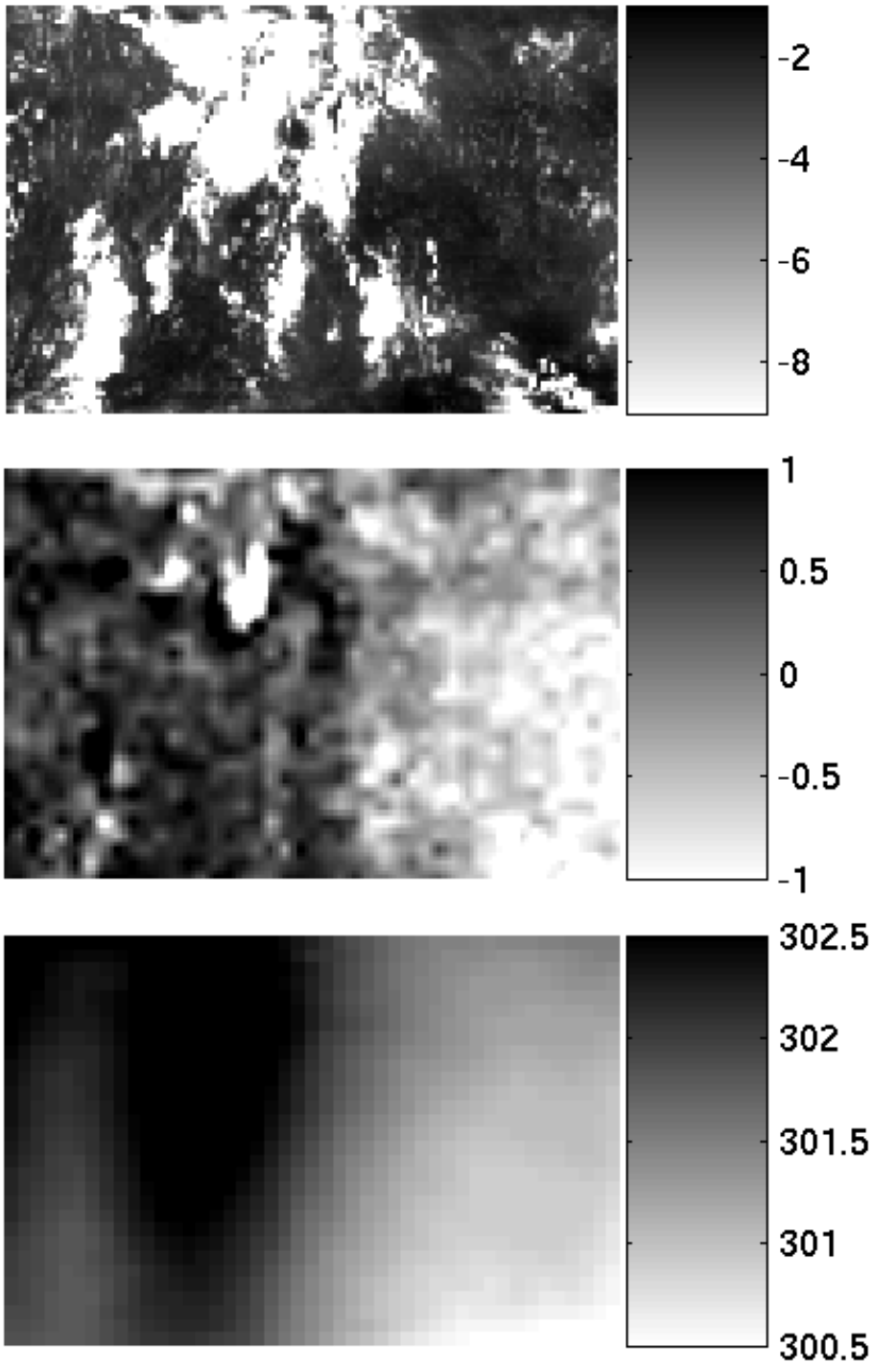


Figure 6-6: AIRS 2390.1 cm^{-1} angle-corrected relative brightness temperatures (°K) near Hawaii (upper image), the corresponding SC cloud-cleared temperatures (middle image), the sea surface temperature (lower image)

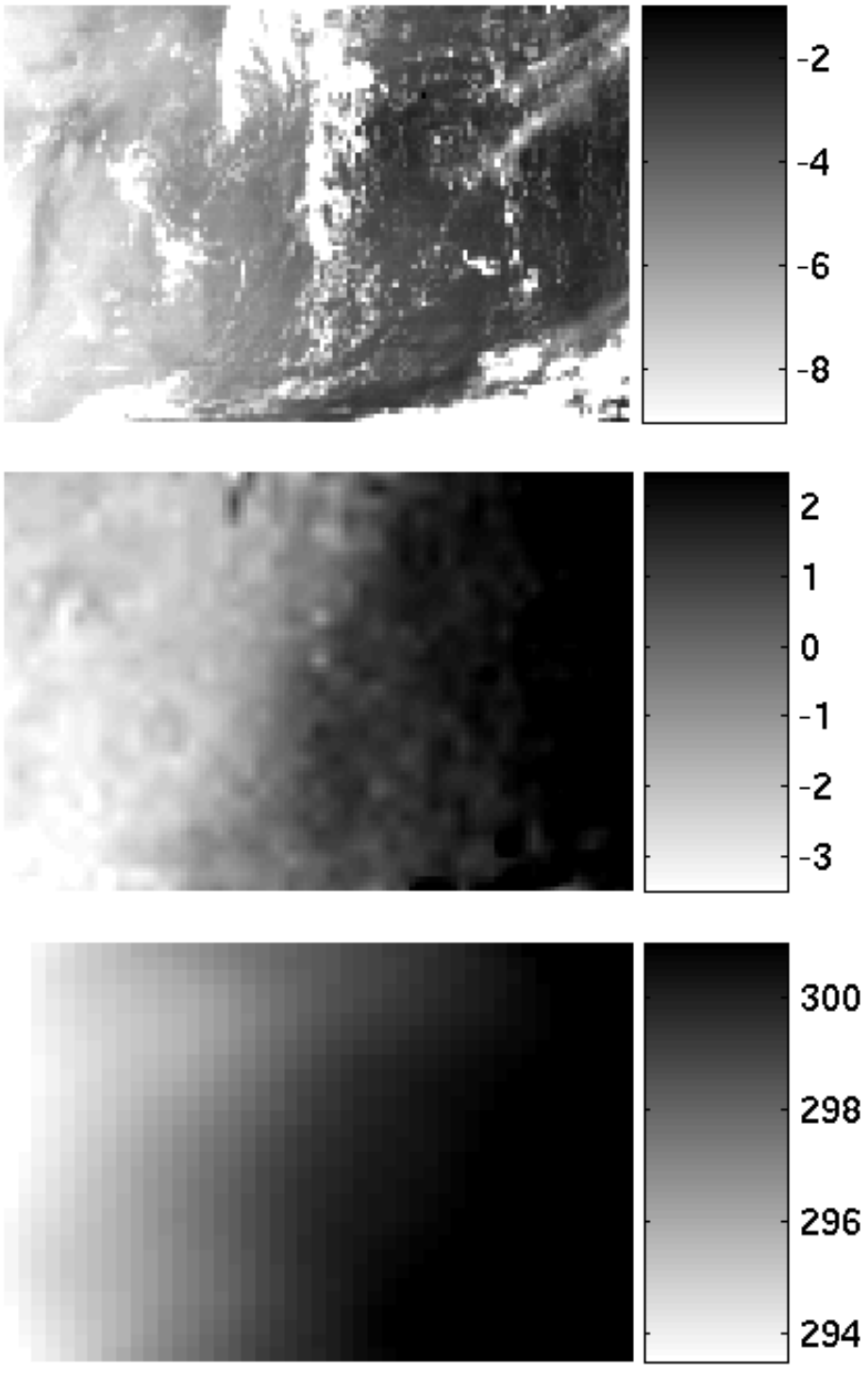


Figure 6-7: AIRS 2399.9 cm^{-1} angle-corrected relative brightness temperatures (°K) in Southwestern Indian Ocean (upper image), the corresponding SC cloud-cleared temperatures (middle image), the sea surface temperature (lower image)

Scan angle (degrees from nadir)	AIRS vs ECMWF (RMS °K)	Percentage in clearest 40.6 percent
All	0.44	40.6
0–10	0.49	40.6
10–20	0.58	40.1
20–30	0.43	43.0
30–40	0.38	37.6
40–48	0.36	33.5

Table 6.6: RMS TB discrepancies at 2390.1 cm^{-1} between AIRS and ECMWF/SARTA as a function of scan angle

clear FOVs.

6.3 NCEP sea surface temperature

The previous validation scheme is based on the training and testing with the same type of data, ECMWF/SARTA simulated cloud-cleared TB, as ground-truth. Sea surface temperature (SST) is also a good indicator of cloud-clearing performance for the window channels [14] because SST should not be affected by clouds. Unlike ECMWF/SARTA data set which involves radiative transfer program to simulate cloud-cleared spectra from physical parameters, SST offers a very simple yet reliable validation tool. The SST data is provided by National Centers for Environmental Prediction (NCEP) as a part of numerical weather prediction model. The proposed cloud-clearing algorithm can easily accommodate SST retrieval with a minor modification; instead of estimating 4 NAPC scores of cloud-clearing perturbation, the linear regression estimates a single scalar, SST. Thus, in the cloud-clearing algorithm described in Figure 5-3, the output of linear estimator should be SST in order to adapt the algorithm to predict SST.

For SST retrievals, 1755 and 1365 golfballs collected on January 3, April 9, July 14 in 2003 were used for the training and testing respectively. All the data used in this evaluation are located in ocean between 40 North and 40 South in latitude. Only one third of the golfballs close to nadir are used. Table 6.7 summarizes the data

Type	Description
Coverage	24 granules in ocean (each granule covers ∼1800 km by ∼400 km)
Date	January 3, April 9, and July 14, 2003
Level 2 version	3.5.0
Scan angle	$\|\theta\| < 16°$
Latitude	Up to 40° North and South
Orbit	Ascending (daytime)
Number of golfballs	1755 golfballs for training, 1365 different golfballs for testing
Channels analyzed	294 channels including 4- and 15-μm sounding channels, 8-μm window channels, and water vapor channels
SST data	Provided by National Centers for Environmental Prediction (NCEP)

Table 6.7: Data specification used for estimation of NCEP sea surface temperature

specification used for this evaluation. Furthermore, about 29% of all pixels which passed quality assurance test are used for training and testing. Figure 6-8 shows RMS differences with respect to the NCEP SST for three different approaches to retrieving SST. The horizontal axis is percentage of quality assurance (QA) passed pixels, and pixels are rank ordered using cloud-cleared minus observed TB for an 8-μm (1217 cm^{-1}) window channel. First, SST is calculated by AIRS version 3.5.0 cloud-cleared TB's, denoted as the red curve. SST is calculated from two window channels and the zenith angle, ϕ:

$$\widehat{SST} = TB_{1231} + 0.28 + 1.2(TB_{1231} - TB_{1227})^2 + 1.49sec(\phi) \tag{6.1}$$

where TB_{1227} and TB_{1231} refer to the brightness temperature at AIRS channel 1227 and 1231 respectively, and ϕ is a zenith angle [3]. Second, the stochastic cloud-clearing algorithm trained with AIRS cloud-cleared TB [10] yields the blue curve, using the same SST-retrieval calculation with 2 window-channel brightness temperatures. It follows the AIRS result. Third, SST is directly estimated by the stochastic cloud-clearing algorithm, which is trained with NCEP SST. This scheme bypassed the estimation of cloud-cleared TB's, thus outperforming the first two. Also shown is

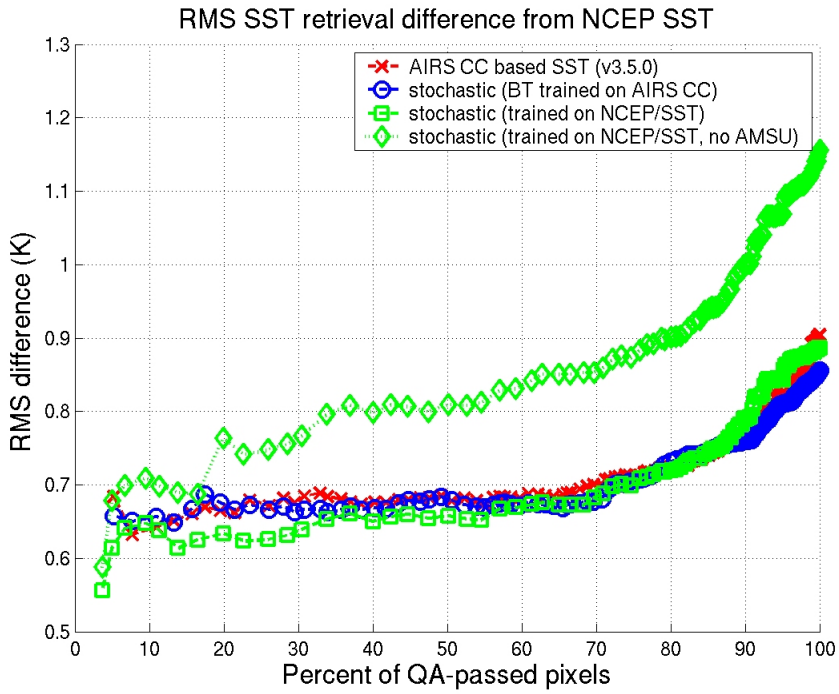


Figure 6-8: SST retrieval difference

the performance of the stochastic SST retrieval without using AMSU. It is clear that microwave observations are critical for SST retrieval as expected.

6.4 Comparison with physical clearing

6.4.1 Using correlation between cloud-cleared image and Visible channel 3

AIRS science team developed a physical method for cloud-clearing, involving the estimation of cloud attributes [42]. In this section, a new SC algorithm is both trained and tested on the same type of AIRS-team physical cleared data. The purpose of this analysis is to quantify the stochastic cloud-clearing performance when trained with physical CC data. The core algorithm used for the cloud-clearing is still the same as before except only one-pass estimator (only C or D) is used instead of a set of four estimators (A, B, C, and D) in Figure 5-3. The reason for neglecting the correction

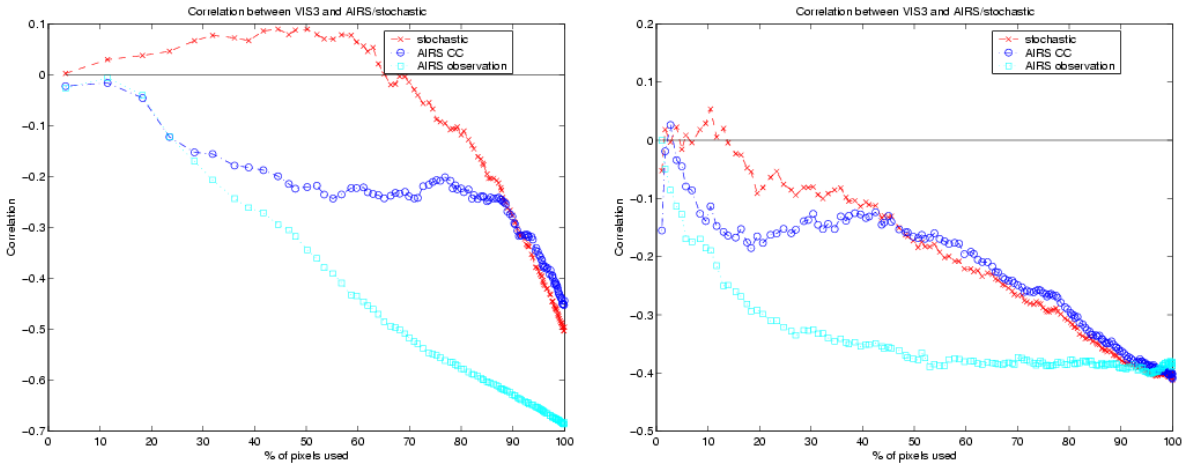


Figure 6-9: Correlations with AIRS visible channel 3: Note that *stochastic* (red curve) refers to the correlation between AIRS visible channel 3 and stochastic cloud-cleared 8-μm channel with respect to a percentage of pixels that are rank-ordered using cloud-cleared minus observed TB for an 8-μm window channel; *AIRS CC* (blue curve) refers to the correlation between AIRS visible channel 3 and AIRS-team physical cloud-cleared 8-μm channel. The left image is for granule #91 (southern Indian Ocean) on September 6, 2002, and the right image for granule #144 (east of England) on the same date.

of nonlinear scan angle effect on the cloud impact in this analysis is to highlight the difference of the cloud-clearing performance of a simple stochastic method versus a more complex physical one. If the nonlinear scan angle effect has been accounted for by employing the full SC algorithm in Figure 5-3, it would only enhance, if any, the performance of the SC algorithm. The specification of the data set used in this analysis is the same as the previous SST analysis in Section 6.3 (see Table 6.7) except that 294-channel AIRS-team CC TB's are used as the ground-truth instead of a scalar SST value. The AIRS-team cloud-cleared product used in this analysis is version 3.5.0. Each plot in Figure 6-9 shows three correlation plots:

1. Correlation between the AIRS visible channel 3 (covering 0.71-0.92 μm) and stochastic cleared (SC).

2. Correlation between the AIRS visible channel 3 and AIRS-team physical cleared.

3. Correlation between the AIRS visible channel 3 and AIRS observed TB (unprocessed).

Left panel is for granule #91 (southern Indian Ocean) on September 6, 2002, and the right panel is for granule #144 (east of England) on the same day. Since visible channel is sensitive to clouds, the high correlation of a certain image with visible-channel image is regarded as having clouds in it. The x-axis is a percentage of pixels used to calculate the correlations; pixels are rank-ordered using cloud-cleared minus observed TB for an 8-μm window channel. The left image shows much less correlation for AIRS stochastic CC than for physical CC. Obviously the observed TB has the highest correlation with the visible channel. For the right graph, the correlation difference is not as striking as in the left one, but still evident, especially for the clearest 40% or less pixels.

6.4.2 RMS relative to baseline for stochastic cloud-clearing

Images in Figure 6-10 show the cloud-clearing results in a different fashion: The leftmost images are AIRS visible channel 3 (0.71-0.92 μm) to visualize the cloud patterns. (All the data is daytime.) The second images are AIRS observed brightness temperature images at 13.9 μm (peaking at 2.9 km), 13.1 μm (peaking at 1.7 km), and 8.2 μm (peaking at 0.2 km), respectively from left to right. The third images are stochastic cloud-cleared brightness temperature images for corresponding channels. The fourth images are masks which pass AIRS QA-test. (Blue means cloudy.) The fifth images are 2-dimensional 3rd-order polynomial fits, so-called baselines, to the QA-pass SC pixels. The rightmost images are stochastic cloud-cleared brightness temperatures minus the polynomial fit. A few points are worth mentioning. First, all the SC images show feasible compensation for the cloud perturbation. Especially for the 13.9 μm-channel at which weighting peak is 2.9 km, most of the partly cloudy regions (and even some heavy clouds) seem to be reasonably compensated. For the 8.2 μm window channel, only light clouds are corrected. Second, the spatially-smooth baseline images model the true temperature field; hence, the residual image between a SC image and its baseline is important: (1) to visualize the characteristics of the SC algorithm, and (2) to assess of the RMS deviation as an indicator of SC performance. It is noted that all three residual images for the 13.9 μm-channel look mostly white

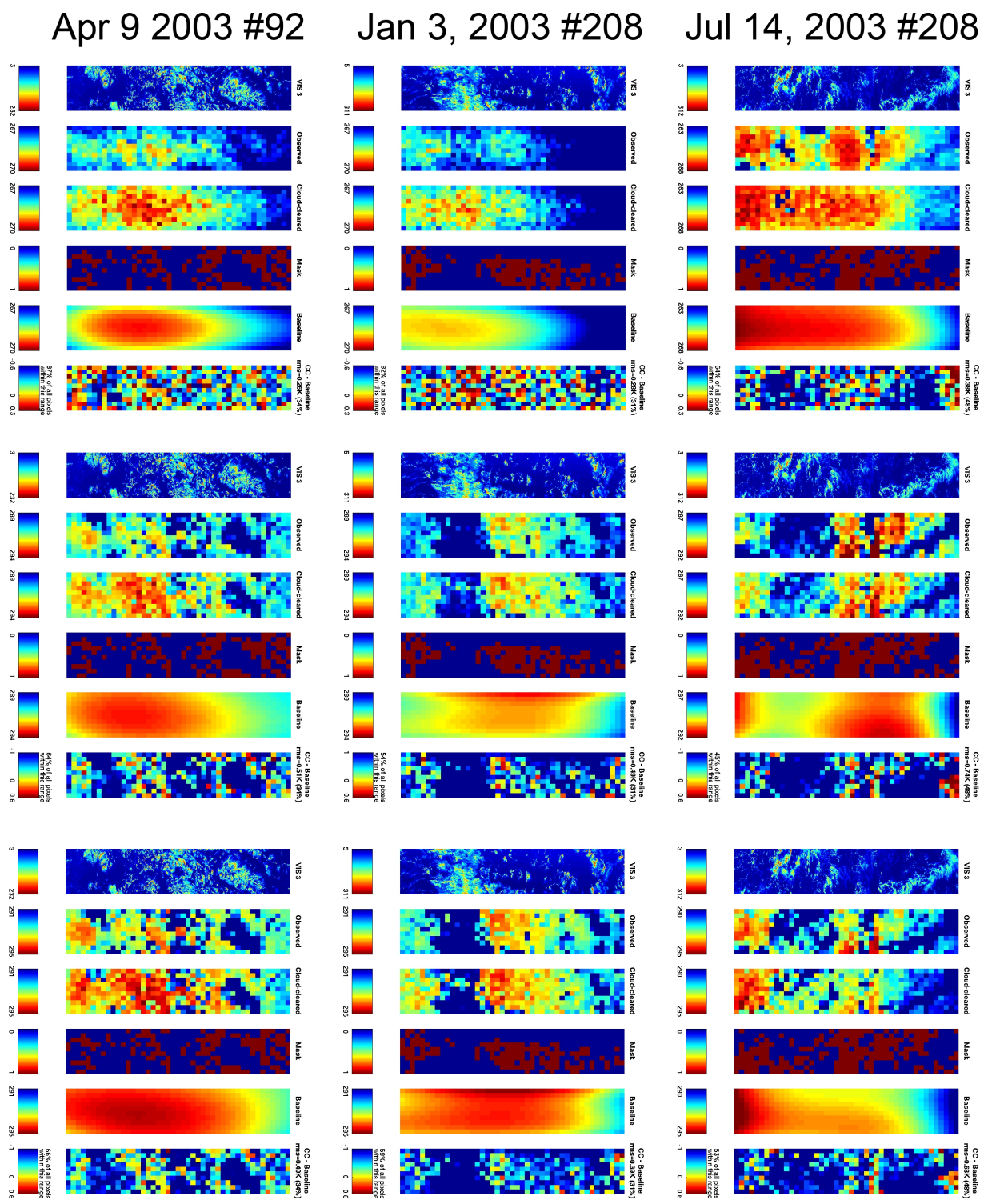


Figure 6-10: Stochastic cloud-cleared images: April 9, 2003 granule #92 is located at 20° South, 63° East; both January 3, 2003 granule #208 and July 14, 2003 granule #208 are located at 10° South, 105° West.

Data used	13.9 μm (WF peak $\sim$2.9 km)	13.1 μm (WF peak $\sim$1.7 km)	8.2 μm (WF peak $\sim$0.2 km)
April 9, 2003 #92	0.38 (48%)	0.74 (48%)	0.63 (48%)
January 3, 2003 #208	0.28 (31%)	0.49 (31%)	0.39 (31%)
July 14, 2003 #208	0.26 (34%)	0.51 (34%)	0.49 (34%)

Table 6.8: Stochastically cleared brightness temperatures (° K RMS) with respect the baseline and the percentage pixels used for RMS analysis

Gaussian whereas the residuals for the channels peaking lower in the atmosphere exhibit underestimated regions as deep blue. Also, the cloud-clearing performance at extreme scan angles appears as good as at nadir. Table 6.8 summarizes the RMS with respect the baseline and the percentage pixels used for RMS analysis. RMS values with respect to baseline are less than 0.74 K for all cases. RMS reflects the cloud residual left in the cloud-cleared image. It can be seen that 13.9-μm channel peaking higher in the atmosphere exhibits less cloud residual than the other channels. Interestingly, the RMS deviation for 8.2-μm channel peaking lower in the atmosphere is slightly less than 13.1 μm-channel.

6.4.3 Comparison of stochastic cloud-clearing and physical cloud-clearing using RMS relative to baseline

The correlation comparison in Section 6.4.1 is based on only two granules of data. In this section, the SC algorithm is applied to 14 granules for the comparison to the alternative physical algorithm for more complete quantitative evaluation. Table 6.9 summarizes the data specification used for this performance comparison.

For this analysis, we used 14 granules or 14129 QA-passed golfballs that pass both SC algorithm's best-78% test and AIRS team QA-flag. The 2-dimensional 3rd-order polynomial fits (baselines) to the least-corrected quarter of the all QA-pass cloud-cleared golfballs were obtained separately for stochastic-cleared image and physical-cleared image. The difference between the local baseline and cloud-cleared image reveal fine-scale structures of the cloud residuals. The RMS relative to local baseline

Type	Description
Date	August 28 2005
Granules used	#2, #3, #18, #19, #33, #66, #84, #99, #132, #148, #165, #214, #216, #217 (14129 QA-passed golfballs in total)
Level 2 version	4.0.9
Scan angle	All scan angle used
Channels analyzed	314 channels including 4- and 15-μm sounding channels, 8-μm window channels, and water vapor channels (See Appendix A for the full channel listing.)
Land definition	Land is defined by land fraction $>$ 0.8 and elevation $<$ 0.5 km

Table 6.9: Data specification used for the evaluation of selected channels for RMS deviation relative to local baseline

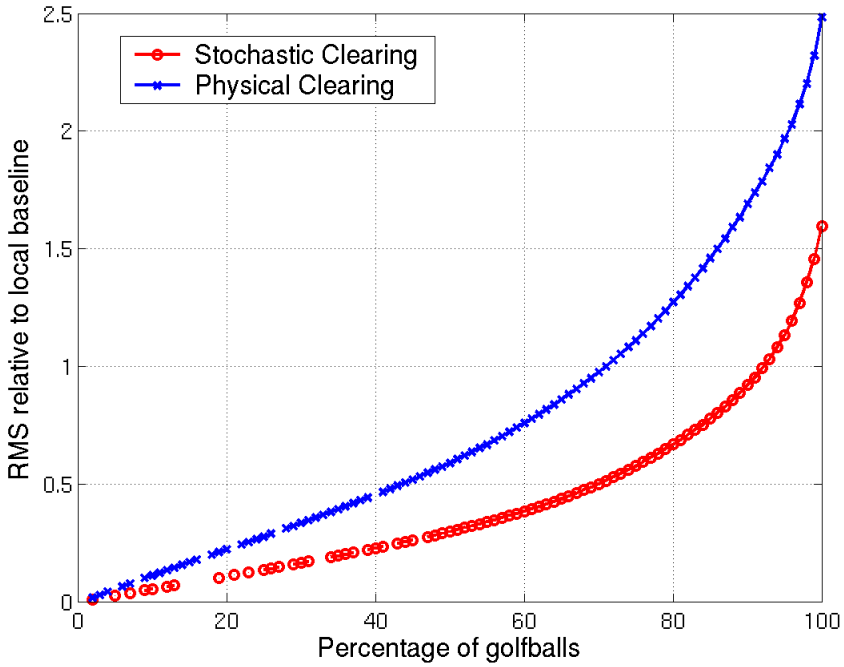


Figure 6-11: RMS relative to local baseline (K) for stochastic cloud-clearing and physical cloud-clearing

is shown in Figure 6-11, for SC and physical results. For both stochastic and physical algorithm, the golfballs are separately rank-ordered within each granule using its own cloud-cleared minus observed TB for an 4-μm window channel (2223 cm^{-1}). These RMS values are then combined to yield a total RMS results versus the percentage of golfballs included. SC results reveal much less RMS (about a half to 65%) than physical results. For the same RMS, SC algorithm can cloud-clear much more percentage of golf balls (about twice for up to 0.3 K RMS) than physical algorithm.

6.5 Summary

The SC algorithm is shown to have an excellent agreement with the numerical weather prediction model, either ECMWF/SARTA cloud-cleared TB or NCEP SST. The significance of these results is four-fold. First, the ECMWF and AIRS global TB fields are in remarkable agreement, and therefore presumably quite accurate. Second, they suggest that AIRS and its successor polar and geosynchronous hyperspectral sounders will provide TB data of sufficient accuracy and global coverage to materially improve the quality of operational and scientific numerical weather predictions. Third, since the cloud-clearing performance reported here is nearly independent of viewing angle and therefore of the diameter of the FOV, high spatial resolution may not be essential for good cloud-clearing performance; the area of the 14-km nadirial FOV of AIRS increases by more than a factor of three at the highest scan angles. Finally, these cloud-clearing methods developed for selected channels can probably be adapted to correcting the remainder, which have many potential applications such as monitoring CO_2 variability, trace gases, cloud properties, and humidity profiles.

The unique contribution of this research to cloud-clearing methods involves development of data-trained stochastic models for the effects of clouds on infrared TB's. This contrasts with conventional cloud-clearing approaches employing physical models for clouds and radiative transfer. Although physical models work well with instruments having modest numbers of channels, such models currently have insufficient accuracy to extract all the information available from hundreds or thousands of in-

frared channels. Stochastic methods apparently access this information better by virtue of the very large training data sets and computation resources now available, provided that suitable nonlinear stochastic models are utilized.

Chapter 7

Stability Analysis

7.1 Motivation

The Atmospheric Infrared Sounder (AIRS) is a novel instrument which has been operational since May 4, 2002. The quality of the AIRS observed radiances and their derivative products is being actively validated and improved by the AIRS science team [34]. The Advanced Microwave Sounding Unit (AMSU), which has been on several different satellites since 1998, on the other hand, is known to be very stable within tenths of a degree K. The unique configuration that the AIRS operates in synchronism with AMSU [2] (see Figure 2-2) allows us to analyze AIRS stability relative to AMSU, and validate the quality of the cloud-cleared AIRS FOVs in the following fashion: The AIRS cloud-cleared brightness temperature spectra are put into a linear regression estimator that predicts AMSU brightness temperatures. The difference of the AMSU estimate and the true AMSU, when averaged over many FOVs, will be able to reveal the AIRS' instrumental drift relative to AMSU, and its dependence on diurnal cycle, latitude, scan angle, etc. The significance of this stability analysis is two-fold. First, this can serve as another validation scheme for the AIRS cloud-clearing algorithm because microwave frequencies are capable of almost total cloud penetration. Second, the study of drift of AIRS instrument is important in its own right in order to further improve the AIRS stability.

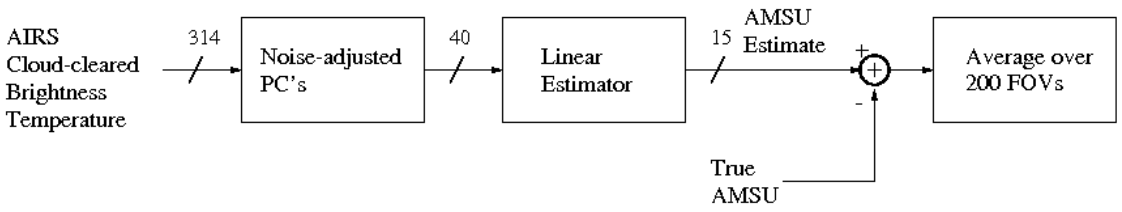


Figure 7-1: Block diagram of stability analysis algorithm

7.2 Method

Figure 7-1 illustrates the algorithm for estimating AMSU brightness temperatures. Noise-adjusted principal component scores of AIRS cloud-cleared brightness temperature for the selected 314 channels are calculated. This set of selected 314 channels is listed in Appendix A. The 40 most significant NAPC scores are fed to a linear regression estimator that predicts AMSU brightness temperatures for all 15 channels. By employing the NAPC transform, the number of inputs to the linear regression estimator is reduced from 314 to 40, which has two important implications: First, the degrees of freedom in the linear regression estimator decrease, thus allowing more stable training of the linear estimator. Second, the truncation of the least significant NAPCs reduces the noise as seen in Section 3.3.1.

The cloud-cleared AIRS data is from the output of stochastic cloud-clearing algorithm discussed in the previous chapter. For the stability analysis, AMSU estimation deviation, i.e. the true AMSU minus the predicted AMSU, is averaged over 200 nearby FOVs. Table 7.1 specifies the data used for this evaluation. For this analysis, the same 10 stratifications as in Section 6.2 are used, depending on land/sea, lower or upper latitude, and day/night/day+night. The number of golfballs used for the training is different for each category, but is more than 1000 for all cases. For all categories, 13,960 golfballs are used in total for the training, and 13,088 are used for the testing.

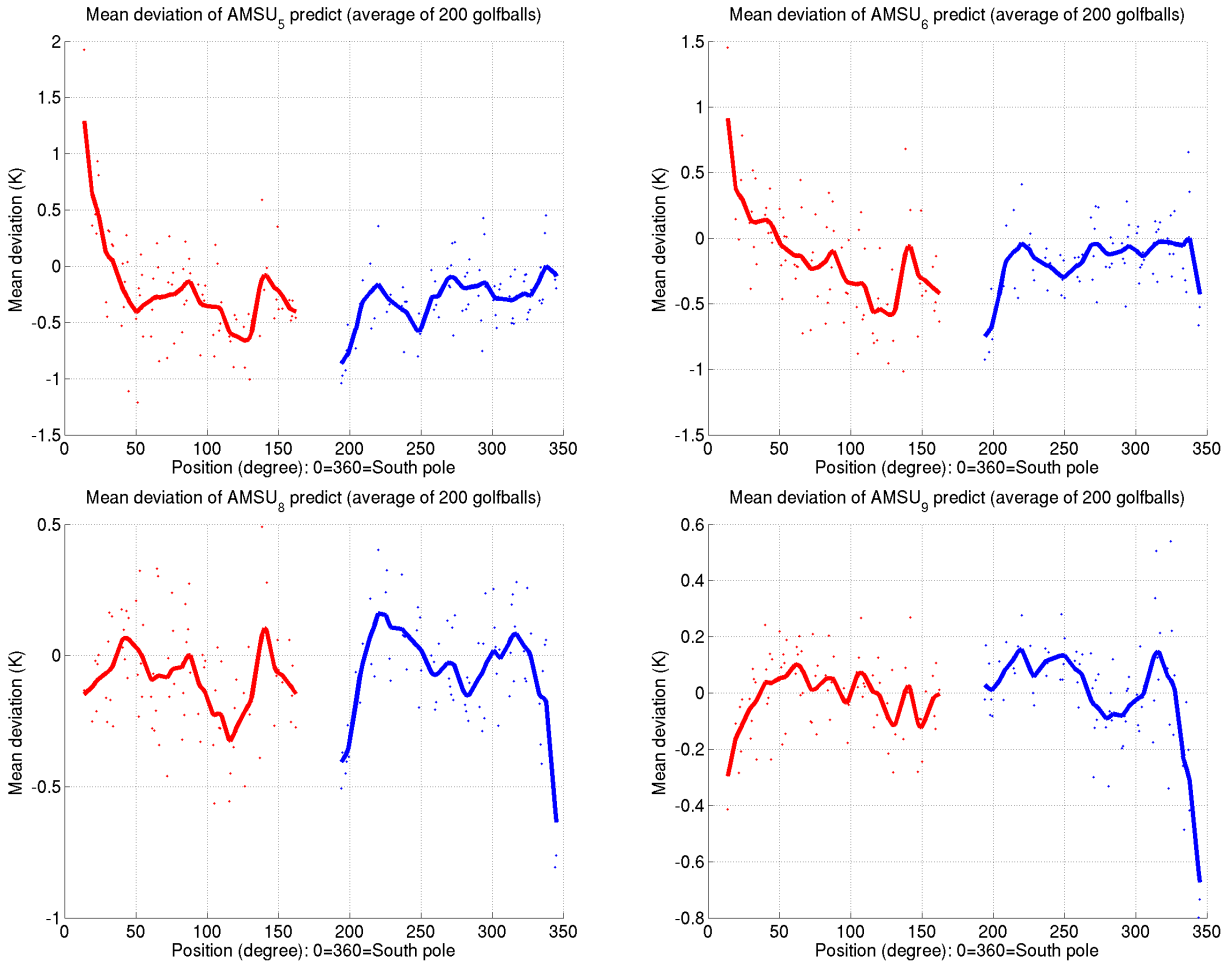


Figure 7-2: Mean deviation (the true AMSU minus the predicted AMSU) for AMSU channels 5 (top left), 6 (top right), 8 (bottom left), and 9 (bottom right)

Type	Description
Coverage	Global grid
Date	August 21 2003 (used for training), and August 23 2003 (used for testing)
Scan angle	All scan angle used
Latitude	Up to 70° North and South
Orbit	Both ascending and descending
Channels analyzed	314 channels including 4- and 15-μm sounding channels, 8-μm window channels, and water vapor channels. See Appendix A for the full channel listing.
Land definition	Land is defined by land fraction $>$ 0.8 and elevation $<$ 0.5 km

Table 7.1: Data specification used for the evaluation of selected channels relative to AMSU

7.3 Results

Figure 7-2 shows the mean deviations (the true AMSU minus the predicted AMSU) of predicted AMSU with respect to true AMSU, for channels 5, 6, 8, and 9. An x-axis is the location of the satellite with respect to Earth's south pole, i.e. 0 to 180 meaning South pole to North pole in an ascending orbit, and 180 to 360 meaning North pole to South pole in a descending orbit. AIRS data on August 21, 2003 is used for training, and the data on August 23, 2003 is used for testing. It is important to note that the training is done by 78% clearest[1] pixels, and the evaluation is based on 28% clearest[1] pixels. The initial study shows that when trained with 28% clearest set and tested with the separate 28% clearest set, the results entail many outliers; these extremes are very large positive deviations at the edge of heavy clouds, meaning that the cloud effects were underestimated in the stochastic cloud-clearing algorithm, and that such small error due to the underestimation of cloud perturbation can be amplified in the AMSU estimator.

Each point in the plots in Figure 7-2 represents average mean deviation (true AMSU brightness temperature minus its predict) over 200 FOVs[2] in one granule,

[1] X% clearest pixels are determined by the same acceptance thresholds used in the stochastic cloud-clearing algorithm in Section 6.2

[2] Only 200 FOVs are used out of 1350 overall FOVs because 28% of FOVs are available and some

i.e. 30 scan-lines times 45 FOVs in one scan-line. Overlaid in the figures are the low-pass filtered version of the mean deviations, using a Gaussian function with the standard deviation of 4° of latitude. For relatively transparent AMSU channels 5 and 6, deviations are in ~ -0.3 and ~ -0.15 K range, respectively, with a common distinct latitudinal pattern. Especially the latitudinal pattern in an ascending orbit for channel 5 and 6 shows a striking similarity, thus may be useful to calibrate AIRS measurements or modify slightly the AIRS cloud-clearing corrections. These channels also show a small negative bias, meaning AIRS cloud effects are overestimated. More opaque AMSU channels 8 and 9 show much less deviation and bias. When the mean deviations are averaged over all latitude within $\pm 60°$ (or points over 30–150 and 210–330 in the graph) for both ascending and descending orbits, the resulting average deviations are -0.2627 K, -0.1833 K, -0.0792 K, 0.0065 K for AMSU channel 5, 6, 8 and 9, respectively. Hence, the AIRS instrument is seen to be fairly stable relative to AMSU instrument although there exists a very small negative deviation. In this latitude range, the differences between the mean deviations and its low-pass filtered curve exhibit a Gaussian density[3], for all AMSU channels shown. Extreme south region consistently shows worse agreement than the extreme north region partly because it is deep winter in the southern hemisphere when this data is collected, and sea ice was not adequately modeled in the cloud-clearing algorithm. The dependence of the AIRS/AMSU drift upon latitude was sought in this analysis; yet, a thorough investigation of its dependence upon other parameters can be done in a similar fashion, and is left to a future work.

of them may still be missing if elevation is higher than 500 m.

[3]The goodness-of-fit to the Gaussian density is done with Bera-Jarque parametric hypothesis test. The Bera-Jarque test uses the skewness and kurtosis of a given data samples to determines if the null hypothesis that the data is normal is rejected. Significance level of 5% was used in this analysis.

Chapter 8

Conclusions

8.1 Summary

A generic methodology is presented for detecting and compensating anomalies from hyperspectral imagery. Noise-Adjusted Principal component (NAPC) analysis and blind signal separation technique such as the Iterative Order and Noise (ION) estimation algorithm are particularly useful in characterization of signal and estimation/removal of artifacts. Various computationally efficient methodologies are also addressed to cope with nonlinearities. An anomaly compensation technique has been applied to specific problems that exhibit different stochastic models for an anomaly. The performance of each anomaly compensation has been evaluated and validated. Hyperspectral anomalies in AIRS observation covered in this thesis are

1. Cloud impact in hyperspectral brightness temperature (TB) fields.
2. Noisy channels.
3. Scan-line miscalibration.

Estimation of the cloud impact using the proposed statistical algorithm is proven successful and comparable or superior to an alternative physical approach, using four different validation schemes:

1. The RMS cloud-clearing errors with respect to ECMWF data are calculated

2. Global cloud-cleared images permit visual inspection of the clearing performance in various geographical regions.

3. Sea surface temperatures provided by NCEP are used to validate the stochastic clearing results.

4. The stochastic algorithm is compared with an alternative physical algorithm, using the correlations with a visible cloud-sensitive channel.

Noisy channels and miscalibrated scan-lines are also well compensated or removed using the proposed algorithms.

8.2 Contributions

The main contribution of this study are two-fold.

First, a generic approach to characterizing hyperspectral data and to detecting and compensating anomalies are presented. This approach is based on novel signal-processing-based techniques utilizing principal component analysis and blind signal separation. A number of approaches are discussed to cope with nonlinearities in most of the remote-sensing applications without incurring a heavy computational burden.

Second, the stochastic cloud-clearing algorithm was designed based on the anomaly compensation and nonlinear estimation techniques. AIRS stochastic cleared (SC) TB's are sufficiently consistent with ECMWF analysis fields that a large fraction of AIRS golfballs could probably be profitably assimilated into operational global models in the near future. The performance of the SC algorithm is shown to be comparable to that of the physical algorithm in the SST-based validation scheme. What differentiates the SC algorithm from the physics-based alternative is that the SC algorithm exploits the cloud information embedded in the given data itself without relying on the underlying physics of cloud impact, which may be challenging to model accurately. Once trained with ensembles of observations and estimates, the SC algorithm runs very fast because it only involves matrix additions and multiplications, whereas the

physical method generally requires iteration between estimation of physical parameters (retrieval) and calculation of observed radiance (radiative transfer) until a certain convergence criterion is met. The operational SC program is implemented with 664 lines of Matlab script, and can cloud-clear an entire day of AIRS data for 314 channels within 20 minutes on a moderately equipped PC. Stochastic cloud-clearing algorithms also appear to function well a significant fraction of the time even if no FOV is fully clear, reducing incentives for employing alternate strategies that rely only on totally clear FOVs. It is found that the cloud-clearing performance at extreme scan angles is nearly as good as the nadirial results; therefore hole-hunting using high spatial resolution may not be essential for good cloud-clearing performance.

8.3 Suggestions for future work

This section addresses possible improvements and refinements to the anomaly compensation theories and the current stochastic cloud-clearing algorithm.

8.3.1 Improvements on the anomaly compensation

- *Optimization of combining spectral and spatial processing.* Spectral and spatial processing may be merged or iterated to that the signal-to-noise ratio (SNR) of an anomaly of interest is enhanced. This thesis lacks a thorough study of optimally combining the two processes in the spatial and spectrum domains and a study of absolute theoretical limits.

- *More extensive spectral processing techniques.* More spectral processing techniques may be used instead of or in combination with the techniques discussed in this thesis. For example, independent component analysis (ICA) may replace PCA.

8.3.2 Stochastic cloud-clearing algorithm

- *Joint cloud-clearing and retrieval.* Cloud-clearing usually precedes retrieval of physical parameters, for example, vertical temperature and water vapor profiles. It would be worthwhile to study the performance and the implication of joint clearing/retrieval relative to the separate approach.

- *Analysis and mapping of trace gases.* In Figure 6-1 it was seen that a group of channels appears to be affected by one or more variable trace gases. Using a blind signal separation techniques and assuming the orthogonality of each trace gas space, it may be feasible to successfully separate the contribution of the each trace gas from the given observation. Thus, interestingly these channels may allow us to monitor the trace gases, and map their concentrations globally.

- *Estimation of* CO_2 *variability.* CO_2, although present in only minute amounts (0.036%), is nevertheless a meteorologically important constituent of air because it is an efficient absorber of energy emitted by Earth and thus influences the heating of the atmosphere [30]. Although the proportion of carbon dioxide in the atmosphere is relatively uniform, there is a steady increase in its concentration over the last century due to human activities. AIRS spectrum covers 3.7–15.4-μm, and different frequencies interact with carbon dioxide differently. Hence, the carbon dioxide variability may be estimated and monitored over a long period of time.

- *Physical explanation of the behavior of 10 different cases in the ECMWF/SARTA validation.* This study does not include the complete analysis and physical reasoning of why 10 different cases (based on latitude, land/sea, and day/night/day+night) behave the way they do. For example, for ocean, there exists little difference on the SC performance for daytime and nighttime whereas, for land, the difference is large.

- *Use of stripe-removing algorithm.* For the present cloud-clearing algorithm in Chapters 5 and 6, we did not apply the stripe-removing algorithm in Sec-

tion 3.3.3. Streak patterns due to miscalibrated scan-lines are less evident with more recent version of AIRS data. It may be interesting to compare the cloud-clearing performance of using the original raw TB versus the stripe-removed data.

- *Thorough AIRS stability analysis.* A thorough investigation of the dependence of the AIRS/AMSU drift upon various parameters can be done. This may permit improvements on the AIRS stability, or useful insights on the performance of the AIRS cloud-clearing algorithm.

- *Embracing the outliers.* The current SC algorithm does not adequately treat outliers, thus it was both trained and tested with the latitude lower than $\pm 70^{\circ}$ and surface elevation lower than 500 m. One may be able to better analyze and characterize outliers, and come up with a method to accommodate the outliers.

- *Better model for nonlinearities.* The current SC algorithm accommodates nonlinearities in simple ways based on stratification and multiplicative scan angle correction. Refinements of the current methods to deal with nonlinearity should yield accuracy improvements and permit extension to more AIRS channels.

- *Optimum architecture for the SC algorithm.* For example, the current warmest/coldest FOV selection and averaging protocols are ad hoc. Exploring all possible SC algorithm architectures in order to find an optimum for a specific retrieval problem is combinatorially challenging, so one future task might be to develop efficient design-of-experiment approaches for this search process.

- *Comparison with different stochastic approach.* Blackwell demonstrated a successful stochastic cloud-clearing algorithm based on Projected Principal Component (PPC) analysis and neural networks using simulated AIRS data [5]. Whereas the SC algorithm presented in the thesis works in a stratified fashion, using a different estimator for FOVs having different physics (land versus ocean, low latitude versus high latitude, and day versus night), Blackwell's

method replies on a single neural-network estimator for all cases. It would be worthwhile to compare these two algorithms on the same data.

- *More study on the near lack of angular dependence on SC performance.* The performance of SC algorithm does not degrade noticeably for the extreme viewing angle, or larger FOV. One relatively easy confirmation of this result would be to process "super golfballs" consisting of a 3-by-3 average of golfballs near nadir and check the performance for this case.

- *Extending to future sounding systems.* The current cloud-clearing algorithm can be applied to similar future sounding systems including the NPOESS Cross-track Infrared Microwave Sounding System (CrIMSS) and the GOES-R Hyperspectral Environmental Suite (HES).

Appendix A

314 Selected AIRS Channels

This appendix lists 314 selected AIRS channels used in the evaluation of stochastic cloud-clearing (Sections 6.1, 6.2) and the AIRS/AMSU stability analysis (Chapter 7).

Channel index	AIRS channel number	SRF centroid frequency (cm^{-1})	Wavelength (μm)	Channel index	AIRS channel number	SRF centroid frequency (cm^{-1})	Wavelength (μm)
1	101	674.414	14.828	2	102	674.673	14.822
3	103	674.931	14.816	4	104	675.189	14.811
5	105	675.448	14.805	6	106	675.707	14.799
7	107	675.966	14.794	8	108	676.226	14.788
9	109	676.485	14.782	10	110	676.745	14.777
11	111	677.005	14.771	12	112	677.265	14.765
13	113	677.526	14.760	14	114	677.786	14.754
15	115	678.047	14.748	16	116	678.308	14.743
17	117	678.570	14.737	18	118	678.831	14.731
19	119	679.093	14.726	20	120	679.355	14.720
21	123	680.142	14.703	22	124	680.404	14.697
23	125	680.667	14.691	24	126	680.930	14.686
25	127	681.194	14.680	26	128	681.457	14.674
27	129	681.721	14.669	28	130	681.985	14.663
29	131	687.601	14.543	30	135	688.680	14.521
31	136	688.950	14.515	32	137	689.220	14.509
33	138	689.491	14.503	34	139	689.762	14.498
35	140	690.033	14.492	36	141	690.304	14.486
37	142	690.576	14.481	38	143	690.847	14.475
39	144	691.119	14.469	40	145	691.391	14.464
41	146	691.664	14.458	42	147	691.936	14.452
43	148	692.209	14.447	44	149	692.482	14.441
45	150	692.755	14.435	46	151	693.029	14.429
47	152	693.303	14.424	48	153	693.576	14.418

49	154	693.851	14.412	50	155	694.125	14.407
51	156	694.400	14.401	52	157	694.674	14.395
53	158	694.949	14.390	54	159	695.225	14.384
55	160	695.500	14.378	56	161	695.776	14.372
57	162	696.052	14.367	58	163	696.328	14.361
59	164	696.604	14.355	60	165	696.881	14.350
61	166	697.158	14.344	62	167	697.435	14.338
63	168	697.712	14.333	64	169	697.990	14.327
65	170	698.267	14.321	66	171	698.545	14.315
67	172	698.824	14.310	68	173	699.102	14.304
69	174	699.381	14.298	70	175	699.660	14.293
71	176	699.939	14.287	72	177	700.218	14.281
73	178	700.498	14.276	74	179	700.777	14.270
75	180	701.057	14.264	76	181	701.338	14.258
77	182	701.618	14.253	78	183	701.899	14.247
79	184	702.180	14.241	80	185	702.461	14.236
81	186	702.742	14.230	82	187	703.024	14.224
83	188	703.306	14.219	84	189	703.588	14.213
85	190	703.870	14.207	86	191	704.153	14.201
87	192	704.436	14.196	88	193	704.719	14.190
89	194	705.002	14.184	90	195	705.285	14.179
91	196	705.569	14.173	92	197	705.853	14.167
93	198	706.137	14.162	94	199	706.422	14.156
95	200	706.706	14.150	96	201	706.991	14.144
97	202	707.276	14.139	98	203	707.562	14.133
99	204	707.847	14.127	100	205	708.133	14.122
101	206	708.419	14.116	102	207	708.706	14.110
103	208	708.992	14.105	104	209	709.279	14.099
105	210	709.566	14.093	106	211	709.853	14.087
107	212	710.141	14.082	108	213	710.429	14.076
109	214	710.716	14.070	110	215	711.005	14.065
111	216	711.293	14.059	112	217	711.582	14.053
113	218	711.871	14.047	114	219	712.160	14.042
115	220	712.449	14.036	116	221	712.739	14.030
117	222	713.029	14.025	118	223	713.319	14.019
119	224	713.609	14.013	120	225	713.900	14.008
121	226	714.191	14.002	122	227	714.482	13.996
123	228	714.773	13.990	124	229	715.065	13.985
125	230	715.357	13.979	126	231	715.649	13.973
127	232	715.941	13.968	128	233	716.233	13.962
129	234	716.526	13.956	130	235	716.819	13.951
131	236	717.112	13.945	132	237	717.406	13.939
133	239	717.994	13.928	134	240	718.288	13.922
135	241	718.582	13.916	136	242	718.877	13.911
137	243	719.172	13.905	138	244	719.467	13.899

139	245	719.763	13.893	140	246	720.059	13.888
141	247	720.354	13.882	142	248	720.651	13.876
143	249	720.947	13.871	144	250	721.244	13.865
145	251	721.541	13.859	146	252	721.838	13.854
147	253	722.135	13.848	148	254	722.433	13.842
149	255	722.731	13.836	150	258	723.626	13.819
151	259	723.925	13.814	152	260	724.224	13.808
153	262	724.824	13.796	154	263	725.123	13.791
155	264	725.424	13.785	156	265	725.724	13.779
157	268	726.627	13.762	158	269	726.928	13.757
159	270	727.230	13.751	160	273	728.137	13.734
161	274	728.439	13.728	162	275	728.055	13.735
163	277	728.660	13.724	164	281	729.873	13.701
165	282	730.177	13.695	166	283	730.481	13.690
167	286	731.395	13.673	168	287	731.700	13.667
169	293	733.536	13.633	170	297	734.765	13.610
171	298	735.073	13.604	172	300	735.690	13.593
173	302	736.308	13.581	174	306	737.546	13.558
175	307	737.856	13.553	176	308	738.167	13.547
177	311	739.100	13.530	178	316	740.660	13.501
179	317	740.973	13.496	180	318	741.286	13.490
181	319	741.599	13.484	182	320	741.913	13.479
183	322	742.541	13.467	184	331	745.380	13.416
185	336	746.967	13.387	186	345	749.839	13.336
187	1387	1294.562	7.725	188	1392	1297.427	7.708
189	1402	1303.194	7.673	190	1407	1306.096	7.656
191	1872	2187.850	4.571	192	1877	2192.412	4.561
193	1882	2196.993	4.552	194	1887	2201.592	4.542
195	1892	2206.211	4.533	196	1897	2210.848	4.523
197	1900	2213.640	4.517	198	1901	2214.572	4.516
199	1902	2215.505	4.514	200	1903	2216.439	4.512
201	1904	2217.373	4.510	202	1905	2218.309	4.508
203	1906	2219.245	4.506	204	1907	2220.181	4.504
205	1908	2221.119	4.502	206	1909	2222.057	4.500
207	1910	2222.996	4.498	208	1911	2223.936	4.497
209	1912	2224.877	4.495	210	1913	2225.818	4.493
211	1914	2226.760	4.491	212	1915	2227.703	4.489
213	1916	2228.647	4.487	214	1917	2229.592	4.485
215	1918	2230.537	4.483	216	1919	2231.483	4.481
217	1920	2232.430	4.479	218	1921	2233.377	4.478
219	1923	2235.275	4.474	220	1924	2236.225	4.472
221	1925	2237.176	4.470	222	1926	2238.128	4.468
223	1927	2239.080	4.466	224	1928	2240.033	4.464
225	1929	2240.987	4.462	226	1930	2241.942	4.460
227	1931	2242.897	4.459	228	1932	2243.854	4.457

229	1933	2244.811	4.455	230	1934	2245.769	4.453
231	1935	2246.728	4.451	232	1936	2247.687	4.449
233	1937	2248.648	4.447	234	1938	2249.609	4.445
235	1939	2250.571	4.443	236	1940	2251.533	4.441
237	1941	2252.497	4.440	238	1942	2253.461	4.438
239	1943	2254.426	4.436	240	1944	2255.392	4.434
241	1945	2256.359	4.432	242	1946	2257.327	4.430
243	1947	2258.295	4.428	244	1948	2259.264	4.426
245	1949	2260.234	4.424	246	1950	2261.205	4.422
247	1951	2262.177	4.421	248	1952	2263.149	4.419
249	1953	2264.122	4.417	250	1954	2265.096	4.415
251	1955	2266.071	4.413	252	1956	2267.047	4.411
253	1957	2268.024	4.409	254	1958	2269.001	4.407
255	1959	2269.979	4.405	256	1960	2270.958	4.403
257	1961	2271.938	4.402	258	1962	2272.918	4.400
259	1963	2273.900	4.398	260	1964	2274.882	4.396
261	1966	2276.849	4.392	262	1967	2277.834	4.390
263	1968	2278.819	4.388	264	1969	2279.805	4.386
265	1970	2280.793	4.384	266	1971	2281.781	4.383
267	1972	2282.770	4.381	268	1973	2283.759	4.379
269	1974	2284.750	4.377	270	1975	2285.741	4.375
271	1981	2291.707	4.364	272	1987	2297.702	4.352
273	1989	2299.708	4.348	274	1991	2301.716	4.345
275	1995	2305.744	4.337	276	1997	2307.763	4.333
277	1999	2309.785	4.329	278	2027	2310.702	4.328
279	2029	2312.530	4.324	280	2031	2314.361	4.321
281	2033	2316.195	4.317	282	2035	2318.031	4.314
283	2037	2319.871	4.311	284	2039	2321.714	4.307
285	2045	2327.260	4.297	286	2059	2340.308	4.273
287	2061	2342.184	4.270	288	2063	2344.063	4.266
289	2065	2345.945	4.263	290	2067	2347.831	4.259
291	2069	2349.720	4.256	292	2071	2351.611	4.252
293	2073	2353.506	4.249	294	2077	2357.305	4.242
295	2081	2361.117	4.235	296	2083	2363.027	4.232
297	2087	2366.858	4.225	298	2089	2368.778	4.222
299	2091	2370.701	4.218	300	2093	2372.628	4.215
301	2095	2374.557	4.211	302	2097	2376.490	4.208
303	2099	2378.426	4.204	304	2101	2380.365	4.201
305	2103	2382.308	4.198	306	2105	2384.253	4.194
307	2107	2386.202	4.191	308	2109	2388.154	4.187
309	2111	2390.110	4.184	310	2113	2392.068	4.180
311	2115	2394.030	4.177	312	2117	2395.995	4.174
313	2119	2397.964	4.170	314	2121	2399.936	4.167

Table A.1: 314 selected AIRS channels

Appendix B

Selected Cloud-Clearing Source Code

This appendix lists selected Matlab scripts and functions that were used in the thesis. Additional source code not listed here may be obtained from the author: cycho@alum.mit.edu.

B.1 Cloud-clearing algorithm

B.1.1 linear-CC-test.m

```
function [AIRS_CC,varargout] = linear_CC_test(input_to_CC, channelnumber, CC_pass, CC_parameter, estimator);
%LINEAR_CC_TEST
%   This is a core algorithm for stochastic cloud-clearing, using linear regression
%
%   Usage:
%     [AIRS_CC,varargout] = linear_CC_test(input_to_CC, channelnumber, CC_pass, CC_parameter, estimator)
%
%   Input:
%     input_to_CC: structure-format inputs to the CC algorithm ('AIRS', 'AMSU', 'scanang', 'landfrac', 'lat',
%                  'lon', optional 'more_input', optional 'correction_pc')
%     channelnumber: AIRS channel index being used (1-2378)
%     CC_pass: either 'initial', 'multiplicative_scanangle'
%     CC_parameter: optional 'AMSU_input_channel', 'num_output_pc_AIRS_correction', 'verbose_flag'
%                    - AMSU_input_channel: AMSU channels used for both training and testing
%                                          default = [5 6 8 9 10]
```

```
%                   - num_output_pc_AIRS_correction: number of output NAPC to keep
%                                                    default = 4
%                   - verbose_flag: when turned on, intermediate results are shown
%                                   default = off
%     estimator: structure-format estimator coefficients
%
%   Output:
%     AIRS_CC: AIRS cloud-cleared brightness temperature
%     varargout: corrections
%

%   (c) Copyright 2005 Remote Sensing and Estimation Group M.I.T.
%   Written by Chuck Cho <cycho@mit.edu>

more off

AIRS    = input_to_CC.AIRS;
AMSU    = input_to_CC.AMSU;
scanang = input_to_CC.scanang(:)';
landfrac= input_to_CC.landfrac(:)';
lat     = input_to_CC.lat(:)';
lon     = input_to_CC.lon(:)';

% input_to_CC.more_input specified?
if isfield(input_to_CC,'more_input')
  more_input = input_to_CC.more_input;
  if size(more_input,1) == size(AMSU,2)
    more_input = more_input'
  elseif size(more_input,2) ~= size(AMSU,2)
    error('"more input" size does not match')
  end
  disp('extra input detected.');
else
  more_input = [];
end

% input_to_CC.correction_pc specified?
if isfield(input_to_CC,'correction_pc')
  correction_pc = input_to_CC.correction_pc(:)';
elseif strcmp(CC_pass,'multiplicative_scanangle')
  error('If CC_pass is "multiplicative_scanangle", input_to_CC.correction_pc must be specified.')
end

% CC_parameter.AMSU_input_channel specified? (default=5,6,8,9,10)
if isfield(CC_parameter,'AMSU_input_channel')
```

```
  AMSU_input_channel = CC_parameter.AMSU_input_channel;
else
  AMSU_input_channel = [5 6 8 9 10];
end

% CC_parameter.num_output_pc_AIRS_correction specified? (Number of output PC correction: default 4)
if isfield(CC_parameter,'num_output_pc_AIRS_correction')
  num_output_pc_AIRS_correction = CC_parameter.num_output_pc_AIRS_correction;
else
  num_output_pc_AIRS_correction = 4;
end

% CC_parameter.verbose_flag turned on? (default off)
% If turned on, display intermediate results for debugging
if isfield(CC_parameter,'verbose_flag')
  verbose_flag = CC_parameter.verbose_flag
else
  verbose_flag = 0;
end

clear input_to_CC;

% load frequency
L2_chan_prop_v6_6_8_freq;
freq = frequency(channelnumber);

% load WF height
load weighting_peak.mat weighting_peak_in_height
height = weighting_peak_in_height(channelnumber);

numgolfball = size(AIRS,2);

height_for_averaging = find(height >= 1 & height <= 3 & freq > 2000 & freq < 2400);
if length(height_for_averaging) == 0
  error('No 4um channels peaking 1-3km');
end

if verbose_flag
  disp(['# of 1-3km 4um channels = ' num2str(length(height_for_averaging))])
end

% load AMSU noise variances
load amsu_noise

numchannel = length(channelnumber);
```

```
AIRS_warmest      = zeros(size(AIRS,1),size(AIRS,2));
AIRS_warmest_4avg = zeros(size(AIRS,1),size(AIRS,2));
AIRS_warmest_9avg = zeros(size(AIRS,1),size(AIRS,2));
AIRS_coldest      = zeros(size(AIRS,1),size(AIRS,2));

tmp1 = squeeze(mean(AIRS(height_for_averaging,:,:),1));
if size(tmp1,1) == 9
  tmp1 = tmp1';
end

[tmp2 tmp3] = sort(tmp1,2);

% average warmest 15-km pixels
for i=1:numgolfball
  AIRS_warmest(:,i)      = AIRS(:,i,tmp3(i,9));
  AIRS_warmest_4avg(:,i) = mean([AIRS(:,i,tmp3(i,9)) AIRS(:,i,tmp3(i,8)) ...
                                 AIRS(:,i,tmp3(i,7)) AIRS(:,i,tmp3(i,6))],2);
  AIRS_warmest_9avg(:,i) = mean(AIRS(:,i,:),3);
  AIRS_coldest(:,i)      = AIRS(:,i,tmp3(i,1));
end
clear tmp*

AIRS_all_warmests = { AIRS_warmest, AIRS_warmest_4avg, AIRS_warmest_9avg };

for i=1:length(AIRS_all_warmests)

  mean_AIRS = estimator{i}.mean_AIRS;
  mean_AIRS_dif = estimator{i}.mean_AIRS_dif;
  mean_AMSU = estimator{i}.mean_AMSU;
  mean_correction = estimator{i}.mean_correction;
  AIRS_pct = estimator{i}.AIRS_pct;
  AIRS_dif_pct = estimator{i}.AIRS_dif_pct;
  correction_pct = estimator{i}.correction_pct;
  P = estimator{i}.P;

  % normalize
  zm_AIRS = AIRS_all_warmests{i} - mean_AIRS*ones(1,numgolfball);
  load airs_noise_1_10_03; noise = airs_noise(channelnumber);
  norm_AIRS = zm_AIRS ./ (sqrt(noise) * ones(1,numgolfball));

  AIRS_dif = AIRS_all_warmests{i} - AIRS_coldest;
  zm_AIRS_dif = AIRS_dif - mean_AIRS_dif*ones(1,numgolfball);
  norm_AIRS_dif = zm_AIRS_dif ./ (sqrt(noise) * ones(1,numgolfball));
```

```
  zm_AMSU = AMSU - mean_AMSU*ones(1,numgolfball);
  norm_AMSU = zm_AMSU ./ (sqrt(amsu_noise') * ones(1,numgolfball));

  number_input_pc_AIRS_warmest = 7;
  number_input_pc_AIRS_dif = 3;

  input1 = AIRS_pct(1:number_input_pc_AIRS_warmest,:) * norm_AIRS;
  input2 = AIRS_dif_pct(1:number_input_pc_AIRS_dif,:) * norm_AIRS_dif;
  input3 = norm_AMSU(AMSU_input_channel,:);

  if strcmp(CC_pass,'initial')
    input = [input1; input2; input3; sec(scanang*pi/180); landfrac; more_input];
  elseif strcmp(CC_pass,'multiplicative_scanangle')
    input = [input1; input2; input3; sec(scanang*pi/180); sec(scanang*pi/180).*correction_pc; ...
             landfrac; more_input];
  else
    error('invalid CC_pass');
  end

  if verbose_flag
    disp(['number of inputs to nnet is ' num2str(size(input,1)) ])
  end

  output = P * input;
  correction = correction_pct(1:num_output_pc_AIRS_correction,:)' * output + ...
               mean_correction*ones(1,size(input,2));

  AIRS_CC_tmp{i} = correction + AIRS_all_warmests{i};
end

AIRS_CC = zeros(size(AIRS_CC_tmp{1}));

channels_0_to_5   = find(height <= 5);
channels_5_to_10  = find(height >  5 & height <= 10);
channels_above_10 = find(height >  10);

AIRS_CC(channels_0_to_5  ,:) = AIRS_CC_tmp{1}(channels_0_to_5  ,:);
AIRS_CC(channels_5_to_10 ,:) = AIRS_CC_tmp{2}(channels_5_to_10 ,:);
AIRS_CC(channels_above_10,:) = AIRS_CC_tmp{3}(channels_above_10,:);

if strcmp(CC_pass,'initial')
  foo = P * input;
  varargout{1} = foo(1,:);
end
```

```
if strcmp(CC_pass,'multiplicative_scanangle')
  foo = P * input;
  varargout{1} = foo(1,:);
  varargout{2} = correction_pct(1:num_output_pc_AIRS_correction,:)' * foo + ...
                 mean_correction*ones(1,numgolfball);
end
```

B.1.2 linear-CC-train.m

```
function [estimator,varargout] = linear_CC_train(input_to_CC, AIRS_CC, channelnumber, CC_pass, CC_parameter);
%LINEAR_CC_TRAIN
%   This is a core algorithm to train stochastic cloud-clearing, using linear regression
%
%   Usage:
%     [estimator,varargout] = linear_CC_train(input_to_CC, AIRS_CC, channelnumber, CC_pass, CC_parameter)
%
%   Input:
%     input_to_CC: structure-format inputs to the CC algorithm ('AIRS', 'AMSU', 'scanang', 'landfrac', 'lat',
%                  'lon', optional 'more_input', optional 'correction_pc')
%     AIRS_CC: AIRS cloud-cleared brightness temperature
%     channelnumber: AIRS channel index being used (1-2378)
%     CC_pass: either 'initial', 'multiplicative_scanangle'
%     CC_parameter: optional 'AMSU_input_channel', 'num_output_pc_AIRS_correction', 'verbose_flag'
%                   - AMSU_input_channel: AMSU channels used for both training and testing
%                                         default = [5 6 8 9 10]
%                   - num_output_pc_AIRS_correction: number of output NAPC to keep
%                                                    default = 4
%                   - verbose_flag: when turned on, intermediate results are shown
%                                   default = off
%
%   Output:
%     estimator: structure-format estimator coefficients
%     varargout: corrections
%

%   (c) Copyright 2005 Remote Sensing and Estimation Group M.I.T.
%   Written by Chuck Cho <cycho@mit.edu>

more off

AIRS    = input_to_CC.AIRS;
AMSU    = input_to_CC.AMSU;
scanang = input_to_CC.scanang(:)';
```

```
landfrac= input_to_CC.landfrac(:)';
lat     = input_to_CC.lat(:)';
lon     = input_to_CC.lon(:)';

% input_to_CC.more_input specified?
if isfield(input_to_CC,'more_input')
  more_input = input_to_CC.more_input;
  if size(more_input,1) == size(AMSU,2)
    more_input = more_input'
  elseif size(more_input,2) ~= size(AMSU,2)
    error('"more input" size does not match')
  end
  disp('extra input detected.');
else
  more_input = [];
end

% input_to_CC.correction_pc specified?
if isfield(input_to_CC,'correction_pc')
  correction_pc = input_to_CC.correction_pc(:)';
elseif strcmp(CC_pass,'multiplicative_scanangle')
  error('If CC_pass is "multiplicative_scanangle", input_to_CC.correction_pc must be specified.')
end

% CC_parameter.AMSU_input_channel specified? (default=5,6,8,9,10)
if isfield(CC_parameter,'AMSU_input_channel')
  AMSU_input_channel = CC_parameter.AMSU_input_channel;
else
  AMSU_input_channel = [5 6 8 9 10];
end

% CC_parameter.num_output_pc_AIRS_correction specified? (Number of output PC correction: default 4)
if isfield(CC_parameter,'num_output_pc_AIRS_correction')
  num_output_pc_AIRS_correction = CC_parameter.num_output_pc_AIRS_correction;
else
  num_output_pc_AIRS_correction = 4;
end

% CC_parameter.verbose_flag turned on? (default off)
% If turned on, display intermediate results for debugging
if isfield(CC_parameter,'verbose_flag')
  verbose_flag = CC_parameter.verbose_flag
else
  verbose_flag = 0;
end
```

```
clear input_to_CC;

% load frequency
L2_chan_prop_v6_6_8_freq;
freq = frequency(channelnumber);

% load WF height
load weighting_peak.mat weighting_peak_in_height
height = weighting_peak_in_height(channelnumber);

numgolfball = size(AIRS,2);

height_for_averaging = find(height >= 1 & height <= 3 & freq > 2000 & freq < 2400);
if length(height_for_averaging) == 0
  error('No 4um channels peaking 1-3km');
end

if verbose_flag
  disp(['# of 1-3km 4um channels = ' num2str(length(height_for_averaging))])
end

% load AMSU noise variances
load amsu_noise

numchannel = length(channelnumber);

AIRS_warmest      = zeros(size(AIRS,1),size(AIRS,2));
AIRS_warmest_4avg = zeros(size(AIRS,1),size(AIRS,2));
AIRS_warmest_9avg = zeros(size(AIRS,1),size(AIRS,2));
AIRS_coldest      = zeros(size(AIRS,1),size(AIRS,2));

tmp1 = squeeze(mean(AIRS(height_for_averaging,:,:),1));

[tmp2 tmp3] = sort(tmp1,2);

% average warmest 15-km pixels
for i=1:numgolfball
  AIRS_warmest(:,i)      = AIRS(:,i,tmp3(i,9));
  AIRS_warmest_4avg(:,i) = mean([AIRS(:,i,tmp3(i,9)) AIRS(:,i,tmp3(i,8)) ...
                                 AIRS(:,i,tmp3(i,7)) AIRS(:,i,tmp3(i,6))],2);
  AIRS_warmest_9avg(:,i) = mean(AIRS(:,i,:),3);
  AIRS_coldest(:,i)      = AIRS(:,i,tmp3(i,1));
end
clear tmp*
```

```
AIRS_all_warmests = { AIRS_warmest, AIRS_warmest_4avg, AIRS_warmest_9avg };

for i=1:length(AIRS_all_warmests)

  % normalize
  mean_AIRS = mean(AIRS_all_warmests{i}')';
  zm_AIRS = AIRS_all_warmests{i} - mean_AIRS*ones(1,numgolfball);
  load airs_noise_1_10_03; noise = airs_noise(channelnumber);
  norm_AIRS = zm_AIRS ./ (sqrt(noise) * ones(1,numgolfball));

  AIRS_dif = AIRS_all_warmests{i} - AIRS_coldest;
  mean_AIRS_dif = mean(AIRS_dif')';
  zm_AIRS_dif = AIRS_dif - mean_AIRS_dif*ones(1,numgolfball);
  norm_AIRS_dif = zm_AIRS_dif ./ (sqrt(noise) * ones(1,numgolfball));

  mean_AMSU = mean(AMSU')';
  zm_AMSU = AMSU - mean_AMSU*ones(1,numgolfball);
  norm_AMSU = zm_AMSU ./ (sqrt(amsu_noise') * ones(1,numgolfball));

  [pc,xx,yy,zz] = princomp(norm_AIRS');
  AIRS_pct = pc';
  AIRS_eigenvalues = yy;

  [pc,xx,yy,zz] = princomp(norm_AIRS_dif');
  AIRS_dif_pct = pc';
  AIRS_dif_eigenvalues = yy;

  correction = reshape(AIRS_CC - AIRS_all_warmests{i}, numchannel, numgolfball);
  mean_correction = mean(correction')';
  zm_correction = correction - mean_correction * ones(1,numgolfball);
  [pc,xx,yy,zz] = princomp(zm_correction');
  correction_pct = pc';
  correction_eigenvalues = yy;

  num_input_pc_AIRS_warmest = 7;
  num_input_pc_AIRS_dif = 3;

  input1 = AIRS_pct(1:num_input_pc_AIRS_warmest,:) * norm_AIRS;
  input2 = AIRS_dif_pct(1:num_input_pc_AIRS_dif,:) * norm_AIRS_dif;
  input3 = norm_AMSU(AMSU_input_channel,:);

  if strcmp(CC_pass,'initial')
    input = [input1; input2; input3; sec(scanang*pi/180); landfrac; more_input];
  elseif strcmp(CC_pass,'multiplicative_scanangle')
```

```
    input = [input1; input2; input3; sec(scanang*pi/180); sec(scanang*pi/180).*correction_pc; ...
             landfrac; more_input];
  else
    error('invalid CC_pass');
  end

  if verbose_flag
    disp(['number of inputs to nnet is ' num2str(size(input,1)) ])
  end

  output = correction_pct(1:num_output_pc_AIRS_correction,:) * zm_correction;

  if verbose_flag
    disp('Calculating linear regression')
  end
  P = output * pinv(input);

  estimator{i}.mean_AIRS = mean_AIRS;
  estimator{i}.mean_AIRS_dif = mean_AIRS_dif;
  estimator{i}.mean_AMSU = mean_AMSU;
  estimator{i}.mean_correction = mean_correction;
  estimator{i}.AIRS_pct = AIRS_pct;
  estimator{i}.AIRS_eigenvalues = AIRS_eigenvalues;
  estimator{i}.AIRS_dif_pct = AIRS_dif_pct;
  estimator{i}.AIRS_dif_eigenvalues = AIRS_dif_eigenvalues;
  estimator{i}.correction_pct = correction_pct;
  estimator{i}.correction_eigenvalues = correction_eigenvalues;
  estimator{i}.P = P;

end

if strcmp(CC_pass,'initial')
  foo = P * input;
  varargout{1} = foo(1,:);
end

if strcmp(CC_pass,'multiplicative_scanangle')
  foo = P * input;
  varargout{1} = foo(1,:);
  varargout{2} = correction_pct(1:num_output_pc_AIRS_correction,:)' * foo + ...
                 mean_correction*ones(1,numgolfball);
end
```

B.1.3 test-all-passes.m

```
function [AIRS_CC,clear_parameter,varargout] = test_all_passes(input_to_CC, channelnumber, threshold, ...
                                                               estimator, CC_parameter);
%TEST_ALL_PASSES
%   This script "cloud-clear"s given input using three-pass CC algorithms (calling linear_CC_test.m)
%   (1) initial pass: initial pass to obtain cloud-correction
%   (2) 'multiplicative_scanangle' pass: has an additional input to CC algorithm -- cloud-correction
%       multiplied by secant of scan angle
%   (3) clear-pixel pass: CC for clear pixels
%
%   Usage:
%     [AIRS_CC,clear_parameter,varargout] = test_all_passes(input_to_CC, channelnumber, threshold, estimator,
%                                                           CC_parameter)
%
%   Input:
%     input_to_CC: structure-format inputs to the CC algorithm ('AIRS', 'AMSU', 'scanang', 'landfrac', 'lat',
%                  'lon', optional 'more_input', optional 'correction_pc')
%     channelnumber: AIRS channel index being used (1-2378)
%     threshold: structure-format threshold specification:
%                threshold{i}.channel1: 1st channel number for i-th threshold
%                threshold{i}.channel2: 2nd channel number for i-th threshold
%                threshold{i}.value1: 1st channel upper-bound thresholdvalue (%)
%                threshold{i}.value2: 2nd channel upper-bound thresholdvalue (%)
%                threshold{i}.lower_threshold_value1: 1st channel lower-bound thresholdvalue (%)
%                threshold{i}.lower_threshold_value2: 2nd channel lower-bound thresholdvalue (%)
%     estimator: structure-format estimator coefficients
%     CC_parameter: optional 'AMSU_input_channel', 'num_output_pc_AIRS_correction', 'verbose_flag'
%                   - AMSU_input_channel: AMSU channels used for both training and testing
%                                         default = [5 6 8 9 10]
%                   - num_output_pc_AIRS_correction: number of output NAPC to keep
%                                                    default = 4
%                   - verbose_flag: when turned on, intermediate results are shown
%                                   default = off
%
%   Output:
%     AIRS_CC: AIRS cloud-cleared brightness temperature
%     clear_parameter: structure-format parameters, mostly for debugging purpose
%                      clear_parameter{i}.abs_threshold_value1 = abs_threshold_value1;
%     varargout: corrections for all channels (for debugging purpose)

%   (c) Copyright 2005 Remote Sensing and Estimation Group M.I.T.
%   Written by Chuck Cho <cycho@mit.edu>
```

```
if isfield(estimator,'prefix') & isfield(estimator,'dir')
  disp('Load estimator...');
  load([estimator.dir estimator.prefix '_A.mat'],'estimator_initial');

  load([estimator.dir estimator.prefix '_B.mat'],'estimator_multipl_scanangle');

  for i=1:length(threshold)
    load([estimator.dir estimator.prefix '_C' num2str(threshold{i}.value1) '.mat'],'estimator_most_clear');
    estimator_most_clear_all{i} = estimator_most_clear;
  end
else
  estimator_initial = estimator{1};
  estimator_multipl_scanangle = estimator{2};
  for i=1:length(threshold)
    estimator_most_clear_all{i} = estimator{i+2};
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%% first pass (estimator A): initial
CC_pass = 'initial';
[AIRS_CC_estimate_initial,correction_1st_pc]=linear_CC_test(input_to_CC, channelnumber, CC_pass, ...
                                          CC_parameter, estimator_initial);

%%%%%%%%%%%%%%%%%%%%%%%%% second pass (estimator B): multiplicative scan angle correction
CC_pass = 'multiplicative_scanangle';
input_to_CC.correction_pc = correction_1st_pc;
[AIRS_CC_estimate_multipl_ang,correction_1st_pc,correction_all_channel]=linear_CC_test(input_to_CC, ...
  channelnumber, CC_pass, CC_parameter, estimator_multipl_scanangle);

%%%%%%%%%%%%%%%%%%%%%%%%% third pass (estimator C):
% most clear pixels using correction-thresholds for two channels
CC_pass = 'multiplicative_scanangle';

input_to_CC.correction_pc = correction_1st_pc;
numgolfball = length(correction_1st_pc);

for i=1:length(threshold)

  threshold_channel1 = threshold{i}.channel1;
  threshold_channel2 = threshold{i}.channel2;
  threshold_value1 = threshold{i}.value1; % in percent
  threshold_value2 = threshold{i}.value2;
  lower_threshold_value1 = threshold{i}.lower_threshold_value1;
  lower_threshold_value2 = threshold{i}.lower_threshold_value2;
```

```
ch1_index = find(channelnumber == threshold_channel1);
ch2_index = find(channelnumber == threshold_channel2);

for current_threshold = -30:0.05:0
  if length(find(correction_all_channel(ch1_index,:) < current_threshold)) > ...
     lower_threshold_value1/100*numgolfball
    break
  end
end
abs_lower_threshold_value1 = current_threshold;

for current_threshold = -30:0.05:0
  if length(find(correction_all_channel(ch2_index,:) < current_threshold)) > ...
     lower_threshold_value2/100*numgolfball
    break
  end
end
abs_lower_threshold_value2 = current_threshold;

for current_threshold = 50:-.005:-20
  if length(find(correction_all_channel(ch1_index,:) < current_threshold & ...
                 correction_all_channel(ch1_index,:) >= abs_lower_threshold_value1)) ...
           < threshold_value1/100*numgolfball
    break
  end
end
abs_threshold_value1 = current_threshold;

for current_threshold = 50:-.005:-20
  if length(find(correction_all_channel(ch2_index,:) < current_threshold & ...
                 correction_all_channel(ch2_index,:) >= abs_lower_threshold_value2)) ...
           < threshold_value2/100*numgolfball
    break
  end
end
abs_threshold_value2 = current_threshold;

clear_parameter{i}.abs_threshold_value1 = abs_threshold_value1;
clear_parameter{i}.abs_threshold_value2 = abs_threshold_value2;
clear_parameter{i}.abs_lower_threshold_value1 = abs_lower_threshold_value1;
clear_parameter{i}.abs_lower_threshold_value2 = abs_lower_threshold_value2;

most_clear_index{i} = find(correction_all_channel(ch1_index,:) < abs_threshold_value1 & ...
                           correction_all_channel(ch1_index,:) >= abs_lower_threshold_value1 & ...
                           correction_all_channel(ch2_index,:) < abs_threshold_value2 & ...
```

```
                          correction_all_channel(ch2_index,:) >= abs_lower_threshold_value2 );

  yield{i} = length(most_clear_index{i}) / numgolfball;

  input_to_CC_clear.AIRS = input_to_CC.AIRS(:,most_clear_index{i},:);
  input_to_CC_clear.AMSU = input_to_CC.AMSU(:,most_clear_index{i});
  input_to_CC_clear.correction_pc = input_to_CC.correction_pc(most_clear_index{i});
  input_to_CC_clear.scanang = input_to_CC.scanang(most_clear_index{i});
  input_to_CC_clear.landfrac = input_to_CC.landfrac(most_clear_index{i});
  input_to_CC_clear.lat = input_to_CC.lat(most_clear_index{i});
  input_to_CC_clear.lon = input_to_CC.lon(most_clear_index{i});

  AIRS_CC_most_clear{i}=linear_CC_test(input_to_CC_clear, channelnumber, CC_pass, CC_parameter, ...
                       estimator_most_clear_all{i});
end

AIRS_CC{1} = AIRS_CC_estimate_initial;
AIRS_CC{2} = AIRS_CC_estimate_multipl_ang;
for i=1:length(threshold)
  AIRS_CC{i+2} = AIRS_CC_most_clear{i};
  clear_parameter{i}.yield = yield{i};
  clear_parameter{i}.clear_index = most_clear_index{i};
end

if nargout == 3
  varargout{1} = correction_all_channel;
end
```

B.1.4 train-all-passes.m

```
function estimator = train_all_passes(input_to_CC, AIRS_CC, channelnumber, threshold, save_parameter, ...
                                      CC_parameter)
%TRAIN_ALL_PASSES
%   This script is to train stochastic CC estimators for three-pass CC algorithms (calling linear_CC_train.m)
%   (1) initial pass: initial pass to obtain cloud-correction
%   (2) 'multiplicative_scanangle' pass: has an additional input to CC algorithm -- cloud-correction
%                                        multiplied by secant of scan angle
%   (3) clear-pixel pass: CC for clear pixels
%
%   Usage:
%     estimator = train_all_passes(input_to_CC, AIRS_CC, channelnumber, threshold, save_parameter,
%                                  CC_parameter)
%
```

```
%   Input:
%     input_to_CC: structure-format inputs to the CC algorithm ('AIRS', 'AMSU', 'scanang', 'landfrac', 'lat',
%                  'lon', optional 'more_input', optional 'correction_pc')
%     AIRS_CC: AIRS cloud-cleared brightness temperature
%     channelnumber: AIRS channel index being used (1-2378)
%     threshold: structure-format threshold specification:
%                threshold{i}.channel1: 1st channel number for i-th threshold
%                threshold{i}.channel2: 2nd channel number for i-th threshold
%                threshold{i}.value1: 1st channel upper-bound thresholdvalue (%)
%                threshold{i}.value2: 2nd channel upper-bound thresholdvalue (%)
%                threshold{i}.lower_threshold_value1: 1st channel lower-bound thresholdvalue (%)
%                threshold{i}.lower_threshold_value2: 2nd channel lower-bound thresholdvalue (%)
%     CC_parameter: optional 'AMSU_input_channel', 'num_output_pc_AIRS_correction', 'verbose_flag'
%                   - AMSU_input_channel: AMSU channels used for both training and testing
%                                         default = [5 6 8 9 10]
%                   - num_output_pc_AIRS_correction: number of output NAPC to keep
%                                                    default = 4
%                   - verbose_flag: when turned on, intermediate results are shown
%                                   default = off
%
%   Output:
%     estimator: structure-format estimator coefficients

%   (c) Copyright 2005 Remote Sensing and Estimation Group M.I.T.
%   Written by Chuck Cho <cycho@mit.edu>


if isfield(save_parameter,'save_flag')
  save_flag = save_parameter.save_flag;
  save_dir  = save_parameter.save_dir;
  save_prefix = save_parameter.save_prefix;
else
  save_flag = 0;
end

%%%%%%%%%%%%%%%%%%%%%%%%% first pass (estimator A): initial
CC_pass = 'initial';
[estimator_initial,correction_1st_pc]=linear_CC_train(input_to_CC, AIRS_CC, channelnumber, CC_pass, ...
                                                      CC_parameter);

%%%%%%%%%%%%%%%%%%%%%%%%% second pass (estimator B): multiplicative scan angle correction
CC_pass = 'multiplicative_scanangle';
input_to_CC.correction_pc = correction_1st_pc;
[estimator_multipl_scanangle,correction_1st_pc,correction_all_channel]=linear_CC_train(input_to_CC, ...
AIRS_CC, channelnumber, CC_pass, CC_parameter);
```

```
%%%%%%%%%%%%%%%%%%%%%%%% third pass (estimator C):
% most clear pixels using correction-thresholds for two channels
CC_pass = 'multiplicative_scanangle';

input_to_CC.correction_pc = correction_1st_pc;
numgolfball = length(AIRS_CC);

for i=1:length(threshold)

  threshold_channel1 = threshold{i}.channel1;
  threshold_channel2 = threshold{i}.channel2;
  threshold_value1 = threshold{i}.value1; % in percent
  threshold_value2 = threshold{i}.value2;
  lower_threshold_value1 = threshold{i}.lower_threshold_value1;
  lower_threshold_value2 = threshold{i}.lower_threshold_value2;

  ch1_index = find(channelnumber == threshold_channel1);
  ch2_index = find(channelnumber == threshold_channel2);

  for current_threshold = -30:0.05:0
    if length(find(correction_all_channel(ch1_index,:) < current_threshold)) > ...
       lower_threshold_value1/100*numgolfball
      break
    end
  end
  abs_lower_threshold_value1 = current_threshold;

  for current_threshold = -30:0.05:0
    if length(find(correction_all_channel(ch2_index,:) < current_threshold)) > ...
       lower_threshold_value2/100*numgolfball
      break
    end
  end
  abs_lower_threshold_value2 = current_threshold;

  for current_threshold = 50:-.005:-20
    if length(find(correction_all_channel(ch1_index,:) < current_threshold & ...
                   correction_all_channel(ch1_index,:) >= abs_lower_threshold_value1)) ...
              < threshold_value1/100*numgolfball
      break
    end
  end
  abs_threshold_value1 = current_threshold;
```

```
  for current_threshold = 50:-.005:-20
    if length(find(correction_all_channel(ch2_index,:) < current_threshold & ...
                   correction_all_channel(ch2_index,:) >= abs_lower_threshold_value2)) ...
              < threshold_value2/100*numgolfball
      break
    end
  end
  abs_threshold_value2 = current_threshold;

  most_clear_index = find(correction_all_channel(ch1_index,:) < abs_threshold_value1 & ...
                          correction_all_channel(ch1_index,:) >= abs_lower_threshold_value1 & ...
                          correction_all_channel(ch2_index,:) < abs_threshold_value2 & ...
                          correction_all_channel(ch2_index,:) >= abs_lower_threshold_value2 );

  yield = length(most_clear_index) / numgolfball;

  input_to_CC_clear.AIRS = input_to_CC.AIRS(:,most_clear_index,:);
  input_to_CC_clear.AMSU = input_to_CC.AMSU(:,most_clear_index);
  input_to_CC_clear.correction_pc = input_to_CC.correction_pc(most_clear_index);
  input_to_CC_clear.scanang = input_to_CC.scanang(most_clear_index);
  input_to_CC_clear.landfrac = input_to_CC.landfrac(most_clear_index);
  input_to_CC_clear.lat = input_to_CC.lat(most_clear_index);
  input_to_CC_clear.lon = input_to_CC.lon(most_clear_index);
  AIRS_CC_clear = AIRS_CC(:,most_clear_index);

  estimator_most_clear_all{i}=linear_CC_train(input_to_CC_clear, AIRS_CC_clear, channelnumber, CC_pass, ...
                              CC_parameter);
end

estimator{1} = estimator_initial;
estimator{2} = estimator_multipl_scanangle;
for i=1:length(threshold)
  estimator{i+2} = estimator_most_clear_all{i};
end

if save_flag
  save([save_dir save_prefix '_A.mat'],'estimator_initial');
  save([save_dir save_prefix '_B.mat'],'estimator_multipl_scanangle');
  for i=1:length(threshold)
    estimator_most_clear = estimator_most_clear_all{i};
    save([save_dir save_prefix '_C' num2str(threshold{i}.value1) '.mat'],'estimator_most_clear');
  end
end
```

Appendix C

Cloud-clearing performance

C.1 Cloud-clearing performance results for 10 cases

Each page in this Appendix has following five figures:

1. 1-km-bin average RMS difference for cloud-cleared brightness temperature (TB) relative to ECMWF/SARTA plotted against weighting function peak in km.

2. 1-km-bin average RMS difference for cloud-cleared TB when AMSU is eliminated in both training and testing.

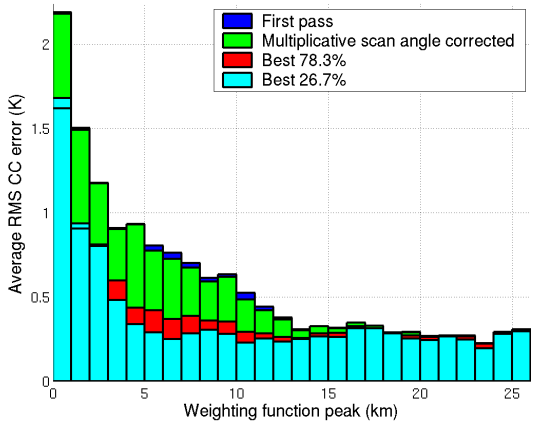
3. RMS cloud-clearing difference for all 314 channels[1] using best ∼28% golfballs[2].

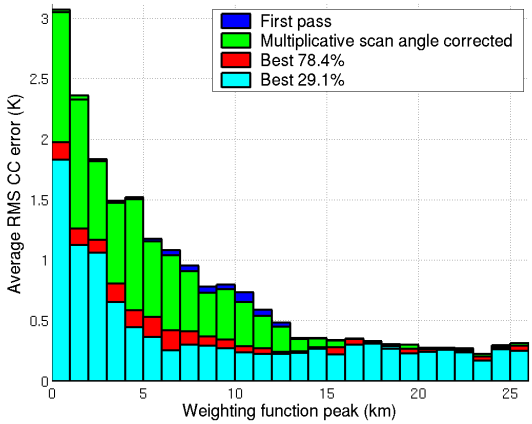
4. RMS cloud-clearing difference for all 314 channels[1] using best ∼28% golfballs, if AMSU is unavailable.

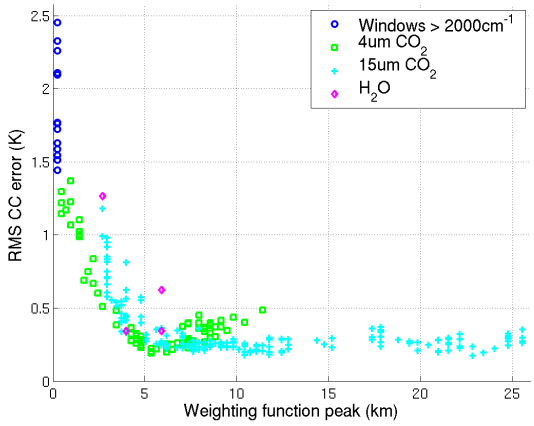
5. 1-km-bin average RMS difference comparisons for best ∼78% and ∼28% with and without AMSU contribution.

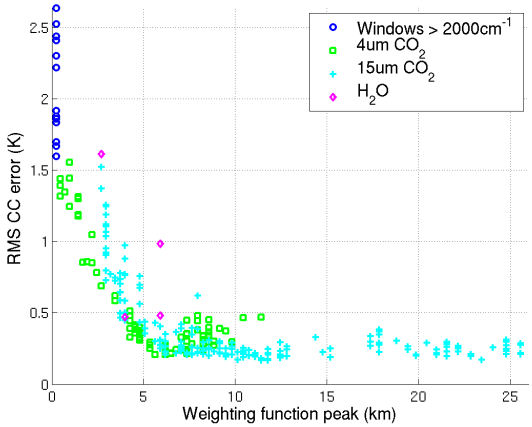
[1] See Appendix A for the full channel listing.

[2] The "best" X% golfballs refer to ones which pass the dual-channel threshold test, thus presumably least cloudy data.









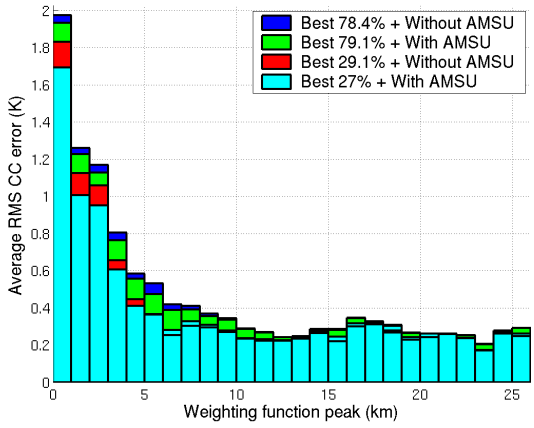
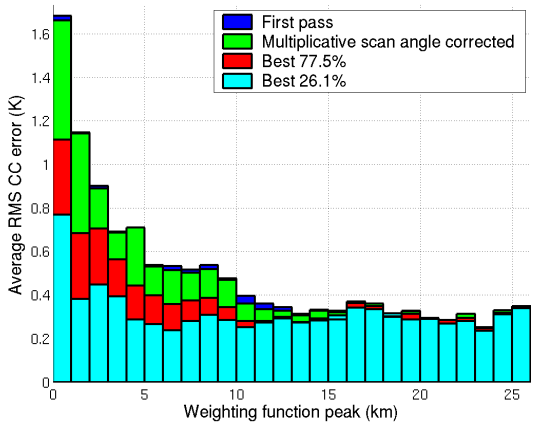
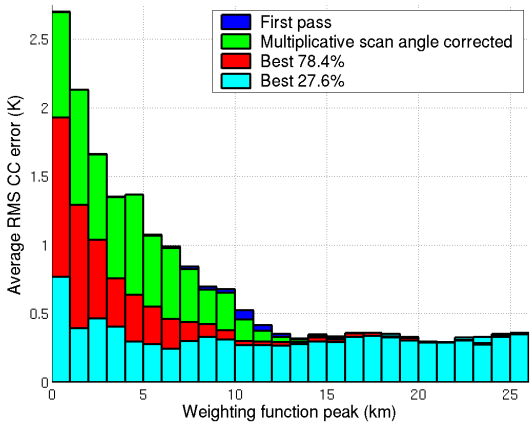
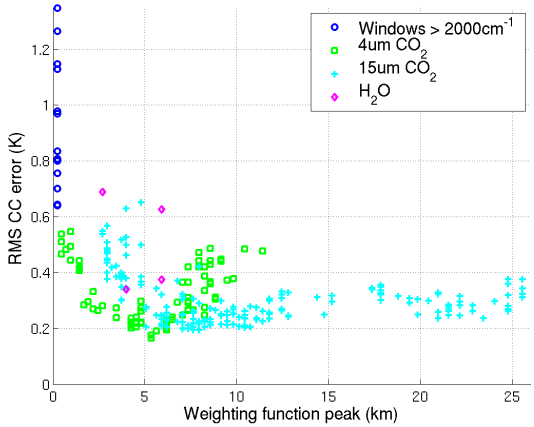
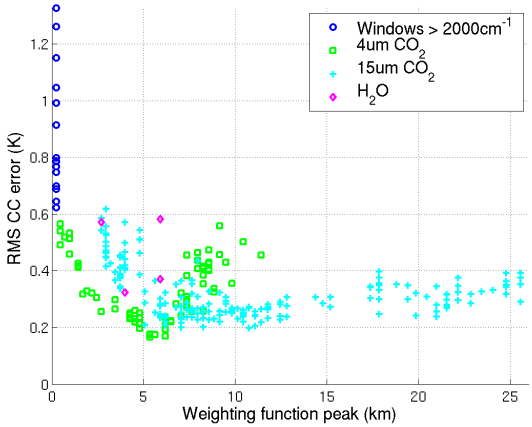


Figure C-1: Stochastic cloud-clearing results: land, |latitude| < 40, day









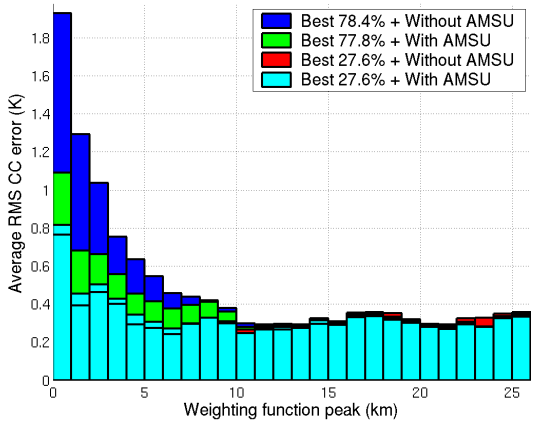
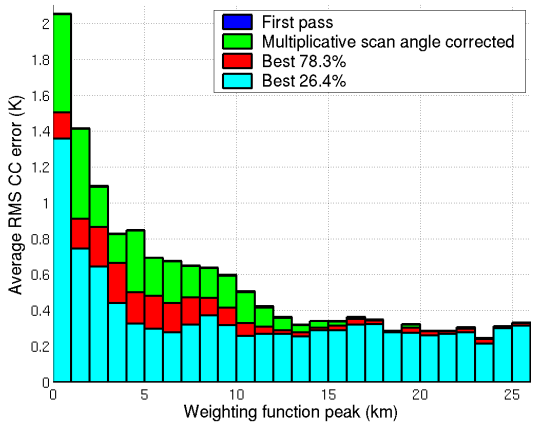
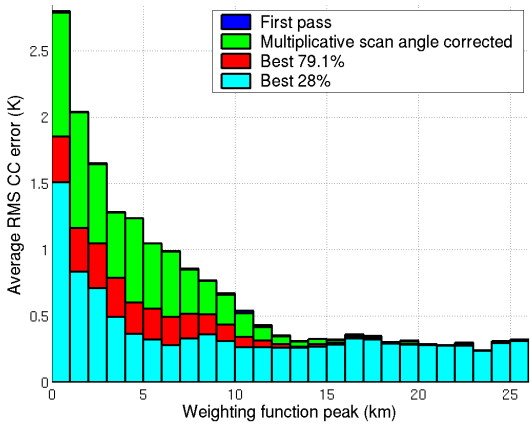
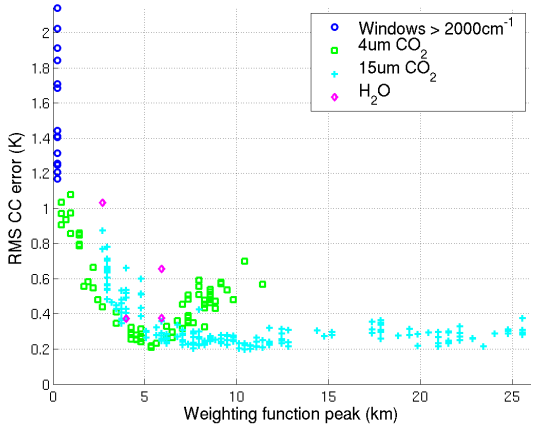
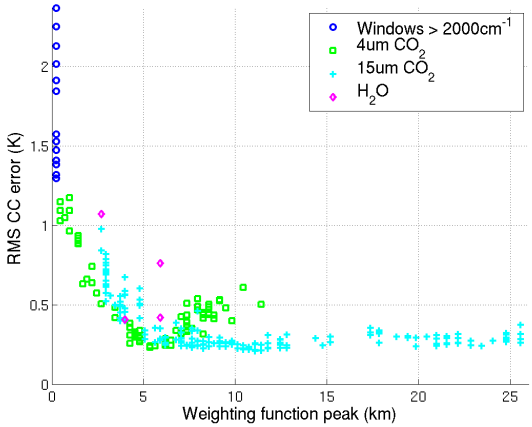


Figure C-2: Stochastic cloud-clearing results: land, |latitude| < 40, night









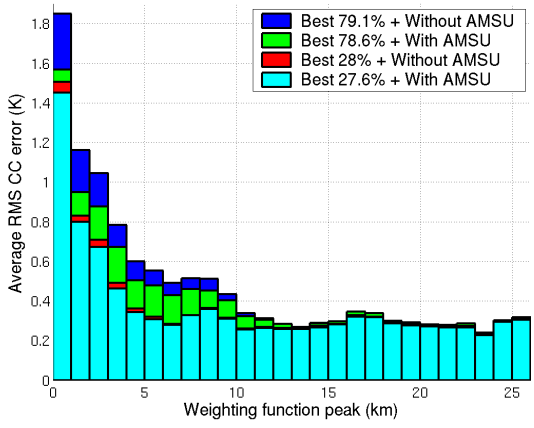
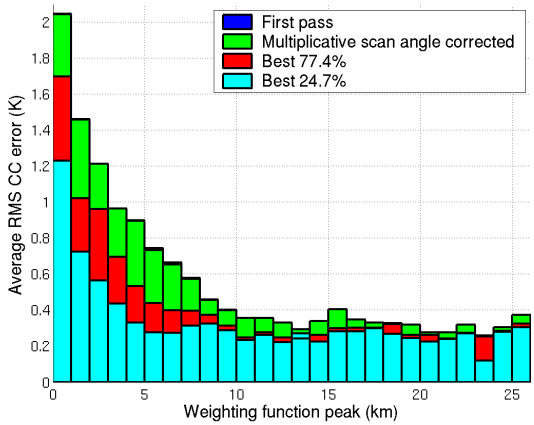
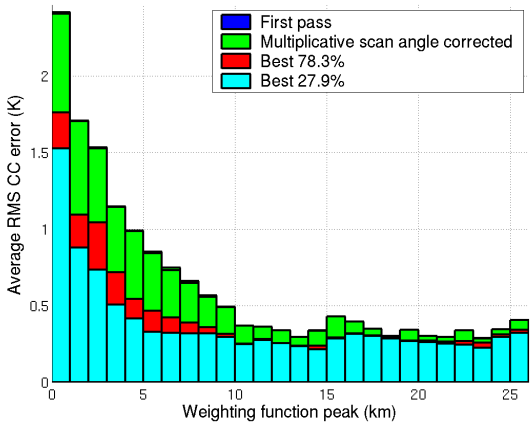
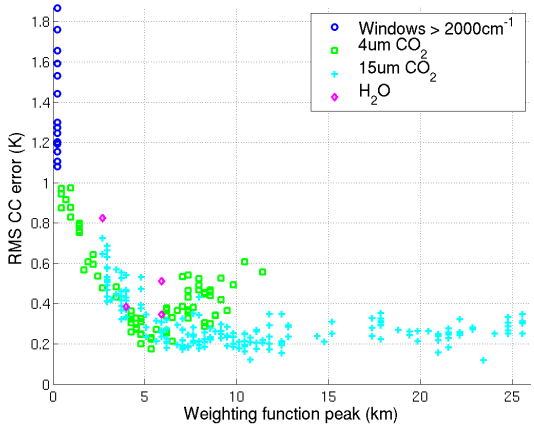
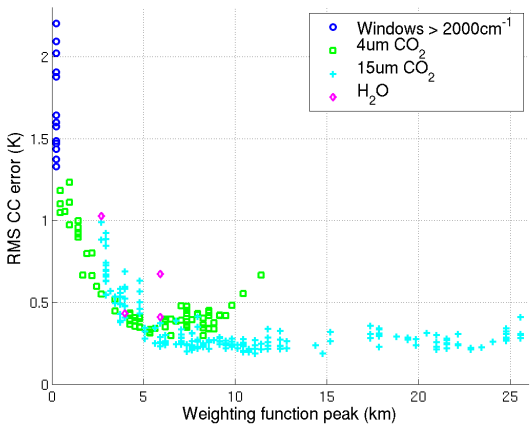


Figure C-3: Stochastic cloud-clearing results: land, |latitude| < 40, day + night









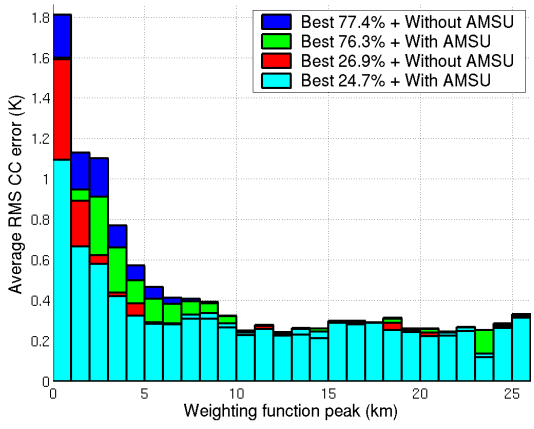
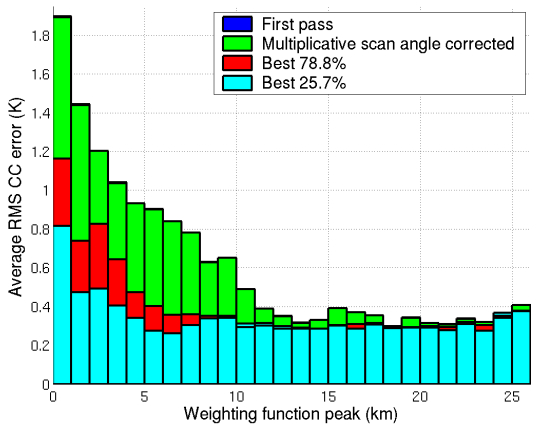
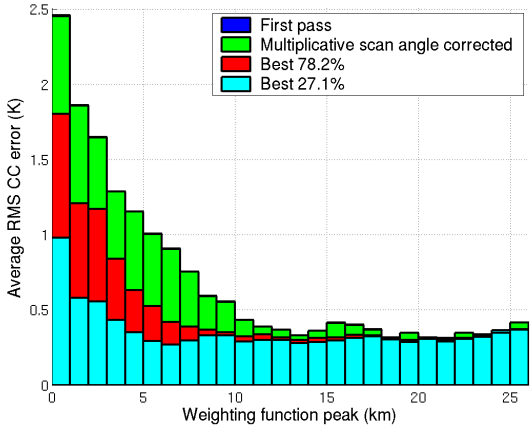
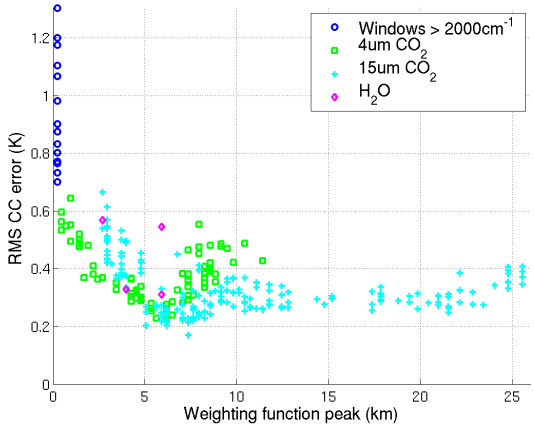
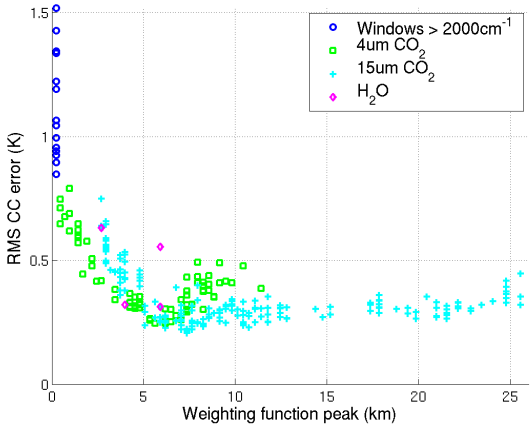


Figure C-4: Stochastic cloud-clearing results: land, $30 < |\text{latitude}| < 70$, day









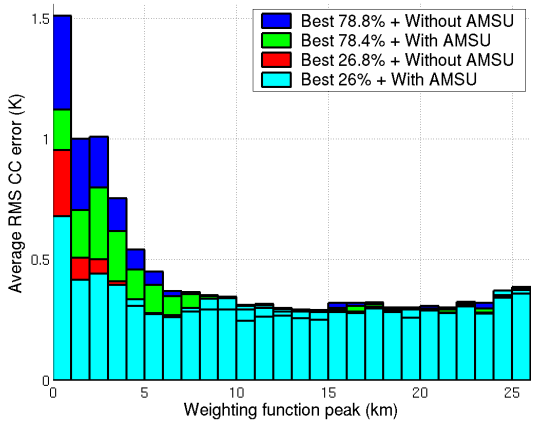
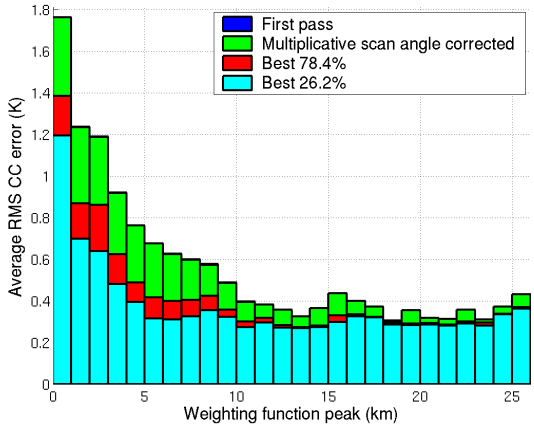
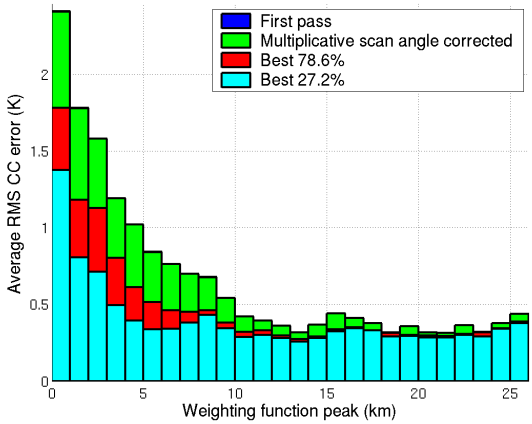
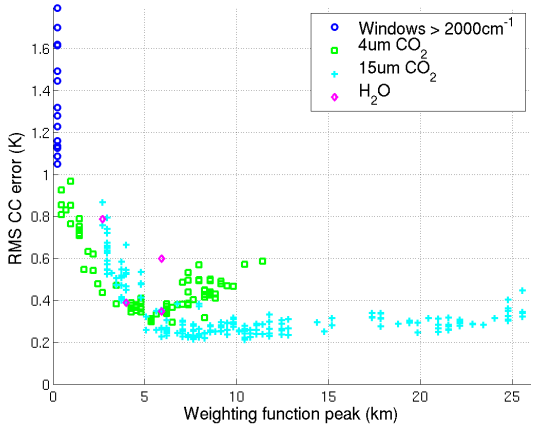
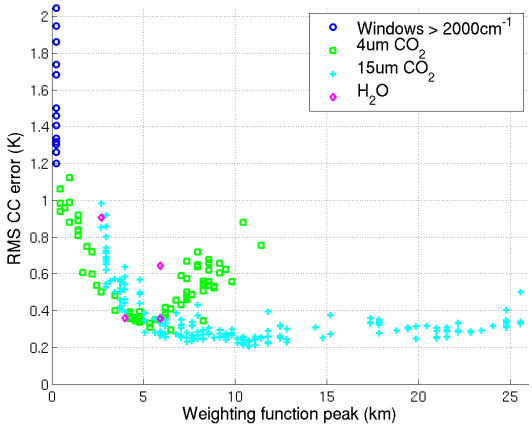


Figure C-5: Stochastic cloud-clearing results: land, $30 < |\text{latitude}| < 70$, night









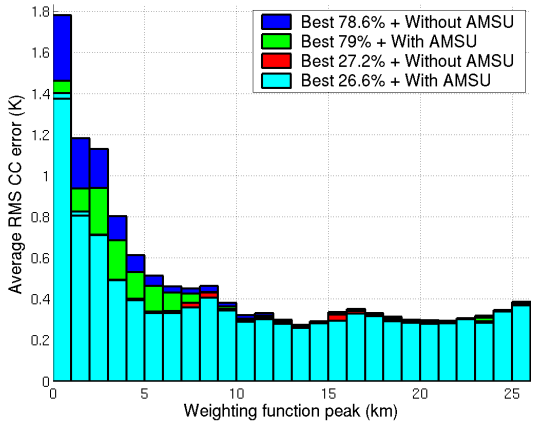
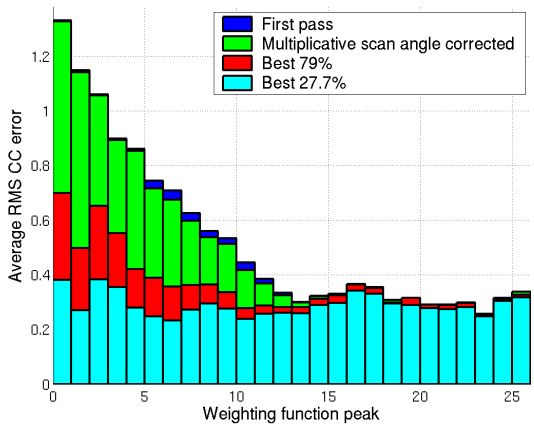
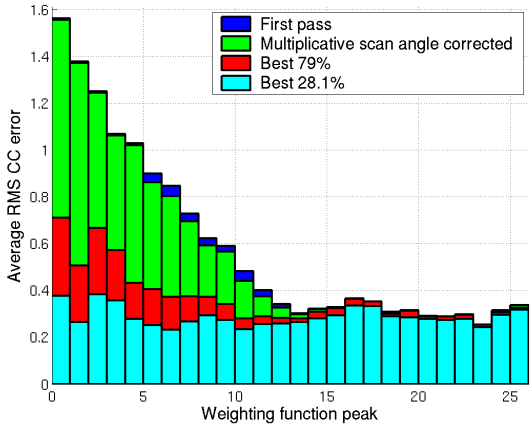
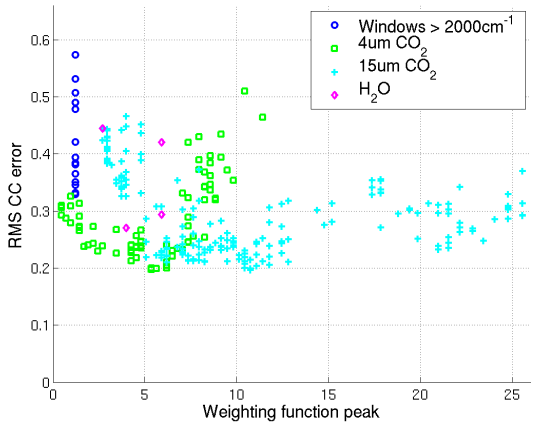
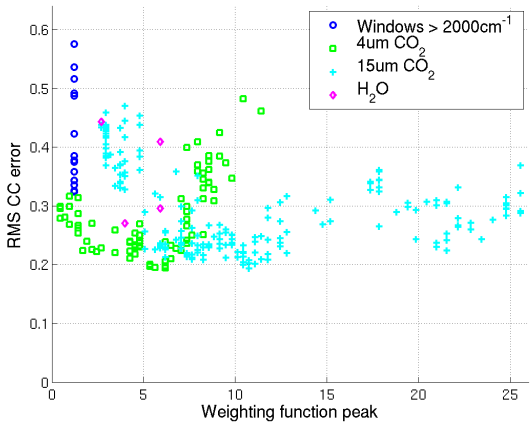


Figure C-6: Stochastic cloud-clearing results: land, $30 < |\text{latitude}| < 70$, day + night









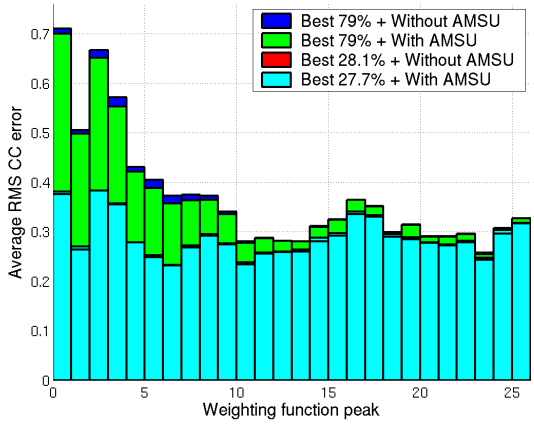
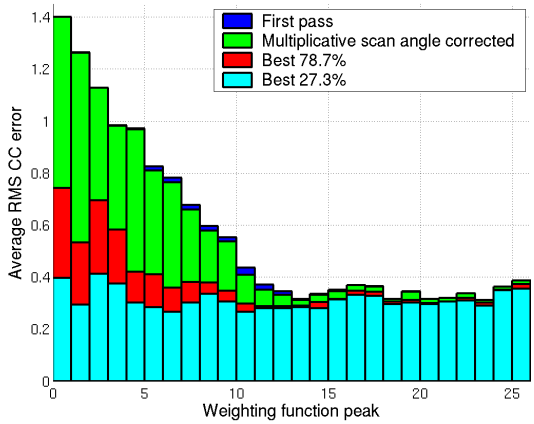
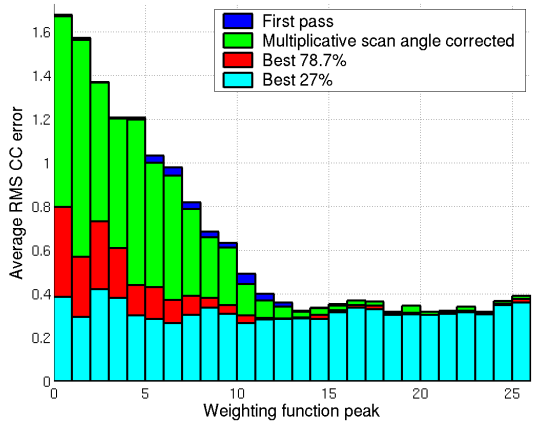
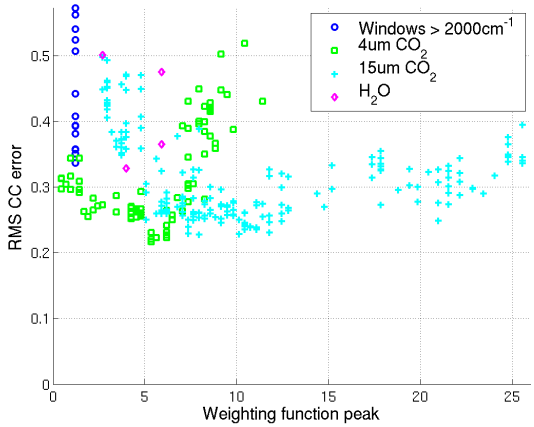
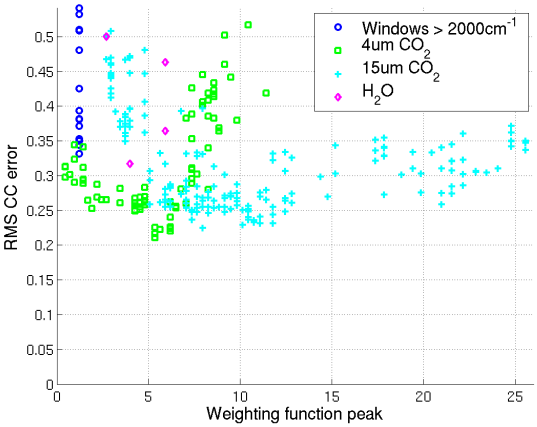


Figure C-7: Stochastic cloud-clearing results: sea, |latitude| < 40, day









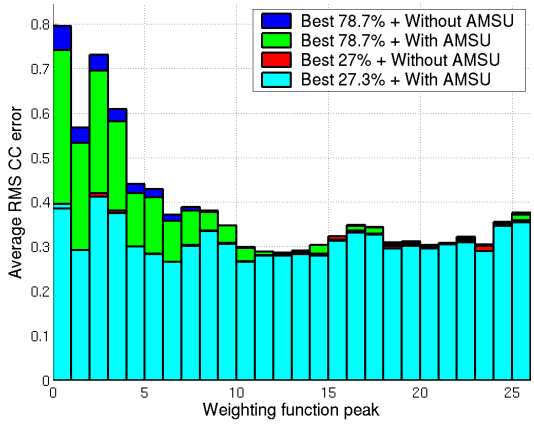
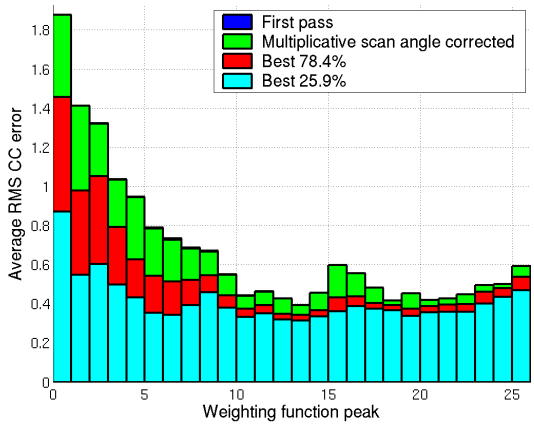
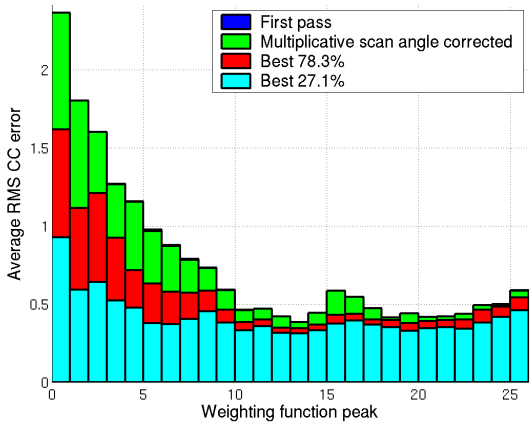
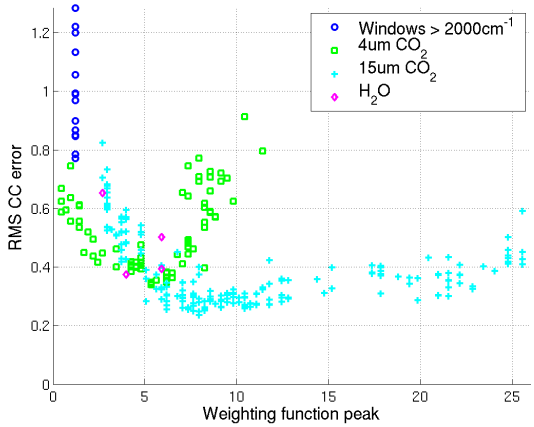
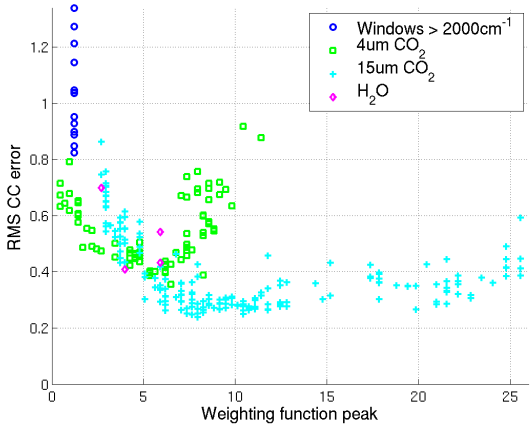


Figure C-8: Stochastic cloud-clearing results: sea, |latitude| < 40, night









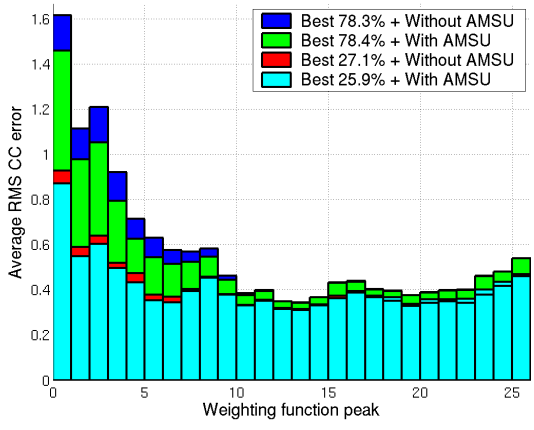
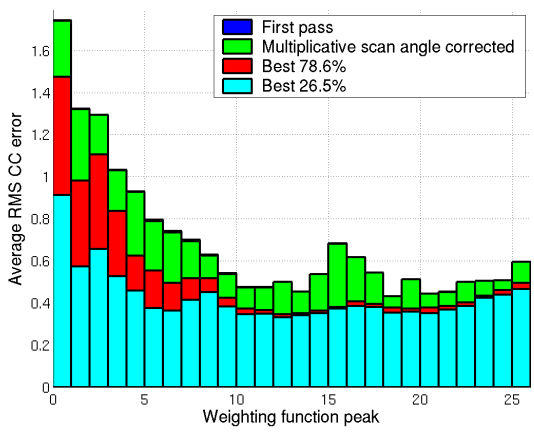
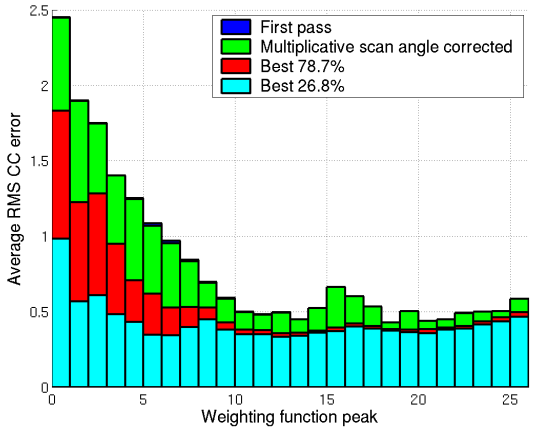
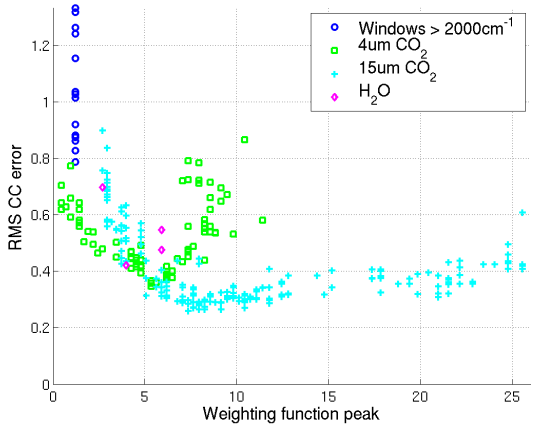
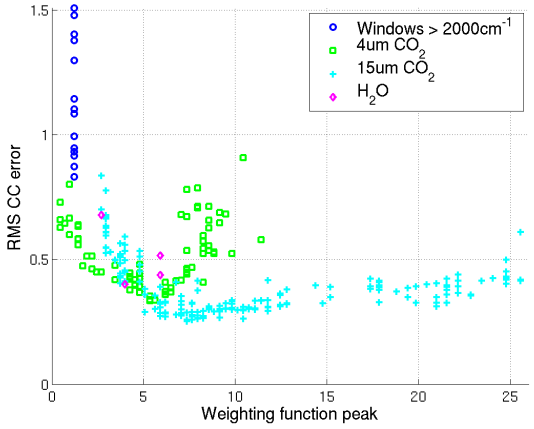


Figure C-9: Stochastic cloud-clearing results: sea, $30 < |\text{latitude}| < 70$, day









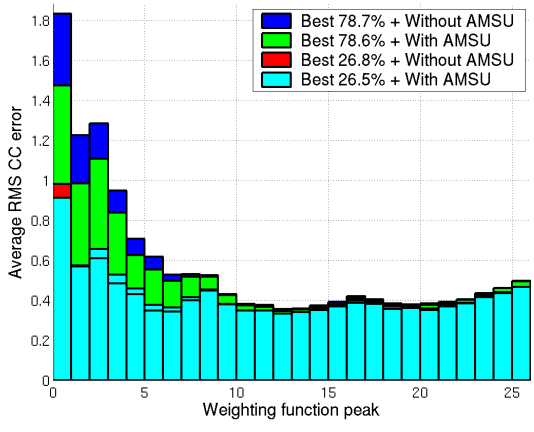


Figure C-10: Stochastic cloud-clearing results: sea, $30 < |\text{latitude}| < 70$, night

Bibliography

[1] A. Andretta, C. Serio, M.M. Bonzagni, R. Rizzi, and V. Cuomo, "A three-step cloud clearing procedure for infrared sounder measurements," *Int. J. Remote Sensing*, vol 11, no. 8, pp. 1349–1368, 990.

[2] H.H. Aumann et al., "AIRS/AMSU/HSB on the Aqua Mission: Design, Science Objectives, Data Products, and Processing Systems," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 2, Feb. 2003.

[3] H.H. Aumann, "High Spectral Resolution Cloud Filters Using 20030103.142 as Example," *Report to AIRS Science Team Meeting*, Jan. 2004.

[4] H.H. Aumann, M.T. Chahine, C. Gautier, M.D. Goldberg, E. Kalnay, L.M. McMillin, H. Revercomb, P.W. Rosenkranz, W.L. Smith, D.H. Staelin, L.L. Strow, and J. Susskind, "AIRS/AMSU/HSB on the Aqua Mission: Design, Science Objectives, Data Products, and Processing Systems," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 2, pp. 253–264, Feb. 2003.

[5] W.J. Blackwell, "Retrieval of Cloud-Cleared Atmospheric Temperature Profiles from Hyperspectral Infrared and Microwave Observations," Sc.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, June 2002.

[6] W.J. Blackwell, "Retrieval of Atmospheric Temperature and Moisture Profiles from Hyperspectral Sounding Data Using a Projected Principal Components Transform and a Neural Network," *Proceedings of the 2003 IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, pp. 2078–2081, July 2003.

[7] C.R. Cabrera-Mercader, "Robust Compression of Multispectrtal Remote Sensing Data," Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1999.

[8] M.T. Chahine, "Remote sounding of cloudy atmospheres. I. The single cloud layer," *J. Atmos. Sci.*, vol. 31, pp. 233–243, 1974.

[9] M.T. Chahine, "Remote sounding of cloudy atmospheres. II. Multiple cloud formations," *J. Atmos. Sci.*, vol. 34, pp. 744–757, 1977.

[10] M.T. Chahine, H. Aumann, M. Goldberg, L. McMillin, P. Rosenkranz, D. Staelin, L. Strow, and J. Susskind, "AIRS Level 2 Algorithm Theoretical Basis Document Version 2.2: AIRS-Team Retrieval for Core Products and Geophysical Parameters," 2001.

[11] Carri W. Chan, "Noise-adjusted Principal Component Analysis to Detect Faulty AIRS Frequency Channels," UROP Report, Spring 2003.

[12] S. Chandrasekhar. *Radiative Transfer*, Dover Inc., New York, 1960.

[13] F.W. Chen, "Global Estimation of Precipitation Using Opaque Microwave Bands," Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2004.

[14] C. Cho, and D.H. Staelin, "Stochastic Cloud Clearing of Hyperspectral Radiances Observed by the Atmospheric InfraRed Sounder (AIRS) on the Aqua Satellite", *Proceedings of the 13th Conference on Satellite Meteorology and Oceanography*, 2004.

[15] C. Cho, and D.H. Staelin, "AIRS observations versus numerical weather predictions of cloud-cleared radiances," *Journal of Geophysical Research Atmospheres*, submitted.

[16] E. Crist, C. Schwartz and A. Stocker, "Pairwise adaptive linear matched-filter algorithm," *Proc. DARPA Adaptive Spectral Reconnaissance Algorithm Workshop*, Jan. 1999.

[17] C.F. Ferrara, "Adaptive Spatial/Spectral detection of subpixel targets with unknown spectral characteristics," *Proc. SPIE*, vol. 2235, pp 82–93, 1994.

[18] E. Fishbein, C.B. Farmer, S.L. Granger, D.T. Gregorich, M.R. Gunson, S.E. Hannon, M.D. Hofstadter, S.-Y. Lee, S.S. Leroy, and L.L. Strow, "Formulation and validation of simulated data for the Atmospheric Infrared Sounder (AIRS)," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 2, pp. 314–329, Feb. 2003.

[19] C. Gautier, S. Yang, and M.D. Hofstadter, "AIRS/Vis Near IR Instrument," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 2, pp. 330–342, Feb. 2003.

[20] A.A. Green, M. Berman, P. Switzer, and M.D. Craig, "A Transformation for Ordering Multispectral Data in Terms of Image Quality with Implications for Noise Removal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 26, no. 1, Jan. 1988.

[21] J. Grossmann, et al, "Hyperspectral analysis and target detection system for the adaptive-spectral reconnaissance program (ASRP)," *Proc. SPIE*, vol. 3372, Apr. 1998.

[22] H.T. Haskett and A.K. Sood, "Adaptive real-time endmember selection algorithm for sub-pixel target detection using hyperspectral data," *Proc. 1997 IRIS Specialty Group Camouflage, Concealment, Deception*, Oct. 1997.

[23] K. Herring, "Blind Separation of Noisy Multivariate Data Using Second-Order Statistics," S.M. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, December 2005.

[24] A. Hyvärinen, "Survey on Independent Component Analysis," *Neural Computing Surveys*, vol. 2, pp. 94–128, 1999.

[25] I.T. Joliffe, *Principal Component Analysis*, Springer-Verlag New York, Inc., New York, NY, 2002.

[26] J. Lee, C.-Y. Liu, H.-L. Huang, T.J. Schmit, X. Wu, W.P. Menzel, and J.J. Gurka, "Optimal Cloud-Clearing for AIRS Radiances Using MODIS," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1266–1278, June 2005.

[27] J. Lee, "Blind Noise Estimation and Compensation for Improved Characterization of Multivariate Processes," Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, March 2000.

[28] J. Lee and D.H. Staelin, "Iterative Signal-Order and Noise Estimation for Multivariate Data," *IEE Electronics Letters*, vol. 37, no. 2, pp. 134–135, 18 January 2001.

[29] J.B. Lee, "Enhancement of High Spectral Resolution Remote-Sensing Data by a Noise-Adjusted Principal Components Transform," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 3, May 1990.

[30] F.K. Lutgens and E.D. Tarbuck, *The Atmosphere*, Prentie Hall, New Jersey, 2001.

[31] D.W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp, 431–441, June 1963.

[32] A. Mueller, "Iterative Blind Seperation of Gaussian Data of Unknown Order," M.Eng. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer, June 2003.

[33] A. Mueller and D.H. Staelin, "Blind Separation of Noisy Gaussian Multivariate Signals," *IEEE Transactions on Signal Processing*, submitted.

[34] T.S. Pagano, H.H. Aumann, D.E. Hagan, and K. Overoye, "Prelaunch and In-Flight Radiometric Calibration of the Atmospheric Infrared Sounder (AIRS)," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 2, Feb. 2003.

[35] I.S. Reed and X. Yu, "Adaptive Multiple-Band CFAR Detection of an Optical Pattern with Unknown Spectral Distribution," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, Oct 1990.

[36] S.M. Schweizer and José M.F. Moura, "Efficient Detection in Hyperspectral Imagery," *IEEE Transactions on Image Processing*, vol. 10, no. 4, Apr. 2001.

[37] D.H. Staelin, "Receivers, Antennas, and Signals in Communication and Sensing Systems," Course note for 6.661 at Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, August 2001.

[38] D.W.J. Stein, S.G. Beaven, L.E. Hoff, E.M. Winter, A.P. Schaum, and A.D. Stocker, "Anomaly Detection from Hyperspectral Imagery," *IEEE Signal Processing Magazine*, Jan. 2002.

[39] A. Stocker, "Stochastic Expectation Maximization (SEM) Algorithm," *Proc. DARPA Adaptive Spectral Reconnaissance Algorithm Workshop*, Jan. 1999.

[40] G. Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press, Wellesley, MA, 1993.

[41] L.L. Strow, S.E. Hannon, S. De Souza-Machado, H.E. Motteler, and D. Tobin, "An overview of the AIRS radiative transfer model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 2, pp. 303–303, Feb. 2003.

[42] J. Susskind, C.D. Barnet, and J.M. Blaisdell, "Retrieval of atmospheric and surface parameters from AIRS/AMSU/HSB data in the presence of clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 2, pp. 390–409, Feb. 2003.

[43] A.S. Willsky, G.W. Wornell, and J.H. Shapiro, " Stochastic Processes, Detection and Estimation," Course notes for 6.432 at Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Fall 1998.

[44] H.E.M. Viggh, "Surface Prior Information Reflectance Estimation (SPIRE) Algorithms," Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Aug. 2001.

[45] http://asl.umbc.edu/pub/packages/sarta.html

[46] http://aviris.jpl.nasa.gov/html/aviris.cube.html

[47] http://disc.gsfc.nasa.gov/AIRS/documentation.shtml

[48] http://redhook.gsfc.nasa.gov/∼imswww/pub/imswelcome/

[49] http://www.nco.ncep.noaa.gov/pmb/nwprod/analysis/