Atmospheric compensation for SeaWiFS images of Lake Superior utilizing spatial information

Kirk Knobelspiesse

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Atmospheric Compensation for
SeaWiFS Images of Lake Superior
Utilizing Spatial Information

Kirk D. Knobelspiesse
B.S. Rochester Institute of Technology (1998)

A thesis submitted to the
Chester F. Carlson
Center for Imaging Science
College of Science
In partial fulfillment of the requirements
For the Master of Science Degree
At the Rochester Institute of Technology
Rochester, New York

September 27, 2000

Signature of Author ____________________________
Kirk D. Knobelspiesse

Accepted by ____________________________
Coordinator, M.S. Degree Program Date 9/27/2000
Atmospheric Compensation for SeaWiFS Images of Lake Superior Utilizing Spatial Information

I, Kirk D. Knobelspiesse, hereby grant permission to the Wallace Memorial Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author

Kirk D. Knobelspiesse

Date 9/27/00
The M.S. Degree of Kirk D. Knobelspiesse has been examined and approved by the thesis committee as satisfactory for the thesis requirement for the Master of Science degree in Imaging Science.

Dr. Anthony Vođacek, Thesis Advisor

Dr. John R. Schott

Mr. Rolando Raqueño

9/27/00

Date
ABSTRACT

Several assumptions are made with the established atmospheric compensation algorithm for images from the SeaWiFS remote sensing platform. One of these assumptions, the existence of Case I (optically clear) ocean water, cannot be made for images of Lake Superior. A modification to the established atmospheric compensation algorithm is presented, where empirical information and external spatial data are utilized to compensate for the atmosphere in all regions of the lake.

The established SeaWiFS atmospheric compensation algorithm uses a form of the Dark Object Subtraction (DOS) method. SeaWiFS has two Near-Infrared (NIR) bands used for atmospheric compensation. At these wavelengths, Case I water has no water leaving radiance. Therefore, radiance that reaches the sensor is due to atmospheric scattering alone. This NIR signal is used to determine the atmosphere type in that region of the image, which is used, in turn, to correct for the atmospheric effects in all bands.

The alternative algorithm defines Lake Clear Water (LCW) as the inland analogy to Case I water. However, unlike Case I water, LCW has water leaving radiance in the SeaWiFS NIR bands. Because of the oligotrophic (nutrient starved) nature of Lake Superior, it is reasonable to assume that this radiance is a constant determined by ground measurements. The atmospheric effect, then, is the difference between the expected water leaving radiance and that measured at the sensor. Like the established algorithm, this NIR signal is used to correct for the atmospheric effect in all bands in LCW regions.

To implement the algorithm, an unsupervised classification method is used to map LCW and non-LCW regions in an image. Since the NIR signal in non-LCW regions is unusable, the NIR signal is extrapolated from neighboring LCW regions. This extrapolation is aided by meteorological data. Using look up tables created from the MODTRAN atmospheric model, an atmospheric type is calculated for each pixel in the image, and used for atmospheric effect subtraction in all bands.

Results of this alternative atmospheric compensation algorithm were compared to optical water profile data gathered on several cruises in Lake Superior.
ACKNOWLEDGEMENTS

I would personally like to thank all those whose guidance, knowledge and support helped me complete this thesis and my degree requirements. First, and foremost, I would like to thank my parents, Ernest and Mary Knobelspiesse, for their never-ending support. My thesis advisor, Tony Vodacek, was an endless supply of good ideas and guidance. Rolando Raqueño and John Schott’s vast experience were very helpful with using MODTRAN and revising my thesis. Nina Raqueño, Bryce Nordgren, and Scott Brown gave tons of help by providing data and software expertise. John Klatt, Niki Wilson, Joe Kloiber, Andy Fordham and all the other members of the DIRS group were great to have around to bounce ideas off of. I would also like to thank Harvey Rhody, Jeff Pelz and John Arney for their help in the early stages of this thesis. Finally, I appreciate the efforts of all the members of the Center for Imaging Science to make my stay here enjoyable and educational.

From the Michigan Technical University, I would like to thank Sarah Green, Judith Budd and Daniel Warrington for their help with the KITES project and Lake Superior data. I would like to thank the KITES project and its backers, the National Science Foundation and the National Oceanic and Atmospheric Administration, for their funding of my work. Finally, I would like to thank the National Atmospheric and Space Administration for providing SeaWiFS imagery.
# TABLE OF CONTENTS

Abstract iv

List of Figures viii

List of Tables x

1. Introduction 1

2. Background 4
   2.1 Apparent Reflectance 4
   2.2 Inherent Optical Properties 4
   2.3 Atmospheric Compensation of Satellite Imagery 7
   2.4 SeaWiFS Radiative Transfer and Atmospheric Compensation 8
   2.5 SeaWiFS Atmospheric Compensation Failure 12
   2.6 Other Approaches 13

3. Approach 15
   3.1 Objectives and Design Criteria 15
   3.2 Algorithm Overview 16
   3.3 Relationship to established SeaDAS algorithm 17
   3.4 Creation of Algorithm Constants 19
      3.4.1 Spectrally Weighted Exo-atmospheric Irradiance 19
      3.4.2 Historically known Lake Clear Water (LCW) regions 21
      3.4.3 Near Infrared (NIR) band reflectance of LCW 26
      3.4.4 Radiative Transfer Geometry 28
         3.4.4.1 Geometry definitions 29
         3.4.4.2 Per-pixel geometry determination 29
      3.4.5 MODTRAN Look Up Table (LUT) creation 32
   3.5 The Alt_SeaWiFS software 36
      3.5.1 Alt_SeaWiFS Software, Part One 36
         3.5.1.1 Opening SeaWiFS files 36
         3.5.1.2 Determining Lake Superior Region 36
         3.5.1.3 Land Mask Determination 39
         3.5.1.4 Opening and preparing the Historical LCW image 42
         3.5.1.5 Ending Part One of Alt_SeaWiFS software 42
      3.5.2 User Assisted Image to Image Registration 42
      3.5.3 Alt_SeaWiFS Software, Part Two 44
         3.5.3.1 Performing image registration 44
         3.5.3.2 ISODATA classification 45
         3.5.3.3 Merging classified image and Historical LCW img 48
         3.5.3.4 Atmospheric Spectra Extrapolation 50
         3.5.3.5 MODTRAN LUT processing 51
         3.5.3.6 Fitting LUT's to image data 56
         3.5.3.7 Removing Atmospheric Effects from VIS bands 57
3.5.3.8 Bio-optical parameter creation 58
3.5.3.9 Output to HDF file 59
3.5.4 Alt_SeaWiFS conclusion 60

4. Results 61
4.1 Qualitative Data Assessment 61
  4.1.1 Latitude and Longitude Images 62
  4.1.2 Lake Superior Region Mask Image 63
  4.1.3 Land Mask Images 63
  4.1.4 Registered Historical LCW Images 65
  4.1.5 Image Classification 66
  4.1.6 Merged Class image and Historical LCW map 66
  4.1.7 NIR band LCW to non-LCW data extrapolation 67
  4.1.8 LUT type images 69
  4.1.9 Final image results 70
4.2 Direct Data Comparison 72
4.3 Day to Day Comparisons 74

5. Conclusions 83

6. Recommendations 84
  6.1 Better LCW reflectance values 84
  6.2 Better White Cap Reflectance Model 84
  6.3 Higher Resolution Wind Data 85
  6.4 Better HDF output 87
  6.5 Better use of image flags 87
  6.6 Better LUT resolution 87
  6.7 Smaller Gaussian kernels 88
  6.8 Better Cloud Mask 88

References 90

Appendix A: Calculation of Lake Superior SeaWiFS fly-over time 92
Appendix B: Master File variables 93
Appendix C: Alt_SeaWiFS software flowcharts 95
Appendix D: Classification 96
Appendix E: Alt_SeaWiFS installation manual 105
# LIST OF FIGURES

## 2. Background
- 2.1: Possible photon paths
- 2.2: SeaWiFS radiative transfer
- 2.3: SeaWiFS 412nm band image after atmospheric correction

## 3. Approach
- 3.1: Exo-atmospheric Irradiance and SeaWiFS band spectral responses
- 3.2: Backscatter from various KITES data points and pure water
- 3.3: Relationship between interpolated 765nm backscatter value and water depth
- 3.4 Lake Superior Transects and Bathymetry
- 3.5: Lake Superior Bathymetry Image
- 3.6: Hydro-Rad calculated reflectances, LCW regions
- 3.7: LCW reflectance, weighted by SeaWiFS Spectral Sensitivity
- 3.8: Radiative Transfer Geometry
- 3.9: Zenith and Azimuth angle calculation geometry
- 3.10: Geographic data element description
- 3.11: Lake Superior region masks
- 3.12: Smoothed Histogram
- 3.13: Finding Maximum Point #1
- 3.14: Suppression of group #1 pixels
- 3.15: Finding Maximum Point #2
- 3.16: Finding Optimum Threshold Value
- 3.17: An image of clusters with like spectral character
- 3.18: The LCW region identification process
- 3.19: 2D Gaussian extrapolation
- 3.20: LUT Interpolation
- 3.21: LUT type chosen for each image region

## 4. Results
- 4.1: Images used for qualitative analysis
- 4.2: Per pixel latitude and longitude for SeaWiFS image of Lake Superior region
- 4.3: SeaWiFS image, all areas except Lake Superior region removed with mask
- 4.4: Land Mask Images
- 4.5: Tweaked Land Mask Image
- 4.6: Registered Historical LCW mask images
- 4.7: Image of Classes
- 4.8: Merged Class Images
- 4.9: Extrapolated Band 7 images
- 4.10: Image LUT values
- 4.11: Water Leaving Radiance, band 3
- 4.12: Direct Radiance Comparison
- 4.14: Corrected Images from 1999, Days 232, 239 & 241
- 4.15: Per Pixel Variance, 1998, Days 94, 95 & 99
4.16: Per Pixel Variance, 1999, Days 232, 239 & 241 79
4.17: Histogram of images from 1998, Days 94, 95 & 99 80
4.18: Histogram of images from 1999, Days 232, 239 & 241 80
4.19: Plot values of specific locations for 1999, Days 232, 239 & 241 81
4.20: Locations of plot points for Figure 4.19 81

Appendix D: Classification
D.1: Step 1 of ISODATA classification 98
D.2: Data points grouped into classes 98
D.3: Class with too few pixels removed 99
D.4: Class center shifting 99
D.5: Splitting of class with large standard deviation 100
D.6: Start of next iteration 100
D.7: Morphological Dilation 101
D.8: 8-connected pixels 103
LIST OF TABLES

3. Approach
3.1: Spectrally weighted exo-atmospheric irradiance, per band 21
3.2: Meteorological Parameters 34
3.3: LUT Parameters 34
3.4: LUT parameter constants 35
3.5: LUT geometry 35
3.6: ISODATA classification parameters 46
3.7: LUT data elements 53
1.0 INTRODUCTION

The main purpose of this thesis is to create an atmospheric compensation algorithm for SeaWiFS images of Case II water in the Laurentian Lake Superior. Funding for the thesis is provided in part by the Keweenaw Interdisciplinary Transport Experiment in Superior (KITES). KITES is a National Science Foundation (NSF) and National Oceanic and Atmospheric Administration (NOAA) backed project to study the physical, chemical and biological phenomena associated with the Keweenaw current in Lake Superior. Because of the large temporal and spatial scales associated with a study of Lake Superior, remote sensing of Chlorophyll-a (Chl-a), Total Suspended Solids (TSS), and Colored Dissolved Organic Matter (CDOM) is useful. However, first results from SeaWiFS produced physically meaningless values for these parameters. The source of the error can be traced to the atmospheric compensation algorithm, which often predicts negative amounts of light leaving the surface of the water (Curran and Novo 1988).

The Sea-viewing Wide Field-of-view Sensor (SeaWiFS) is a space based remote sensing platform intended for ocean observations. SeaWiFS is a multispectral sensor, with eight bands in the visible and near infrared (NIR) segments of the electromagnetic spectrum. Launched in 1997, the instrument is the successor to the Coastal Zone Color Scanner (CZCS), which operated from 1978 to 1986 (Feldman 1999).

As part of NASA’s Mission to Planet Earth (MTPE), SeaWiFS has the goal of quantitatively imaging the ‘Primary Production’ of organic matter by the algae and bacteria that make up the bottom of the food chain in the ocean (Feldman 1999). To do this, the effects of the atmosphere between the sensor and the water must be removed. This is atmospheric compensation. Once the atmospheric effects have been removed, we have data about the amounts of light leaving the surface of the water. By using empirically derived ratios between different bands, measures of various bio-optical parameters can be determined. These bio-optical parameters are useful to the limnologist studying Lake Superior.
Atmospheric compensation involves understanding several types of atmospheric constituents, and the ways electromagnetic radiation can interact with them. Once the physics behind these interactions is understood, the remote sensing scientist needs to determine the quantity of each constituent in a particular image to perform an atmospheric compensation. For some constituents, however, an accurate measurement of quantity and type is difficult to achieve. Thus, many algorithms have been created to perform an atmospheric compensation using image data alone. The image processing system established for SeaWiFS images, called SeaDAS (SeaWiFS Data Analysis System), uses a modified Dark Object Subtraction (DOS) method. This method assumes that the radiance reaching the sensor in several bands is due to atmospheric effects alone, because no radiance is leaving the surface. Using this atmosphere signal, SeaDAS determines the atmospheric constituents in that segment of the image. With this, the atmospheric compensation is performed for all image bands (Gordon 1997). However, atmospheric compensation for visible and NIR imagery of water can be particularly difficult. Often, 80%-90% of the signal reaching the top of the atmosphere is due to atmospheric scattering, while only the remaining 10-20% of the signal is due to scattering in the water.

The DOS method works for SeaWiFS images of deep ocean water because clear water absorbs nearly all NIR radiation. Therefore, clear water has no water leaving radiance in the NIR. However, in turbid coastal or inland water, there are often enough particles to reflect NIR radiance back to the sensor. In the ocean sciences community, clear, non-reflecting water is called Case I water, and turbid, reflecting water is called Case II water. In Case II regions, the DOS method wrongly attributes reflected photons to atmospheric effects. Therefore, SeaDAS overestimates the effect of the atmosphere, and creates an overly aggressive atmospheric compensation (Curran and Novo 1988). The result is an underestimation of the water leaving radiance values in the visible wavelength bands. This underestimation can be so severe as to cause a prediction of negative water leaving radiance values in some bands!
The proposed atmospheric compensation solution is designed specifically for Lake Superior. It divides the lake into two regions, one representing areas of uniform reflectance in the NIR (Lake Clear Water, LCW) and the other representing non-uniform reflectance (non-LCW). The atmospheric signal is then separated from the hydrospheric signal in LCW areas. This signal is extrapolated to all non-LCW regions of the lake using meteorological data. Using look up tables created from the MODTRAN atmospheric propagation model, the atmospheric properties are found. Once these are known, the water leaving radiance can be determined.
2.0 BACKGROUND

Prior to any discussion about the proposed SeaWiFS atmospheric compensation algorithm, several topics must first be considered. First, the final product of our remotely sensed data will be a measure of various water constituents, so we must understand all of the factors in the water and atmosphere that affect radiative transfer between these constituents and the sensor. The atmospheric compensation algorithm used by SeaDAS will be considered. This will include an overview of the theory of the algorithm and some specifics about its implementation. The causes and effects of the standard atmospheric compensation failure over Lake Superior will be discussed. Finally, a brief overview of alternative atmospheric compensation algorithms will be given.

2.1 Apparent Reflectance

A common unit used in radiative transfer discussions is the apparent reflectance. For our sake, apparent reflectance refers to the magnitude of flux reaching the sensor due to the "reflectance" from a particular object below. This can include both the reflectance from the surface of a body of water transmitted through the atmosphere and the flux scattered toward the sensor in the atmosphere. The total apparent reflectance in a scene is then the sum of the constituent apparent reflectances. The relationship between apparent reflectance and radiance is shown by Schott's "Magic Pi" derivation (Schott 1997).

\[ \rho = \frac{\pi L}{E_0 \cos \theta} \]  

(1)

where \( L \) is the radiance reaching the sensor, \( E_0 \) is the exo-atmospheric irradiance, and \( \theta \) is the angle between the normal to the surface and the sun (the solar zenith angle).

2.2 Inherent Optical Properties

The interaction between electromagnetic waves and the transmissive bodies through which they propagate can be described by several inherent optical properties (IOP's).
These properties are constant for a particular wavelength regardless of light magnitude, and can be used to determine atmospheric or hydrospheric constituents. Inherent optical properties neglect the effects of multiple scattering, and are additive over several layers of atmosphere and/or water. The following IOP's are described with more detail by Gordon (Gordon 1993).

The extinction coefficient describes the fraction of the power removed from a ray of electromagnetic energy per unit length in a medium. Removal could come from both scattering and absorption within the medium.

$$c(\lambda) = \frac{dQ(\lambda)}{dlQ_o(\lambda)} = a(\lambda) + b(\lambda)[m^{-1}]$$

(2)

where \(dQ(\lambda)\) is the power at wavelength \(\lambda\) lost within the medium due to both scattering and absorption, \(Q_o(\lambda)\) is the starting power, and \(dl\) is the thickness of the medium. \(a(\lambda)\) is the power at wavelength \(\lambda\) per unit length lost within the medium to absorption, and \(b(\lambda)\) is the power per unit length at wavelength \(\lambda\) lost within the medium to scattering.

The scattering coefficient describes the magnitude of power lost per unit length from the ray due to scattering.

$$b(\lambda) = \int_0^{2\pi} \beta(\lambda, \epsilon - \epsilon')d\Omega(\epsilon')[m^{-1}]$$

(3)

The \(d\Omega(\epsilon')\) term represents the entire sphere surrounding the scattering particle, while \(\beta(\lambda, \epsilon - \epsilon')\) is the volume scattering function, which describes the direction and magnitude of scattering about the particle. The volume scattering function is the fraction of the power traveling in direction \(\epsilon\) that is scattered to direction \(\epsilon'\). It can be expressed:

$$\beta(\lambda, \epsilon - \epsilon') = \frac{d^2Q(\lambda, \epsilon')}{dlld\Omega(\epsilon')Q_o(\lambda)}$$

(4)

where the \(d^2\) represents a second derivative: over the length term, \(dl\), and the solid angle term, \(d\Omega\).
The extinction coefficient increases exponentially with thickness, $l$ of the medium.

According to Grum (1979) the fraction of the power lost over the length $l$ can be defined:

$$\frac{dQ_o}{Q_o} = -\beta_c(l)dl$$

(5)

where $\beta_c$ is the extinction coefficient due to absorption and scattering. If we follow a beam along a path of length $L$, with a power of $Q_o$ at the start and $Q_L$ at the finish, we have:

$$\int_0^L \frac{dQ}{Q} = \int_0^L \beta_c(L)dl = \ln \left( \frac{Q_L}{Q_o} \right) = \ln Q_L - \ln Q_o = \ln \left( \frac{Q_L}{Q_o} \right) = \int_0^L -\beta_c(L)dl$$

(6)

This can be simplified to the exponential function:

$$\frac{Q_L}{Q_o} = e^{-\int_0^L \beta_c (L) dl}$$

(7)

The extinction coefficient is an expression of all power lost over length $dl$, and is the sum of the power lost to scattering and the power lost to absorption. This relationship can be used to determine the absorption coefficient:

$$a(\lambda) = c(\lambda) - b(\lambda) [m^{-1}]$$

(8)

where $a(\lambda)$ is the absorption coefficient.

It is important to note that extinction coefficients are additive. Therefore, we can have the sum of a scattering coefficient due to aerosol scattering and a scattering coefficient due to Rayleigh scattering equaling a total atmospheric scattering coefficient:

$$b_r(\lambda) = b_s(\lambda) + b_r(\lambda)$$

(9)
Another useful property is the single scattering albedo, which is the probability that a photon interaction in a body will involve scattering. It is really derived from previously defined IOP's:

$$\omega_s(\lambda) = \frac{b(\lambda)}{c(\lambda)}$$  \hspace{1cm} (10)

where $$\omega_s$$ is the single scattering albedo for a particular wavelength, $$\lambda$$.

A final property that is often used is transmittance, which is measured over large distances, rather than dl. Transmittance is the portion of radiance that is passed through a medium:

$$t = \frac{L_o}{L_i}$$  \hspace{1cm} (11)

where $$L_i$$ is the input radiance, $$L_o$$ the output radiance and $$t$$ the transmittance.

Transmittance values range between zero (no transmittance) and one (full transmittance). Unlike IOP's, transmittance values are not additive, but multiplicative.

### 2.3 Atmospheric Compensation of Satellite Imagery

To convert a signal detected from space into a measure of some ground characteristic, we must understand how that signal reached the detector. This is done with the radiative transfer equation, which describes the path of electromagnetic energy from the source to the detector, and includes the effects of attenuation in between. The radiative transfer equation describes all paths a photon could take between source and detector. Figure 2.1
shows examples of these paths.

If we are attempting to use remote sensing to measure the quantity of some water constituent, we need to fulfill two criteria. First, that constituent must have some characteristic that will affect the radiative transfer system. In other words, it needs some sort of apparent reflectance effect. Second, this effect must be strong, and unique, enough to distinguish it from other reflectance effects. A problem with satellite remote sensing of bodies of water in the visible wavelengths is that any signal received from the water accounts for only about 10% of the total signal (Andre and Morel 1991). The rest is due to scattering caused by air molecules (Rayleigh scattering) and aerosol particles (aerosol scattering). Therefore, to measure the reflectance effects of a particular water parameter, we must first remove the atmospheric effects. Since the atmosphere accounts for so much of the total signal, an accurate atmospheric compensation is paramount.

2.4 SeaWiFS Radiative Transfer and Atmospheric Compensation

The radiative transfer for SeaWiFS (and its predecessor, CZCS) is described in a simplified form from what is presented in Figure 2.9, above (Gordon 1997). The total, top of the atmosphere signal, $\rho(\lambda)$, expressed as apparent reflectance, is the sum of the apparent reflectances at the sensor due to atmospheric scattering, water leaving radiance, direct reflection from the water surface, and reflection from whitecaps on the water surface.

\[
\rho_s(\lambda) = \rho_{\text{scat}}(\lambda) + t(\lambda)\rho_g(\lambda) + t(\lambda)\rho_{\text{ww}}(\lambda) + t(\lambda)\rho_w(\lambda)
\]  

(12)

where $\rho_{\text{scat}}(\lambda)$ is the apparent reflectance due to atmospheric scattering, $t(\lambda)\rho_g(\lambda)$ is the direct reflectance off the water surface (glint) times the atmospheric transmittance, $t(\lambda)\rho_{\text{ww}}(\lambda)$ is the reflectance from whitecaps times the atmospheric transmittance, and $t(\lambda)\rho_w(\lambda)$ is the water leaving radiance (expressed as a reflectance) times the atmospheric transmittance (Gordon 1997). A diagram of this simplified radiative transfer system is presented in Figure 2.2.
Several assumptions are made in this radiative transfer description. First, multiple scattering effects between water and the atmosphere are ignored. Liquid water has a strong absorption in the NIR, so the magnitude of the water leaving radiance is very small, and the radiance that is scattered down again and reflected back is negligible (Gordon 1997). Likewise, multiple scattering effects between any of the other components of the system are also ignored. Multiple scattering effects between atmospheric components are included, but all of those effects are covered with the $\rho_{\text{scat}}$ term, and will be dealt with later.

In equation 8, the sensor product is the apparent reflectance at the top of the atmosphere, $\rho_s(\lambda)$, and the desired product is the reflectance due to water leaving radiance, $\rho_w(\lambda)$. This leaves the glint, whitecap reflectance, transmittance and atmospheric scattering as terms to determine to compute an accurate measure of water leaving radiance. Glint, the direct solar reflectance off the surface of the ocean, is highly directional. Therefore, the sensor is overwhelmed with glint when in its angular path. Otherwise, the glint term has a very minimal effect. In practice, areas overwhelmed by glint are easily noticed, and discarded. The SeaWiFS sensor has the ability to point away from glint, so for our purposes it can be ignored in our radiative transfer equation. The percent whitecap coverage is a function dependent upon wind speed. The spectrally dependent whitecap reflectance is also well known, so calculating the whitecap input to the radiative transfer equation is a matter of knowing the wind speed (Koepke 1984) (Monahan and O'Muircheartaigh 1986). The
transmittance of the atmosphere can be determined accurately with the surface atmospheric pressure (Gordon, Brown and Evans 1988). This leaves the atmospheric scattering as the unknown quantity. To further understand the scattering in the atmosphere, $\rho_{\text{scat}}$, we split it down into components.

$$\rho_{\text{scat}}(\lambda) = \rho_r(\lambda) + \rho_a(\lambda) + \rho_{\text{ra}}(\lambda)$$

(13)

where $\rho_r(\lambda)$ is the single scattering due to the molecules of the atmosphere (Rayleigh scattering) alone, $\rho_a(\lambda)$ is the single scattering due to aerosol size particles alone, and $\rho_{\text{ra}}(\lambda)$ is the multiple scattering effect between the two. Multiple scattering refers to photons first scattered by one particle, then another, before reaching the sensor. The magnitude of the multiple scattering, then, is dependent upon both single scattering magnitudes. Like transmittance, Rayleigh scattering is well understood and can be determined with the surface atmospheric pressure. Aerosol scattering, on the other hand, is not easily predicted. Since the multiple scattering term depends on the amount of aerosol scattering, it is also an unknown.

This leaves a radiative transfer equation that has too many unknowns ($\rho_r(\lambda), \rho_a(\lambda)$ and $\rho_{\text{ra}}(\lambda)$) to solve. The SeaWiFS/CZCS solution to this is to make several assumptions about the reflective properties of water in several bands. The SeaWiFS/CZCS solution assumes that the apparent reflectance of water in the NIR is negligible. The aerosol and multiple scattering terms are the only unknowns left, so the signal received at the sensor can be used to determine the aerosol type and magnitude. Once and aerosol type and magnitude have been chosen, its effects can be removed from the signal in the visible wavelength bands, thus yielding the water apparent reflectance.

The NIR bands used for the SeaWiFS atmospheric compensation are 865nm and 765nm. The parameter $\varepsilon(765,865)$ is a ratio between the apparent reflectance of the single scattering aerosol effects in both bands. The current approach is based upon single scattering aerosols, as this was used for the SeaWiFS predecessor, CZCS.

$$\varepsilon(765,865) = \frac{\rho_a(765)}{10 \rho_a(865)}$$
where $\rho_{sa}(\lambda)$ is the sum of the single scattering parameters $\rho_{s}(\lambda)$ and $\rho_{a}(\lambda)$.

The relationship between the single scattering aerosol apparent reflectance, $\rho_{sa}(\lambda)$, and the multiple scattering terms is shown with $K(\lambda, \rho_{sa}(\lambda))$:

$$\rho_{s}(\lambda) + \rho_{m}(\lambda) = K[\lambda, \rho_{sa}(\lambda)]\rho_{sa}(\lambda)$$

The value for $K(\lambda, \rho_{sa}(\lambda))$ described above is dependent upon the aerosol type, and is the deviation from a linear relationship between the single and multiple scattering reflectances. The SeaWiFS ratio term that includes multiple scattering combines equations 10 and 11.

$$\epsilon(765,865) = \frac{K[865, \rho_{as}(865)][\rho_{a}(765) + \rho_{m}(765)]}{K[765, \rho_{as}(765)][\rho_{a}(865) + \rho_{m}(865)]}$$

The K ratio above is an unknown quantity, so $\epsilon(765,865)$ is calculated for several aerosol types. An average value for $\epsilon(765,865)$ is found, and the model $\epsilon(765,865)$ values farthest from the average are found. A new average is then calculated without the edge $\epsilon(765,865)$ values. This process is repeated until only the closest four aerosol $\epsilon(765,865)$ values are used to calculate an average $\epsilon(765,865)$ value.

Once a $\epsilon$ value has been calculated for a pixel, the aerosol reflectance for all bands must be found. This is done by finding the aerosol reflectance predicted by the models closest to the average $\epsilon$ value, and combining their effect in the visible spectrum in proportion to their closeness to the average (Gordon 1997).

One final note is that this is a pixel by pixel process. A separate solution is found for every pixel in the image, regardless of the solution for neighboring pixels. As we will see later, this contrasts with the alternative atmospheric compensation algorithm, where data from neighboring pixels is used to improve the solution.
2.5 SeaWiFS Atmospheric Compensation Failure

The atmospheric compensation algorithm established for SeaWiFS and presented above is known to fail over Case II water, such as the Laurentian Lake Superior. Lake Superior is known to be very clear, but compared to deep ocean water, it is not. Therefore, the turbidity of Lake Superior invalidates the nil NIR water leaving radiance approximation (Curran and Novo 1988). When this happens, the backscattered radiance from the water is incorrectly identified as aerosol scattered radiance. The result is an overestimation of actual aerosol scattering. Since this large scattered radiance is subtracted from the signal, the result is an underestimation of actual water leaving radiance values in visible wavelength bands. In severe cases, the predicted water leaving radiance values are negative. Figure 2.3 is an illustration of this catastrophic failure.

Figure 2.3: SeaWiFS 412nm band image after atmospheric correction using established algorithm

This 412nm band image shows the drastic effects of the standard atmospheric correction algorithm, where the pink and red regions represent predictions of 'negative' water leaving radiance.
2.6 Other Approaches

Because of the failure of the standard SeaWiFS processing in coastal waters, several groups of researchers have also considered alternatives to the established atmospheric compensation approach. Some of these alternatives address absorbing aerosols in Case I water, while others address solutions for Case II water. Much of the research is being conducted presently, and is in review. Several of the alternative solutions for Case II water are described below.

The Siegel algorithm, developed by David Siegel at the University of California, Santa Barbara, assumes the magnitude of sediment backscatter co-varies with that due to chlorophyll. This is a valid assumption for Case I water and improves the solution in Case II water. However, chlorophyll and sediment do not have the same scattering spectrum, so the solution is not completely appropriate (Siegel 2000). This algorithm has been adopted in the latest version of the SeaWiFS SeaDAS processing system. However, the comparisons used in this thesis were with SeaDAS processing that did not include this change, as it occurred too late in this project's timeline to be incorporated.

The Aeroplus method, developed at the University of Rhode Island, looks at pixels corrected with the standard algorithm that have a negative water leaving radiance in the 412nm band. The aerosol model magnitude for these pixels is adjusted in an iterative fashion until the water leaving radiance in the 412 band is brought to zero. The approach is not physically based, and destroys any information contained within the 412nm band, but is a pragmatic "quick fix" (O'Reilly, Yoder and Schollaert 2000).

The Hu and Carder algorithm is intended for shallow regions of clear water. They found that water bottom reflectance affects SeaWiFS atmospheric correction only in regions shallower than four meters (assuming clear water). Therefore, correction can be performed by extending the established algorithm's correction from nearby regions with a depth greater than four meters. Of course, this requires the presence of optically clear water, which is not the case in Lake Superior (Hu, Muller-Karger, Carder and Lee 1998).
Ruddick, Ovidio and Rijkeboer (2000) have an algorithm that holds several parameters constant over a region of study. They assume that the ratios between NIR water reflectances, and the ratios between atmospheric scattering reflectances in the NIR bands remain constant over the region of study. The parameterization of these ratios is accomplished by examining scatter plots, using in situ data, or default values. Without in situ values or default values, a Case I type water region is needed in some part of the study area to choose ratio values from scatter plots (Ruddick 2000).

Land & Haigh, 1997, approach the problem by solving for atmospheric and hydrospheric constituents in one iterative algorithm. Therefore, the product is not L2 water leaving radiance data, but L3 water constituent data. This approach is heavily dependent on model accuracy. In the first paper (Land and Haigh 1996) the algorithm accurately calculated water leaving radiance values in some Case II regions, but had difficulty differentiating between atmospheric aerosol types. By allowing relative humidity to vary in the model, a higher degree of accuracy was achieved (Land and Haigh 1997).
3.0 APPROACH

3.1 Objectives and Design Criteria

The main objective of this thesis is to create a regionally appropriate atmospheric compensation algorithm for SeaWiFS images of Lake Superior. The algorithm is to be implemented as an alternative process within the SeaDAS system. The Alt_SeaWiFS algorithm completed several tasks:

1. First and foremost, the algorithm did not make the clear water assumption, which is the main reason for the standard algorithm's failure.

2. The algorithm was implemented in computer code. Research Systems Inc.'s Interactive Data Language (IDL) was used for several reasons. IDL supports the HDF scientific data file format used for SeaWiFS imagery, and is well suited for image processing. IDL is also already in use in the facilities at the Center for Imaging Science. In addition to IDL, a suite of remote sensing tools called The Environment for Visualizing Images (ENVI) were used. ENVI offers image processing software for remote sensing written in IDL. It is also already in use at the Center for Imaging Science.

3. The software was designed to process SeaWiFS level 1B images (not corrected for atmospheric effects) to SeaWiFS L2 images (corrected for atmospheric effects) that have the same data structure as images processed with the established atmospheric correction software, SeaDAS.

4. The algorithm operates on a single image without temporally dependent external data other than that typically supplied. Some atmospheric compensation methods make use of external data such as *in situ* data and LIDAR, which is not available for our images. Non-temporally dependent data, such as knowledge of historic LCW regions, was calculated. Daily meteorological data maps, supplied with SeaWiFS imagery, were used.

5. The proposed algorithm produced an image that can be used for further calculations of hydrospheric constituents.

6. The proposed algorithm acknowledged the difference in scale between expected SeaWiFS data use and Lake Superior use. Users of Lake Superior
SeaWiFS imagery will typically examine phenomenon occurring on a much smaller scale than open ocean imagery users.

7. The established algorithm is a pixel-by-pixel solution. However, this neglects the high probability that the atmospheric effect in adjacent pixels is correlated. The proposed algorithm recognized the importance of spatial information and utilized it in a way that does not degrade the radiometric fidelity of the solution.

8. The proposed algorithm acknowledged the environmental differences between the open sea and Lake Superior, such as differences in expected atmospheric constituents.

### 3.2 Algorithm Overview

The proposed algorithm expands on concepts of the established algorithm with key differences. It starts by dividing an image into two regions. The Lake Clear Water (LCW) region is assumed to have uniform water leaving radiance values in the NIR, while the non-LCW region does not. The boundaries of these regions are not permanent, so they are calculated for each individual image using a clustering algorithm.

The signal in LCW regions represents the atmospheric effect on a known water leaving radiance. This signal can be used to determine the atmospheric constituents, but is first extrapolated to all (non-LCW) regions of the lake. This extrapolation utilizes meteorological data, which will show what directions to emphasize.

In determining the atmosphere effect, the proposed algorithm avoids the clear water assumption of no water leaving radiance in the NIR and assumes an experimentally derived constant for radiance. Using this as input, the radiance reaching the sensor is predicted for a number of atmospheric situations using the MODTRAN atmospheric propagation model. This Look Up Table (LUT) is created using aerosols expected in the Lake Superior region. To determine the atmosphere for a particular LCW point in an
image, its NIR radiance is matched to an atmosphere type and magnitude in the LUT. This knowledge is then used to subtract for its effects in all wavelength bands.

Detailed descriptions of each of the steps in the proposed algorithm follow. For a flowchart of these steps, see Appendix C. Appendix D contains a flowchart of the steps in the established algorithm.

3.3 Relationship to established SeaDAS algorithm

NASA's SeaWiFS Data Analysis System (SeaDAS) is the software provided to process, display, and perform other functions on SeaWiFS imagery. SeaDAS includes the atmospheric compensation algorithms established for use with SeaWiFS images, and is the primary tool for their implementation (Baith and Lindsay 1999).

As we recall from equation (12), the two unknowns in the SeaWiFS atmospheric compensation algorithm are the scattering effects of the aerosols in the atmosphere and the water leaving radiance. Rearranged so the unknowns are isolated on the left side of the equation, (12) is:

$$\rho_a(\lambda) + \rho_{wa}(\lambda) + t(\lambda)\rho_a(\lambda) = \rho_I(\lambda) - t(\lambda)\rho_g(\lambda) - t(\lambda)\rho_{wc}(\lambda) - \rho_r(\lambda)$$

As noted previously, the aerosol single scattering and multiple scattering effects are removed in SeaDAS using the clear water approximation, which is not adequate for our purposes. However, determination of other unknowns, such as the Rayleigh scattering effect, the white cap reflectance, the atmospheric transmission and the glint, are adequate. Originally, we intended to use SeaDAS processing to remove the effects on the right side of the equation above, and our algorithm to determine the aerosol type and magnitude needed for the left. The input to our algorithm would have been an apparent reflectance in all eight bands after all of Level 1b processing (removal of sensor artifacts) and some Level 2 processing (atmospheric compensation). This input is expressed:
\[ \rho_i(\lambda) = \rho_i(\lambda) - t(\lambda) \rho_g(\lambda) - t(\lambda) \rho_{wc}(\lambda) - \rho_r(\lambda) \]  

(18)

where \( \rho_i(\lambda) \) is the apparent reflectance input to the proposed algorithm for a point within the \( \lambda \) band of the image.

Combining equations (17) and (18), we have:

\[ \rho_a(\lambda) + \rho_{ca}(\lambda) + t(\lambda) \rho_w(\lambda) = \rho_i(\lambda) \]  

(19)

In practice, this would be implemented by executing the Level 1 and Level 2 processing on the image, and exporting the values for \( \rho_i(\lambda) \) as they are calculated. However, we would encounter problems in the inverse modeling step of the algorithm. (See section 3.4.5: MODTRAN Look Up Table (LUT) Creation). Fitting atmospheric signal spectra from LCW regions to MODTRAN LUT’s would be trying to fit data in one form to that in another. Due to SeaDAS processing, LCW atmospheric spectra would have Rayleigh scattering and other atmospheric effects removed, while the MODTRAN LUT’s would not. This could be corrected several ways. We could adjust the atmospheric spectra so it includes the effects corrected in SeaDAS, but this would be counterproductive. We could also adjust the MODTRAN LUT’s so they do not include Rayleigh scattering and other atmospheric effects. It seems simplest to avoid all SeaDAS Level 2 processing and directly apply the results to the LUT’s. In this manner, we could have more control over the atmospheric compensation and more power to make the solution regionally appropriate.

This leaves us to correct for the non-atmospheric effects removed in SeaDAS processing but not accounted for in MODTRAN LUT’s. The most important of these effects is the white cap reflectance, \( \rho_{wc} \). It seems reasonable to assume, however, that the relationship between white cap reflectance and its driving factor, wind speed, is not the same for salt water and fresh water. Since SeaDAS processing assumes salt water, our algorithm could
be made even more regionally appropriate by solving for fresh water. Determining the relationship between wind speed and white cap reflectance for fresh water would require an extensive amount of data collection. Since this has not been done, the wind speed/white cap reflectance relationship for salt water will be used. However, the relationship is easy to change in the software.

3.4 Creation of algorithm constants

Several constants are required for processing images in this alternative atmospheric correction algorithm. While their use will be described in more detail later, explanations of their creation follows.

3.4.1 Spectrally Weighted Exo-atmospheric Irradiance (also used in SeaDAS)
Knowledge of exo-atmospheric irradiance is necessary to convert radiance values, used in some parts of the algorithm, to apparent reflectance values, and back again. (see section 2.1: Apparent Reflectance) A number of sources provide plots of exo-atmospheric irradiance at each wavelength. Ours was taken from the data used by the MODTRAN radiative transfer software (United States of America, Air Force Research Laboratory 1998).

These irradiance values, however, are not immediately useful for our calculations. The radiance values with which they will be processed represent the spectral sensitivity of each SeaWiFS band. Spectral sensitivity plots are available at (Barnes 2000). Figure 3.1 is a plot of these spectral sensitivity values, weighted by the solar exo-atmospheric irradiance.
To perform calculations with both irradiance and radiance values, we must convert them to the same units and spectral response. To calculate exo-atmospheric irradiance band values that we can use with radiance values, we multiply the spectral sensitivity of each band by the exo-atmospheric irradiance function. The integral of each exo-atmospheric band value is related to the amount of irradiance at the top of the atmosphere. However, it must be normalized by the integral of the sensor spectral response:

$$E_i = \frac{\int s_i(\lambda)E_o(\lambda)d\lambda}{\int s_i(\lambda)d\lambda}$$

(20)

where $E_i$ is the band value for the exo-atmospheric irradiance for band $i$, $s_i(\lambda)$ is the spectral sensitivity of band $i$ at wavelength $\lambda$, and $E_o(\lambda)$ is the exo-atmospheric irradiance at wavelength $\lambda$. As for units, our radiance is expressed as:
where \( [mW] \) is unit of flux, \( [cm^2] \) is the area of the receptor, \( [sr] \) is the unit of the solid angle, and \( [\mu m] \) is the unit of the wavelength. The exo-atmospheric irradiance, however, is expressed as:

\[
\left[ \frac{W}{m^2} \right] \text{ per nm}
\]

where \( [W] \) is unit of flux, \( [m^2] \) is the area of the receptor, and \( [nm] \) is the unit of the wavelength. To account for the difference in units, a scaling factor of 1000x must be applied to the exo-atmospheric irradiance values to use them with the radiance values supplied with SeaWiFS.

The spectrally weighed exo-atmospheric irradiance values, for each band, scaled to SeaWiFS units, are shown in Table 3.1

<table>
<thead>
<tr>
<th>Band 1</th>
<th>Band 2</th>
<th>Band 3</th>
<th>Band 4</th>
<th>Band 5</th>
<th>Band 6</th>
<th>Band 7</th>
<th>Band 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>125.149</td>
<td>146.680</td>
<td>167.355</td>
<td>162.958</td>
<td>164.731</td>
<td>144.329</td>
<td>108.515</td>
<td>93.501</td>
</tr>
</tbody>
</table>

3.4.2 Historically known Lake Clear Water (LCW) regions (not used in SeaDAS)

The algorithm presented here requires splitting the lake into two regions: Lake Clear Water, and non Lake Clear Water. We define LCW as regions where the Near-Infrared (NIR) reflectance is a constant. This assumption is reasonable given the ognitrophic nature of Lake Superior. While the extent of these LCW regions varies with time, some parts of the lake will always have LCW. We call these regions ‘Historical LCW’. Since we do not know the daily extent of LCW areas prior to the use of each image, it is important to know the Historical LCW regions so that we can determine the full LCW extent for that image.
We base our knowledge of Historical LCW regions on bathymetry. Using data collected on KITES cruises in 1999 with a HydroScat-2 backscatter meter, a relationship between bathymetric depth and backscatter was found. Backscatter near the surface is directly related to water leaving radiance, and thus reflectance. Therefore, we can extend this relationship to one between bathymetry and reflectance. If we can find the bathymetric depth where the NIR backscatter levels off to some constant, we have a rough estimate of which regions in the lake are historically LCW.

Unfortunately, the HydroScat-2 backscatter meter does not measure NIR backscatter. The instrument has two bands, at 470nm and 676nm. We must extrapolate to estimate a backscatter in the first SeaWiFS NIR band, 765nm. For our purposes, the SeaWiFS 765nm band is more important than the SeaWiFS 865nm, as it is more likely to be affected by near shore turbidity. Our data were gathered between June 30, 1999 and July 16, 1999 and consist of two collects for each transect. To calculate 765nm backscatter, average values for the top several meters of water backscatter at 470nm and 676nm were found for each transect location. To find the 765nm backscatter value, an average value for backscatter was found for all collects for each location at 470nm and 676nm and these values were linearly extrapolated to the NIR. An illustration of this is shown in Figure 3.2.
Figure 3.2: Backscatter from various KITES data points and pure water (data from Hobi Labs Backscatter meter)

Figure 3.2 shows the backscatter measured with the Hobi Labs HydroScat-2 Backscatter meter on several data collects in southern Lake Superior in 1999. Data points corresponding to 470nm and 676nm were measured by the instrument, while data points at 765nm and 865nm were linearly extrapolated from shorter wavelength values. Figure 3.3 shows backscatter decreasing for all data sets as wavelength increases. This is to be expected, as water reflectance is well known to decrease as wavelength increases from visible to NIR wavelengths. Reflectance is directly related to backscatter, as the higher the quantity of backscattered photons, the higher the reflectance. Each plot shows the backscatter for a different water type. Values in black represent theoretical values for pure water. Therefore, it is to be expected that pure water reflects less than Lake Superior water, which has various amounts of reflecting sediment and other constituents. The other three plots represent the backscatter at three locations. In the legend is given the ratio between the expected reflectance by the backscatter value and the noise equivalent reflectance of the SeaWiFS sensor at that band. This ratio shows where the NIR nil reflectance assumption of the established SeaWiFS atmospheric correction algorithm holds. If the ratio is less than one, then the signal given by the reflectance is too small to
be noticed by the sensor – it may as well be zero. If it is greater than one, then the signal can be detected, thus invalidating the nil reflectance assumption. As expected, pure water has ratio values less than one, and each of the Lake Superior points has a value greater than one for at least the 765nm band. (Note that this ratio is an oversimplification, as it does not account for atmospheric or geometric effects).

To find our measure of the water depth needed to show a uniform backscatter value, we plotted the backscatter at 765nm versus the depth of that data point, in Figure 3.3:

![Figure 3.3: Relationship between interpolated 765nm backscatter value and water depth, for four KITES data transects measured June 30, 1999 to July 16, 1999](image)

Although it is more obvious in some transects than others, backscatter seems to level off between values of 0.001 and 0.002. The depth at which this happens occurs between 125m and 250m. The rate of backscatter decline is not identical among sampling transects, as they each have their own unique bathymetry. The Eagle Harbor transect, in particular, stands out. This transect is somewhat unique in that it covers a region of rapidly increasing depth. Therefore, sediment typically restrained to shallower regions occurs in the deep regions of this transect as they are geographically close to shore.

Figure 3.4 is a bathymetry map of Lake Superior, with transect locations superimposed.
A threshold at 200m was used to create a binary LCW map. Regions with a greater depth than the threshold were given a value of one, representing Historical LCW regions, while everything else was left at zero. Figure 3.5 is the binary bathymetry threshold image used in the algorithm. To protect against scenarios where rapidly increasing depth means sediment travels into deep water regions, a morphological erosion has been applied to the bathymetry image.
3.4.3 Near Infrared (NIR) band reflectance of LCW (not in SeaDAS)

The next piece of external data we need to determine is the reflectance in the LCW regions of the lake. As mentioned previously, we are assuming the NIR reflectance to be a constant over the entire LCW region. The value of this constant is derived from ground measurements. A Hobi Labs HydroRad-4 hyperspectral oceanographic radiometer was used to determine LCW reflectance (HOBI Labs 2000). The HydroRad-4 can measure irradiance or scalar irradiance depending on the configuration of each channel, and various other IOP's and Apparent Optical Properties (AOP's) from combinations of these channels. We intend to measure the water reflectance, which is the ratio of the irradiance directed upwards by the irradiance directed downwards.

\[
\rho_w = \frac{E_u}{E_d}
\]

where \( \rho_w \) is the reflectance of the water at a specific point, \( E_u \) is the irradiance upwelling to that point, and \( E_d \) is the irradiance downwelling to that point.

The HydroRad-4 is an instrument that is lowered into the water so that optical parameters can be measured at different depths. However, we need to know the reflectance at the surface. To do this, we take several measurements slightly beneath the surface of the water, and extrapolate their values to determine an expected surface reflectance. Next, we divide the upwelling irradiance by the downwelling irradiance for every measured wavelength value. Figure 3.6 is the plot of \( \rho_w \) versus wavelength.
Figure 3.7: Hydro-Rad calculated reflectances, LCW regions

Figure 3.6 shows how the noise in the reflectance measurement increases dramatically as the reflectance values proceed farther into the NIR. This is due to the relatively large water absorption in the NIR, which depresses both the upwelled and downwelled radiance. (Technically, the sensor noise remains relatively constant across the spectrum, but the signal is decreasing, so it is the signal to noise ratio that is actually decreasing in the NIR). To account for this noise, the reflectance spectrum was averaged by a kernel nine spectral samples wide.

Now we have a measure of reflectance at every wavelength. Like the exo-atmospheric irradiance, we need to weight these reflectances by the spectral sensitivity of each NIR band. Figure 3.7 shows the weighted reflectance of SeaWiFS NIR bands 7 and 8.
Figure 3.7: LCW reflectance, weighted by SeaWiFS Spectral Sensitivity

Normalization is performed in the same manner as with the exo-atmospheric irradiance. The integral of a band’s spectral spectrally weighed reflectance is divided by the integral of the spectral response of that band.

$$
\rho_{wi} = \frac{\int s_i(\lambda) \left[ \frac{E_{\text{ex}}(\lambda)}{E_{\text{atm}}(\lambda)} \right] d\lambda}{\int s_i(\lambda)} 
$$

(24)

Note that the spectral character in the Band 8 plot shown above is most likely due to noise in the signal (see Figure 3.7) and not spectral reflectance character of the water.

3.4.4 Radiative Transfer Geometry

The geometry associated with radiative transfer changes in different regions of the image. To appropriately create a top of the atmosphere radiance for each LUT element, and to remove the effects of the atmosphere in the visible bands, we must have some means of determining the geometry in each part of the image, and some means for assessing the effect of that geometry.
3.4.4.1 Geometry definitions

Four parameters can be used to describe the radiative transfer associated with an image of a flat scene (a lake, for example). These parameters are the solar zenith angle, $\sigma$, the solar azimuth angle, $\phi_s$, the sensor zenith angle, $\theta$, and the sensor azimuth angle, $\phi_d$. The zenith angle is the declination angle between the normal to the earth surface and the angle of the radiation source (the sun) or the radiation detector (the satellite sensor). The azimuth angle is the ground projected direction of the solar or sensor declination, measured clockwise from due North. Figure 3.8 is a drawing of these angles (Schott 1997).

**Figure 3.8: Radiative Transfer Geometry**

3.4.4.2 Per-pixel geometry determination

Determining the geometry at every point in the image can be a complicated task. First, we need to know the time and location of the image, as it will tell us the solar geometry scenario. The SeaWiFS sensor is located on a platform in Low Earth Orbit (LEO) in descending mode. The equator crossing time is noon, and with a 98.9 minute period, the sensor passes over the lake at 11:47am Local Standard Time (see Appendix A for the calculation) (Gregg, et al. 1993).

A noon flyover time means that the sun is at the daily highest (lowest zenith angle). Therefore, the solar zenith angle is dependent upon the latitude of the ground point and
the time of the year. Since it is noon, the sun will be directly south, so the solar azimuth angle will be 180°. Likewise, the sensor zenith angle will depend on its angle of inclination, which for SeaWiFS is 20°. Also due to the noon fly over, the azimuth angle is 180°, turning to 0° after the sensor has passed.

However, this all applies only in the ground path of the sensor. Points off-axis will have changes in their geometry as their degree longitude distance from the ground path increases. (Actually, the geometry will also change with the degree latitude distance. As we increase or decrease in latitude, the solar zenith angle will change. This effect is relatively minor, and our image is larger in the longitude dimension. Therefore, it is ignored for the purposes of computational brevity.) The off-axis geometry change is described in Figure 3.9:

**Figure 3.9: Zenith and Azimuth angle calculation geometry**

![Diagram of Zenith and Azimuth angle calculation geometry]

The diagram in Figure 3.9 shows the geometry of our images. Point A is the location, in orbit, of our satellite. It is nadir to the ground location B, directly beneath it. The satellite is pointed in the direction of C. Lines BC and CE are both on the surface of the earth, while lines AB and CD are normal to the surface pointing up towards the satellite or
down to the core of the earth, respectively. The dotted line is the path of the satellite's forward motion, due north, and the shaded line is the direction of the sweeping motion of the scanning imager.

The geometry for point C, then, is relatively straightforward. The sensor zenith angle is the declination angle BAC (set at 20°) and the sensor azimuth angle is the angle from North at (N)CB. The off-axis scenario is highlighted in green. The longitude difference between point C and point E is given with the angle CDE. The sensor zenith angle is then BAE (greater than 20°) and the sensor azimuth angle is the angle from North (of E) at (N')EB.

In a given off-axis scenario, we know the angles CDE and BAC, and the length of lines AB and CD. Our zenith angle can be derived from BAE and our azimuth angle can be derived from BEC. We can determine both of these if we know the length of lines BC, BE and CE:

$$BC = AB \tan (BAC)$$

$$CE = \left( \frac{DC \pi \text{CDE}}{180} \right)$$ (the only part of our calculation where we don't assume a flat earth)  

$$BE = \sqrt{BC^2 + CE^2}$$

Our sensor azimuth angle is then:
\[ \theta = \tan^{-1}\left( \frac{BE}{AB} \right) \]
\[ \theta = \tan^{-1}\left( \frac{\sqrt{BC^2 + CE^2}}{AB} \right) \]
\[ \theta = \tan^{-1}\left( \frac{\sqrt{[AB \tan(BAC)]^2 + \left[ \frac{CD \pi CDE}{180} \right]^2}}{AB} \right) \]

given: \( BAC = 20^\circ, AB = 7.05 \times 10^5 \text{ m}, CD = 6.37 \times 10^6 \text{ m} \)
\[ \theta = \tan^{-1}\left( \sqrt{6.584 \times 10^{10} + \left[1.11 \times 10^5 \frac{CDE}{7.05 \times 10^5} \right]^2} \right) \]

(28)

Our sensor zenith angle is then:
\[ \phi_d = 90 + BEC \]
\[ \phi_d = 90 + \tan^{-1}\left( \frac{BC}{CE} \right) \]
\[ \phi_d = 90 + \tan^{-1}\left( \frac{AB \tan(BAC) 180}{CD \pi CDE} \right) \]

given: \( BAC = 20^\circ, AB = 7.05 \times 10^5 \text{ m}, CD = 6.37 \times 10^6 \text{ m} \)
\[ \phi_d = 90 + \tan^{-1}\left( \frac{2.309}{CDE} \right) \]

(29)

3.4.5 MODTRAN Look Up Table (LUT) creation for each geometric scenario
Given the geometric constraints of our imaging scenario, the MODTRAN radiative transfer model is used for inverse modeling (United States of America, Air Force Research Laboratory 1998). While the model use will be described more in depth below, its creation will be explained here.

MODTRAN was developed by the US Air Force to pool atmospheric propagation knowledge into a single software package. We used MODTRAN version 4.0 to create a
Look Up Table (LUT) of expected top of the atmosphere radiance values when given an LCW water reflectance and a variety of atmospheric conditions. These atmospheric conditions were chosen for their appropriateness to the Lake Superior region, and represent rural-continental, rather than marine conditions.

Four Look Up Tables were created for use in this algorithm. Each LUT represents the same sets of atmospheric parameters, but different geometric scenarios. The SeaWiFS images we are using are large enough that the solar and sensor geometry change significantly in different regions of the image. In an attempt to account for the different geometry of the image, we have created four LUT's which represent geometric differences between points of identical latitude but changing longitude. This is in the interest of computational brevity, as creating LUT's for both latitude and longitude changes would require weeks of computational effort.

We are more concerned about longitude changes than latitude changes for two reasons. First, our Lake Superior image spans more degrees in the longitude dimension than the latitude dimension. Lake Superior spans from about 97°W to about 83°W, but only from about 46°N to 49°N. Second, sensor geometry is more apt to vary in the longitude dimension than the latitude dimension. Since SeaWiFS is in a polar orbit, the latitudinal declination is apt to remain constant for points on the ground, while longitudinal declination will vary with the distance from the sensor’s ground path.

The four LUT geometric configurations represent a point directly on the sensor ground track (0°), and off that track by three longitude degree increments (3°, 6° and 9°). This range was chosen because it adequately represents the maximum degree off the ground track (9°) that a region in a usable SeaWiFS image could have. The total longitudinal width of the image is about 14°, but the image must be relatively centered so that the edges are not lost. The maximum longitudinal distance between the center of the lake and the ground track was determined to be about 3°. It is important to note that the geometric effect of a longitudinal shift is assumed identical regardless of its direction. This is
reasonable because angular changes are identical in either direction and symmetric scattering functions are assumed.

Once the algorithm is run, the four LUT’s are resampled so that a new LUT is created for each column in the image. This, in effect, creates a LUT for each pixel that estimates the geometric effects for its distance from the sensor ground track.

The atmospheric parameters used in the model were chosen from statistical analysis of meteorological data supplied with SeaWiFS imagery. Thirty SeaWiFS meteorological data files from 1998 and 1999 were chosen. These meteorological files were from the summer months, and represent the time of the year of optimum SeaWiFS data usage. Table 3.2 is a table of these meteorological data values:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure</td>
<td>1006.61</td>
<td>25.15</td>
<td>Millibars</td>
</tr>
<tr>
<td>Precipitable Water</td>
<td>18.39</td>
<td>7.68</td>
<td>Kg per m²</td>
</tr>
</tbody>
</table>

Therefore, a range of two Standard Deviations about the mean should be sufficient to model 95% of all possible atmospheric scenarios. Table 3.3 shows the parameters actually used to create the LUT’s.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value range</th>
<th>Number of increments</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aerosol Visibility</td>
<td>1.0-100.0</td>
<td>3</td>
<td>percentage</td>
</tr>
<tr>
<td>Elevation</td>
<td>0.015-2.25</td>
<td>3</td>
<td>km</td>
</tr>
<tr>
<td>Water Vapor</td>
<td>0.05-2.25</td>
<td>3</td>
<td>g per cm²</td>
</tr>
</tbody>
</table>

Each LUT has only twenty-seven possible combinations of these parameters, thus limiting the resolution of different scenarios. The small size, however, does allow for a computationally efficient proof of concept.
Several parameters were held constant across all LUT scenarios. These included solar/sensor geometry (a separate LUT was created for each geometric scenario), global location, viewing date and time, and aerosol type. Table 3.4 is a table of these constants:

**Table 3.4: LUT parameter constants**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Crossing Time</td>
<td>11:47 local time, 17.783 GMT</td>
</tr>
<tr>
<td>Longitude</td>
<td>88°W</td>
</tr>
<tr>
<td>Latitude</td>
<td>47.5°N</td>
</tr>
<tr>
<td>Atmospheric Type</td>
<td>Continental Rural</td>
</tr>
<tr>
<td>Sensor Altitude</td>
<td>705km</td>
</tr>
<tr>
<td>Day of Year</td>
<td>214</td>
</tr>
</tbody>
</table>

The geometry associated with each LUT is calculated for points 3, 6, and 9 longitude degrees west or east of the sensor ground path. The details of these calculations is presented in section 3.4.4, and their results are shown in Table 3.5.

**Table 3.5: LUT geometry**

<table>
<thead>
<tr>
<th>Longitude degree distance from sensor ground path</th>
<th>Sensor Zenith Angle</th>
<th>Sensor Azimuth Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°</td>
<td>160°</td>
<td>180°</td>
</tr>
<tr>
<td>3°</td>
<td>149.167°</td>
<td>127.572°</td>
</tr>
<tr>
<td>6°</td>
<td>134.608°</td>
<td>111.40°</td>
</tr>
<tr>
<td>9°</td>
<td>124.313°</td>
<td>104.383°</td>
</tr>
</tbody>
</table>

Angles are presented in this table in "MODTRAN type" format. The zenith angle is measured from the line drawn at normal to the ground surface to the direction of the sensor observation. Therefore, nadir looking instrument will have a sensor zenith angle of 180°. The azimuth angle is measured clockwise from due north to the ground projection of the sensor/source location on the ground. Therefore, an object with a zenith due east of the ground point will have an azimuth angle of 90°.

Finally, it is important to note that the solar geometry is held constant for each LUT, and is determined from the ground spot location, the time of day and day of year.
3.5 The Alt_SeaWiFS software

At this point we have defined the parameters needed for Alt_SeaWiFS. Now the steps of the software will be explained in detail.

3.5.1 Alt_SeaWiFS Software, Part One

Instead of prompting the user for a number of files to be used for processing, Alt_SeaWiFS uses a text file where all input and output filenames are specified. This is the MASTER_FILE, and is where user supplied variables, such as the input filename, are stored. (See Appendix B).

3.5.1.1 Opening SeaWiFS files (‘openfiles’ function)

Prior to any data processing in the software, images must be read into arrays of variables. SeaWiFS images are stored in the scientific data format called Hierarchical Data Format (HDF), which allows several types of data, along with their annotation and units, in a single file (Fanning 1999) (NCSA 2000). A standard L1B SeaWiFS HDF file contains the eight band image data, arrays of geographic reference data, telemetry and other sensor data, image data quality flags, and so on. To conserve computer memory, not all data contained within an HDF file is opened and read, rather, parts are opened, processed, and output again when finished. The ‘openfiles’ function, then, opens the MASTER_FILE, reads the variables, opens the HDF image, and queries simple image related data from the HDF file, such as size, arrays of latitude and longitude values corresponding to image pixels, and so on. The outputs from these queries are passed in variable form to the main program.

3.5.1.2 Determining Lake Superior Region (‘makemaskregion’ function)

When the atmospheric correction software is started, the image data is immediately read into ENVI image variables. This does not mean that the data has been read into memory, rather, the data's location is acknowledged to facilitate a quick transfer to memory.
SeaWiFS images, intended for oceanographic studies, can be quite large. Most files are about 150-200Mb in size, and contain eight band images that are about 3000x1500 pixels. The Lake Superior region in one of these images can be comparatively small - about 500x300 pixels. Therefore, it makes sense to only read the Lake Superior region of the image into data variables, and thus reduce memory usage and computation time.

However, the Lake Superior region of a SeaWiFS image is not always located in the same image area. Therefore, we must use the geographic data contained within the SeaWiFS HDF file to create a mask of the Lake Superior region. Data evaluated with the atmospheric correction software will be limited to areas within the Lake Superior region mask.

The six SeaWiFS HDF geographic data elements each contain a one-dimensional floating point array. Each array contains the longitude or latitude of the column of image pixels at the start, middle or end of each scan line. Figure 3.10 is a graphical description.

**Figure 3.10: Geographic data element description**

![Geographic data element description](image)

Using the geographic data arrays, a two dimensional array is created for both latitude and longitude values that has the same dimensions as the image and contains either a latitude or longitude for every pixel. Each 2D array is created on a (horizontal) scan line by scan line basis. Each line in the 2D array contains three data points - a latitude or longitude for the starting pixel, the middle pixel and the ending pixel. Pixels between these data points
are filled by interpolating using a second order polynomial fit. This is repeated for every scan line in both the latitude and longitude data arrays.

Once the latitude and longitude arrays have been constructed, creating a "region mask" is simply a matter of creating a binary image where a value of 1 is assigned to pixels that contain latitude and longitude values within some specified (in the MASTER_FILE) range, and a value of 0 is set for elsewhere.

The resulting binary image contains a polygon that may not have horizontal or vertical edges. To simplify data input and output, the polygon is extended so that it has edges that are horizontal and vertical. This allows the description of the image region to be reduced to four values: west edge pixel location, east edge pixel location, north edge pixel location and south edge pixel location. An example of both the original region polygon, and its extended area, is given in Figure 3.11.

**Figure 3.11: Lake Superior region masks**
The ‘makemaskregion’ function outputs to the main program a structure containing the value of the edges of this calculated Lake Superior region. These values are used upon opening new image files – only the region they specify is loaded.

3.5.1.3 Land Mask Determination (‘makemaskland’ function)

Once we have found the Lake Superior region, we must identify which pixels represent water, and which represent land or clouds. In SeaWiFS band eight (centered at 865nm), the reflected radiance varies greatly between water pixels and cloud or land pixels. This makes it easy to use a frequency histogram to select a threshold value between the expected water value and the expected land/cloud value. Figures 3.12 – 3.16 show, graphically, how this threshold selection occurs.

Figure 3.12 is a theoretical smoothed histogram of band 8 values. Smoothing is performed to remove noise and center the highest points of each pixel type about the center range of their frequencies.

**Figure 3.12: Smoothed Histogram**

*Frequency Histogram of Band 8 (865nm)*

The next step of the thresholding algorithm is to choose the first maximum point – the pixel value with the highest frequency. This value is the center of pixel group #1 (in this case, land/cloud pixels).
Figure 3.13: Finding Maximum Point # 1

Next, we need to find the center of pixel group #2 (water pixels). We cannot simply find the next highest frequency value, as it may be a point representing group #1 pixels. Therefore, we suppress all values within some range, [x], of the value [max #1] by setting them to zero.

Figure 3.14: Suppression of group # 1 pixels

Now we can find the highest frequency value and expect it to represent pixels in group #2.
Now we can determine the optimum pixel value to set as a threshold to distinguish between groups one and two. This value is chosen as the point equidistant from \( \text{max #1} \) and \( \text{max #2} \).

Careful selection of the value \( [x] \) is important for proper determination of the threshold value. If \( [x] \) is too small, then the pixels in group #1 may not all be suppressed, and \( \text{max #2} \) may be attributed to a group #1, rather than group #2 pixel value. To account for this
problem, the Alt_SeaWiFS software has predetermined ranges for $[\text{max } \#1]$ and $[\text{max } \#2]$. If either max falls outside its range, or if the distance between each maximum is smaller than some value, a default threshold value is used. This default value was chosen as an average of threshold values for several sets of images.

Once the threshold value has been chosen, a simple thresholding operation is applied to the band 8 image. A binary result is created, where image pixels greater than the threshold are set to zero, and image pixels less than the threshold are set to one.

The 'makemaskland' function finishes by exporting the land mask image to an ENVI variable, to be used by the user for registration, and by subsequent functions in their processing.

3.5.1.4 Opening and preparing the Historical LCW image ('openlcw' function)
The Historical LCW image, derived from a bathymetry file, consists of two binary images in HDF format. The first image is a binary land mask image, similar to the land mask image produced above. The difference lies in the projection of each image - the image land mask shows a projection due to sensor geometry, while the LCW image shows a geospatially registered projection. The 'openlcw' function exports a binary land mask from the Historical LCW file to ENVI to use for user assisted registration, and a binary deep water mask, where values of one represent pixels deeper than 200m.

3.5.1.5 Ending Part One of Alt_SeaWiFS software.
Part One of the Alt_SeaWiFS software ends by saving all opened variables and compiled routines to a temporary file. Also, a text instruction file is opened to aid the user in the next step, image to image registration.

3.5.2 User Assisted Image to Image Registration
Ideally, a lengthy software processing scheme should require user input at one stage only: the beginning. This would enable the user to start the process and then leave to work on other things. This was one of the design goals of the Alt_SeaWiFS software. An
automated solution was found for almost every element of the algorithm. One step, however, is most accurate with some form of user interaction: image to image registration.

The Historical LCW landmap (presented in section 3.5.1.3: Land Mask Determination) is a two-dimensional data set. The value associated with each pixel location in the landmap must have a corresponding pixel use point in the imagery being processed by Alt_SeaWiFS. However, due to the geometry of orbital remote sensing, the spatial representation of our image area by SeaWiFS may be different from standard geospatial representation. The result is a Historical LCW landmap that is skewed differently than the SeaWiFS image. If we are to combine these two data sets, we must warp one so its data points register with the other.

Some effort was put into finding a way to automate the registration process. However, a robust technique, that was immune to effects of clouds, accurate to within several pixels, and computationally possible, was not found. Perhaps another thesis could be devoted to this very interesting problem. Since we can accomplish our registration goals using user input, this approach was taken.

Unfortunately, this user input cannot be performed prior to the execution of the Alt_SeaWiFS algorithm. Both the SeaWiFS image and the Historical LCW image must be prepared for user registration. Therefore, the Alt_SeaWiFS algorithm operates in two parts. Part one reads the image files and creates binary land mask images to use for image to image registration. The software then saves all variables, and pauses to allow the user to register the images and save his or her result. Part two then performs the rest of the atmospheric correction. Part one should take about ten minutes to process on a Sun Microsystems Ultra-Sparc 10 workstation, and part two about 45 minutes.

A second user driven function was added to the Alt_SeaWiFS software at a later date. Since was land mask routine was not successful (see results) the option for user tweaking
of the threshold value used for masking was given. This routine operates prior to the completion of the first stage of Alt_SeaWiFS, and is explained in more detail below.

Originally, we intended to geospatially register the SeaWiFS image to ground map units (the Historical LCW image is geospatially registered). This presents a problem, however, when we output our data. Standard SeaWiFS Level 2 images are not geospatially registered, so if we want to make our output look like standard output, we must reverse our geospatial registration prior to finishing. It makes more sense, then, to warp our geospatially registered Historical LCW image to the particular registration of the image being processed. That way we can output our data in the form of a standard processed L2 image, yet still use Historical LCW imagery.

Image to image registration is performed using ENVI tools. The user selects nine or more concurrent points on both images. These points are chosen for their known locations - islands, edges of bays, peninsulas, and other easily identifiable points. The user selects the Ground Control Points (GCP's) and saves them to an ASCII file, to be used in part two of Alt_SeaWiFS.

3.5.3 Alt_SeaWiFS Software, Part Two

Now that part one of Alt_SeaWiFS has been completed, we have constructed a land mask, prepared the LCW bathymetry image for warping, and opened and prepared all the data from the HDF files. Now we are ready to start part two, the meat of the algorithm. Part two is more computationally intensive than part one, taking about forty-five minutes to complete on a Sun Microsystems Ultra Sparc 10. When started, the software re-reads the MASTER_FILE and reloads the variables and compiled modules from part one.

3.5.3.1 Performing image registration ('reg_lcw' function)

Between steps one and two, the user selects nine or more GCP's to use for image registration. These GCP's are used in this step to actually perform the registration. However, the user selected GCP's to register the LCW land map to the image land map. The algorithm, however, needs LCW bathymetry data. Image registration is actually performed on the LCW bathymetry image, but using GCP's selected by the user for the
LCW landmask image. The LCW landmask and LCW bathymetry images represent the same projections, so a registration performed on one would apply the same to the other. The product of the 'reg_lcw' function is a LCW bathymetry map projected to the warp of our SeaWiFS image.

3.5.3.2 ISODATA classification ('makeclassmap' function)
The first step in finding the full extent of LCW regions is to perform an unsupervised clustering algorithm on the multispectral image. This will group regions of like spectral character. This spectral character represents the effects of both atmospheric and hydrospheric constituents. Our goal is to identify regions of like atmospheric character, so spectral bands that best show atmospheric character (NIR bands 7 and 8) will be the only bands used in the clustering algorithm. Originally, band 1 (centered about 412nm), was also going to be used for classification. It is well known that, like the NIR bands, band 1 reflectance should be relatively independent from hydrospheric constituent variances. Unfortunately, band 1 is very susceptible to atmospheric multiple scattering effects. This means that differences in radiative transfer geometry will produce significant changes to the band 1 top of the atmosphere (TOA) radiance. Since a SeaWiFS image contains a range of geometric scenarios (see section 3.4.4: Radiative Transfer Geometry), a band 1 image will show significant geometric effects upon radiance. Therefore, much of the variance in data values in a band 1 image is not due to atmospheric effects. Since variance due to atmospheric effects best helps us classify the different atmospheric regions in the lake, band 1 was not used.

The ISODATA unsupervised classification method is used to perform the clustering. Rather than writing a new ISODATA classification algorithm, the ISODATA classification algorithm supplied with ENVI was used. Several of its characteristics influenced its choice. First, ISODATA is an unsupervised classification routine, and does not fit spectral data to known material spectra. Instead, ISODATA classifies image data upon its variance alone, which is ideal since we have no external spectral data. Second, spectral character changes may not be represented by sharp boundaries, the ISODATA algorithm is useful in its ability to identify regions of gradual change. The algorithm can
select a variable number of classes, which is useful with changing ranges of spectral data in different images. The final classification result must be a series of classes that contain connected pixels. This is necessary because pixels that have similar spectral character but are not near each other may be the result of different atmospheric and hydrospheric constituents. Therefore, a connected components algorithm must be performed, so that classes that contain several regions are split into separate groups. (see Appendix D: Classification) An example of a classified image is presented in Figure 3.17.

Figure 3.17: An image of clusters with like spectral character

Note how the shape of each cluster shows the type of constituent that dominates spectral character. The Southwestern regions of the lake are overwhelmed with spectral character due to turbid water, and thus show shapes that look like turbidity patterns. The East-central regions are dominated by atmospheric constituents and are shaped like atmospheric patterns.

To create the image used in Figure 3.17, the ISODATA classification algorithm was performed on the 765nm and 865nm bands with equal weighting. The parameters chosen for the ISODATA classification routine are shown in Table 3.6:
Table 3.6: ISODATA classification parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$, number of starting classes</td>
<td>75</td>
</tr>
<tr>
<td>$\theta_s$, the maximum standard deviation value for each class</td>
<td>Calculated from image statistics - equal to standard deviation of total image, divided by ten.</td>
</tr>
<tr>
<td>Minimum number of classes</td>
<td>60</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>5</td>
</tr>
<tr>
<td>$\theta_N$, Minimum number of data points in a class</td>
<td>10</td>
</tr>
<tr>
<td>Maximum percentage of pixels that can change classes per iteration</td>
<td>20%</td>
</tr>
<tr>
<td>$\theta_C$, Maximum allowable distance between two class centers before they are 'lumped' together</td>
<td>Calculated from image statistics - equal to standard deviation of total image, divided by one hundred.</td>
</tr>
<tr>
<td>Maximum number of classes that can be merged in a single iteration</td>
<td>2</td>
</tr>
</tbody>
</table>

The above classification parameters were chosen for their ability to produce an image with a large number of classes, while avoiding the effects of noise. Next, a clumping algorithm was used to remove small, isolated pixels of a particular class, and assign those pixels to adjacent classes. This clumping algorithm was also supplied with ENVI as part of a suite of post-classification tools. A connected components algorithm was used to split non-adjacent class segments into separate clusters. This was essential, as each class is intended to represent a particular hydrospheric and atmospheric optical scenario. We are confident that this scenario remains constant if class pixels are connected. However, we cannot be sure that a pixel set as one class in one region of the image would always represent the same hydrospheric and atmospheric scenario in a different region of the lake. Rather, the total spectral values producing a particular class may represent equal summations of different atmospheric effects. Therefore, pixels identified as a particular class that are not connected are split into two separate classes.
The product of the 'makeclassmap' function is an image of integer values, where the value assigned to each pixel is a unique class identification number.

3.5.3.3 Merging classified image and Historical LCW image ('merge_class' function)

The image has now been divided into like clusters. However, we still do not know which clusters represent LCW and the effects of the atmosphere above that region, and which do not. To do this we must also utilize the binary Historical LCW image to derive a map of LCW and non-LCW clusters.

This data merging was performed by multiplying the binary Historical LCW image by the image produced from the 'makeclassmap' function. The values in the image result are the classes that represent LCW regions. The LCW region is the total area encompassed by each class left after the above image multiplication. Essentially, if a particular cluster touches on a historically known LCW region, the entire cluster will be designated as representing an LCW region. In this way the LCW regions of the lake will be expanded beyond the historically known LCW regions. Therefore, it is important to conservatively estimate historical LCW regions. Figure 3.18 graphically shows the above process.
Figure 3.18: The LCW region identification process

An arbitrary "historical" LCW map (top left) is combined with the classified image from Figure 3 to form a map consisting of classes that touch LCW regions.

Correct identification of LCW regions is very important. Non-LCW regions that are labeled as LCW regions will be expected to have the same water leaving radiance as true LCW regions. Therefore, their atmospheric compensation solution will be incorrect. The error may not be limited to just that region, however, since it will be used to extrapolate a
solution to areas identified as non-LCW. LCW regions that are identified as non-LCW will adversely affect the total solution, requiring a longer extrapolation from LCW to non-LCW regions.

3.5.3.4 Atmospheric Spectra Extrapolation ('extrap' function)
Once we have identified the LCW regions in an image, we know where we can find a solution with our methods, and where we need to extrapolate this solution. Originally, we intended to find the solution in the LCW regions, then extrapolate that solution to non-LCW regions. However, this presents several problems. First, how does one extrapolate a solution that is not numeric, but a combination of atmosphere mixtures and magnitudes? Second, and perhaps more important, how can we be sure that the relationship between solution and spectral character is linear? Changing from atmosphere type A to atmosphere type B may represent a drastic change in atmosphere radiance effect when the same change between atmosphere types C and D may not. For this reason, we defer finding a solution in LCW regions until we extrapolate the data used to find it to non-LCW regions.

The atmospheric scenario for a particular LUT pixel is determined by fitting a MODTRAN LUT to the spectra of the NIR bands in LCW regions. Since we cannot make this fit in non-LCW regions of the image, we determine the atmospheric LUT value by extrapolating NIR spectra from LCW regions to non-LCW regions, and fitting the MODTRAN LUT to these extrapolated values.

The extrapolation is performed using a Gaussian weighting function shaped by the direction of the wind. Essentially, the value of each pixel in non-LCW regions is determined by multiplying a 2D Gaussian function, centered at the pixel in question, by the LCW data for the image. The non-LCW pixel value is the integral of this result, normalized by the integral of the Gaussian function (in LCW regions only) itself. Figure 3.19 is a diagram of this extrapolation.
To find the value of the center pixel, denoted as a red star in Figure 3.19, a 2D Gaussian function is multiplied by the LCW pixel values in the image. The double integral of these values is divided by the double integral of the Gaussian values in LCW regions. This is expressed mathematically as:

$$nonLCW(x,y) = \frac{\int_{i=0}^{m-1} \int_{j=0}^{n-1} GAUS\left(\frac{i-x}{b}\right)GAUS\left(\frac{j-y}{d}\right)LCW(i,j)didi}{\int_{i=0}^{m-1} \int_{j=0}^{n-1} GAUS\left(\frac{i-x}{b}\right)GAUS\left(\frac{j-y}{d}\right)binLCW(i,j)didi}$$

(30)

where $nonLCW(x,y)$ is the value calculated for a with a particular non-LCW water pixel, $m$ is size of the image in the $x$ dimension, $n$ is the size of the image in the $y$ dimension, $GAUS((i-x)/b)$ is the value of a Gaussian function centered at point $x$ and scaled by $b$ for location $i$, $LCW(i,j)$ is the value of a LCW pixel (set to zero for land and non LCW regions) and $binLCW(x,y)$ is a binary image, equal to one in LCW regions and zero in land and nonLCW regions.

The shape of the Gaussian function, controlled by the scaling factors $b$ and $d$, is determined by the wind direction and magnitude for the image. We assume that the atmospheric type and magnitude upwind or downwind from the atmosphere of a
particular location is more apt to be similar than the atmosphere in a lateral direction. Therefore, the wind direction, calculated from meteorological data included with the SeaWiFS image, is used to aid the LCW data extrapolation. It is important to note that the spatial resolution of our meteorological data is on the order of one degree latitude and longitude, so $b$ and $d$ are held constants across the image.

Extrapolation is repeated for every water pixel in non-LCW regions of the lake, for both NIR band images. Obviously, the farther the extrapolation from LCW to non-LCW regions of the image, the greater the potential error in the solution. Unfortunately, implementation of equation (30), for the entire non-LCW pixel regions of the lake is prohibitively computationally intensive. Therefore, non-LCW pixel values are calculated for only a select number of pixels, and the results are interpolated between all non-LCW pixels. The number of non-LCW pixels selected for extrapolation directly affects the computation time and extrapolation quality. The Alt_SeaWiFS software user has the option of selecting several extrapolation modes in the MASTER_FILE, including a slow, accurate extrapolation and a fast, less accurate extrapolation.

3.5.3.5 MODTRAN LUT processing ('make_searchable_lut' function)

Once the spectral data from LUT regions have been extrapolated to all areas of the lake, that data must be converted to an atmospheric type and magnitude. This is to be performed using inverse modeling techniques and the MODTRAN atmospheric propagation model.

As stated previously, we have created several MODTRAN LUT's for different geometric viewing scenarios. We use these LUT's for two purposes. First, we create a LUT that represents expected top of the atmosphere (TOA) radiances for each atmospheric LUT scenario in NIR regions. This is used to determine the atmospheric LUT scenario for each pixel in the image. Using the same LUT's, we correct for the atmospheric effects in the visible (VIS) bands for each pixel using our knowledge of the atmosphere there.
Our MODTRAN LUT contains radiative transfer data about several optical atmospheric parameters. Parameters used in our processing are expressed in Table 3.7:

### Table 3.7: LUT data elements

<table>
<thead>
<tr>
<th>LUT parameter name</th>
<th>LUT parameter calculation and use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Reflected Radiance, $L_r$</td>
<td>TOA radiance from direct reflection of solar radiance off a 100% ground reflector. (see equation (27))</td>
</tr>
<tr>
<td>Tau-2, $t_2$</td>
<td>The atmospheric transmittance from the ground to the sensor.</td>
</tr>
<tr>
<td>Upwelled radiance, $L_u$</td>
<td>The radiance scattered in the atmosphere to the sensor.</td>
</tr>
<tr>
<td>Downwelled radiance, $L_d$</td>
<td>The solar radiance scattered in the atmosphere to the ground target (Skylight).</td>
</tr>
</tbody>
</table>

Two other parameters used in our LUT processing, but not supplied by the LUT itself, is the solar zenith angle ($\sigma$) and the sensor zenith angle ($\theta$) for each location. These are calculated for each location (as shown in section 3.4.4: Radiative Transfer Geometry).

Prior to our direct LUT processing, we must derive the nadir transmission from the direct reflected radiance term, $L_r$:

$$
L_r = \frac{E_s \cos(\theta) t_1 t_2 r}{\pi}
$$

(31)

where $E_s$ is the exo-atmospheric irradiance, $t_1$ is the transmission through the atmosphere from the sun to the ground target, $t_2$ is the transmission through the atmosphere from the ground to the sensor, and $r$ is the ground reflectance (calculated by the LUT with a value of 1.0). The nadir transmission is the transmission through the smallest distance of the atmosphere: normal to the ground surface. It is derived as follows:

The atmospheric transmission is defined as the exponent of the negative optical depth of the atmosphere (Schott 1997).
The optical depth is the product of the atmospheric absorption coefficient times the transmission distance.

\[ \delta = \beta_a z \]  

(33)

where \( \beta_a \) is the absorption coefficient of the atmosphere, in units of inverse meters [m\(^{-1}\)] and \( z \) is the length, in meters, of atmospheric transmission.

Combining (32) and (33) we have a definition of atmospheric transmission as follows:

\[ t = e^{-\beta_a z} \]  

(34)

The path length of the atmosphere increases with the zenith angle.

\[ z' = z \sec(\theta) \]  

(35)

where \( z' \) is the length, in meters, of the ground to sensor path, where the sensor has a zenith angle of \( \theta \). The ground to sensor transmission, then, is a combination of (34) and (35).

\[ t_2 = e^{-\beta_a z} = e^{-\beta_a z \sec(\theta)} \]  

(36)

Therefore, the relationship between \( t_2 \) and \( t \) can be defined by combining equations (34) and (36).

\[ t_2 = e^{\ln(t) \sec(\theta)} \]  

(37)

We need to calculate \( t \) from our equation for \( L_r \), so equation (31) becomes:

\[ L_r = \frac{E_s \cos(\theta) e^{\ln(t) \sec(\theta)} e^{\ln(t) \sec(\theta)}}{\pi} \]  

(38)

note how \( r \) is no longer included, since it has no effect. Solving for \( t \), we have
Knowing $t$ allows us to actually calculate $t_1$ and $t_2$, which are not included in our LUT. (see equation (37), use as an analogy for $t_1$).

We also need to calculate the reflectance of the wind driven foam on the surface of the water. The latest version of SeaWiFS white cap reflectance derivation is (Frouin et al. 1996).

$$r_{wc} = 1.6225 \times 10^{-7} w_s^{3.52} \quad (40)$$

Where $r_{wc}$ is the white cap reflectance, and $w_s$ is the wind speed, in meters per second.

Now we have all the elements to create our radiative transfer equation. Our radiative transfer equation is

$$L_{Tn}(\lambda_i) = L_{rn}(\lambda_i)r_w(\lambda_i) + L_{un}(\lambda_i) + L_{dn}(\lambda_i)t_{2n}(\lambda_i) + L_{rn}(\lambda_i)t_{2n}(\lambda_i) + L_{dn}(\lambda_i)r_{wc}t_{2n}(\lambda_i) \quad (41)$$

where $L_T$ is the total TOA radiance for NIR band $\lambda_i$ and LUT element $n$, $L_r$ is the direct reflected radiance for NIR band $\lambda_i$ and LUT element $n$, $r_w$ is the LCW reflectance as calculated in section 3.4.3, $L_u$ is the upwelled radiance for NIR band $\lambda_i$ and LUT element $n$, $L_d$ is the downwelled radiance for NIR band $\lambda_i$ and LUT element $n$, $t_2$ is the transmission from equations (37) and (39), and $r_{wc}$ is the white cap reflectance from (40).

Equation (39) allows us to predict the expected TOA radiance for each atmospheric scenario. Calculations for each LUT element are performed, creating our 'searchable' LUT, used to match to scene TOA radiances to determine atmospheric type.

In practice, a 'searchable' LUT is calculated for each geometric scenario. This involves making a LUT's for each column in the image. (Recall that we are accounting for
variations in radiative transfer geometry in the longitude dimension only.) The average longitude distance from the sensor ground track is calculated for each column of pixels. Using a second order polynomial fit, the value for each LUT element at every geometric scenario is determined from the four input LUT's (0°, 3°, 6°, 9°). The output of the 'make_searchable_lut' is really a number of LUT's (one for each data column) that each account for the geometry of that column of pixels and is presented in total, TOA radiance. Figure 3.20 is an illustration of LUT usage.

**Figure 3.20: LUT Interpolation**

3.5.3.6 Fitting LUT's to image data ('match_data_to_lut' procedure)

The extrapolated image data now is fit to the 'searchable' MODTRAN LUT's. While there are several methods of LUT fitting, such as the AMOEBA algorithm (Press, 1986), we only need to fit the two NIR bands to the LUT. Therefore, we can take the simplified approach of calculating the total error between the scene NIR band values and the values for each LUT member. The LUT member with the smallest error is then chosen as the
most appropriate atmospheric type and magnitude. The LUT element value for each image region will be stored in an image of LUT element subscripts, as shown in Figure 3.21.

Figure 3.21: LUT type chosen for each image region

Figure 3.21: LUT type chosen for each image region

(note: image color denotes LUT subscript number only)

3.5.3.7 Removing Atmospheric Effects from VIS bands ('correct_image' function)
Now that an atmospheric type has been chosen for every pixel in the lake, its effects in the VIS bands must be removed. This is done by again opening the four input LUT's and calculating the sensor and solar zenith angles for each image column. Formula (39) is again used, but this time it is solved for $r_w$ and calculated for each VIS band for the LUT type determined in the previous step.

$$ r_w(\lambda_i) = \frac{L_{Tn}(\lambda_i) - L_{un}(\lambda_i) - L_{rn}(\lambda_i)r_{wc} - L_{dn}(\lambda_i)r_{wc}t_{2n}(\lambda_i)}{L_{rn}(\lambda_i) + L_{dn}(\lambda_i)t_{2n}(\lambda_i)} $$

(42)

where the units are defined in the same manner as in (41) but are calculated for visible bands, $\lambda_i$. 
Standard SeaWiFS L2 output is in radiance, rather than reflectance, units, so we must convert using equation (1):

\[ L_w(\lambda_i) = \frac{r_w(\lambda_i)E_o \cos(\sigma)}{\pi} \]

The output from the 'correct_image' function is an atmospherically corrected image for each of the six VIS bands.

3.5.3.8 Bio-optical parameter creation ('make_bio_opt_imgs' function)
Standard SeaWiFS L2 images contain several bio-optical parameter images. To adequately mimic the standard SeaWiFS L2 output, the Alt_SeaWiFS algorithm must also produce these bio-optical parameter images. However, no new research was directed to these bio-optical parameter algorithms. The latest SeaWiFS algorithms were used.

The first bio-optical parameter that was created was the CZCS type pigment. CZCS pigment is calculated using the following formula:

\[ CZCSpig = e^{0.696 - 2.085 \ln \left( \frac{L_w(band 3)}{L_w(band 5)} \right)} \]

where the CZCS pigment is in units of [mg m\(^{-3}\)], and is calculated using the water leaving radiance \( L_w \) for band three (centered at 490nm) and band five (centered at 555nm). If the calculated value for CZCS pigment is less than 2.0 [mg m\(^{-3}\)], then the equation for CZCS pigment changes to:

\[ CZCSpig' = 1.280 \left[ CZCSpig \left( \frac{L_w(band 2)}{L_w(band 5)} - 0.163 \right) \right]^{(1/2)} \]

where \( CZCSpig' \) is the new CZCS pigment value, and band two is centered about 443nm. (Aiken, et. al. 1995)
The diffuse attenuation coefficient (see section 2.2: Inherent Optical Properties) at 490nm (band 3) is calculated from a ratio of bands three and five. (Hooker et. al. 1999)

\[
K(490) = 0.016 + 0.15645 \left[ \frac{L_w(\text{band3})}{L_w(\text{band5})} \right]^{-1.5401}
\]  

(46)

The Chlorophyll-a algorithm used in Alt_SeaWiFS was version four of the OC4 algorithm (Barnes, 2000):

\[
Chl - a = 10^{[0.366 - 3.067R + 1.930R^2 + 0.649R^3 - 1.532R^4]}
\]

(47)

where Chl-a is expressed in units of [\mu g / L], and:

\[
R = a \log_{10} \left[ \frac{L_w(\text{band2}) > L_w(\text{band3}) > L_w(\text{band4})}{L_w(\text{band5})} \right]
\]

(48)

Finally, an Epsilon ratio image between the two NIR bands is calculated for L2 output:

\[
\varepsilon = \frac{L_T(\text{band7})}{L_T(\text{band8})}
\]

(49)

3.5.3.9 Output to HDF file ('out_to_hdf' function)

The final step of the Alt_SeaWiFS software is to output the calculated data to a SeaWiFS type L2 HDF file. This was done by simulating the data sets produced by the established algorithm and filling them with images produced from Alt_SeaWiFS. This is implemented in three steps.

First, data sets that remain the same after processing from L1B to L2 imagery are simply copied to the new HDF file. Examples of these data sets are the sensor telemetry, date of capture, and so on.
Next, non-imagery data sets that do change from L1B to L2 processing were created and saved to the HDF file. Examples of these data sets include logs of L2 processing software version and time, and so on.

Finally, imagery calculated from the Alt_SeaWiFS, such as ground radiance, CZCS pigment and so on are saved to the HDF file.

3.5.4 Alt_SeaWiFS conclusion

When Alt_SeaWiFS software completes its processing, a success report is given for user analysis, and if chosen in the MASTER_FILE, images are output to ENVI variables that the user can use to assess the quality of the atmospheric correction. The next section, the Results, will describe how these output variables are used.
4.0 Results

The atmospheric correction validity for the Alt_SeaWiFS was examined three different ways. The first method was a qualitative examination of processing results after each step of the software. This allows us to determine if trends in the data are being recognized, if the results are within expected ranges, and help identify the sources of various problems. Next, the results from images of several consecutive days were compared qualitatively. The quantity of reflecting water constituents should not change radically over a short span of days. Therefore, if the atmospheric compensation algorithm is working properly, the water leaving reflectance should remain constant. Finally, the results from an image were compared directly to ground measurements taken at the same time as the satellite overpass.

4.1 Qualitative Data Assessment

Our qualitative assessment of the Alt_SeaWiFS algorithm consists of examining the imagery produced at every step. Two image were chosen for this purpose. An image from day 94 of 1998 was used because it represents a best case atmospheric scenario: virtually no cloud obscuration, no obvious high atmospheric aerosol concentration plumes, and no airplane contrails. The image from day 99 of the same year was not so ideal. It has a large section obscured by clouds, numerous airplane contrails and high aerosol concentration plumes. It is also not directly underneath the satellite ground track, so the viewing angles are higher than in day 94. These images present the best and worst case scenarios possible for Lake Superior, and are thus useful for qualitative assessment. Figure 4.1 shows the uncorrected band 6 images from each day.
4.1.1 Latitude and Longitude Images

The first images produced by Alt_SeaWiFS were maps of latitude and longitude values at every pixel. (see section 3.5.1.2: Determining Lake Superior Region) Examples of these maps are shown in Figure 4.2:

**Figure 4.2: Per pixel latitude (left) and longitude (right) for SeaWiFS image of Lake Superior region**

In Figure 4.2, the brighter the pixel value, the larger the latitude or longitude value. Thus, the latitude value increases from South to North, and the longitude value increases from West to East. Note that these changes do not occur uniformly across the image, but due to
the curvature of the earth occur more rapidly in some regions than others. For example, the longitude value changes faster in the northern part of the image than the southern. This occurs because the actual distance spanned by a single longitude degree is greater in southern regions than northern regions.

4.1.2 Lake Superior Region Mask Image

The latitude and longitude images make sense, and are successful in providing the information needed to create a region mask. Figure 4.3 is an image where all areas except the Lake Superior image are suppressed by the mask:

**Figure 4.3: SeaWiFS image, all areas except Lake Superior region removed with mask**

The region mask appears to successfully identify the Lake Superior region without clipping any elements of the lake's image.

4.1.3 Land Mask Images

The next step is to make a mask to divide the land and cloud pixels from the water pixels. This is done according to the steps presented in section 3.5.1.3. Figure 4.4 are land mask images from our two dates.
The land mask is successful for the day 94 image. It includes all areas of the lake, and successfully distinguishes between both islands and water pixels, turbid water and land pixels.

The mask is less successful for the day 99 image. While water pixels were successfully separated from land pixels, many of the cloud pixels were not masked out. While this becomes more apparent as we view more image results, we can conclude that a pixel is dominated by clouds if known visible hydrospheric effects (plumes) are not noticeable. This is the case in some of the South Western parts of the day 99 image.

Because of the difficulties in land masking, the option was added to Alt_SeaWiFS to allow the user to alter the threshold between land/clouds and water. While this is not a sophisticated land/cloud masking solution, it yields a mask that produces less processing error. (Note that the rest of the Day 99 results in this section include images that did not use this tweaked land mask).
Figure 4.5: Tweaked Land Mask Image

Day 99 Land Mask after user tweaking

(the red arrow is the tip of the Keweenaw peninsula)

4.1.4 Registered Historical LCW Images

The next step is the user registration between parts I and II of the Alt_SeaWiFS software. The best way to see if this step was successful is to examine the warped Historical LCW bathymetry image to determine if it was registered to the image land map. Figure 4.6 is the registered LCW region images, superimposed on the land mask.

Figure 4.6: Registered Historical LCW mask images
As you can see, both LCW regions fall within the water region from the land masks. Since both registrations have placed the Historical LCW regions in the same place, we deem the registration warping successful.

4.1.5 Image Classification
The next step is the ISODATA classification. The goal of this step was to produce a large number of classes, determined by spectral character in the NIR bands. Figure 4.7 are the class images for each day, where the color of each pixel has no physical meaning, but represents a particular class.

![Figure 4.7: Image of Classes](image)

The classified images from day 94 and day 99 show some differences. The day 94 classes seem to be more compact and continuous. They seem to represent gradual changes in spectral character with differences in atmospheric or hydrospheric constituents. The day 99 image, on the other hand, seems noisier. This is due to the greater range of spectral character present in the day 99 image. This greater range is due to cloud pixels that were not removed with the land/cloud mask.

4.1.6 Merged Class image and Historical LCW map
The next step in Alt_SeaWiFS is to combine the data from the Historical LCW region with the image of classes. (See section 3.5.3.3: Merging classified image and Historical LCW image) Figure 4.8 is the merged class image for both days:

**Figure 4.8: Merged Class Images**

As expected from the analysis of the class images, the day 94 merged class area extends farther than the day 99 image. The day 94 image had larger class regions due to the smaller range in spectra for the NIR bands of that image. This means that the data extrapolation from LCW to non-LCW regions will not be as far as in the image from day 99. Some of the day 94 results are counter-intuitive. The classes that snake around in the eastern section of the lake probably do not represent LCW regions in the lake, as they are very close to shore in a region where points farther from the shore have not been identified as LCW. The day 99 image does not extend as far as the day 94 image, requiring a larger NIR extrapolation. This extrapolation may be spreading the improper land/cloud masking. Many of the cloud regions in the southwestern section of the lake were not removed with the mask, so some of their values were selected as LCW classes. In the extrapolation stage of Alt_SeaWiFS, these cloud spectra will be extended to previously cloud free sections of the lake.

4.1.7 NIR band LCW to non-LCW data extrapolation
The next step is to extrapolate data from the regions identified as LCW to regions identified as non-LCW. This is accomplished as shown in section 3.5.3.4. Image results from band 7 of this step are shown in Figure 4.9:

Figure 4.9: Extrapolated Band 7 images

Day 94 Band 7 Extrapolated Image

Day 99 Band 7 Extrapolated Image

Several conclusions can be drawn from these extrapolated images, as it is easy to distinguish between LCW and non-LCW regions. Since the LCW regions show actual
image data, they contain much more noise than non-LCW regions, which are taken by averaging pixel values from an area. Both images show how the extrapolation is unresponsive to high frequency spatial effects. While this is useful to suppress image noise, it is excessive with respect to our scenario. Adverse effects of our extrapolation are especially prevalent in the South West portions of the day 94 image, where LCW pixels are noticeably darker than non-LCW pixels. The result of this inappropriate extrapolation could mean selection of the wrong MODTRAN LUT atmospheric scenario. It also means that the corrected images may show the boundaries between LCW and non-LCW regions, which we do not want to enhance in our imagery. High frequency LCW details, such as the airplane contrails in the central regions of the day 99 image, are also lost. The result of this loss could mean that MODTRAN LUT fitting, which may remove the effects of the high frequency information, would not remove them from non-LCW regions.

4.1.8 LUT type images
Next, the Alt_SeaWiFS software constructs a searchable LUT from given MODTRAN LUT's, and fits the extrapolated image data to them. (See section 3.5.3.5: MODTRAN LUT processing). Figure 4.10 shows the LUT value associated with each pixel in our images:

![Figure 4.10: Image LUT values](image)

Except for a few isolated pixels, the entire image from day 94 was assigned to a single atmospheric type. The day 99 image, on the other hand, shows at least six different atmospheric types, distributed in a rather chaotic fashion. The main difference between
the extrapolated images used for the LUT fitting was the range of their values. The day 94 image had a small range of spectral value differences, while the day 99 image had a much larger range, increased by the mistakenly limited cloud masking.

The day 94 image may well have had a single atmospheric type throughout the entire image. Figure 4.9 shows an image without obvious atmospheric spatial character, and without large scale spectral differences between regions. However, due to the excessive smoothing in the extrapolation of LCW to non-LCW regions of the image, we would expect a selection of different LUT types in the South West portions of the image. Since the LUT type was constant throughout, and since we know our LUT is limited in resolution, we conclude that our problems stem from lack of LUT resolution, and not uniformity of atmospheric type. As shown below, this problem may not be catastrophic, as the final results fall within expected data ranges. However, increasing the LUT resolution would not be a challenging task.

Although the final results from the day 99 image are less promising, the LUT image shows that the Alt_SeaWiFS software is working. The LUT image for day 99 shows different atmospheric types in the South Western regions of the lake, and LUT types that change for small scale effects like the airplane contrails in the central regions of the lake.

4.1.9 Final image results
Following calculation of per pixel LUT type, the effects of that LUT type are removed from the VIS band images. (See section 3.5.3.7: Removing Atmospheric Effects from VIS bands) Band 3 (centered about 490nm) images for day 94 and 99 are presented in Figure 4.11:
Figure 4.11: Water Leaving Radiance, band 3

Day 94 final result:
Band 3 water radiance

Day 99 final result:
Band 3 water radiance
The band 3 water leaving radiance image from day 94 is promising. It shows no obvious spatial signatures from the atmospheric correction (as would be hoped with a uniform LUT type selection) and is falls within the expected data ranges (see figures below). The day 99 image is less useful. As expected, the inadequately masked regions in the South Western regions of the lake exhibit spatial character not expected for hydrospheric signals. It is interesting to note that very small values were predicted for a large portion of the cloud region. In this case, it seems that the atmospheric signal was removed, but since it so dominated the total signal, none was left. As expected, many of the atmospheric effects were not removed. Since they were not properly modeled in the MODTRAN LUT, they were not totally removed. A good example of this is the airplane contrails in the center of the lake. A final note of interest is the lighter region surrounding dark clouds in the North central regions of the day 99 image. These suggest that Alt_SeaWiFS attempted to remove the atmospheric signal of the clouds, but was unsuccessful. However, due to blurring in the Alt_SeaWiFS correction (as mentioned in sections 4.1.7 and 4.1.8) the correction spread to neighboring pixels, in effect over-correcting their values.

We have now finished looking at our intermediate images in a qualitative sense, which helps us determine the quality of the algorithm step by step. Next, we examine the quality of the calculated data directly, by comparing it to actual ground measured data.

4.2 Direct Data Comparison

The most logically sound method of remote sensing data verification is to compare the calculated ground leaving radiance to the actual ground leaving radiance. The surface data is called ground ‘truth’. However, gathering ground truth can be expensive, and is typically collected over a small region. (If the gathering process were not expensive, a main purpose of remote sensing – providing cheap data over a large range – would be invalid).
Our ground truth was gathered as part of the KITES project in the Keweenaw current region of Lake Superior. An optical water profiler, the Satlantic SPMR (Satlantic 2000) was configured to mimic the spectral sensitivity of the SeaWiFS sensor in visible bands, and deployed on several transects.

Data from the Satlantic profiler was gathered and converted to water leaving radiance values. The profiler gathers upwelling radiance values at various depths. The values close to the surface were propagated to a surface upwelling radiance, and these values were compared to surface radiance values calculated by the Alt_SeaWiFS algorithm and the SeaDAS algorithm. A plot of these values for a particular date and location are shown below, in Figure 4.12. The data was gathered nine kilometers from the shore in the Eagle Harbor Transect, running approximately north of Eagle Harbor, Michigan. The data was gathered on the 241st day of 1999 (August 28th).

Figure 4.12: Direct Radiance Comparison

![Figure 4.12: Direct Radiance Comparison](image)
SeaWiFS has a reflective accuracy goal of 5% after atmospheric compensation. Therefore, the equivalent radiance value was used as error bars around the ground truth data. If the Alt_SeaWiFS algorithm is successful, then the calculated values should fall within these error bars.

As we can see in the plot, the results from SeaDAS, the established atmospheric compensation algorithm, severely underestimate the water leaving radiance. Alt_SeaWiFS, however, produces values within the error goals established for the SeaWiFS project, at least for bands 3-6. Bands 1 and 2 overestimate the water leaving radiance, but are accurate to within about ten reflectance units.

If the spectral accuracy for the Alt_SeaWiFS software remains constant for all image points and dates, then we can conclude that some derived image products will be more accurate than others. The CZCS type pigment product uses bands 3 and 5 for pigment concentrations greater than 2.0 [mg m$^{-3}$]. For concentrations less than 2.0, bands 2 and 5 are used. Therefore, calculation of the CZCS type pigment will be more accurate for concentrations above 2.0 than for concentrations below. The diffuse attenuation coefficient is calculated using bands 3 and 5 for all cases. Therefore, we should expect an accurate diffuse attenuation coefficient calculated image. Finally, the Chlorophyll-a image product is calculated using band 5 and the largest value of bands 2, 3 or 4. If band 2 is continuously overestimated, as in Figure 4.12, then we can expect an overestimation of Chl-a in our imagery.

**4.3 Day to Day Comparisons**

As mentioned previously, a problem with using ground truth to verify the accuracy of an image data set is that it is a verification for one pixel at one point in time. In reality, many thousands of pixel data sets are being produced, and their quality is not necessarily the same as the ground truth pixel. One method of verification without ground truth is to compare successive image days. In theory, the water constituents that produce water
leaving radiance should not change drastically on a daily basis. Therefore, the water leaving radiance values should remain almost identical on a day to day basis.

Two sets of image data were used. The first set is from days 94, 95 and 99 in 1998. The second is from days 232, 239 and 241 in 1999. The availability of series of usable image days was limited by the infrequent occurrence of cloudless days over Lake Superior.

To simplify the verification process, only one band was compared between each image. Band 3, centered about 490nm, was chosen because that band is used in many bio-optical parameter calculation algorithms.
Figures 4.13 and 4.14 show the atmospherically corrected band 3 images.

**Figure 4.13: Corrected Images from 1998, Days 94, 95 & 99**

- **Band 3 Lw, 1998 day 94**
- **Band 3 Lw, 1998 day 95**
Figure 4.14: Corrected Images from 1999, Days 232, 239 & 241
It is difficult to compare three images side by side to verify their constant nature. What we need to do is individually compare each pixel that represents a specific location, and see how that pixel changes over the three days. To do this, we need to register all three images to the same projection and place them into a three dimensional array. Once that has been done, statistics can be calculated for each pixel location. The per pixel variance was calculated and is shown in 4.15 and 4.16. Note that calculation of the variance requires at least three data points, so if a region of one image was masked out due to clouds, the variance was not calculated for that point.
In both sets of images the proximity to clouds seems to affect the variance of the time dependent pixel series. One of the goals of the SeaWiFS project is to have water leaving radiance calculations to within five reflectance units of the truth value. Using the spectrally sampled exo-atmospheric irradiance values, we can calculate the radiance equivalent of 5% reflectance for each band. (see equation (1)). For band three, this value is 2.665 (mW/cm2/μm). The images above show variances less that that value in all regions, suggesting that the consistency of Alt_SeaWiFS falls within SeaWiFS goals.
Another method of image to image verification is to examine the histograms for each image. If we overlay these histograms on the same plot, they should overlap. Figures 4.17 and 4.18 are the histograms of each image series.

**Figure 4.17: Histogram of images from 1998, Days 94, 95 & 99**

![Histogram of images from 1998, Days 94, 95 & 99](image)

**Figure 4.18: Histogram of images from 1999, Days 232, 239 & 241**

![Histogram of images from 1999, Days 232, 239 & 241](image)

The histograms in all three images do not entirely overlap, however, if we recall the SeaWiFS accuracy goal in radiance units (2.665 (mW/cm²/μm)), we realize that most of the pixels in all three days fall within the accuracy goal of each other.
A final analysis is to take specific pixels, and see how the value produced at that pixel changes on a daily basis. This was done for the series of images from 1999, and is shown in a plot in Figure 4.19. The locations of these values is shown in Figure 4.20.

**Figure 4.19: Plot values of specific locations for 1999, Days 232, 239 & 241**

Band 3 Lw values, 1999 Day 232, 239 & 241

**Figure 4.20: Locations of plot points for Figure 4.19**
Note that the locations chosen for this spectral plot were limited to pixels not masked by clouds in all three images. Regardless, Figure 4.19 shows even greater promise, as the change for each pixel seems uniform for each day’s image. This has several implications. It could mean that the weather was producing a uniform change to the water constituents, producing a uniform reflectance change. It could also mean that the atmospheric correction itself could be producing uniform changes to the image. Either way, it shows us that the solution produced is uniform throughout each image, implying a high fidelity in relative (throughout one image) solution values.

We have shown that the Alt_SeaWiFS atmospheric compensation routine is much more effective in Lake Superior than the established SeaDAS routine. We have also shown results that fall within the SeaWiFS accuracy goals for most bands and images. However, several improvements could be made to Alt_SeaWiFS to further enhance its accuracy. The next section is a more detailed conclusion, and recommended changes.
5.0 Conclusions

The Alt_SeaWiFS code presented here is a proof of concept of an algorithm intended to satisfy the goals of section 3.1. Accomplishment of these goals is presented below.

The Alt_SeaWiFS algorithm avoids the ‘clear water assumption’ by splitting the lake into two regions, one where NIR water reflectances are known and measured, another where they are unknown. The atmospheric correction is found by inverse modeling techniques, where the difference between ground and top of the atmosphere radiance is used to determine the atmospheric type. Once this is known, its effect can be removed from VIS band images, and a solution extrapolated to unknown reflectance regions.

The Alt_SeaWiFS algorithm requires the ancillary data provided with SeaWiFS imagery and several constants and files that do not change on an image to image basis.

The results of Alt_SeaWiFS are HDF files that are in the same format as an HDF file produced by standard SeaDAS atmospheric correction software.

Alt_SeaWiFS acknowledges the differences between Lake Superior imagery and ocean imagery. Since the scale is smaller, data extrapolation is utilized to fill in data points previously lacking.

The data extrapolation and classification elements of the Alt_SeaWiFS software utilize the data available in a pixel’s neighborhood.

Environmental differences, such as aerosol type, are incorporated in the algorithm.

While Alt_SeaWiFS has major shortcomings which prevent immediate research use, several simple changes could be made to make it usable. Section 6.0 lists some of these recommended changes.
6.0 Recommendations

Several recommendations could be made for further development of the Alt_SeaWiFS software. Most of these changes would be easy to implement in the Alt_SeaWiFS software, but require some amount of outside research. Some of the changes are purely structural with regards to the software, but were not incorporated due to time constraints.

6.1 Better LCW reflectance values

The NIR water reflectance in LCW regions was calculated using a hyperspectral oceanographic radiometer (see section 3.4.3: NIR band reflectance of LCW). As illustrated previously, the data from this instrument, in NIR regions, is so dominated by noise that a significant amount of smoothing is required to produce a measure of reflectance in the SeaWiFS NIR bands.

This issue could be avoided by improving the measurement methods. The original radiometer measurements were not intended, at the time, to be used to determine NIR reflectance in LCW regions. The same instrument could be used with a longer integration time to reduce the noise, or another instrument that measures reflectance directly (such as ASD’s portable spectrometer) could be utilized.

New NIR reflectance values could be incorporated into Alt_SeaWiFS without changing the existing software. Expected reflectance values are specified in the MASTER_FILE, and the new values could be added to this file.

6.2 Better White Cap Reflectance Model

The current model for wind driven white cap reflectance is based on measurements made in coastal regions of the Atlantic Ocean (Koopke 1984) (Monahan and O’Muircheartaigh
1986). The validity of these reflectance models for SeaWiFS data has been questioned, and the newest version of the SeaDAS code applies a 60% reduction to the results from the white cap model (Frouin et. al. 1996). In addition, it is reasonable to expect the white cap reflectance to be different in fresh water Lake Superior. A field campaign to measure the white cap reflectance as a function of wind speed could improve the atmospheric correction of the Alt_SeaWiFS software, and make it more appropriate for the Lake Superior region.

The Alt_SeaWiFS radiative transfer code would need to be changed to incorporate an improved reflectance model. Both the ‘make_searchable_lut’ and the ‘correct_image’ functions utilize the current white cap model, and would need to be changed to incorporate a new one.

6.3 Higher Resolution Wind Data

The current method of determining wind speed utilizes meteorological data supplied with SeaWiFS images. The spatial resolution of this data is one degree latitude or longitude. The Alt_SeaWiFS software averages the several data values that encompass the Lake Superior region to determine a single wind speed and direction value. This has two possible negative implications, as the wind speed and direction is used at two points in the software. Increasing the spatial information about wind speed and direction so that a uniform value is not chosen for the entire lake could improve both the NIR data extrapolation and the white cap reflectance correction. This could be implemented by making a wind speed ‘image’, where the wind speed and direction is calculated for each pixel location in the lake. During processing for a particular image pixel, the value for that pixel location in the wind speed image could be used instead of the total image wind speed.

The first wind speed and direction use is in the extrapolation of NIR image data from LCW to non-LCW regions of the image. The shape of the Gaussian averaging kernel
used in the data extrapolation is determined by the wind speed and direction. Currently, the same kernel is used for the entire image. The extrapolation might be more appropriate if this Gaussian kernel where modified to local conditions, thus permitting a more appropriate data extrapolation. In practice, this would modify equation (30) from:

\[
nonLCW(x, y) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} GAUS\left(\frac{i-x}{b}\right) GAUS\left(\frac{j-y}{d}\right) LCW(i, j) didj}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} GAUS\left(\frac{i-x}{b}\right) GAUS\left(\frac{j-y}{d}\right) binLCW(i, j) didj}
\]

(30)

where \(b\) and \(d\) are global values related to the wind speed and direction, to:

\[
nonLCW(x, y) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} GAUS\left(\frac{i-x}{b(x)}\right) GAUS\left(\frac{j-y}{d(y)}\right) LCW(i, j) didj}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} GAUS\left(\frac{i-x}{b(x)}\right) GAUS\left(\frac{j-y}{d(y)}\right) binLCW(i, j) didj}
\]

(50)

where \(b(x)\) and \(d(y)\) are derived from the wind speed and direction calculated for pixel \((x, y)\).

Higher resolution wind speed could also improve the inverse modeling efforts illustrated in sections 3.5.3.6 and 3.5.3.7. Since wind speed determines the quantity of surface foam, (and thus white cap reflectance), a per-pixel measure could improve the accuracy of the modeling by tailoring the expected reflectance for each pixel. However, since we do not make a searchable LUT of TOA radiances for every pixel location in the image, these effects would have to be applied at the model fitting stage of the inverse modeling ('match_data_to_lut'), rather than at the searchable LUT creation ('make_searchable_lut'). High resolution wind data could also be applied at the 'correct_image' function, simply by using the image, rather than global, value.

Paul J. Roebber is a scientist at the University of Wisconsin – Milwaukee, producing real time meteorological maps of the Great Lakes Region, and may be a source of this higher resolution data (Roebber 2000).
6.4 Better HDF output

A minor problem discovered in the verification of data produced by the Alt_SeaWiFS software was the inability of SeaDAS to open an HDF image produced by Alt_SeaWiFS as a valid L2 SeaWiFS file. This problem appears to stem from the identification and naming of the HDF data elements within the output L2 file, and does not affect the data itself. In fact, SeaDAS can open the alternative L2 files as HDF files.

A function of the SeaWiFS HDF file format design allows the processing software to make comments to HDF values within the output file for use in verification of processing success. While this is not essential to proper atmospheric correction, it may be useful for future users if implemented.

6.5 Better use of image flags

Flag images are an important part of the established atmospheric correction routine, as they identify regions of failure or success for various algorithms. The Alt_SeaWiFS software makes an image of flags showing the inverse modeling quality. However, there are many more parameters that could be identified in a flag image, and the Alt_SeaWiFS software would probably benefit from more extensive and detailed flag maps.

6.6 Better LUT resolution

Sections 4.1.8 and 4.1.9 illustrate the need for more detailed MODTRAN LUT’s. The LUT’s currently in use were created as a proof of concept. While they begin to perform the necessary tasks, they cannot adequately model the total possibilities encountered in a typical Lake Superior SeaWiFS image. Therefore, to make the Alt_SeaWiFS software more robust, the number of atmospheric scenarios modeled in the LUT’s must increase from the current 27 to several times that value. The sampling points for the current LUT’s
represent the mean value of the parameter in question, and the value two standard deviations above and below that value. It would be best, then, to increase the sampling of parameter value within one standard deviation of the average, to more adequately model the expected scenarios. This would involve calculating a larger MODTRAN run, but would require no changes to the Alt_SeaWiFS software, as new LUT’s are incorporated by simply placing them in the proper directory.

Another option would be to interpolate atmospheric LUT values between the two closest LUT elements chosen for a pixel. To implement this, the two closest LUT elements for a given pixel must be determined. When correcting for the atmospheric effects in the visible bands, the atmospheric propagation parameters for each LUT need to be found. The parameters used to correct a particular pixel, then, would be interpolated between the two LUT parameter values. This should yield a higher resolution of atmospheric correction possibilities without increasing the size of the LUT’s.

### 6.7 Smaller Gaussian kernels

Section 4.1.7 demonstrates the need for higher resolution LCW data extrapolation. This could be accomplished by reducing the size of the Gaussian averaging kernels used in the extrapolation. The relationship between wind speed and the Gaussian scaling factors \( b \) and \( d \) (see equation (30)) used in Alt_SeaWiFS was arbitrary, so making changes would not be a problem.

### 6.8 Better Cloud Mask

As illustrated in many parts of 4.1, an accurate method of masking out the cloud dominated pixels is essential for many stages of the atmospheric correction process. The Alt_SeaWiFS code makes use of a simple single band thresholding algorithm, but much research has been done on more advanced techniques of cloud masking (Darzi 1992).
Multiple band techniques could be used to determine spectral differences between water and cloud/land pixels, or several thresholded bands could be combined.

At a later date, a modification was made to the cloud masking process, allowing the user to add his input to determining the threshold value, thereby improving the cloud masking process.
REFERENCES


Warrington, Daniel. Personal communication, email. 22 March, 2000. Dswarrin@mtu.edu.
Appendix A: Calculation of Lake Superior SeaWiFS fly-over time (Gregg, et al. 1993).

Given:
- Orbit Period, $p = 98.9$ minutes
- Inclination, $i = 98.25^\circ$, $i' = i-90 = 8.25^\circ$
- Latitude of start, $\varphi_s(t_0) = 0^\circ$
- Latitude of overpass, $\varphi_s(t-t_0) = 47.5^\circ$
- Time of start, $t_0 = 12.00$ local time
- Great circle distance from $\varphi_s(t_0)$ to $\varphi_s(t-t_0)$, $D = 360 \lambda t / p = 3.64 (t-t_0) = 3.64 t$

Unknowns:
- Time of overpass, $t$

Relationship:
\[
\sin(\Psi_s(t-t_0)) = \cos \delta \sin(\Psi_s(t_0)) + \sin \delta \cos(\Psi_s(t_0)) \cos(i')
\]
Where $\delta = 3.64 (t-t_0)$

Solution:
\[
\begin{align*}
\sin(47.5^\circ) &= \cos \delta \sin(0^\circ) + \sin \delta \cos(0^\circ) \cos(8.25^\circ) \\
\sin(47.5^\circ) &= \sin \delta \cos(8.25^\circ) \\
\delta &= \sin^{-1}\left(\frac{\sin(47.5^\circ)}{\cos(8.25^\circ)}\right) \\
\delta &= 48.16^\circ
\end{align*}
\]

Therefore, $t - t_0 = 13.23$ minutes, so the local fly over time is 11:47am.
### Appendix B: Master File variables

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Filename</strong></td>
<td>Input L1B SeaWiFS filename</td>
</tr>
<tr>
<td>Outfilename</td>
<td>Filename of output L2 SeaWiFS file</td>
</tr>
<tr>
<td>Lcwfilename</td>
<td>Filename of Historical LCW file</td>
</tr>
<tr>
<td>Lutfilename0</td>
<td>LUT filename for ground track</td>
</tr>
<tr>
<td>Lutfilename3</td>
<td>LUT filename for 3° off ground track</td>
</tr>
<tr>
<td>Lutfilename6</td>
<td>LUT filename for 6° off ground track</td>
</tr>
<tr>
<td>Lutfilename9</td>
<td>LUT filename for 6° off ground track</td>
</tr>
<tr>
<td>Vars_filename</td>
<td>Filename for temporary storage of variables between parts I and II</td>
</tr>
<tr>
<td>Rout_filename</td>
<td>Filename for temporary storage of compiled routines between parts I and II</td>
</tr>
<tr>
<td>Gcpfilename</td>
<td>Filename of Ground Control Points (GCP’s) chosen between parts I and II</td>
</tr>
<tr>
<td>Instr_filename</td>
<td>Filename for user instructions between parts I and II</td>
</tr>
<tr>
<td>Clean</td>
<td>Value determines which images are left in ENVI memory after processing, 0 to keep everything, 1 to remove all but most important images, 2 to remove all images</td>
</tr>
<tr>
<td>Verbose</td>
<td>If set to one, Alt_SeaWiFS prints a running commentary</td>
</tr>
<tr>
<td>Wedge</td>
<td>Longitude value of Western edge of Lake Superior region</td>
</tr>
<tr>
<td>Edge</td>
<td>Longitude value of Eastern edge of Lake</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Nedge</td>
<td>Latitude value of Northern edge of Lake Superior region</td>
</tr>
<tr>
<td>Sedge</td>
<td>Latitude value of Southern edge of Lake Superior region</td>
</tr>
<tr>
<td>Ref</td>
<td>Array of expected LCW reflectance values</td>
</tr>
<tr>
<td>Eo</td>
<td>Array of spectrally weighted exo-atmospheric irradiance values</td>
</tr>
<tr>
<td>Extrap_fast</td>
<td>Variable to set speed of NIR data extrapolation, 0 for full processing, 1 for 10x speed processing</td>
</tr>
<tr>
<td>Softwareid</td>
<td>String containing Alt_SeaWiFS version number, for output in HDF file</td>
</tr>
</tbody>
</table>

Note: variable names in italics represent variables typically chosen by the user.
Appendix C: Alt_SeaWiFS software flowcharts

Alt_seawifs calling program: part 1

Variables set in MASTER_FILE

- latitudes, longitudes
- image filenames
- bathymetry image filenames
- auxiliary data filenames
- hdf_data structure

Open files function
- opens and reads image HDF data

Make mask region function
- find pixel coordinates of lake superior region

Make mask land function
- create binary land mask image

Find land mask size
- land mask size

Output land mask image to ENVI
- ENVI image object

Open L.C. image
- opens L.C.W image bathy and shoreline file

L.C.W set function
- final manipulation of L.C.W image

Save all compiled routines and variables for part II
- Save all

Display text instructions
Appendix D: Classification

The SeaWiFS atmospheric compensation algorithm presented here uses the ISODATA unsupervised classification to help differentiate between Lake Clear Water (LCW) and non-LCW regions in the image. It also uses several classification post processing routines, such as clustering and the connected components algorithm.

D.1 The ISODATA Unsupervised classification algorithm

The Iterative Self-Organizing Data Analysis Techniques (ISODATA) Algorithm is an unsupervised classification algorithm, implying that it does not require training to establish the spectral characteristics of each class. It is an iterative process that divides the pixels among a set number of classes, and adjusts this distribution until some convergence criteria has been met or some number of iterations have occurred. The ISODATA algorithm is similar to the K-Means unsupervised classification algorithm, but allows more control over the specifics of the classification.

The ISODATA algorithm about eight key steps, as illustrated in Tou and Gonzalez (1974). Parameters that are set with the ISODATA algorithm are the following:

\[ K = \text{the number of class centers} \]
\[ \theta_N = \text{the minimum number of data points in a class} \]
\[ \theta_S = \text{the maximum standard deviation value for each class} \]
\[ \theta_C = \text{the maximum allowable distance between two class centers before they are 'lumped' together} \]
\[ L = \text{the maximum number of pairs of classes that can be 'lumped' in an iteration} \]
\[ I = \text{the number of iterations} \]

Step 1:
Randomly select $K$ number of class centers. In a two dimensional parameter space, this would look like Figure D.1:

**Figure D.1: Step 1 of ISODATA classification**

1. **Step 2:**
   Assign each data point to its closest class center, as illustrated in Figure D.2:

**Figure D.2: Data points grouped into classes**

2. **Step 3:**
   Delete classes with fewer samples than $\theta_N$, as illustrated in Figure D.3:
Step 4:
Shift each class center to the center of its cluster, as illustrated in Figure D.4:

Step 5:
Find the average distance, $D_j$, and standard deviation, $\sigma_j$, for each class, $j$, from the cluster center.

Step 6:
If $\sigma_j > \theta_s$ for class $j$, split it into two classes, as illustrated in Figure D.5:
Figure D.5: Splitting of class with large standard deviation

Step 7:
Find the distance between each class center, $D_{ij}$. Rank these values from smallest to largest, and discard all values greater than $\theta_C$. Combine each pair of the first $L$ classes.

Step 8:
If this is the last iteration, end the classification. Otherwise, return to step 2. Figure D.6 is what our data would look like after step 2 on the second iteration:

(Duda and Hart 1973) (Tou and Gonzalez 1974)
D.2 Post Classification Clustering and Grouping routines

Often the results of a classification algorithm are noisy, with holes in some class areas and isolated pixel classes in others. This is due to noise in the image data itself. The occurrence of noisy classification results could be reduced by minimizing the noise in the original image data, using a low frequency passing filter, for example. However, this will blur the original data and reduce the accuracy of the classification, since neighboring pixel data will merge with actual data from a pixel. Another means to reduce classification noise is to operate on the results directly, using morphological processes. It is these morphological processes that are used as post-classification grouping and clustering routines (Tou and Gonzalez 1974).

Morphology, as it applies to image processing, is the process of using logical operators to perform specific tasks. A basic morphological process is the dilation routine, where the region encompassing a binary value is expanded. In the execution of a morphological dilation routine, each pixel in the image is checked. If the value of the pixel is 1, the result at that location is left the same. If the value of the pixel is 0, but the value of at least one of the neighborhood pixels is 1 (see section D.3), then the value of the pixel is changed to 1. If the value of a pixel is 0, and all of its neighborhood pixels is also 0, then the pixel value is left at 0. Figure D.7 is an example of the result of a morphological dilation routine.

![Figure D.7: Morphological Dilation](image)

In the above diagram, the red pixels represent pixels with an initial binary value of one. Blue pixels represent pixels with the initial value of zero and final value of one.
after a morphological dilation routine. Black pixels have an initial value of zero and are unchanged after the dilation.

A morphological erosion routine works much the same way as a dilation, however, the area of the binary value of one is decreased, rather than increased (Gonzalez and Woods 1993).

Completion of several morphological operations can have other effects. A morphological opening procedure is an erosion followed by a dilation. This results in an image where isolated pixels with a value of one, and isthmuses of value one between larger areas of one, are removed. The 'Sieve' routine in ENVI, the software package used for image processing in this thesis, is a morphological opening algorithm implemented on each class individually. A morphological closing procedure, on the other hand, is a dilation followed by an erosion. Closing tends to remove small groups of isolated pixels and smooth the edges or larger groups. ENVI's 'Clump' post-classification routine uses a closing algorithm (Gonzalez and Woods 1993) (Better Solutions Consulting 1997).

D.3 The Connected Components Algorithm
The Connected Components Algorithm identifies regions in a binary image that contain pixels of uniform value and are 'connected' in the sense that they all share a neighbor with at least one other pixel in the region.

The '8-neighbors' of a particular pixel are the eight pixels above, below, left, right and diagonal to that pixel. A group of pixels are '8-connected' if a series of 8-neighbor pixels all share a uniform value (Gonzalez and Woods 1993). An example of this type of pixel description is given in Figure D.8.
The Connected Components Algorithm is used in our processing to split classes identified with ISODATA into classes of connected components. This is essential, as each class is intended to represent a specific hydrospheric and atmospheric reflectance scenario, which sum to a top of the atmosphere radiance value. However, a class containing non-localized elements may contain several hydrospheric and atmospheric reflectance scenarios. The spectral character of these scenarios may sum to the same atmospheric radiance value. To avoid this, each class in split into localized region classes in an attempt to limit each class to a single hydrospheric and atmospheric reflectance scenario.

The Connected Components Algorithm operates on a binary image, $I$ of dimensions $m$ and $n$, as follows:

1. The algorithm starts by creating an array, $I'$ with identical dimensions as the binary image. A counting variable, $c$, is created and initialized to 0.

2. The first pixel, in the upper left of the image ($m=0$ and $n=0$) is examined. If $I(0,0)=0$, then $I'(0,0)$ is set to 0. If $I(0,0)=1$, then $c=c+1$ and $I'(0,0)=c$. 

---

**Figure D.8: 8-connected pixels**

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>P6</td>
<td>P7</td>
<td>P8</td>
<td>P9</td>
<td>P10</td>
</tr>
<tr>
<td>P11</td>
<td>P12</td>
<td>P13</td>
<td>P14</td>
<td>P15</td>
</tr>
<tr>
<td>P16</td>
<td>P17</td>
<td>P18</td>
<td>P19</td>
<td>P20</td>
</tr>
</tbody>
</table>

In the above diagram, the 8-neighborhood pixels for P8 are pixels P2, P3, P4, P9, P14, P13, P12, and P7. If all bold, red pixels have the same value, then P8 is 8-connected with P9, P14 and P13. Therefore, a connected components group could be formed with pixels P8, P9, P10, P13, P14 and P17. Note that P1 and P6 would form a different connected components group, since they have the same value but are not a neighbor of any of the elements from the first group.
3. Each pixel in the first row is then examined. If \( I(m,0) = 0 \), then \( I'(m,0) = 0 \). If \( I(m,0) = 1 \) and \( I(m-1,0) = 0 \), then \( c = c+1 \) and \( I'(m,0) = c \). If \( I(m,0) = 1 \) and \( I(m-1,0) = 1 \), then \( I'(m,0) = I'(m-1,0) \).

4. Repeat for each row in the image, checking not just \( I(m-1,n) \) for 1, but also \( I(m-1,n-1) \) and \( I(m,n-1) \). The result should be an image, \( I' \), filled with 0's in the 0 regions in \( I \), and integer values for each connected component region, in order of its discovery. (Rhody 1999)

Recall, however, that our classified image is not binary - it has an integer value for each class. To split each class into connected component classes, the Connected Components Algorithm must be run many times - once for each class. In the software, a binary image is created for each class determined by the ISODATA algorithm. In this binary image, the pixels identified as being elements of the particular class are given a value of 1, and everything else a value of 0. The Connected Components algorithm is then performed, splitting the class into its localized regions. Each of these regions is then given a unique class value. These steps are then repeated for each class, resulting in a new classified image containing many more classes than before.
Appendix E: Alt_SeaWiFS installation manual

README for ALT_SEAWIFS Version 1.0 atmospheric correction package.

ALT_SEAWIFS is an atmospheric compensation/correction software package designed specifically for SeaWiFS images of Lake Superior. The software was written in Research Systems Inc.'s (RSI) Interactive Data Language (IDL) as a module to add to the remote sensing visualization package called The Environment for Visualizing Images (ENVI).

This file contains the background and purpose of the software along with instructions for its installation and use.

Kirk D. Knobelspiesse
Digital Imaging and Remote Sensing Laboratory
Rochester Institute of Technology
knobelspiesse@hotmail.com
www.rit.edu/~kdk2963/research.html

TABLE OF CONTENTS

PART I: Introduction and Background
PART II: Uncompressing the software bundle
PART III: Software 'Parts List'
PART IV: Setting up alt_seawifs
PART V: Running alt_seawifs
PART VI: Troubleshooting
PART VII: Version changes

PART I: Introduction and Background

This project was conducted as part of the Master's Thesis of Kirk D. Knobelspiesse in the Digital Imaging and Remote Sensing Laboratory (DIRS) at the Rochester Institute of Technology. Funding for the project comes from DIRS and the Keneewaw Interdisciplinary Transport Experiment in Superior (KITES), whose funding comes from the National Science Foundation and
the National Oceanographic and Atmospheric Administration.

This project started because of a recognized failure of the established SeaWiFS atmospheric correction method in coastal and inland bodies of water. The abstract of the author's thesis dissertation follows:

"Several assumptions are made with the established atmospheric compensation algorithm for the SeaWiFS remote sensing platform. One of these assumptions, the existence of Case I clear ocean water, cannot be made for Lake Superior. A modification to the algorithm is presented, where empirical information and external spatial data are utilized to compensate for the atmosphere in all regions of the lake.

"The algorithm defines Lake Clear Water (LCW) as the inland analogy to Case I water. However, unlike Case I water, LCW has water leaving radiance in the SeaWiFS Near Infra-Red (NIR) 765nm and 865nm bands. Because of the oligotrophic nature of Lake Superior, it is reasonable to assume that the radiance is a constant determined by ground measurements. The atmospheric effect, then, is the difference between the expected water leaving radiance and that measured at the sensor. To implement the algorithm, an unsupervised classification method is used to map LCW and non-LCW regions in an image. Using look up tables created from atmospheric models, atmospheric types are calculated for each class in LCW classified regions. These are then interpolated to non-LCW regions of the lake. This interpolation will be aided by meteorological data. Results of the proposed atmospheric compensation will be compared to optical water profile data gathered on several cruises in Lake Superior."

This software processes a SeaWiFS HDF GAC Level 1-B file to a SeaWiFS HDF Level 2 file. SeaWiFS level 1-B files are available from the NASA Goddard DAAC. located on the web at: http://daac.gsfc.nasa.gov/data/dataset/SEAWIFS/index.html

To process Level 1-A SeaWiFS files to Level 1-B, the SeaWiFS Data Analysis System (SeaDAS) is needed. It can be found at: http://seadas.gsfc.nasa.gov/

The Alt_SeaWiFS software package was written with IDL version 5.2.1 for ENVI version 3.2. For more information about either of these, please visit the RSI web page at http://www.rsinc.com

Finally, for more information about this thesis, please visit the author's web page at http://www.rit.edu/~kdk2963/research.html for the slides from a series of presentations. To get a copy of the thesis itself, contact the author at

kdk2963@rit.edu (before October, 2000)
knobelspiesse@hotmail.com (after October, 2000)
or his thesis advisor, Dr. Anthony Vodacek, at vodacek@cis.rit.edu

______________________________________________________________

PART II

First you must uncompress the alt_seawifs.tar.gz file. The 'gzip' utility was used to compress the file. So type
gunzip alt_seawifs.tar.gz

Now that you have uncompressed the download file, you need to split the 'tape' file into its components. To do this, type

```
$ tar -xvf alt_seawifs.tar
```

For more information on tar, refer to the unix MAN files

PART III
SOFTWARE 'PARTS LIST'

Once the download file has been uncompressed, it should consist of six files in a main directory, three files in a 'data' directory, and thirty three files in an 'idl_files' directory.

Main Directory:
- `alt_seawifs_envi.men`: This file contains the customized ENVI menu setup that includes buttons for the alt_seawifs software. See Part IV for more information.
- `MASTER_FILE`: contains the filenames of the input L1A SeaWiFS HDF files, meteorological files, and any data files called by the software, along with the expected filenames of the output files. Setting of these filenames is done this way to facilitate batch processing. It is run as a script from the software.
- `README_ALT_SEAWIFS`: this file...
- `swf`: A script file containing the commands to compile all the necessary IDL files for execution of the software.

'Data' Directory
- `alt_seawifs.lut`: The Look Up Table (LUT) of the MODTRAN simulated atmospheric optical parameters under different conditions.
- `alt_seawifs_instructions.txt`: This file is called from the software after the completion of Part I, but before the execution of Part II. It contains the instructions for user input needed before Part II. The contents of this text file are displayed in a widget window.
- `superior_lcw.hdf`: An HDF file containing two binary images: One showing the landmap for the Lake Superior region, and another showing the regions of Lake Superior whose depth is greater than 200m (known LCW).

'idl_files' Directory
- This directory contains all of the IDL files needed to run this software.

For further documentation on these, see the HTML help files or the thesis data flowcharts. (As of 7/25/00, not yet created)

PART IV
SETTING UP ALT_SEAWIFS

The Alt SeaWiFS software requires no major configuration or compilation. Several steps must be taken, however, to ensure it works properly with ENVI.
CHANGING PATH SETTINGS:

- Once a directory for alt_seawifs has been chosen, the MASTER_FILE and swf files need to be changed to reflect file locations on your system. In the MASTER_FILE, several path locations start with '/cis/grad/kdk2963/idl/' which are the file locations on my system. Change these occurrences to the proper path in your system. In swf, the first line sets the path for the IDL files to compile. Set this path to how it occurs on your system.

TO MAKE AN ENVI SEAWIFS ATMOSPHERIC CORRECTION MENU BUTTON:

- Under the 'System' pull down menu on the ENVI menu bar, select 'Edit Current Configuration'.
- A list of buttons will appear. Select the 'User Defined Files...' button.
- The 'Default ENVI Menu File' window displays the path and filename of the file that sets the appearance of the ENVI file. You will need to edit the file listed here to include the new menu options, or change the chosen menu file to that supplied in these files: 'alt_seawifs_envi.men'
- If you intend to use the supplied menu file, simply change the path and filename in the box to the appropriate location and name. The menu file lists the SeaWiFS Atmospheric Correction buttons under the 'Transforms' menu option.
- If you have already customized your ENVI menu file, you may wish to simply change the customized file rather than replace it. To do this, open the .men file in your favorite text editor.
- The .men file looks like the following:

```
0 {Top Level Button name}
  1 {Second level button name}
    2 {Third level button name} {called function name}
```

- So, to make two buttons, one for each stage of the atmospheric correction processing, insert the following text under the top level menu button of your choosing:

```
1 {SeaWiFS alt. atm. correction, Part I} {alt_seawifs} {alt_seawifs} {separator}
1 {SeaWiFS alt. atm. correction, Part II} {alt_seawifs_2} {alt_seawifs_2}
```

PART V
RUNNING ALT_SEAWIFS

The Alt_seawifs software runs in two parts. The first part identifies the Lake Superior region in the HDF image, and prepares a land mask. After the first part finishes, the user must perform an image registration between supplied LCW data and the HDF image. (For instructions, please refer to the 'alt_seawifs_instructions.txt' file in the 'data' directory). The second part then operates using the ground control
points (GCP's) selected by the user, and produces an output L2 HDF file, and if selected by the user, images for ENVI.

On the author's Sun Sparc-10 workstation, step 1 takes about ten minutes to process, and step 2 takes between 30min and 1 hour to process.

Considerable effort could have been placed into GUI's and other methods for making the use of alt_seawifs more intuitive, but in the interests of time, the author has decided to limit user interaction to supplying data within the MASTER_FILE.

The MASTER_FILE contains all the information that the alt_seawifs software needs to run. Before each processing run, the user must specify several filenames within the MASTER_FILE that the software will use. These filenames are:

- The HDF image filename - set the part in quotes to the location of the SeaWiFS L1B file you wish to process.

  HDF image filename:
  filename='/dirs/home/kdk2963/data/seawifs/l1b/S1998139185458.L1B_HNSG'

- The location and filename you intend to use for the output image.

  Corrected HDF image filename:
  outfilename='/dirs/home/kdk2963/data/seawifs/l2/S1998139185458.L2_alt_HNSG'

- The location of the LCW image map (should remain constant)

  LCW map filename:
  lcwfilename='cis/grad/kdk2963/idl/alt_seawifs/data/superior_lcw.hdf'

- The location of the weather data file downloaded with the SeaWiFS image

  Meteorological data filename:
  ancilfilename='/dirs/home/kdk2963/data/seawifs/access/S199813918_NCEP.MET'

- The location of the LUT (should remain constant)

  Look Up Table filename:
  lutfilename='cis/grad/kdk2963/idl/alt_seawifs/data/alt_seawifs.lut'

- Temporary file locations (should remain constant)

  Filename for temporary storage of variables between parts 1 & 2
  (placed in working directory...)

  vars_filename='cis/grad/kdk2963/alt_seawifs_vars.tmp'

  Filename for temporary storage of routines between parts 1 & 2
  (placed in working directory...)

  rout_filename='cis/grad/kdk2963/alt_seawifs_routines.tmp'

- Anticipated filename of GCP's to select between parts 1&2 (should remain constant)

  Filename of ground control points selected between parts 1 & 2
  gcpsfilename='cis/grad/kdk2963/gcp pts'

- Instructions filename (should remain constant)

  Filename of instructions for part II. displayed in widget
  instr_filename='cis/grad/kdk2963/data/alt_seawifs_instructions.txt'
Once you have set all the elements in MASTER_FILE accordingly, start ENVI. The menu buttons should appear under the 'Transforms' option. However, you first need to compile all the IDL procedures. To do this, you must run the 'swf' script. To run a script in ENVI, type at the command line the 'at' (a) symbol followed by the script file. In this case, you would use 'a swf'. You may need to play with the path (ie. @idl/swf) to get it working correctly.

Now you should be able to select "Alt SeaWiFS Atm. Correction. Part I" from the ENVI menu. The program should start, and you should see a running commentary at the command line. At the end of step I, follow the instructions in the window, then start step II.

And that's it!

PART VI
TROUBLESHOOTING

As this is the first version of ALT_SEAWIFS, a full verification and debugging process has not yet been completed. Therefore, this troubleshooting section is empty. However, if you have problems, please write me at kdk2963@rit.edu!

PART VII
VERSION CHANGES

Version 1.1. August 15, 2000
-bugs were fixed in output generation of HDF files. Changes made to outtohdf.pro so that all SDS data from L1b image is transferred to the L2 image.
-note: For some reason, SeaDAS still cannot open a L2 Alt_SeaWiFS file as a standard 'SeaWiFS' file. Instead, it must be opened as an 'HDF' file

Version 1.2. September 9, 2000
-an interactive thesholding routine was added to help in creating the land/cloud mask. The name of this routine is TWEAKTHRESH.PRO
Changes were also made to MAKEMASKLAND.PRO