Precipitation measurements using 54 and 183 GHz AMSU satellite observations

by

Antonio Fuentes Loyola

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of Bachelor of Science in Electrical Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 21, 1999

© Massachusetts Institute of Technology, 1999. All Rights Reserved.

Author ........................................
Department of Electrical Engineering and Computer Science
May 21, 1999

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

Accepted by ..........
Arthur C. Smith
Chairman, Department Committee on Graduate Students
Precipitation measurements using 54 and 183 GHz AMSU satellite observations

by

Antonio Fuentes Loyola

Submitted to the Department of Electrical Engineering and Computer Science on May 21, 1999, in partial fulfillment of the requirements for the degrees of Bachelor of Science in Electrical Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science

Abstract

Brightness temperatures from the Advanced Microwave Sounding Unit (AMSU) on the NOAA-15 satellite respond in part to precipitation. In two separate experiments, the Tb’s from AMSU were input to a simple neural network to yield precipitation estimates. This neural network was trained with data from the Next Generation Radar (NEXRAD). In the first experiment, the 183±1 and 183±7 GHz AMSU-B humidity channels were used as inputs to the network. In the second experiment, AMSU-A temperature profile channels 4-7 were used instead, all with frequencies near 54 GHz. For both experiments, the training and test data included a major frontal passage in the eastern United States and a hurricane.

The precipitation estimates from the first experiment, which had an RMS discrepancy of 0.015 inches/15min relative to NEXRAD, were used to define nonoverlapping geographic regions centered on each significant precipitation cell. The total estimated precipitation was then integrated in each of these regions, and these rain integrals, along with the areas of each of these regions, were input to a second neural net trained with NEXRAD-based precipitation integrals over the same regions. NEXRAD missed regions where 2.3 percent of the AMSU-based estimated rain fell, and AMSU missed regions where 11.8 percent of NEXRAD-estimated rain fell. After processing using the second neural network, the regional RMS error between AMSU and NEXRAD-based precipitation estimates (m^3s^{-1}) was 2.4 dB.

The second experiment, in which AMSU-A data was used to estimate precipitation, involved pre-processing the Tb’s to remove scan angle dependencies and surface emissivity effects. The scan angle dependency was removed by subtracting cross-scan averages and adding averages at nadir. The surface emissivity effects were removed using Linear Least Squares estimation, based on the AMSU-A window channel, which is most sensitive to the surface. Separately, AMSU-B data at 183±7 GHz was used in combina-
tion with an adaptive threshold technique to define regions of precipitation. These regions in every AMSU-A channel were then set to zero and later filled with two-dimensional interpolation, which allowed for isolation of the effects of rain in AMSU-A. Once these perturbations were separated they were input to a neural net, which trained with NEXRAD data. The net yielded estimates with an RMS error of 0.008 inches/15 min, significantly lower than that of the first experiment. At the regional level, a much better agreement of AMSU and NEXRAD-based rain was achieved, even without the use of a second neural net, such as the one used in the first experiment.

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

First I would like thank professor David Staelin for his guidance and support over the past year, and for letting me be a part of his research team. His knowledge, expertise, ideas and suggestions were invaluable in completing this work.

I would also like to thank Carlos Cabrera and Fred Chen for their contributions and suggestions to many parts of this thesis. I also wish to thank the people at the Remote Sensing Group at the MIT Research Laboratory of Electronics, for their constant support. They made this last year at MIT a very enjoyable experience.

I also owe much to my friends, who helped me through the hardest times, especially Paola Nasser, Nitza Basoco and my best friend Ivan Gonzalez-Gallo. Their unfailing support and unconditional friendship made this semester the most amazing time of my life.

Most of all, and most importantly, I wish to thank my family, Claudia, Rodrigo, Oscar, Pablo, mi Vieja, and Pa, for always being there for me. I would be nothing without them.

Financial support from NASA contract NAS5-31376 and from Lincoln Laboratories purchase order CX-22325 is acknowledged.
Contents

1 Introduction .......................................................... 13
   1.1 Motivation ........................................................................................... 13
   1.2 Problem Statement ............................................................................ 14
   1.3 Prior Work ......................................................................................... 16
   1.4 Outline of the thesis ........................................................................... 17

2 Background Information ............................................ 19
   2.1 Data sets used for this study ............................................................... 20
      2.1.1 Advanced Microwave Sounding Unit (AMSU) ......................... 20
      2.1.2 Next Generation Radar (NEXRAD) ........................................... 26
   2.2 Mathematical tools used for this work .............................................. 28
      2.2.1 Linear Least Mean Squares Estimation .................................. 28
      2.2.2 Neural Networks ........................................................................ 30
      2.2.3 Spatial interpolation of Satellite and Radar data .................... 32
      2.2.4 Interpolation of satellite data into isometric coordinates ............ 33
   2.3 Software and Hardware Resources .................................................. 37

3 Precipitation estimation algorithms and implementation .................... 39
3.1 Description of the data sets using for training and testing.........................40

3.2 Precipitation measurements using 183 GHz observations..........................43
   3.2.1 Basis for precipitation estimation from data at 183 GHz ..................43
   3.2.2 Neural Net implementation......................................................45
   3.2.3 Regional corrections of the precipitation estimate........................47
   3.2.4 Precipitation estimation error..................................................49

3.3 Precipitation measurements using 54 GHz observations..........................51
   3.3.1 Basis for precipitation estimation from data at 54GHz ..................51
   3.3.2 Scan angle dependence ..........................................................55
   3.3.3 Surface emissivity .................................................................57
   3.3.4 Development of 54GHz precipitating cloud perturbations...............60
   3.3.5 Neural net implementation.......................................................64
   3.3.6 Regional corrections..............................................................66
   3.3.7 Precipitation estimation error..................................................66

4 Conclusion

4.1 Thesis achievements .........................................................................69

4.3 Future research possibilities..............................................................70

Appendix A. MATLAB implementation of satellite image interpolation algorithm73

Appendix B. MATLAB implementation of region separation algorithm .........80

Appendix C. MATLAB implementation of 54 GHz data pre-processing and
neural net

Appendix D. Output for Hurricane Georges, Sep 30, 1998 ........................89

Bibliography

91
List of Figures

Figure 2.1. (a) AMSU scheme for measuring brightness temperature. (b) Window channels for AMSU-A and AMSU-B. ........................................ 24

Figure 2.2. AMSU-A and AMSU-B data for same date as in Figure 2.1. (b) AMSU-A, 52.8 GHz, 53.596 GHz, 54.4 GHz, 54.9 GHz. AMSU-B, 183 ± 1 GHz, 183 ± 7 GHz. ........................................ 25

Figure 2.3. NEXRAD image corresponding to precipitation accumulation over 15 min. ............................................................................................ 27

Figure 2.4. Neural net with five inputs, and one output. ........................................ 30

Figure 2.5. Region of interpolation where the image fc(x,y) is obtained from the closest four pixels in the original image. (From Lim, 1990).......33

Figure 2.6. Horizontal distance from nadir in a satellite measuring distance. Satellite path is perpendicular to drawing. (a) Description. (b) Geometric Representation. ................................................................. 35

Figure 2.7. (a) Horizontal and vertical projection of each satellite spot onto a horizontal plane representing the surface of the earth. (b) Positioning of the spots in a isometric grid .................................................. 37

Figure 3.1. Data sets used for training/testing. ......................................................... 41

Figure 3.2. AMSU-B images for rainband structure (October 7, 1998). (a) Window channel, (b) 183 ± 1 GHz, (c) 183 ± 7 GHz, (d) NEXRAD radar data. ......................................................... 44
Figure 3.3. Nonlinear relationship between brightness temperatures from the two 183 GHz channels used to train the neural net and the NEXRAD rain rate. .......................................................... 46

Figure 3.4. Data for a cold front, Oct 7, 1998. (a) NEXRAD precipitation data, lowpass filtered to 15km. (b) Precipitation estimates using neural network. (c) Region definition for rainmap in (b). ....................... 48

Figure 3.5. Scheme for estimating regional precipitation described in Section 3.1. Both neural nets are included in this diagram. ........................................ 49

Figure 3.6. (a) Performance of the first neural net. (b) Performance of the system when the second neural net is added. ........................................ 50

Figure 3.7. AMSU-A Temperature sounding channels used for the precipitation estimation algorithm with NEXRAD data........................................... 54

Figure 3.8. Illustration of the scan-angle dependency and its correction. .......... 56

Figure 3.9. Reduction of surface effect in data from channel at 52.8 GHz. .......... 59

Figure 3.10. Flowchart showing algorithm to obtain adaptive threshold .......... 62

Figure 3.11. Precipitating cloud perturbations. (a) AMSU-B 183 ±7 GHz. (b) Region where (a) falls below adaptive threshold. (c) Cloud perturbations using (b) on 52.8 GHz. ............................. 63

Figure 3.12. Pre-processing of AMSU-A imagery prior to input to Neural Network. .. 64

Figure 3.13. Performance of the Neural Net estimator for Cold Front. (a) NEXRAD precipitation data, lowpass filtered to 15km. (b) Precipitation estimates using neural network. (c) Region definition for rainmap in (b). .............................. 65

Figure 3.14. Performance of neural network. (a) Output, compared to NEXRAD precipitation data at a pixel level. (b) Regional precipitation integrals, compared to those of NEXRAD. x’s correspond to training set, o’s to validation set. ................................................................ 67

Figure D.1. Output of precipitation estimator with data from Hurricane Georges, Sep 30, 1998 ................................................................. 90
List of Tables

Table 2.1  AMSU channels used for this work ................................................. 23
Table 3.1  Data sets used for training, testing and validation ....................... 42
Table 3.2  AMSU-A temperature sounding channels used to estimate
            precipitation ...................................................................................... 53
Chapter 1

Introduction

This chapter outlines the motivation driving the research, discusses the prior work done in the research area, and provides an outline for the thesis.

1.1. Motivation.

Rain provides much of the water required for human existence, and it is without a doubt one of the environmental events with greatest impact on human life. Rain variations can cause floods and droughts, which can deeply affect the economy and the quality of life. Accurate assessment of precipitation is necessary for improved forecasting, and understanding of weather variability and the hydrologic cycle.

The first measurements of precipitation were made centuries ago using rain gauges, and this method is still in use today. However, measurement by rain gauges, like other ground based methods, provides only with scattered samples of rain and is certainly not sufficient for global rainfall estimation. The use of radar now enables for rain observations over large areas, but coverage is still limited to land, which constitutes only about
30% of the terrestrial surface.

Remote sensing from space, a technology currently under rapid growth, would allow for global precipitation estimates by recording radiant energy from space use it to estimate precipitation. In this scheme, an orbiting satellite records energy at wavelengths that range from the visible to the microwave range. They have all been utilized to estimate rain, but recent estimates come from microwave observations, which penetrate clouds and interact with water droplets. This comes as a consequence of the empirical relationship of precipitation and temperature at different levels of the atmosphere, which are available through microwave remote sensing.

1.2. Problem Statement.

The humidity and temperature profile channels of the Advanced Microwave Sounding Unit are known to respond in part to precipitation. In this work, the brightest temperatures from AMSU-A at four center frequencies and AMSU-B at two center frequencies were combined to yield precipitation estimates over the continental United States. In order to do this, precipitation was assumed to be highly related to atmospheric humidity and temperature at different altitudes. This assumption is confirmed by looking at ground based measurements and their corresponding brightness temperatures measured from space. Moreover, it was assumed that information from four AMSU-A channels and two AMSU-B channels were sufficient to produce reasonably accurate precipitation estimates.

This thesis addresses the possibility of creating reasonably accurate weather maps
from satellite observations through the following questions:

- Can rain be estimated from microwave brightness temperatures measured through passive remote sensing?
- If so, what set of brightness temperatures gives the best estimate?

Instead of calculating precipitation using the physical relationships between rain and atmospheric brightness temperatures and humidity, a neural net was trained to learn them based on NEXRAD rain estimates. The choice of inputs for the learning process of the network, and for improving the accuracy of the rain estimate, is one of the main issues addressed in this thesis.

The desired features of this estimator can be summarized as follows:

- The estimator must be able to understand information from multiple variables (inputs) that are assumed to be related to precipitation (in this case, the brightness temperatures).
- Moreover, the estimator must be able to perform mapping of the output, based on the input variables, in a two dimensional space.
- Lastly, the estimator must be able to process large amounts of data, to produce global precipitation estimates.
1.3. Prior work.

Estimation of surface rainfall from data obtained using remote sensing started many years ago with the use of brightness temperatures at visible and microwave frequencies. One approach has been to relate individual cloud brightness temperature to rain in order to obtain pixel-wise estimates; other approaches are based on analysis of cloud type and their change in time. These methods constituted the first attempt to use remote sensing to estimate precipitation, but have in general limited accuracy and resolution in time and space (Hsu, et al, 1997).

Moreover, precipitation has been estimated from imagery at infrared and microwave frequencies, and in general these methods work by choosing inputs from which precipitation can be modeled with the least amount of uncertainty. Grody (1998) has developed algorithms that can yield precipitation estimates from data at microwave frequencies based on a single model for precipitation formation over land.

A relatively new and promising approach of solving the physical problem of precipitation formation is the use of artificial neural networks. Neural networks have been used in a wide range of applications, such as cloud classification, temperature retrievals, etc. Zhang and Scofield (1994) made the first attempt to estimate heavy convective rainfall and recognize clouds using neural nets. Hsu, et al (1997) estimated precipitation using neural networks trained with infrared imagery, but the limitations in resolution and coverage of their input data limited the research to the Florida peninsula and Japan. In addition, Tsintikidis, et al (1997) estimated rainfall using neural nets and brightness temperatures from the Special Sensor Microwave Imager (SSM/I) at several frequencies, and reported a significant reduction in the error as compared with estimates using linear regression.
The purpose of this paper is to apply neural networks to perform precipitation retrieval from AMSU remotely sensed data. Section 1.4 describes the outline of the thesis.

1.4. Outline of the thesis.

Chapter 2 provides some basic background information relevant for this thesis, including a description of the data sets and the mathematical tools that were utilized for this work; Chapter 3 describes the two algorithms that were used to implement the rain estimator; and Chapter 4 summarizes the result of this research, its contributions, and makes suggestions for future research.
Chapter 2

Background Information

In this chapter, both the data sets used for the study and the mathematical models used to manipulate them will be explained in a general way. The first subsection contains a general explanation of the data sets utilized for the estimation of rainfall. There are two sources of data for the study: a set of brightness temperatures for the estimation of precipitation, and a set of measured precipitation for the same region used for validation of the estimate. The brightness temperatures are measurements from the Advanced Microwave Sounding Unit (AMSU) on the NOAA-15 polar orbiting satellite, and the precipitation measurements originate from the NOAA Next Generation Radar (NEXRAD) network.

In order to obtain a more visually pleasing and accurate rainfall estimate using the above data sets and the procedures explained later in this paper, the brightness temperatures were interpolated to obtain results in consistent coordinates. This interpolation scheme is discussed in this chapter, in a separate section dedicated to mathematical models, together with the basic concepts of Least Squares Linear estimation and Neural Networks, which were used to estimate precipitation.
2.1. Data Sets used for the study.

2.1.1. Advanced Microwave Sounding Unit (AMSU).

The Advanced Microwave Sounding Unit is one of the latest instruments designed to sound terrestrial atmospheric temperature and moisture at different altitudes. This sounding is done passively, that is, a receiving device records radiant energy emitted spontaneously by a source, which in this case is the troposphere. The signal is usually the brightness temperature and its value depends on the physical characteristics of the atmosphere (temperature, composition, clouds, precipitation, humidity, etc.), and the scanning angle. The measured energy is located in the microwave region of the spectrum, as opposed to the infrared, which has been used by other sounding devices. Microwave measurements are more accurate in cloudy regions where infrared instruments have limited reliability, because at microwave frequencies the instrument can effectively see through many kinds of clouds. Currently there are two radiometers which together constitute the Advanced Microwave Sounding Unit installed in NOAA-15, which was launched on May, 1998: AMSU-A and AMSU-B.

AMSU-A is a 15-channel temperature cross-track sounder with a resolution of 48km at nadir. Each scan consists of 30 measurements (separated by 3.333 degrees as seen by the instrument), with one whole scan completed every 8 seconds. Most of the 15 channels have a specific bandwidth and center frequency, while some have individual side-bands. Channels 1 and 2 are window channels, which are sensitive to the terrestrial surface and measure humidity and atmospheric liquid water over oceans. Moreover, channels 4-14 sound the atmospheric temperature from the surface of the earth to an alti-
tude of approximately 2mbar (Staelin, 1989). Each channel is particularly sensitive to temperatures at a certain altitude, and each channel responds best to temperatures in an individual layer of the atmosphere which is from 5 to 10km thick. (Saunders, Hewinson, Stringer, Atkinson, 1995). For example, the more opaque frequencies respond principally to atmospheric temperature at the top of the atmosphere. Table 2.1 contains a summary of the relevant channels used for this work, and their individual applications. Examples of images from data for each of these frequencies are shown in Figure 2.2.

AMSU-B scans the atmosphere at the same time as AMSU-A, but with a better resolution (16 km at nadir). Every scan for AMSU-A is done while three scans for AMSU-B are performed, giving 9 cells of AMSU-B brightness temperatures for each AMSU-A cell. In other words, each scan provides 90 measurements, or earth views. This instrument completes one scan every $8/3$ seconds. AMSU-B contains one window channel and four channels with center frequencies near 183 GHz, which is a resonance frequency of water vapor. For this reason, these four channels serve as moisture sounders. Sounding at these frequencies seems to be unaffected by surface effects and thus, in combination with temperature profile retrievals, these channels can provide water vapor profiles over land or sea. Moreover, low level clouds have little effect in the vicinity of 183 GHz because the water vapor level absorbs most of the radiation above the top of the clouds (Burns, Diak, 1997). This is not the case for high level clouds, for which there is absorption by water vapor and other water particles (hydrometeors), causing brightness temperature deviations that could go all the way to 100K (Adler et al, 1997). For this reason these channels are used to locate and estimate precipitation. The AMSU-B channels
that were used for this work are summarized in Table 2.1, with corresponding images in Figure 2.2. Deficient antennas and RF shielding have led to radio interference that contaminates AMSU-B data sufficiently that only 183 ± 1 and 183 ± 7 GHz data prior to October 13th, 1998 is quantitatively analyzed here. Even before this date these two channels have small angle-dependent biases, which were ignored in this work.
<table>
<thead>
<tr>
<th>Channel</th>
<th>Center Freq (GHz)</th>
<th>Altitude of weighting function peak (km)</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMSU-A</td>
<td>Ch 4 52.8</td>
<td>&lt;1</td>
<td>Atmospheric Temperature Profiles</td>
</tr>
<tr>
<td></td>
<td>Ch 5 53.596 ±0.115</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ch 6 54.4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ch 7 54.9</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>AMSU-B</td>
<td>Ch 16 89</td>
<td>Function of water vapor burden</td>
<td>Window channel(surface effects, precipitation, tropical humidity)</td>
</tr>
<tr>
<td></td>
<td>Ch 18 183.31±1</td>
<td>(variable with location)</td>
<td>Atmospheric water vapor profiles, precipitation, humidity profiles, clouds</td>
</tr>
<tr>
<td></td>
<td>Ch 20 183.31±7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1. AMSU channels used for this work.
Figure 2.1. (a) AMSU scheme for measuring brightness temperature. (b) Window channels for AMSU-A and AMSU-B. Both figures are from data at 89GHz but at different resolutions. Figure corresponds to Hurricane Georges, which struck Florida and the southern US on September 28, 1998.
Figure 2.2. AMSU-A and AMSU-B data for same date as in Figure 2.1. (a) AMSU-A, 52.8 GHz. (b) 53.596 GHz. (c) 54.4 GHz. (d) 54.9 GHz. (e) AMSU-B, 183 ± 1GHz. (f) 183 ± 7GHz. Note resolution of AMSU-B data, and angle dependence of AMSU-A data.
2.1.2. Next Generation Weather Radar (NEXRAD).

Precipitation is measured on a global scale nowadays using radars. Radars detect the presence and location of objects by emitting electromagnetic waves which the objects reflect. The time it takes for these waves to come back to the radar is used to determine the distance to the object. In the case of meteorological measurements, these objects could be water or ice particles, dust, clouds, etc. Most pre-NEXRAD radars, which were used widely in the US and are still being used in many countries, scan only at low elevation angles, which means that weather phenomena can only be measured up to certain altitudes (NOAA, 1998). On the other hand, NEXRAD radars have the advantage that they can detect objects within a large coverage area, and with great detail. In fact, NEXRAD has a volume coverage pattern that extends all the way up to 70,000 feet (Piel, 1995).

However, one radar can only detect meteorological phenomena within a radius of approximately 230 km and several radars are needed to cover extensive regions. In fact, national coverage is provided by 138 radars, which were all installed between 1990 and 1996 (NOAA, 1998). Other countries, such as Korea and Japan, have also made use of this technology in the last 5 years.

The data from these radars is used to develop more than 60 products with a wide range of applications. For instance, they are used to create rainfall maps with great resolution, better than that of most conventional systems. NEXRAD gives detailed images of intensity, location, and movement of precipitation systems, such as the one in Figure 2.3, which has a spatial resolution of $2 \text{Km}^2$. As explained before, this data will be used to validate the precipitation estimator described in this paper.
Figure 2.3. NEXRAD image corresponding to precipitation accumulation over 15 min. Image corresponds to Hurricane Georges, for the same day as in Figure 2.2. Note the spatial resolution of the data.
2.2. Mathematical tools used for this work.

2.2.1. Linear Least Mean Squares Estimation.

One of the most used techniques in probabilistic analysis and estimation is Least Squares Estimation (LSE). According to this method, (Bertsekas, Tsitsiklis, 1999) if $X$ is a random variable and $\hat{X}$ is an estimate (or prediction) of the random variable, and the quality of the estimate is quantified in terms of the mean squared error

$$\text{MSE} = \mathbb{E}[(X - \hat{X})^2]$$  \hspace{1cm} (2.1)

then the value of the estimator that minimizes this error is

$$\hat{X} = \mathbb{E}[X]$$  \hspace{1cm} (2.2)

Suppose now that an experimental value $y$ is observed, of some random variable $Y$, which is correlated to $X$. This additional information can be exploited by using (2.2), conditioned on the observation $Y = y$. In other words, the conditional mean error

$$\text{MSE} = \mathbb{E}[(\hat{X} - X)^2 | Y = y]$$  \hspace{1cm} (2.3)

is minimized using the estimator

$$\hat{X} = \mathbb{E}[X | Y = y].$$  \hspace{1cm} (2.4)

The resulting estimate $\hat{X}$ depends on the observed value $y$. The estimate in (2.4) is called the least-squares estimate of $X$ given the experimental value $y$.

Moreover, if an estimator taking the form of a linear function of the random variable $Y$ is desired, in other words, an estimator of the form

$$\hat{X} = aY + b$$  \hspace{1cm} (2.5)

then, in order to find $a$ and $b$ that minimize the minimum squared error in this case, which
is equal to

\[ \text{MSE} = \mathbb{E}[(X - aY - b)^2] \]  \hspace{1cm} (2.6)

it can be supposed that \( a \) has already been chosen. In order to choose \( b \), it is useful to note

that this situation corresponds to having to choose a constant \( b \) to estimate the random variable \( aX - Y \). Using (2.2), the best choice for estimator is to let

\[ b = \mathbb{E}[X - aY] = \mathbb{E}[X] - a\mathbb{E}[Y] \]  \hspace{1cm} (2.7)

In order to minimize the minimum squared error, the expression

\[ \text{MSE} = \mathbb{E}[(X - aY - \mathbb{E}[X] + a\mathbb{E}[Y])^2] \]  \hspace{1cm} (2.8)

must be minimized with respect to \( a \). Expanding the quadratic term above, the following expression is obtained

\[ \text{MSE} = \sigma_X^2 + a^2 \sigma_Y^2 - 2a \cdot \text{cov}(X, Y) \]  \hspace{1cm} (2.9)

This quadratic expression with respect to \( a \) is minimized when the derivative is zero, which occurs at

\[ a = \frac{\text{cov}(X, Y)}{\sigma_Y^2} \]  \hspace{1cm} (2.10)

With this choice of \( a \), the linear, mean squares estimate is given by

\[ \hat{X} = \mathbb{E}[X] + \frac{\text{cov}(X, Y)}{\sigma_Y^2}(Y - \mathbb{E}[Y]) \]  \hspace{1cm} (2.11)

with a mean squared error equal to

\[ \text{MSE} = \sigma_X^2 + a^2 \sigma_Y^2 - 2a \cdot \text{cov}(X, Y) \]  \hspace{1cm} (2.12)

In Chapter 3, this method is used to remove unwanted perturbations in the data given some prior observations of the data that caused the perturbations.
2.2.2. Neural Networks.

An artificial neural network is a computer model of an input/output process composed of basic units called nodes. These nodes could be connected in several fashions; nodes are organized in layers and are interconnected with other nodes in the same or in other layers. The network usually consists of an input layer, one or more hidden layers, and an output layer. Figure 2.4 depicts an example of a neural net.

![Neural Net Diagram](image)

Figure 2.4. (a) Neural net with five inputs, and one output. The net has two hidden layers and a single output. Here the inputs are the brightness temperatures from different channels and the output an estimation of the precipitation.

The transformation of the input values from the input layer to the hidden layer is continuous and nonlinear. The weights are the parameters for this transformation. For each hidden layer there is another transformation of the data, and finally, the outputs are generated by linearly transforming the outputs of the next-to-last layer to the output layer.
in a similar way, with each transformation having its own set of weights.

Neural networks need to be trained to process the inputs in order to use them in applications. The network can be trained to learn two ways: supervised and unsupervised mode (Haykin, 1994). The supervised mode consists in adjusting the input weights on the nodes in the network such that the output approximates the desired output as much as possible. This would mean that for every set of inputs, the desired output must be known in advance for the training process.

The most used technique for supervised learning is backpropagation. This algorithm uses a gradient descent formula in which the mean square error between the network output and the desired output is minimized (Tsintikidis, 1997). The algorithm keeps track of the error at the output layer and minimizes it by backpropagating to the input layer. The network can be made to keep track at the derivative of the error as well: this measures how much the error changes as the weights are varied. Ideally, the goal is to obtain zero errors; in reality, the network is considered trained after the error is less than a set threshold or the training function has been applied a given number of times. Once the network is trained, it can be used to predict output values for which no desired response is known.

Neural networks are good in the sense that they do not require specific prior knowledge about the process being modeled. In this case, the physics that governs the relationship between brightness temperatures at different altitudes and precipitation would simply be learned by the net. In addition, precipitation would be estimated from relatively instantaneous snapshots of brightness temperatures, without knowledge of history of the meteorological event. Moreover, the net should give a consistent estimate of the precipitation for brightness temperatures all over the world, for which satellite data is available.
Lastly, neural nets could efficiently learn the nonlinear relationship between brightness temperature and precipitation and quickly handle global data. The neural network structure has been mathematically proven to be an universal function approximator which can handle nonlinear functions very accurately (Hsu et al, 1996).

2.2.3. Spatial interpolation of Satellite and Radar data.

One of the problems when developing a pixel-based precipitation estimator using the AMSU satellite data described in Section 1.1 is that AMSU-A, AMSU-B and NEXRAD data based images do not have the same resolution. For this reason, some common resolution must be established before the data can train or be processed by a neural net. When choosing this resolution, the main issues involved are the size of the data and the amount of information lost with interpolation.

The resolution of AMSU-B (~15 Km at nadir) produces very acceptable images, with an acceptable data set size. For this reason, both the AMSU-A and NEXRAD sets were represented using this resolution. In the case of NEXRAD, translation simply involved a mapping to the AMSU-B coordinate system, since resolution of NEXRAD is much greater. However, AMSU-A resolution is lower (~50Km at nadir), so some sort of interpolation is needed to translate the data to the new resolution.

In order to implement this resolution adjustment, AMSU-A data was interpolated using bilinear interpolation. In this method, a new image \( f_c(x,y) \) is obtained from the original image \( f[n_1,n_2] \) by a linear combination of the four closest pixels of the original image to every new location on the new image (Lim, 1990):
\[ f_c(x, y) = (1 - \Delta_x)(1 - \Delta_y)f(n_1, n_2) + (1 - \Delta_x)\Delta_yf(n_1, n_2 + 1) \]
\[ + \Delta_x(1 - \Delta_y)f(n_1 + 1, n_2) + \Delta_x\Delta_yf(n_1 + 1, n_2 + 1) \]  

(2.13)

In this expression,
\[ \Delta_x = (x - n_1), \quad \Delta_y = (y - n_2) \]  

(2.14)

This relations can be understood by looking at figure 2.5 (from Lim, 1990). Once AMSU-A images are interpolated (by a factor of 3 on each dimension) they will have the same size and pixel resolution as AMSU-B images, and they will be easier to handle for the neural net precipitation estimator.

\[ \text{Figure 2.5. Region of interpolation where the image } f_c(x, y) \text{ is obtained from the closest four pixels in the original image. (From Lim, 1990).} \]

2.2.4. Interpolation of satellite data into isometric coordinates.

Once the satellite data (AMSU-A and AMSU-B) have the same resolution, they must be interpolated to remove artifacts due to the position and velocity of the satellite high in the atmosphere, as well as the curvature of the earth. However, although this interpolation still obeys (2.13) and (2.14), the resolution of the original images is not uniform across the image. As it will be explained in this section, satellite images have a better res-
olution closer to nadir due to the finite satellite altitude and the earth's curvature. Since the algorithm operates on images with nonuniform resolution and makes it uniform, it could not be implemented with conventional software packages such as the image processing toolbox in MATLAB.

In order to develop an interpolation scheme for AMSU satellite data, it is necessary to understand which geographic location on the surface of the earth corresponds to each spot in the measurement instrument. In order to do this, each single spot in the AMSU-B data was assigned an x and y coordinate with respect to the first spot in the first scan. Figures 5 and 6 clearly illustrate this concept.

The horizontal location on the surface of the earth of each spot depends exclusively on the scan angle for that particular spot. To prove this, we can start by representing the measuring instrument as a geometric system such as the one shown in Figure 5b. It can be shown with simple vector algebra that the quantities represented in the Figure can be obtained from

\[ z = (r+h)\cos \alpha - \sqrt{r^2 - \sin^2 \alpha \cdot (r+h)^2} \]  

\[ \sin \beta = \frac{z \cdot \sin \alpha}{r} \]  

Which gives

\[ \beta = \arcsin [(r+h)\sin \alpha \cos \alpha - \sin \alpha \sqrt{r^2 - \sin^2 \alpha \cdot (r+h)^2}] \]  

Since the radius of the earth (r) and the satellite altitude (h) are known quantities, the horizontal distance from nadir \( r\beta \) depends only on the scan angle from nadir, here named \( \alpha \). This angle is a known quantity for every spot for AMSU systems (as mentioned in Section 3).
To calculate the vertical location of each spot, (corresponding to the distance along an axis parallel to the path of the orbit), it was noted that if the time between two single measurements is given, and the vertical velocity of the satellite is known (obtained from the orbital period) the vertical displacement with respect to the surface of the earth of two successive spots is

\[
\Delta_y = \text{measurement period} \cdot \text{velocity} \quad (2.18)
\]

\[
\Delta_y = \text{measurement period} \cdot \frac{2\pi r}{\text{orbital period}} \quad (2.19)
\]

Using (2.15) - (2.19), every single spot in an AMSU-based image can be assigned an (x,y) coordinate in a plane corresponding to the terrestrial surface. using this technique, and overlapping a grid with the desired resolution, gives an idea of the actual places on the surface that correspond to each AMSU measurement, as shown in Figure 2.6.

Figure 2.6. Horizontal distance from nadir in a satellite measuring distance. Satellite path is perpendicular to drawing. (a) Description. (b) Geometric Representation.
Once this translation of coordinates was done, a specialized bilinear interpolation procedure was implemented to produce a new pixel in every location in the new grid, resulting from a linear combination of the four closest locations of AMSU measurements (marked with ‘x’ in Figure 2.7). This interpolation algorithm was coded in MATLAB, (see Appendix A) using equations (2.13) and (2.14) and basically works as follows:

- Create an isometric grid, with the desired resolution, with each point given a (x’,y’) coordinate.
- Take a point in this new spatial grid.
- Locate the four closest neighbors from the set of AMSU (x,y) locations and calculate the separation distance.
- Add the distances and normalize each separation by this total. Store quantities on disk.
- Repeat this process until the next row of AMSU (x,y) locations is reached.
- Store all the coefficients on disk. Since there is a regular pattern in the AMSU scan lines, this process is not repeated for every one of them.
- Produce a linear combination of the four closest pixels to every point in the grid, where each coefficient is the normalized distance stored on disk. Produce the output image by repeating this process.
Figure 2.7. (a) Horizontal and vertical projection of each satellite spot onto a horizontal plane representing the surface of the earth. (b) Positioning of the spots in a isometric grid where each pixel is obtained through bilinear interpolation.

2.3. Software and Hardware Resources.

The work in image processing and neural networks was done in a Digital Alpha workstation. This machine operates under Digital UNIX and has 256 Megabytes of RAM. All the operations were performed with MATLAB version 5.2 for UNIX. The neural networks and image processing toolboxes in this software package were used extensively.
Chapter 3

Precipitation estimation algorithms and implementation

In this chapter, two different approaches for the estimation of rain from brightness temperature measurements are explained in detail. The core of each algorithm consists of using neural nets to estimate precipitation, and perform some regional corrections. The neural nets are trained to learn the relationship between brightness temperature and rainfall and these operations are performed in a pixel-wise manner across the data images. The main difference between the two rain estimation approaches explained in this chapter is the data sets used to train the neural net.

Brightness temperatures from passive microwave measurements in the atmosphere can be used to estimate rain because precipitating clouds produce significant contamination against an otherwise relatively uniform background. This effect is in general different over land and over sea (Staelin, Chen, Fuentes, 1998). Over land, precipitating clouds produce brightness temperatures that are colder than the surroundings, with depressions of 100K observed for strong convective systems (Adler, et al, 1990). Over ocean, precipitat-
ing clouds appear warmer than the ‘cold’ sea background.

The first implementation of the neural net precipitation estimator was fed with imagery with brightness temperatures from AMSU-B at $183 \pm 1$ and $183 \pm 7$ GHz. In the second approach, four more inputs were added, resulting from information from AMSU-A imagery at several 54 GHz frequencies. In both cases a regional correction algorithm is implemented using a second neural net; and in both cases the nets are validated with precipitation data from the Next Generation Radar (NEXRAD).

3.1. Description of the data sets using for training and testing.

The neural nets were trained and tested using four data sets for which both AMSU and NEXRAD data was available. They all correspond to weather events with a significant amount of precipitation in the continental United States that occurred in 1998. Unfortunately, interference has degraded measurements taken after October 13th, 1998, so more recent data files were not used. NEXRAD images were obtained for the same day and approximately the same time as the scan time for every convective system. However, NEXRAD data comes in images that correspond to snapshots that are taken every 15 minutes, so there is an eventual time offset between NEXRAD and AMSU that could go all the way up to seven minutes. However, the maximum offset for the data sets used in this study was around 6 minutes. Assuming that large rain cells change and move relatively slowly across the land, this offset should not have a significant influence on the correlation of the radar and satellite data.

This data sets used for training and validation are detailed in Table 3.1, with corresponding NEXRAD images in Figure 3.1.
Figure 3.1. Data sets used for training/testing. Refer to Table 3.1 for data set details. (a-b) Hurricane Georges. (c) Cold Front. (d) Storm over Midwest.
<table>
<thead>
<tr>
<th>Event</th>
<th>Aprox Area</th>
<th>Date (1998)</th>
<th>UTC NEXRAD Snapshot (15 min avg) endtime</th>
<th>UTC First AMSU Scan</th>
<th>MAX time offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hurricane Georges (1)</td>
<td>140,000 Km²</td>
<td>Sept 30</td>
<td>00:45</td>
<td>00:39</td>
<td>6 min</td>
</tr>
<tr>
<td>Hurricane Georges (2)</td>
<td>160,000 Km²</td>
<td>Sept 28</td>
<td>13:45</td>
<td>13:43</td>
<td>2 min</td>
</tr>
<tr>
<td>Cold Front</td>
<td>110,000 Km²</td>
<td>Oct 7</td>
<td>13:45</td>
<td>13:41</td>
<td>4 min</td>
</tr>
<tr>
<td>Storm Over Midwest</td>
<td>280,000 Km²</td>
<td>Jun 11</td>
<td>13:30</td>
<td>13:26</td>
<td>4 min</td>
</tr>
</tbody>
</table>

Table 3.1. Data sets used for training, testing and validation.
3.2. Precipitation measurements using 183 GHz observations.

3.2.1. Basis for precipitation estimation from data at 183GHz

As explained in Chapter 2, low-level clouds have little effect on channels around the 183.31-GHz absorption line. Since microwave radiometry at these center frequencies is able to penetrate low level clouds, precipitation estimations should be more accurate than when measurements from other frequencies are used. On the other hand, high-level clouds over land do contaminate the brightness temperatures because of increased absorption by water vapor within the cloud and the scattering and absorption by water droplets and ice (Burns, Wu, Diak, 1997). These contaminations, which cover completely most precipitation cells observed in NEXRAD data, appear as lower-than-nominal brightness temperatures. Other factors which affect this absorption are particle size, altitude, etc., but their sole presence has the most significant influence on brightness temperature (Grody, 1993).

For this reason, the information for these channels was used to attempt to estimate rainfall. The underlying physical relationships between the brightness temperatures and the rain rate are left for the neural net model to learn. In order to obtain a better estimation of precipitation, two inputs were used, corresponding to data from AMSU-B channels with frequency bands at 183 ± 1 and 183 ± 7 GHz. These two channels seem to be the most affected by precipitation, and moreover, comparison of raw data indicates that 183 ± 7 GHz seems to respond better to the presence of rain, specially for regions with low rain rates and small precipitation systems. On the other hand, 183 ± 1 GHz responds better in the presence of convected ice particles high in the atmosphere, signaling strong precipitating cells; for this reason has the brightness temperatures have lower nominal
value than at 183 ± 7 GHz (Staelin, Chen, Fuentes, 1998). These concepts are shown in Figure 3.2. It displays a typical set of satellite images with the corresponding radar precipitation measurements.

Figure 3.2. AMSU-B images for rainband structure (October 7, 1998). (a) Window channel, (b) 183 ± 1 GHz, (c) 183 ± 7 GHz, (d) NEXRAD radar data. (b) and (c) were the inputs to the neural net; (d) was used as target.
3.2.2. Neural Net implementation.

Initial examination of the brightness temperatures at $183 \pm 1$ and $183 \pm 7$ GHz reveals that their relationship with rain is not linear. Figure 3.3, together with Figure 3.2, indicates that $183 \pm 7$ GHz responds more strongly to rain, in the sense that rain cells can be clearly distinguished as cold spots from relatively uniform warm backgrounds. $183 \pm 1$ GHz, on the other hand, responds to other factors in the atmosphere, like vertical velocities that push humidity upward but at a rate that ultimately does not result in rain. Nevertheless, the information in these two channels seems to be sufficient to make a reasonably good estimate of rain. The non-linear relationships that relate brightness temperature and precipitation (Figure 3.3) are to be simulated using a neural net. The artificial neural network described here basically produces an equation that corresponds to a surface in the Tb-Tb-Rain space, that fits the data in Figure 3.2 with minimum error.

This network was feedforward and had two inputs (brightness temperatures) corresponding to the channels named above, which seem to convey all the necessary information. Since AMSU-B does not seem to have a strong scan angle dependence, the scan angle was not included for this part of the study. However, the second part of this Chapter describes an algorithm in which other AMSU channels (namely from AMSU-A) are utilized in order to increase the number of degrees of freedom (Section 3.2), and where the scan angle plays an important role.

Besides having two inputs, the network had four input nodes, four hidden nodes, and one linear output node. The network was trained using NEXRAD 15-min average precipitation data (in mm/hr). In addition, this network was trained using both a training and a validation vector which had several thousand elements, corresponding to pixels in
two separate sets of images. Of those, approximately 15% were from precipitating cells. The training and validation data sets consisted of two separate kinds of precipitating clouds: a hurricane (Hurricane Georges, which corresponds to images in Chapter 2) and a large rainband across the United States (Figure 3.2).

Figure 3.3. Nonlinear relationship between brightness temperatures from the two 183 GHz channels used to train the neural net and the NEXRAD rain rate.

Figure 3.4b shows the neural net output for this given input. Most precipitation cells were detected by the neural net, within the resolution limitations of AMSU data, for both the training and the validation set and other test sets. However, the network missed around 10% of the rain detected by NEXRAD, and NEXRAD missed around 3% of the rain cells detected by the neural net, as discussed in the next section.
3.2.3. Regional corrections of the precipitation estimate.

Comparison of the NEXRAD and the AMSU-based generated rain map shows that there are some differences in structure and precipitating rate. The differences in cell shape and area are mainly due to the fact that while NEXRAD is more sensitive to water and large ice particles, AMSU is more sensitive to small ice particles (Staelin, Chen, Fuentes, 1998). These ice particles show up in AMSU images when being pushed aloft high in the atmosphere, but show up in NEXRAD images when in the form of large hydrometeors. These two processes are not centered in the same place, (they are usually offset by a distance that could go all the way up to tens of kilometers) and for this reason the two precipitation maps are offset. This motivates the use of some sort of integration or estimation of the total rainfall for a whole convective cell, regardless of its position, and comparison of those rates for the same cells in NEXRAD. This is why the rain maps were corrected at a regional level.

Regionally integrated precipitation rates were calculated by first separating the rain maps which were the output of the first neural net into separate regions. In order to do this, the center of the strongest precipitating cells, which were located using a single threshold, were assigned boundaries, and these were expanded until they reached the boundaries of another region. The boundaries between two regions had in general a lower precipitation rate that was fairly constant across the boundaries, so this method for division works well in the sense that small variations in the shape of the regions did not produce significant changes in the regional precipitation rate. Figure 3.4c shows the result of such division, and Appendix B shows its MATLAB implementation.

The same approach was used to divide NEXRAD precipitation maps into regions.
The regional estimates were used to train a second neural net having two inputs: the total area of each region, and the regional precipitation rate. The net was trained to target the NEXRAD regional estimates (m³sec⁻¹). This neural net had two input neurons, one hidden layer, consisting of only one neuron, and one linear output neuron. This network effectively improves the accuracy of the precipitation estimate at a convective-cell level. Figure 3.5 shows how the entire process works.

Figure 3.4. Data for a cold front, Oct 7, 1998. (a) NEXRAD precipitation data, lowpass filtered to 15km. (b) Precipitation estimates using neural network. (c) Region definition for rainmap in (b). There are 17 regions in this case.
3.2.4. Precipitation estimation error.

The initial neural net created reasonably accurate rain maps that had an RMS error equal to 0.015 inches/15min. This performance is quite reasonable at the convective cell level. As explained before, the regional convective cell shape discrepancy between AMSU and NEXRAD based estimates accounts significantly for this error and is due in large measure to the time offset between the two data sets. When the cells were compared, there was 11.8% of the total rain in the NEXRAD map in regions missed by the AMSU estimate, whereas there was 2.3% of the total rain in the AMSU-based system in regions missed by NEXRAD. Figure 3.6a illustrates the performance of the first neural network.

When the regional correction network is added to the system, a much more accurate measure of rain is obtained. The second neural net was able to predict with great accuracy the regional integrals of rain for all the regions in the initial precipitation map; in this case, the RMS error (at the regional level) was determined to be $0.24 \text{ [log10 (m}^3\text{s}^{-1}\text{ space]}}$. Figure 3.6b shows the performance for 21 different regions. Of those, 10 were used for training; and the rest were used for testing. The figure shows also that the estimate is more accurate for larger rain cells. This is because the fractional offsets in time and space are less significant to the total rain.
Figure 3.6. (a) Performance of the first neural net. (b) Performance of the system when the second neural net is added. Crosses represent test set; circles represent the training set.
3.3. Precipitation measurements using 54 GHz observations.

3.3.1. Basis for precipitation estimation from data at 54GHz.

AMSU-A has 15 channels with center frequencies that vary from 23.8GHz to 89GHz (see Chapter 2). Of those, some serve as window channels because they are most sensitive to land/sea variations. Others, which are of particular interest to the precipitation estimator, have center frequencies centered around 52-54GHz and sound the atmospheric temperature profile from the surface to an altitude of approximately 250 mbar. Unlike AMSU-B channels used in Section 3.2 which respond mainly to water vapor, these channels are used exclusively to estimate temperature profiles of the atmosphere with variable vertical resolution, generally between 5-10 km (Staelin, 1989).

The brightness temperatures measured by the sounder are made up mainly from three contributions (Grody, 1998):

- Surface radiation emissions
- The upwelling atmospheric radiation
- The downwelling radiation which reflects at the surface of the earth.

Besides measuring the mean absorption of oxygen, which as stated in Chapter 2, is a weak function of atmospheric temperature and depends exclusively on the frequency of each channel, the instrument is also sensitive to events that contaminate these brightness temperatures. The upwelling atmospheric transmittance as sensed by the sounder depends mainly on cloud temperature and the presence and size of hydrometeors. These factors are obviously related to rainfall.

Since the formation of rain is a meteorological process that occurs over a thick layer in the atmosphere, an attempt to estimate rain from brightness temperatures emerg-
ing from several altitudes is explored here. It must be noted that the approach described in Section 3.1 used radiances in the water vapor band at high altitudes in the atmosphere, whereas the algorithm described here focuses on temperature weighing functions which extend both lower and higher in the atmosphere.

Table 3.2 lists the four AMSU-A channels that were used to estimate precipitation, together with the window channel used to estimate the surface effects. Figure 3.7 shows what images from the channels at these frequencies look like. Observation of these images reveals that there two important issues which compromise the accuracy of any application that is based on these images:

- The first two channels (namely, 52.8GHz and 53.6GHz) show strong surface emissivity effects. The coastline is clearly visible in these channels, since they respond more strongly to brightness temperatures at lower altitudes, where the atmospheric transmittance is significantly affected by radiation emitted from the surface of the earth. As the altitude in the atmosphere and the oxygen absorption increases, the surface effects seem to diminish, which is evident in the images from 52.8 and 53.6GHz.

- There is a strong scan angle dependence in each of the 54-59 GHz channels. This effect is more evident in the 54.4 and 54.9 GHz channels, since the whole exhibited range of brightness temperatures is less than that of channels at 52.8 and 53.6GHz.
Table 3.2. AMSU-A Temperature sounding channels used to estimate precipitation.

<table>
<thead>
<tr>
<th>AMSU Channel</th>
<th>Center Frequency</th>
<th>Peak sensitivity altitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch 4</td>
<td>52.8 GHz</td>
<td>~1000 mbar</td>
</tr>
<tr>
<td>Ch 5</td>
<td>53.6 ± .115 GHz</td>
<td>~700 mbar</td>
</tr>
<tr>
<td>Ch 6</td>
<td>54.4 GHz</td>
<td>~400 mbar</td>
</tr>
<tr>
<td>Ch 7</td>
<td>54.9 GHz</td>
<td>~250 mbar</td>
</tr>
</tbody>
</table>
Figure 3.7. AMSU-A Temperature sounding channels used for the precipitation estimation algorithm in Section 3.2. (a) 52.8GHz. (b) 53.6GHz. (c) 54.4GHz. (d) 54.9GHz. (e) Window channel, showing land/sea boundaries. (d) NEXRAD precipitation data for same region.
3.3.2. Scan angle dependence.

As suggested by Figure 3.7, the AMSU-A temperature sounding channels exhibit a strong cross-scan angle dependence. This makes the brightness temperatures significantly lower at the extremes of each scan than for the views to nadir. When doing any kind of statistical analysis using this kind of data, a separate implementation for every spot along the orbit is not practical because the data sets are too small, so the data must be corrected to eliminate the cross-scan curvature.

One possible approach to remove this undesirable effect is to define some function of the scan angle (such as a linear function, or the cosine) and make it an additional input to the neural net, and let the network learn the relationship between the scan angle and the variations in brightness temperature. However, since the complexity of the network is to be kept as low as possible, another approach was used in this work.

To reduce the effects caused by the scan angle, a cloud-free region, consisting exclusively of land (no sea boundaries) and in the latitude of interest (in this case, over America, which lies approximately between 25° and 50°) was identified and for each of the AMSU-A channels used in the study, an average of the brightness temperature at each spot along a selected region was computed. These averages are shown in Figure 3.8a, and the continuity of this average confirms that the variation is due exclusively to the scan angle. These averages were then subtracted from each corresponding spot along the orbit, and the average at nadir (where the scan angle effect is assumed to be minimum) was added to the whole image. One corrected scan is shown in Figure 3.8b. This technique proved to work really well, particularly for the channels at 54.4GHz and 54.9GHz, where no information was clearly visible before the correction was implemented (Figure 3.8c,d).
Figure 3.8. Illustration of the scan-angle dependency and its correction. (a) Averages along one scan of a clear-air, only-land region for 54.9 GHz. (b) Result of subtracting (a) from one scan and adding the average at nadir. (c) Original image for 53.6 GHz. (d) Result of reducing the scan angle effect.
3.3.3. Surface emissivity.

In addition to the scan angle dependence, the AMSU-A channels at 52.8 GHz and 53.6 GHz exhibit surface emissivity effects. These channels sound the atmospheric temperature at altitudes where radiation by the surface of the earth still has a great influence in the atmospheric transmittance. For this reason, oceanic regions appear significantly colder than continental regions.

Since the effect of precipitating clouds is also to lower the brightness temperatures with respect to warm backgrounds corresponding to land, precipitating cells appear in the data images with brightness temperatures that are comparable to that of oceans. These effects would confuse a neural network, since both rain cells and ocean are to be interpreted in different ways.

One possible solution to this problem is the implementation of two neural network systems, one for land regions, and one for oceans. Since the transition from one kind of region to the other is very continuous (i.e. regions with coastlines) a third neural network could be trained to handle this tricky ‘intermediate’ regions. A more practical approach is to remove the surface effects from every channel. This can be achieved with several techniques, such as principal components analysis (Chen, 1998) or Linear Estimators.

As explained in Chapter 2, the AMSU-A window channel (89GHz) responds almost exclusively to surface emissions, and is a good indicator of the presence of land. Given the observation of the surface information from 89GHz, and using the Linear Least Squares Estimation technique (explained in Chapter 2), the information on a particular channel without the effect of the surface can be approximated as the error from estimating the surface information at that particular channel. In order to do this, the brightness tem-
 Temperatures on a given channel (ch) are assumed to have two components: a component due to the surface effects (S), and random noise (w):

\[ ch = S + w \] (3.1)

Moreover, the surface effects are assumed to be highly correlated to the information on the 89 GHz channel, which as explained in Chapter 2, responds mainly to the surface effects. Our goal is to remove these surface effects, or in other words, isolate the ‘noise’ component (w) which contains the variations of brightness temperatures that are uncorrelated with the surface effects. With this in mind, and using LLSE, we can write

\[ S = \frac{\text{cov}(ch, 89\text{GHz})}{\text{cov}(ch, ch)} \cdot (89\text{GHz} - \text{mean}(89\text{GHz})) \] (3.2)

which yields

\[ w = ch - \frac{\text{cov}(ch, 89\text{GHz})}{\text{cov}(ch, ch)} \cdot (89\text{GHz} - \text{mean}(89\text{GHz})) \] (3.3)

The images from channels at 54.4GHz and 54.9GHz did not present a significant surface emissivity effect, and therefore were left unprocessed. However, images from channels at 52.8GHz and 53.6GHz were corrected using this technique, with resulting images shown in Figure 3.9. As shown in the Figure, the surface effects were effectively removed, and the result shows only effects due to precipitating clouds.

In order to quantify the quality of this transformation, the brightness temperatures in regions of precipitating clouds were compared before and after the surface effects were removed, and they showed no significant change. Moreover, the brightness temperature difference over land and over sea (around 15K for the original image) was measured to be less than 2K after the land removal process.
Figure 3.9. Reduction of surface effect in data from channel at 52.8 GHz. (a) Original image. Florida, east coast and great lakes are clearly visible. There is a precipitation cell right below both lakes. (b) Result of applying LLSE techniques to reduce the surface emissivity effect. Note that coastline and great lakes have a significantly reduced effect, while the precipitating cell is unaffected. (c) Effect of the surface alone, resulting from subtracted (a) and (b).
3.3.4. Development of 54 GHz precipitating cloud perturbations.

Although AMSU-A channels respond to precipitating clouds, they also respond to other features in the atmosphere that can modify the temperature, such as masses of hot or cold air that form fronts. These effects are still unaccounted for when removing the surface emissivity (as explained in Section 3.3.3) and could affect the quality of any precipitation estimator trained with this data. It was seen in Section 3.3 that the AMSU-B moisture channels respond almost exclusively to precipitating cells, and for this reason, they were used as indicators of the position of rain cells. Once these positions are determined, the effect of these precipitating cells on AMSU-A channels can better be isolated and used to train a neural net to estimate rainfall.

In order to determine the location of rain cells, an adaptive threshold was applied to the information from the 183 ± 7 GHz channel from AMSU-B. As seen earlier, precipitating clouds appear as ‘cold’ regions over a warm, relatively uniform background. These cold spots appear to have brightness temperatures that range from 160 to 245K, whereas the background exhibits temperatures with an average greater than 260K. Application of a single threshold to define precipitation regions proved to be ineffective, since no threshold could consistently separate rain cells for all data sets.

To calculate the adaptive threshold, a lower bound of 245K was established, and all regions with brightness temperatures at 183 ± 7 GHz below this initial threshold were considered to be a precipitation cloud. This is a very safe assumption, since a brightness temperature of 245K is almost 20 degrees lower than the average nominal value for 183 ± 7 GHz. The area of this initial guess for precipitation was calculated and stored on disk. After initial regions of rain were labelled, the threshold was increased by steps of 1K
and a new area with pixels below this threshold was obtained. Increasing the threshold increased the area of the regions labeled as precipitation. However, once the threshold exceeds a certain level, the area of the cell blows up, indicating that the threshold must be lowered. For this reason, when a change of rain cell area equal to 25% was experienced due to an increase in the threshold, the algorithm stops and the final threshold is recorded. Rain is then defined to exist only on those regions where 183 ± 7 GHz goes below the adaptive threshold. A block diagram of this algorithm is shown in Figure 3.10, with the rain cell-location detection results in Figure 3.11.

Once the location of rain cells was defined, each of the AMSU-A channels was set to zero in those regions; then the regions were interpolated using the neighboring values in every direction. The interpolation technique solves Laplace’s equation to produce smooth images, effectively filling the ‘holes’ that used to contain values corrupted by rain. This function is included in the Image Processing toolbox for MATLAB. The objective of this interpolation is to make a first approximation of what every AMSU-A channel would be without the effects of precipitating clouds. Once the rain-free images are obtained, they are subtracted from the originals, producing perturbations which can be assumed to be caused exclusively by rain. One example of these perturbations (for 52.8GHz) is shown in Figure 3.11. As it will be explained in the next section, these ‘perturbations’ are used to train a Neural Network.
Figure 3.10. Flowchart showing algorithm to obtain adaptive threshold to obtain location of rain cell.
Figure 3.11. Precipitating cloud perturbations. (a) AMSU-B 183 ± 7 GHz. (b) Region where (a) falls below adaptive threshold. (c) Cloud perturbations using (b) on 52.8 GHz.
3.3.5. Neural net implementation.

Once the data from AMSU-A channels has been corrected from the effects of the scan angle and surface emissivity (as explained in Sections 3.3.2 and 3.3.3), and the perturbations due to rain have been isolated (Section 3.3.4), they are used as inputs to a Neural Network, similar to the one implemented in Section 3.2.2. This Network is trained to learn the relationships between AMSU-A channel perturbations due to rain and precipitation rate. The whole process by which this data is treated is shown in Figure 3.12. See Appendix C for the MATLAB implementation of the whole process.

\[\text{AMSU-A 54 GHz} \rightarrow \text{Scan Angle Dependence Correction} \rightarrow \text{Removal of surface emissivity effects} \rightarrow \text{Calculation of cloud perturbations} \rightarrow \text{Input to Neural Network}\]

Figure 3.12. Pre-processing of AMSU-A imagery prior to input to Neural Network.

The network was feedforward and had four inputs, corresponding to applying the process described in Figure 3.12 to data from channels at the four frequencies of interest (see Table 3.2). In other words, this network had as inputs the perturbations due exclusively to precipitating clouds in the AMSU-A 54 GHz channels. Since these channels seem to convey a great amount of information to estimate precipitation, no other inputs were deemed necessary for this study.

Besides having four inputs, the network had two input nodes, two hidden nodes, and one linear output node. The simplicity of this network allowed for very fast calculation of precipitation estimations. Moreover, the network was trained using NEXRAD 15-
min average precipitation data. Two sets were used to create the network: a training set from a cold front and a validation set from a Hurricane (see Table 3.1). The sets had several thousand elements, of which approximately 15% corresponded to precipitation.

Figure 3.13b shows the output of the neural net for the Cold Front precipitation system. The location of the rain cells appear to be in agreement with those in the NEXRAD version, shown in Figure 3.13a. Moreover, the cells appear less fragmented than those of the Neural Net output for the algorithm described in Section 3.2, which is a consequence of the limitations in resolution of AMSU-A data (See Chapter 2).

Figure 3.13. Performance of the Neural Net estimator for Cold Front. (a) NEXRAD precipitation data, lowpass filtered to 15km. (b) Precipitation estimates using neural network. (c) Region definition for rainmap in (b). Since rainmaps are more uniform, there is a smaller number of regions than in Figure 3.4 (11 regions).
3.3.6. Regional corrections.

As seen in Section 3.2.3, discrepancies between the neural net output and NEXRAD were seen in terms of cell shape and area. As explained before, this is due to the eventual spatial offset of AMSU-B channels and the actual position of the rain cells. The neural network described here utilizes AMSU-B information to pinpoint the place where the learning process for the neural net takes place (Section 3.3.4) and therefore the output images from this stage also contain a spatial offset with respect to NEXRAD, although it appears to be less significant in this case. The neural network output was considered reasonably accurate at the regional-integral level, (Figure 3.14b) and for this reason, a second network, such as the one implemented for AMSU-B estimates in section 3.2.3, was not considered necessary.

3.3.7. Precipitation estimation error.

The neural net correctly identified the position of the rain cells, and the output image had an RMS error of 0.0082 inches/15min when compared to NEXRAD. This RMS error is significantly lower than that of the implementation using only AMSU-B channels (Section 3.2.4). However, because of the limitations on resolution in AMSU-A data, the network failed to identify cells with areas in the order of (30x30) km$^2$. When the rest of the cells are compared, approximately none of the NEXRAD-measured rain was missed by the AMSU-A based estimator; whereas only 5% of the AMSU-A based estimator was missed by NEXRAD.
Figure 3.14. Performance of neural network. (a) Output, compared to NEXRAD precipitation data at a pixel level. (b) Regional precipitation integrals, compared to those of NEXRAD. x’s correspond to training set, o’s to validation set. Notice that the network performs quite accurately at the regional level.
Chapter 4

Conclusion

This chapter lists the main achievements of the thesis, and a set of possibilities for future research.


In this thesis two approaches for developing a precipitation estimation technique using brightness temperatures obtained through microwave remote sensing were presented. Both systems, which utilize the flexibility and strength of an artificial neural network to estimate rain, were explained in detail in Chapter 3. The first version of the algorithm used brightness temperatures from the water vapor channels in AMSU-B at 183 ± 1 and 183 ± 7 GHz, and the second used brightness temperatures from AMSU-A at four frequencies close to 54 GHz.

The performance of the estimators has been demonstrated using data from a major frontal passage over the United States (see Chapter 3). Moreover, Appendix D shows the performance of both estimators over hurricane Georges. By comparing the error in both
cases, it was noticed that the system based on AMSU-A 54 GHz data performed better, (in
terms of RMS error) both at the pixel-level and at the regional rain cell level. For this rea-
son, it was concluded that AMSU-B data at 183 ± 1 and 183 ± 7 GHz is better for deter-
mining the position of precipitating clouds, whereas AMSU-A data at 54 GHz is better for
estimating the amount of rain.

The main features of the precipitation estimation technique can be summarized as
follows:

- The algorithm estimates the amount of rain with acceptable accuracy, from satellite
data, relative to ground-based methods.
- The algorithm uses the flexibility of neural nets to adapt to the precipitation character-
istics of several regions (depending on latitude).
- The algorithm provides some insight into the way in which the brightness tempera-
tures at different altitudes in the atmosphere have an effect on the amount of rainfall.

It should be clear that since the estimations depend on the brightness temperatures
measured by the satellite, the application of neural networks can only be expected to be as
consistent as the Tb’s it uses.

4.2. Future research possibilities.

A great deal of work is still required to develop and strengthen the methods intro-
duced in this paper. The strategy of using neural networks and brightness temperatures is
very promising; the following represent some of the most significant tasks requiring com-
pletion before accurate precipitation estimation from microwave data becomes a reality.
• Research is necessary to identify which input variables are most informative for the estimation of rain. Here two separate approaches (one with AMSU-A data and other with AMSU-B) were developed. An obvious extension is to combine both AMSU-A and AMSU-B data into a single model, and this approach looks very promising. The selection of the input data represents the main possible research development of this precipitation algorithm.

• An improvement to the algorithm would include training the network to recognize the dependency of brightness temperatures not only to rainfall, but other variables as well, such as ice shape and distribution.

• Moreover, the current algorithm produces estimates that are limited to land surfaces only; an extension could use other precipitation data over oceans to produce a separate estimate. It was seen in Chapter 3 that the surface effects were in fact removed from the brightness temperature; a better estimator could include the effects of land as another input variable.

• The complexity of the neural networks could be made smaller, trying to keep the error at an acceptable level.

• The estimator presented in this thesis is limited by the amount of validation data to the continental United States. The networks could be trained with other sources of precipitation measurements at different latitudes for more global rain estimates.

• Lastly, an implementation of the algorithm in real time would be valuable for weather forecasting and other applications.
Appendix A

Satellite image interpolation code

This code shows the MATLAB implementation of the interpolation algorithm which transforms AMSU satellite imagery into isometric coordinates.
function [N] = interpmap(chdata, dir)

% INTERPAMSU     Interpolate AMSU satellite images
% INTERPAMSU(CH,dir) interpolates data from the Advanced
% Microwave Sounding Unit (AMSU) and translates it into an
% isometric coordinate system. It also orients the output
% depending on the direction of the orbit, with 'u'
% corresponding to UP and 'd' corresponding to DOWN as
% defined by the latitude.
%
% %

% Determine AMSU type and bounds
if size(chdata,1) == 30
    type = 'A';
elseif size(chdata,1) == 90
    type = 'B';
end

bounds = 1:size(chdata,2);

% Create coefficients from bilinear interpolation as explained
% In chapter 2 (See createcoeff.m).
createcoeff(bounds,type);

% Rotate image depending on direction of orbit
%
if dir=='d' % going downwards
    chdatar = rot90(rot90(rot90(chdata(:,bounds))));
else chdatar = rot90(chdata(:,bounds));
end

% Produce output by multiplying coefficients
% Start horizontally
M = zeros(size(newgridy,1),size(newgridx,2));
data = zeros(size(cl));

for h = 1:max(bounds)-min(bounds)+1
    row = chdatar(h,:);

    for i = 1:size(newgridx,2)
        if ruler1(1,i) == 0
            data(1,i) = 0;
            data(2,i) = row(1,ruler2(1,i));
        elseif ruler2(1,i) == 0
            data(1,i) = row(1,ruler2(1,i));
            data(2,i) = 0;
        end
    end
end

74
data(1,i) = row(1,ruler1(1,i));
data(2,i) = 0;
else
data(1,i) = row(1,ruler1(1,i));
data(2,i) = row(1,ruler2(1,i));
end
end

pdata = data.*c1;
M(h,:) = sum(pdata,1);
end

% Now complete interpolation vertically

N = [];
T = zeros(size(newgridy,1), blocksize);
U = zeros(size(newgridy,1), blocksize);
cfactor = [0:blocksize-1]*(dy/yd);

for block = 1:nbloks
    if block==nbloks
        T = [ M(:,1+(block-1)*blocksize:size(newgridx,2))... 
             zeros(size(M,1), blocksize-mod(size(newgridx,2), blocksize)) ];
    else
        T = M(:,1+(block-1)*blocksize:block*blocksize);
    end

    for i = 1:size(newgridy,1)
        if ruler3(i,block) == 0
            U(i,:) = T(ruler4(i,block),:)*c2(i,block);
        elseif ruler4(i,block) == 0
            U(i,:) = T(ruler3(i,block),:)*c2(i,block);
        else
            U(i,:) = T(ruler3(i,block),:).* (cfactor+c2(i,block)) + ...
                     T(ruler4(i,block),:).* (-cfactor+c2(i,block));
        end
    end
    N = [N U];
end

N = N(1:size(newgridy,1),1:size(newgridx,2));
function () = createcoeff(bounds, amsutype)

% This file generates the x-coordinate and the y-coordinate
% for the AMSU-B data. These are calculated with respect to
% the beginning of the orbit. Refer to documentation for
% explanation of how the coordinates are obtained.

load coeffparams;
if ( lastsizebounds ~= max(bounds) - min(bounds) + 1 | ...
    lasttype ~= amsutype )
    if amsutype == 'A'
        spots = 30;
        gridsize = 12;
        a = 3.3*(pi/180);
        operiod = 101.5 * 60;
        speriod = 8;
        mperiod = 3*.019;
    elseif amsutype == 'B'
        spots = 90;
        gridsize = 4;
        a = 1.1*(pi/180);
        operiod = 101.5 * 60; % orbit period (sec)
        speriod = 8/3; % scan period (sec)
        mperiod = .019; % measurement period (sec)
    else error(sprintf...
        ('Incorrect AMSU data type. Please specify `A` or `B`'));
end

% X-coordinates

r = 6371.03; % mean radius of earth (km)
h = 833; % altitude of measuring instrument (km)
x = zeros(1,spots);
% initialize lookup table
tempx = zeros(1,spots/2);

bestbeta = 0;
for spot = 1:spots/2
    alpha = spot*a - (a/2);
    bestdiff = 10;
    for beta = bestbeta:.00001:bestbeta+0.03;
        k = (r/tan(alpha))*sin(beta) - 2*r*(sin(beta/2)).^2;
        if abs(k-h) < bestdiff
            bestdiff = abs(k-h);
            bestbeta = beta;
        end
    end
    tempx(l,spot) = bestbeta.*r;
end
x = [-flip(tempx) tempx];
% Generation of rulers and new grid for x position.

newgridx = [ floor(min(x)/gridsize)*gridsize:gridsize:0] ...
            [gridsize:gridsize:ceil(max(x)/gridsize)*gridsize] ;

ruler1 = zeros(size(newgridx));
ruler2 = zeros(size(newgridx));

index1 = 1;
for index2 = 1:size(newgridx,2)
    if newgridx(index2) > x(index1)
        index1 = index1 + 1;
        ruler1(1,index2) = index1 - 1;
    elseif (index1 == size(x,2) & newgridx(index2) > x(index1))
        ruler1(1,index2) = 0;
    else
        ruler1(1,index2) = index1 - 1;
    end
end

ruler2 = ruler1 + 1;
for index2 = 1:size(newgridx,2)
    if ruler2(1,index2) > size(x,2)
        ruler2(1,index2) = 0;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% horizontal coefficients
% coeff1 is multiplier for element on RIGHT
% coeff2 is multiplier for element on LEFT

coeff1 = zeros(size(newgridx));
coeff2 = zeros(size(newgridx));

for i = 1:size(newgridx,2)
    if ruler1(1,i) == 0
        coeff1(1,i) = 0;
        coeff2(1,i) = 1;
    elseif ruler2(1,i) == 0
        coeff1(1,i) = 1;
        coeff2(1,i) = 0;
    else
        d = abs(x(ruler1(1,i)) - x(ruler2(1,i)));
        d1 = abs(x(ruler1(1,i)) - newgridx(1,i));
        d2 = abs(x(ruler2(1,i)) - newgridx(1,i));
        coeff1(1,i) = d2/d;
        coeff2(1,i) = d1/d;
    end
end
cl = [coeff1; coeff2];

% generation of y coordinates. There are as many rulers as there are columns in the output
r = 6371.03; % radius of earth (km)
h = 833; % height of instrument (km)

yd = (aperiod / operiod) * (360*pi/180) * r; % vert distance (km)
y = [0:max(bounds)-min(bounds)]*yd; % between scans

y = y';

dy = (mperiod / operiod) * (360*pi/180) * r; % vert dist (km)

dy = dy*(spots-l)/size(newgridx,2); % data has more points now. % after x-interpolation

newgridy = [0:gridsize:ceil(max(y)/gridsize)*gridsize];
newgridy = newgridy';

blocksize = floor(gridsize/dy);
nblocks = ceil(size(newgridx,2)/blocksize);

ruler3 = zeros(size(newgridy,1),nblocks);
ruler4 = zeros(size(newgridy,1),nblocks);
coeff3 = zeros(size(newgridy,1),nblocks);
coeff4 = zeros(size(newgridy,1),nblocks);

for j = 0:nblocks-1
    k = j+1;
    y = y + j*(blocksize+l)*dy;
    index3 =1;
    for index4 = 1:size(newgridy,1)
        if newgridy(index4) > y(index3)
            index3 = index3+1;
            ruler3(index4,k) = index3-1;
        elseif (index3==size(y,1) & newgridy(index4)>y(index3))
            ruler3(index4,k) = 0;
        else
            ruler3(index4,k) = index3-1;
        end
    end
    ruler4(:,k) = ruler3(:,k)+1;

    for index4 = 1:size(newgridy,1)
        if ruler4(index4,k) > size(y,1)
            ruler4(index4,k) = 0;
        end
    end

    for i = 1:size(newgridy,1)
        if ruler3(i,k) == 0
coeff3(i,k) = 0;
coeff4(i,k) = 1;
elseif ruler4(i,k) == 0
coeff3(i,k) = 1;
coeff4(i,k) = 0;
else
d = abs(y(ruler3(i,k)) - y(ruler4(i,k)));
d1 = abs(y(ruler3(i,k)) - newgridy(i,l));
d2 = abs(y(ruler4(i,k)) - newgridy(i,l));
coeff3(i,k) = d2/d;
coeff4(i,k) = d1/d;
end
end
c2 = [coeff3 coeff4];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save xy x y dy yd newgridx newgridy blocksize nblocks...
c1 c2 ruler1 ruler2 ruler3 ruler4;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lastsizebounds = max(bounds) - min(bounds) + 1;
lasttype = amsutype;
save coeffparams lastsizebounds lasttype;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear a r h t i ltable alpha beta k spot ...
   bestbeta bestdiff tempx...
   gridsize h spots index1 index2 index3 index4 bounds ...
   coeff1 coeff2 coeff3 coeff4 d d1 d2 i...
   operiod speriod;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
Appendix B

Region definition code

This code shows the MATLAB implementation of the region definition algorithm described in Section 3.3.4.
Region definition code. Both AMSU and NEXRAD files are assumed to be in the same coordinate system (isometric).

See Section 3.3.4 for explanation of the algorithm.

%%% Load files; rename for convenience

```matlab
load /usr/users/afl/thesis.code/data/b982801329.mat
load /usr/users/afl/thesis.code/data/n982801345.mat;
load /usr/users/afl/thesis.code/183GHz/output.mat;

nexrad = nexrad/20;
amsu = prec/20;
```

%%% Threshold definitions for morphological processing

```matlab
thigh = 0.06;  % High threshold
tmed = 0.055;  % Medium threshold
tlow = 0.01;   % Low threshold
```

%%% Pre-process regions

```matlab
amsu_high = double(bwmorph(amsu>thigh, 'diag'));
amsu_high = double(bwmorph(amsu_high, 'fill'));
amsu_med = double(bwmorph(amsu>tmed, 'diag'));
amsu_med = double(bwmorph(amsu_med, 'fill'));
amsu_low = double(bwmorph(amsu>tlow, 'diag'));
amsu_low = double(bwmorph(amsu_low, 'fill'));
```

%%% Define regions

```matlab
regions_amsu = bwlabel(regions_amsu,8);
num_regions = max(max(regions_amsu));
```

%%% Initialize variables for training vector

```matlab
amsurain_max = [];
amsurain_int = [];
amsurain_area = [];
amsurain_area_high = [];
```

```matlab
nexradrain_int = [];
```

```matlab
display = zeros(size(amsu));
```

%%% Process regions, one by one

```matlab
disp('Starting now...');
for i=1:num_regions
    current_region = double(bwmorph((regions_amsu==i), 'close', inf));
    current_region_amsu = current_region.*amsu;
    current_region_nexrad = current_region.*nexrad;
```

82
if length(find(current_region)) > 30;
    amsurain_max = [amsurain_max; max(max(current_region))];
    amsurain_int = [amsurain_int; sum(sum(current_region))];
    amsurain_area = [amsurain_area; length(find(current_region))];
    amsurain_area_high = [amsurain_area_high; ...
                           length(find(current_region.*amsu_high))];
    nexradrain_int = [nexradrain_int; sum(sum(current_region_nexrad))];
end
end
Appendix C

54 GHz based neural net implementation

This code shows the MATLAB implementation of the precipitation algorithm detailed in Section 3.3.
% Precipitation estimator based on 54GHz BT's
% See Section 3

% Files that start with b are AMSU-B BT's
% Files that start with a are AMSU-A BT's
% Files that start with e are topographic elevation
% Files that start with n are NEXRAD precipitation
% They are all assumed to be in the same coordinate system
% See cloudeffect.m for development of cloud perturbations

%%% VALIDATION VECTOR.
load chav; % contains scan averages over clear air
load /data/b982730026;
load /data/a982730026;
load /data/e982730026;
load /data/n982730045;
ch4 = ch4 - repmat(ch4av, [1 size(ch4,2)]) + ch4av(45);
ch5 = ch5 - repmat(ch5av, [1 size(ch5,2)]) + ch5av(45);
ch6 = ch6 - repmat(ch6av, [1 size(ch6,2)]) + ch6av(45);
ch7 = ch7 - repmat(ch7av, [1 size(ch7,2)]) + ch7av(45);
C = cov([ch2(:) ch4(:) ch5(:)]);
V = C(:,1)/C(1,1);
ch4 = ch4 - V(2)*(ch2-mean(ch2(:)));
ch5 = ch5 - V(3)*(ch2-mean(ch2(:)));
[deltach4,deltach5,deltach6,deltach7,mask] = cloudeffect(ch20,ch4,ch5,ch6,ch7);

VV = struct('P',[],'T',[]);
VV.P = [deltach4(:); deltach5(:); deltach6(:); deltach7(:)];
VV.T = nexrad(:)'/20;

%%% TRAINING VECTOR
load chav; % contains scan averages over clear air
load /data/b982801329;
load /data/a982801329;
load /data/e982801329;
load /data/n982801345;
ch4 = ch4 - repmat(ch4av, [1 size(ch4,2)]) + ch4av(45);
ch5 = ch5 - repmat(ch5av, [1 size(ch5,2)]) + ch5av(45);
ch6 = ch6 - repmat(ch6av, [1 size(ch6,2)]) + ch6av(45);
ch7 = ch7 - repmat(ch7av, [1 size(ch7,2)]) + ch7av(45);
C = cov([ch2(:) ch4(:) ch5(:)]);
V = C(:,1)/C(1,1);
ch4 = ch4 - V(2)*(ch2-mean(ch2(:)));
ch5 = ch5 - V(3)*(ch2-mean(ch2(:)));
[deltach4,deltach5,deltach6,deltach7,mask] = cloudeffect(ch20,ch4,ch5,ch6,ch7);

TV = struct('P',[],'T',[]);
TV.P = [deltach4(:)];...
deltach5(:)'; ... 
deltach6(:)'; ... 
deltach7(:)';];
TV.T = nexrad(:)'/20;

limits = [-5 5; -5 5; -1 1; -1 1;];
net = newff(limits, [4 4 1], {'tansig' 'tansig' 'purelin'});
net.trainParam.epochs = 100;
net = init(net);

[net,tr] = train(net,TV.P,TV.T);

prec = sim(net,TV.P);
prec = reshape(prec,[90 length(prec)/90]);
function [varargout] = cloudeffect(pm7, varargin)

% CLOUDEFFECT Display effects of precipitating clouds on AMSU-A channels.
% [DELTACH1,DELTACH2,...] = CLOUDEFFECT(pm7,ch1,ch2,...) calculates an
% adaptive threshold to separate precipitating clouds in the 183+/-7 GHz
% channel. Regions with clouds are eliminated from the rest of the
% channels, and its contents are filled with interpolation. The function
% returns the perturbation on each channel.
%
%%% 0.) Error traps

nchannels = nargin-1;
if isempty(varargin), error('Invalid input'); end;
if nchannels ~= nargout-1,
    error(['Enter the same number of input '...
            'and output channels']);
end;
rows = size(varargin{1},1);
cols = size(varargin{1},2);
for i = 1:nchannels
    if (size(varargin{i},1) ~= rows) | (size(varargin{i},2) ~= cols)
        error('Input matrices must have the same size.');
    else varargout{i} = varargin{i};
    end
end

%%% 1.) Calculate adaptive threshold

tlow = 245;
thigh = 300;
for tb = tlow:thigh,
    dif = sum(sum((pm7<tb+1) - (pm7<tb)));
    if dif/sum(sum(pm7<tlow)) >= .15
        t = tb;
        break;
    end
end

%%% 2.) Select region of precipitating clouds (based on threshold)

bwl = pm7<t;
bw2 = dilate(bwl,[1 1 1;1 1 1;1 1 1]);

%%% 3.) Determine regions to fix and store

% Top edge
    e = find(bw2(1,:) ==1);
    if ~isempty(e), index1 = e([1 find(diff(e)>1)+1]);
        index2 = e([find(diff(e)>1) length(e)]);
    else index1 = [1]; index2 = [];
    end
    bwl(1,:) = 0;

88
Bottom edge
\[ e = \text{find}(bw2(:,end,:) == 1); \]
if \(-\text{isempty}(e)\),
\( \text{index3} = e([1 \text{ find}(\text{diff}(e)>1)+1]); \)
else \(\text{index3} = []; \)
\( \text{index4} = []; \)
end;
\( \text{bwl}(:,end,:) = 0; \)

Left edge
\[ e = \text{find}(bw2(:,1,:) == 1); \]
if \(-\text{isempty}(e)\),
\( \text{index5} = e([1 \text{ find}(\text{diff}(e)>1)+1]); \)
else \(\text{index5} = []; \)
\( \text{index6} = []; \)
end;
\( \text{bwl}(:,1) = 0; \)

Right edge
\[ e = \text{find}(bw2(:,end,:) == 1); \]
if \(-\text{isempty}(e)\),
\( \text{index7} = e([1 \text{ find}(\text{diff}(e)>1)+1]); \)
else \(\text{index7} = []; \)
\( \text{index8} = []; \)
end;
\( \text{bwl}(:,end) = 0; \)

4.) Now fix edges for every channel
for \( u = 1:\text{nchannels} \)
\( \text{curchannel} = \text{varargout}(u); \)

Top edge
for \( k = 1:\text{length(index1)} \)
if \( \text{index1}(k) == \text{index2}(k) \),
\( \text{curchannel}(1,\text{index1}(k)) = (\text{curchannel}(1,\text{index1}(k)-1)+\text{curchannel}(1,\text{index2}(k)+1))/2; \)
else \( \text{curchannel}(1,\text{index1}(k):\text{index2}(k)) = \text{interp1}([\text{index1}(k) \text{ index2}(k)],\ldots \)
\[ \{\text{curchannel}(1,\text{index1}(k)) \text{ curchannel}(1,\text{index2}(k))\}, \]
\( \text{index1}(k):\text{index2}(k); \)
end;

Bottom edge
for \( k = 1:\text{length(index3)} \)
if \( \text{index3}(k) == \text{index4}(k) \),
\( \text{curchannel}((\text{end},\text{index3}(k)) = (\text{curchannel}(\text{end},\text{index3}(k)-1)+\text{curchannel}(\text{end},\text{index4}(k)+1))/2; \)
else \( \text{curchannel}((\text{end},\text{index3}(k):\text{index4}(k)) = \text{interp1}([\text{index3}(k) \text{ index4}(k)],\ldots \)
\[ \{\text{curchannel}(\text{end},\text{index3}(k)) \text{ curchannel}(\text{end},\text{index4}(k))\}, \]
\( \text{index3}(k):\text{index4}(k); \)
end;

Right edge
for \( k = 1:\text{length(index5)} \)
if \( \text{index5}(k) == \text{index6}(k) \),
\( \text{curchannel}(\text{index5}(k),1) = (\text{curchannel}(\text{index5}(k)-1,1)+\text{curchannel}(\text{index6}(k)+1,1))/2; \)
else \( \text{curchannel}(\text{index5}(k):\text{index6}(k),1) = \text{interp1}([\text{index5}(k) \text{ index6}(k)],\ldots \)
\[ \{\text{curchannel}(\text{index5}(k),1) \text{ curchannel}(\text{index6}(k),1)\}, \]
\( \text{index5}(k):\text{index6}(k); \)
end;
end;

%% Left edge
for k = 1:length(index7)
    if index7(k)==index8(k),
        curchannel(index7(k),end) = (curchannel(index7(k)-1,end)+curchannel(index8(k)+1,end))/2;
    else curchannel(index7(k):index8(k),end)=interp1([index7(k) index8(k)],...
        [curchannel(index7(k),end) curchannel(index8(k),end)],
        index7(k):index8(k))’;
    end;
end;

varargout(u) = varargout(u) - roifill(curchannel.*-bw1,bw2);

end;

varargout(nchannels+1) = bw1;
Appendix D

Output for Hurricane Georges

The image on next page shows the output of the precipitation estimators (both schemes presented in Chapter 3) for Hurricane Georges, which struck Florida in September, 1998.
Figure D.1. Output of precipitation estimator to data from Hurricane Georges, Sep 30 1998, (a) Output from 183 GHz based estimator, (b) Output from 54 GHz based estimator, (c) NEXRAD precipitation data for the same day.
References


Diak, G. Kim, D. Whipple, M. Preparing for the AMSU. Bulletin of the American Meteorological Society, 73, 1971-84.


