The Pennsylvania State University

The Graduate School

College of Engineering

ESTIMATION OF ATMOSPHERIC BOUNDARY LAYER DEPTH FROM RAMAN LIDAR USING 2-D SPATIAL FILTERS

A Thesis in

Electrical Engineering

by

Sriram N. Kizhakkemadam

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2002

Sriram N. Kizhakkemadam

We approve the thesis of Sriram N. Kizhakkemadam.

Date of Signature

C. Russell Philbrick
Professor of Electrical Engineering
Thesis Advisor

Lynn Carpenter
Professor of Electrical Engineering.

Kenneth Jenkins
Professor of Electrical Engineering
Head of the Electrical Engineering Department

Abstract

Estimation of the depth of the various layers in the Atmospheric Boundary Layer (ABL) is of interest to the meteorological, air quality and aeronautical community. While previously published literature on this topic concentrated on using wavelet techniques applied to wind profiler or lidar backscatter data, a case is presented for using Raman lidar to describe the ABL depth. The techniques introduced in this thesis complement, as well as offer an alternative to the existing technique of ABL depth estimation using wavelet techniques. The image processing techniques introduced in this thesis demonstrate a capability to detect as well as follow gradients in the ABL time series, to obtain the time series of atmospheric layer boundaries. The algorithm performs edge detection and then uses dynamic programming techniques to perform edge following. The 2-D spatial filters are used for edge detection. These filters utilize information of the current and neighboring profiles to estimate gradients in the current profile. This additional information compared to the 1-D wavelet technique yields a robust gradient detection scheme. The edge following algorithm can also be used with the wavelet technique mentioned in the published literature to obtain a time series of the boundaries between the various layers in the atmosphere. Examples are described showing the traces of multiple layers in the atmosphere with the new technique, which is now available for routine data processing.

Table Of Contents

List of Figures

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Prof. Philbrick for his ceaseless motivation and guidance in the successful accomplishment of this thesis.

Thanks are due to Sachin Verghese, Alex Achey, Corey Slick, Gregg O'Marr, Guangkun (Homer) Li, Steve Esposito, Ginnipal Chadha and Karoline Mulik for their valuable inputs in the form of advice and programs which have been used in this thesis. Collaborative effort with Prof. Kenneth Davis of the Department of Meteorology at Penn State University has helped make great strides in this project.

If I were to find the root cause or person to whom I should be thankful for completing this thesis, I would derive my inspiration from the following Sanskrit verse in the *Brahma Samhita*,

"*Ishvarah paramah Krishnah sat-cid-ananda vigrahah*

Anadir adir govindah sarva karanah karanam"

Translation: "Krsna who is known as Govinda is the Supreme Godhead. He has an eternal blissful spiritual body. He is the origin of all. He has no other origin and He is the prime cause of all causes."

I thank the Supreme personality of Godhead, Lord Krishna and his eternal consort Shrimati Radharani to have inspired me to embark on this venture and pray that this thesis be an offering to their Lotus feet.

Chapter 1

The Atmospheric Boundary Layer

1.1 Introduction

This chapter introduces the Atmospheric Boundary Layer (ABL) and describes its growth/decay. Initially, we will draw on the subject of turbulent flow and boundary layer fluid mechanics to discuss the applicability of these concepts to the layer of atmosphere between the ground to the free troposphere. The influence of the diurnal variation of the solar flux on the Earth's surface will be described because it drives the dynamic processes in the ABL [1]. The role of water vapor as a tracer of the boundary layer dynamics will be the underlying theme of this discussion. Having described the atmospheric layer dynamics, the applications of the ABL and its depth will be discussed. This approach is deemed necessary to determine the choice of instruments used to probe the ABL. The techniques for estimating the ABL depth will be discussed in the following chapters.

1.2 Boundary Layer – A Fluid Mechanics Background

The term boundary layer arises from fluid mechanics [2]. Boundary Layer is an engineering term for the moving fluid layer adjacent to a solid body [3]. Both liquid and gases come under the category of fluids. While gases resist compression, liquids resist both tension and compression. While fluids do not resist shear, they resist the rate of shear, also known as viscosity. In fact, the definition of a fluid is often based on this property of resistance to rate of shear as a material that deforms continuously and

permanently under the application of a shearing stress, no matter how small. Viscosity is formally defined as the resistance to the rate of shear [4],

$$\text{Viscosity} = (\text{Shear Stress}) / (\text{Shear Rate}). \qquad [1.1]$$

Whenever a fluid flows over an immovable surface, the fluid immediately adjacent to the surface does not move (it has the same speed as the surface). This is also called the *'No-Slip'* condition. Figure 1.1 gives a graphical depiction of the velocity of a fluid near a surface. The reason for the occurrence of the no-slip condition can be explained by examining a solid surface at a molecular scale. When a fluid is stationary, its molecules are in a constant state of motion with a random velocity, say v. When a fluid is in motion, there is superimposed on this random velocity a mean velocity, V, sometimes called the bulk velocity, which is the velocity at which fluid moves from one place to another parallel to the surface. At the interface between the fluid and the surface, there exists an interaction between the molecules or atoms that make up the fluid and those that make up the solid. The collision rate and surface roughness result in interactions that are strong enough to reduce the bulk velocity of the fluid to zero. So the bulk velocity of the fluid must change from whatever its value is far away from the solid surface to a value of zero at the surface resulting in the no-slip condition [2]. The no-slip condition requires a region of shearing. Viscosity thus plays a role in determining flow patterns. The velocity of the fluid increases as its distance above the surface increases. Studies have shown that the velocity increases parabolically as the distance from the surface increases. The convexity of the parabolic distribution depends on the speed of

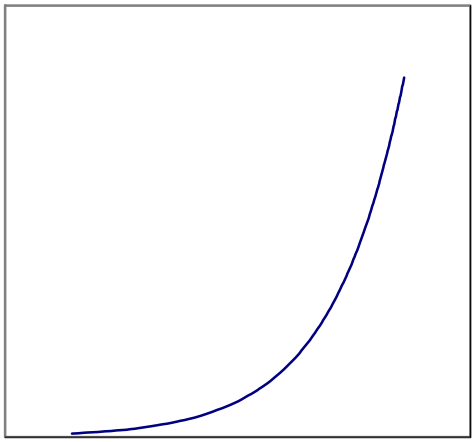


Figure 1.1. Profile of fluid velocity near a surface.

flow in the free flow regime. If a fluid is very resistant to rate of shear (high viscosity), the velocity gradients will tend to be gentle resulting in a relatively great distance between the surface and free-stream and vice-versa. The thickness of the gradient region from the zero speed to the free-stream velocity is called the boundary layer. Due to the asymptotic nature of the distance at which the free stream velocity is reached, the boundary layer is often taken to be the height at which 99% of the free stream velocity is reached. This approximate boundary layer is also known as the Prandtl Boundary Layer [5].

1.3 Atmospheric Boundary Layer

The ground serves as a solid surface and the circulating air above interacts with this “motionless” ground creating friction in a thin layer about 1/10th of troposphere resulting in a boundary layer. This boundary layer is a part of the troposphere (the first layer of the atmosphere above the ground surface). The Atmospheric Boundary Layer is defined as "the part of the troposphere that is directly influenced by the presence of

Earth's surface, and responds to surface forcings with a timescale of about an hour or less" [6]. The flow of wind and the surface friction create shearing forces while the diurnal variation of temperature creates a buoyancy force. Hence, the ABL experiences turbulence due to both mechanical forces and also thermal forces. In addition, the ABL is influenced by the coriolis force due to the Earth's rotation imposed on it. The effective transport of heat, momentum and pollutants is due to turbulence. Thus, the ABL is a turbulent region in a rotating heavily stratified layer [7]. The actual thickness varies from 100's to a few 1000's of meters. The thickness of the ABL depends upon two processes which generate turbulence, surface wind shear and excess buoyant convection is dissipated. The diurnal variation of temperature in the lower troposphere due to local heating of the surface is minimized in the free atmosphere.

1.4 Diurnal Variation of the ABL

Figure 1.2 shows a pictoral representation of the growth of the ABL [6]. From the figure, we observe that the boundary layer structure over land has 4 major components that change over the diurnal cycle. They are the “convective boundary layer” (CBL or the mixed layer), the “residual layer”, the “nocturnal boundary layer” or the “stable boundary layer”, and the “surface layer”. In the presence of clouds, the residual layer can be further subdivided into cloud layer and sub-cloud layer.

1.5 Surface Layer

The surface layer is the lowest region of the boundary layer where the turbulent fluxes and stresses vary by less than 10% of their magnitude [6]. During day, this layer is superadiabatic and statically unstable (potential temperature and water vapor decreasing with altitude and presence of strong wind shear). At night, due to absence of heating it can be statically stable when the surface is cooler than the overlying atmosphere. In this layer we might observe small scale structures such as buoyant vertical plumes and even dust devils. The moisture content of this layer is dependent on the recent history. The wind profile shows a logarithmic increase with height in the surface layer. But, the wind on the surface of the earth is close to zero due to the surface friction.

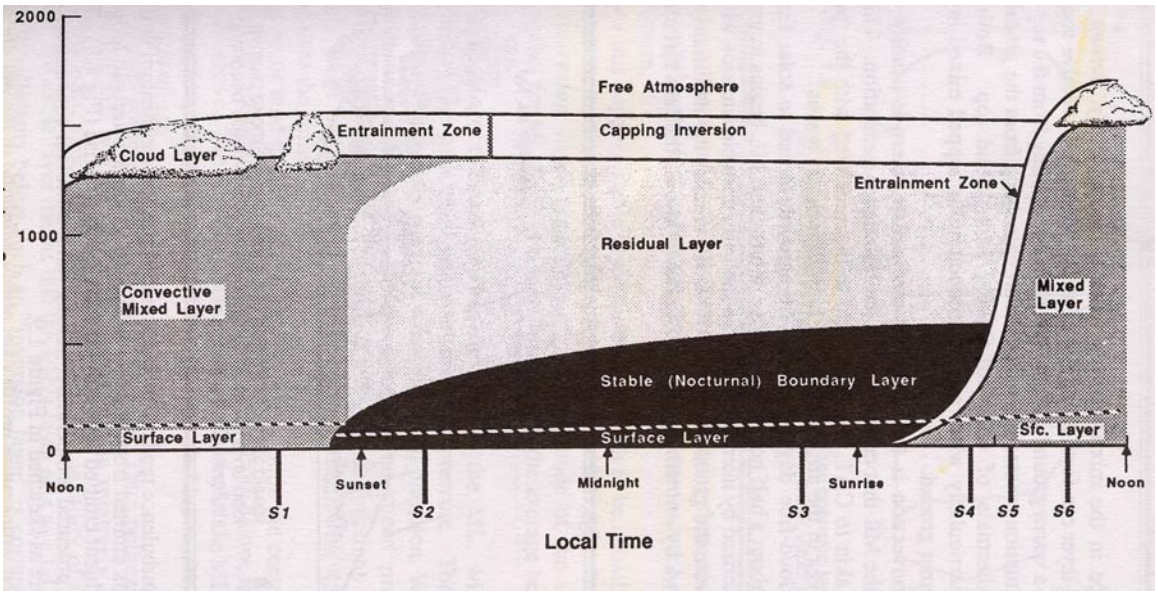


Figure 1.2. Diurnal variation of the boundary layer [6].

1.6 Convective Boundary Layer

The convective boundary layer is the turbulent portion of the atmosphere. It can be divided into three parts, the surface layer (discussed earlier), the mixed layer (comprising the middle 35 to 80%) and the entrainment zone (the top 10% to 60%) [6]. The mixed layer is so called due to the intense mixing in this layer which can be generated mechanically by wind shear or convectively by buoyancy. Of these two sources of mixing, buoyancy gives more uniformity in the vertical distribution. A uniform mixing results in variables like potential temperature and specific humidity having almost constant values over the vertical extant of the mixed layer. The non-instantaneous nature of the mixing process, the entrainment of warm air from the free troposphere, and heating from the warm ground contribute to the deviation of these scalar from a constant value.

The buoyancy generated turbulence can be described based on the solar radiation balance depicted in Figure 1.3 . The prime source of heat in the atmosphere is the solar radiation which is about 344 W/m^2 per day on the average. A major portion of this radiation (51%) is absorbed by the ground. About 30% of the radiation is reflected by the atmosphere to the sky, termed as albedo [8, 9]. The remainder is absorbed by the clouds and other constituents in the atmosphere such as water vapor, ozone, dust etc.

Fluids transfer heat most effectively by convection. Convection is the transfer of heat from one location to another by the ensemble mass motion of the fluid. Hot air masses tend to rise and are replaced by surrounding cooler, more dense air. When air in contact with the surface is heated relative to the cold air overlaying it, the water vapor produced by evaporation at the surface is carried vertically by vigorous convection. As water vapor absorbs radiation, it affects the radiative balance. The final equilibrium

depends on the balance between the radiative and convective effects and is called radiative-convective equilibrium [7].

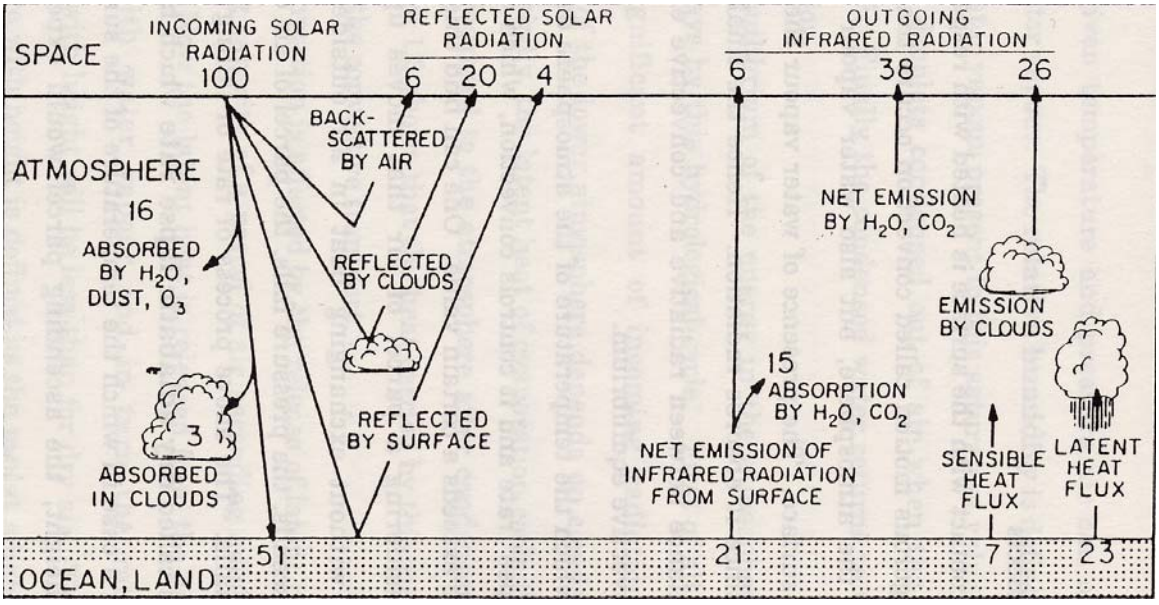


Figure 1.3. Radiation balance for the atmosphere. (From "Understanding Climate Change", U.S National Academy of Sciences, Washington, D.C., Figure 1.6 [10]).

Convection is controlled by the rate of decrease of temperature with height also called as the "lapse rate". In fact, convection occurs only when lapse rate exceeds a certain value. The pressure profile in the atmosphere decreases exponentially and is given by,

$$p(z) = p_0 \exp\left(\frac{-gz}{RT}\right) \qquad [1.1]$$

where p_0 = Ground pressure,
g = Acceleration due to gravity = 9.8 m/s^2,
R = universal gas constant = 8.314 J/mol.K,
T = actual temperature in Kelvin, and
z = altitude variable.

Consider a parcel of air that moves up and down adiabatically, i.e without exchanging heat in air outside the parcel. As the parcel ascends to a region where the

pressure is lower, the parcel of air becomes less dense and expands physically. The kinetic energy of the molecules decreases and hence the temperature decreases. This process, resulting in decreasing temperature due to expansion, is called the adiabatic lapse rate and has a value of -9.8 K/km. If the lapse rate is smaller, the ascending parcel would be warmer and would keep moving upward under its own buoyancy. This unstable situation is the process of turbulent convection. Convection transfers heat upward to reduce the lapse rate till it acquires equilibrium and convection stops. The same idea can be expressed in terms of potential energy. Whenever the adiabatic lapse rate is lower than the lapse rate of surroundings, the potential energy can be reduced by moving parcels adiabatically to different levels (to conserve energy). The equilibrium of energy in the atmosphere is effected in a number of ways by the hydrological cycle in the atmosphere. Convection is also affected by the release of latent heat when moisture condenses to form clouds. The water holding capacity of a parcel of air ascending adiabatically decreases as it cools with height. Latent heat is released from the ascending air parcel saturated with water vapor. Above the region of condensation, the rate of decrease of temperature with height will approach dry adiabatic lapse rate. The rate of decrease of temperature with height in the lower troposphere is defined by the moist adiabatic lapse rate, and has a value that depends on specific humidity, temperature and pressure. For the lower atmosphere, the value is about 4°K/km at 20°C and 5°C/km at 10^0C. The precise values are tabulated in [11].

While the moist adiabatic lapse rate is appropriate for describing upward rising air, the dry adiabatic lapse rate is appropriate for descending air. This follows from the fact that the amount of water vapor a parcel of air can hold increases as the parcel goes

down and so the parcel is always unsaturated. As water vapor is one of the components of buoyant convective motion, water vapor is a tracer for the growth/decay of the convective boundary layer. The daytime variations of mean virtual potential temperature, specific humidity and other scalars are shown in Figure 1.4. The potential temperature is defined as the temperature a parcel of dry air would have if brought adiabatically from its initial state to the standard pressure, selected arbitrarily to be the surface value of 1000 mbar.

1.7 Residual Layer

About half an hour before sunset, the turbulence which was previously present in the well-mixed layer begins to decay due to the decay in the buoyant fluxes. Consequently, a layer gets established with the same initial state variables and same concentration of variables as the previous layer. This layer has neutral stability [6]. Moisture accumulates each night in the residual layer after it has evaporated during the previous day and has been distributed in the mixed layer. The residual layer does not have direct contact with the ground. During the night, the stable nocturnal layer keeps increasing its thickness by modifying the bottom part of the residual layer. The stable layer (to be discussed below) prevents interaction between the surface and the residual layer. Thus the remainder of the residual layer is not affected by turbulent transport of surface related properties [6].

1.8 Nocturnal Boundary Layer

This layer is characterized by stable air with weak thin layers of infrequent turbulence. Temperature inversion is often seen at the top of this layer during calm nights. These inversions are formed due to the surface cooling by infra-red emission. As a result, the ground becomes cooler faster than the air above and we find that the temperature increases with height in nocturnal boundary layer giving rise to an inversion layer (the temperature normally decreases with height in ABL).

The wind profile should ideally exhibit a parabolic behavior due to a surface (ground). Whereas ground level winds are lighter and calmer at night, the winds at higher altitude may accelerate to super geostrophic speeds near the top of the nocturnal boundary layer. The geostrophic wind moves parallel to isobars and occurs above the ABL where the effect of surface friction is absent. Above the top of the nocturnal boundary layer, the wind speed and direction will have a smooth transition to geostrophic values. This layer of peak wind speed close to the top of the stable layer is called low level jet (LLJ) or nocturnal jet [6]. The LLJ usually occurs netween 100 to 800m above the ground. The LLJ can have width spanning hundreds of kilometers and length of a thousand kilometers. Factors that contribute to LLJ include fronts, advective accelerations and sea breezes.

Owing to the stability of the layer, the discharge of pollutants is horizontal and is also known as fanning (horizontal spreading out in thin layers) [6]. The top of the nocturnal boundary layer is poorly defined and blends into the overlying residual layer. It is defined by the presence of very low turbulence intensity compared to the surface value. The stable boundary layer is not restricted to night time only. When warm air advection

occurs over a cooler surface, such as after a warm air frontal passage or near shorelines, then also the stable boundary layer can occur.

Typical vertical profiles of some scalars are shown in Figure 1.4. Due to the presence of an inversion at the top of the ABL, these scalars help us to estimate the depth of the ABL during daytime. The almost constant concentration of the water vapor mixing ratio in the mixed layer is due to the uniform mixing in the turbulent atmosphere during daytime. We also observe a strong gradient in potential temperature profile and a gradient in the water vapor mixing ratio profile at the top of the daytime boundary layer in the entrainment zone.

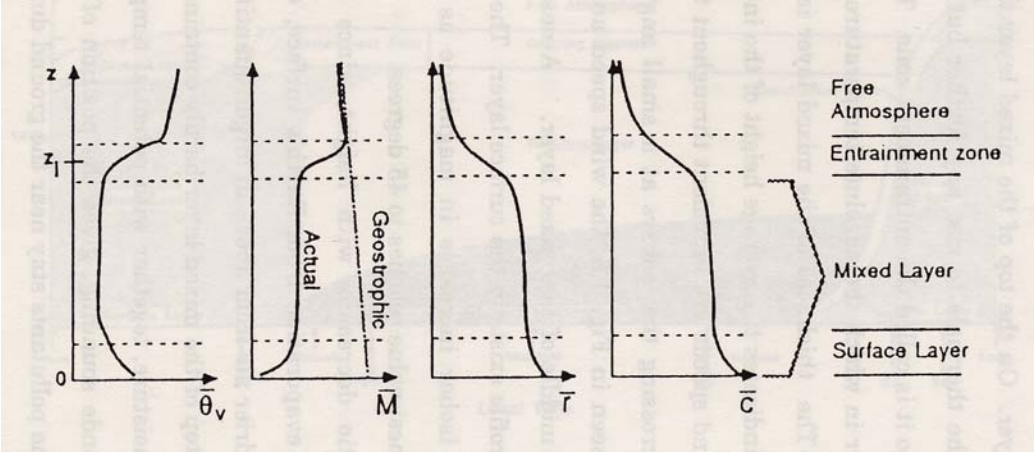


Figure 1.4 Typical daytime variations of mean virtual potential temperature (θ_v), mean horizontal wind speed (M , $M = u^2 + v^2$), water vapor mixing ratio, r, and pollutant concentration, c [7].

1.9 ABL Environment and Its Effects

The ABL properties determine the environment, climate, vegetation, food and quality of air. Some of the many factors important in the ABL include:

1) The weather forecasts of dew, frost, temperature and wind speeds are boundary layer predictions.

2) Pollution mixing and transport depends on the boundary layer dynamics and its evolution. In urban areas, the dispersal of surface generated pollutants and smog depend on the dynamics of the atmospheric boundary layer.

3) Design of structures to withstand wind gusts requires a study of turbulence.

4) Agricultural meteorology and hydrology are concerned with the dry deposition of natural gases and pollutants on crops. The growth of crops is influenced by evaporation, dew and frost formation. The last three factors depend on the intensity of turbulence and energy balance at the surface [1].

5) Aeronautical meteorology: Aircraft are affected by boundary layer meteorological phenomena such as low cloud, low-level jets, intense wind shear leading to high intensity turbulence, especially during landing and take off.

The aforementioned applications elaborated the role of the ABL in our daily life. The depth of the ABL is of particular interest because it serves as a key parameter for evaluation of Numerical Weather Prediction (NWP) models. Accurate estimation of the ABL depth will help provide better weather forecasts. It will also help us to characterize the flow of pollutants dispersed in the boundary layer which consequently affect human health.

Summary

The discussion in this chapter demonstrated that water vapor is a tracer of buoyancy and thermal effects in the atmosphere over the course of 24 hours in a day. As

seen from Figure 1.4, the water vapor mixing ratio exhibits a gradient at the top of the boundary layer which can be used as a marker of ABL height.

Chapter 2

Instrumentation for Atmospheric Water Vapor Profiles

Introduction

From Chapter 1, we concluded that the altitude gradients of the water vapor content that follow the dynamical processes in the atmosphere could provide a technique for continuous monitoring of the thickness of the Atmospheric Boundary Layer (ABL). The thickness of the boundary layer may extend up to 3000 m. Hence, we need remote sensing instruments which can sense the gradients in the water vapor profiles, changes in aerosol distribution or changes in dynamical properties over a specified location. Two instruments, RADAR and LIDAR (Light Detection and Ranging) could provide these capabilities. Raman lidar techniques can give a direct measurement of the water vapor and aerosol distribution. Hence we introduce the Penn State University's LAPS Lidar and its capabilities to measure water vapor profiles which can be used to develop an algorithm for ABL depth. The measurement of the atmospheric water vapor from lidar requires an understanding of Raman scattering and accordingly Section 2.1 will address this topic. The instrumentation of the LAPS Lidar system will then be described briefly in Section 2.2, and Section 2.3 will focus on the calculation of water vapor profiles from the lidar return signal.

2.1 Raman Scattering

If monochromatic radiation of a very narrow frequency band scatters from a molecule, then the molecule's electron cloud will experience an increase in virtual energy equivalent to that of the scattering photon. In most cases, the molecule returns to a

ground state and the energy of the scattered photon is very close to the energy of the incident photon, only slightly shifted by the Doppler velocity of the molecule. This is the classic Rayleigh scattering or elastic scattering case. However, it may happen that the molecule will gain or lose energy relative to the energy of the incident photon. The energy difference between the incident and scattered photons during Raman scattering is characterized by the addition or loss from a change in the rotational or vibrational energy states of the specific molecule. If the molecule gains energy, the scattered photon energy and hence the oscillating frequency of the photon will be reduced. This red-shifted radiation is referred to as Stokes' radiation. On the other hand, if the molecule loses energy, the photon gains energy and consequently oscillates at higher frequency which is termed as blue-shifted or Anti-Stokes' radiation. In the case of vibrational shifts, the intensity of the Stokes' vibrational radiation is about 1000 times smaller and anti-Stokes radiation is about 10 million times smaller than the Rayleigh scattering. The anti-Stokes vibrational scattered radiation is so small because the vibrational states of simple molecules are not likely to be populated at normal atmospheric temperatures. Figure 2.1 illustrates the Stokes and anti-Stokes scattering process.

A unique feature of the Raman scattering process is that the frequency/wavelength shifts are characteristic of the particular molecular species involved, and thus identify the molecular species that caused the scattering. If more than one molecular species exist in the volume, the scattered radiation will have components from all the different species. This suggests that in order to retrieve water vapor profiles from the atmosphere, we can irradiate the atmosphere with a monochromatic source of light (laser beam) and detect the scattered radiation with a very narrowband filter at the

wavelength of the scattered radiation for the given transmitted wavelength. A Raman lidar has capabilities to perform the aforementioned tasks.

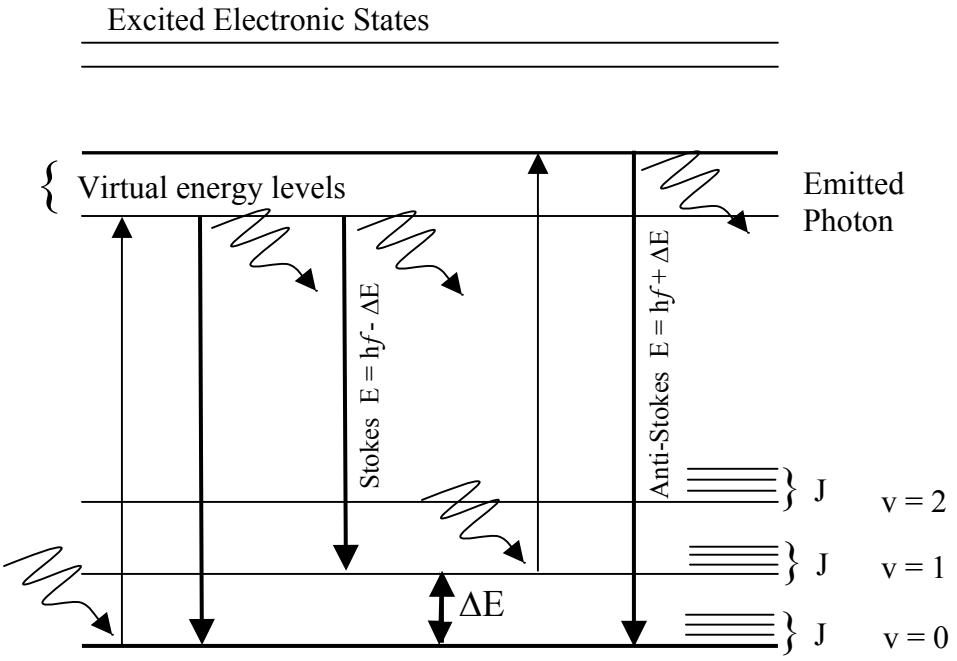


Figure 2.1. Illustration of Raman vibrational scattering [12]

2.2 Lidar Atmospheric Profile Sensor (LAPS) Lidar.

The LAPS lidar is a monostatic, optical remote sensing instrument which employs Raman scattering techniques to investigate properties of the lower atmosphere. The LAPS Lidar was developed at Penn State University [12]. A typical lidar configuration consists of a laser transmitter which sends a pulsed beam to scatter from the molecules and particles in the atmosphere. The backscattered and Raman shifted wavelengths are collected by a receiver consisting of a telescope and detection electronics. The intensity of the received signal is directly proportional to the atmospheric constituents such as

water vapor, ozone and parameters such as temperature and optical extinction. A Raman Lidar uses one or more optical detector channels consisting of filters, beam splitters and other optical components corresponding to each species to be observed. The narrowband output of the detector of each channel illuminates its associated photo multiplier tube (PMT) which generates an electrical signal output. The current output is amplified and quantized by a binary threshold discriminator to yield a logic level signal which is given as an input to a counter to count the number of photons collected from the channel. The measurement is range resolved in 75 m intervals (500 ns bins) and the arrival time of the backscattered light pulse is a measure of the altitude of the scattering volume [14].

The configuration of the various subsystems of the LAPS lidar is given in Table 3.1. The LAPS instrument is contained in two separate housings. The transmitter, receiver and control electronics are inside a weather sealed container, and the control console includes the detector box, the data collection electronics, and a computer used for instrument control, data storage and real time data processing. The transmitter is a high power Nd:YAG laser with a primary wavelength at 1064 nm that is passed through nonlinear crystals, which double and quadruple the primary frequency to produce the output wavelengths at 532 nm and 266 nm. The residual primary wavelength is dumped on a water cooled surface. The details of the transmitter are given in Table 2.2. Figure 2.2 shows the transmitter and receiver of the LAPS instrument. The return signal is collected by a zenith pointing prime focus paraboloid telescope which is focused onto a 1-mm diameter optical fiber. The signal is transferred to a detector box in the console unit by the fiber optic cable. The detector box, shown in Figure 2.3 consists of 8 channels

corresponding to 7 Raman channels and a Rayleigh channel. The details of the 8 channels are given in Table 2.3.

Table 2.1. LAPS Lidar System Configuration [13]

Transmitter	Continuum 9030 –30 Hz 5X Beam Expander	600 mJ @ 532 nm 130 mJ @ 266 nm
Receiver	61 cm Diameter Telescope Focal length – 1.5 m	Fiber optic transfer
Detector	Eight PMT channels Photon Counting	660 and 607 nm – Water Vapor 528 and 530 nm – Temperature 295 and 284 nm – Daytime Water Vapor 277 and 284 nm – Raman/DIAL Ozone 607, 530, and 284 nm – Extinction 532 nm – Backscatter
Data System	DSP 100 MHz	75-meter range bins
Safety Radar	Marine R-70 X-Band	Protects 6° cone angle around beam

Table 2.2. LAPS Transmitter characteristics. [15]]

Laser	Continuum Model 9030 with 5X Beam Expander
Pulse Frequency	30 Hz
Pulse Duration	8 ns
Fundamental Power	1.6 J/Pulse
Power Output at 1064 nm	Dumped into heat sink
Energy Output at 532 nm	800 mJ/Pulse
Energy Output at 266 nm	120 mJ/Pulse - BBO ; 90 mJ/Pulse – KD*P
Beam Divergence (after expansion)	0.1 mrad (FWHM)
Linewidth (1064)	30 GHz (1 cm^{-1})

Figure 2.2 (a) LAPS Transmitter (b) LAPS Receiver (Courtesy C.R.Philbrick)

Table 2.3. LAPS detector channels [13]

Property	**Measurement**	**Altitude**	**Time - Resolution**
Water Vapor	660/607 (H_2O/N_2) 408/387 (H_2O/N_2)	Surface to 5 km Surface to 5 km	Night - 1 min Day & Night - 1 min
Temperature	528/530 353/354 Rotational Raman	Surface to 5 km Surface to 5 km	Night - 10 to 30 min Day & Night - 10 to 30 min
Extinction 530nm	530 nm Rotational Raman	Surface to 5 km	Night - 10 to 30 min
Extinction 607nm	607 nm N_2 Vibrational Raman	Surface to 5 km	Night 10 to 30 min

The optical signals are converted to electrical signals by detecting and amplyfying them through high sensitivity photomultiplier tubes (PMT). Photon counting techniques are used to quantify the backscatter signals. The data collection unit for each channel consists of a photomultiplier tube, amplifier, discriminator, counter and accumulator.

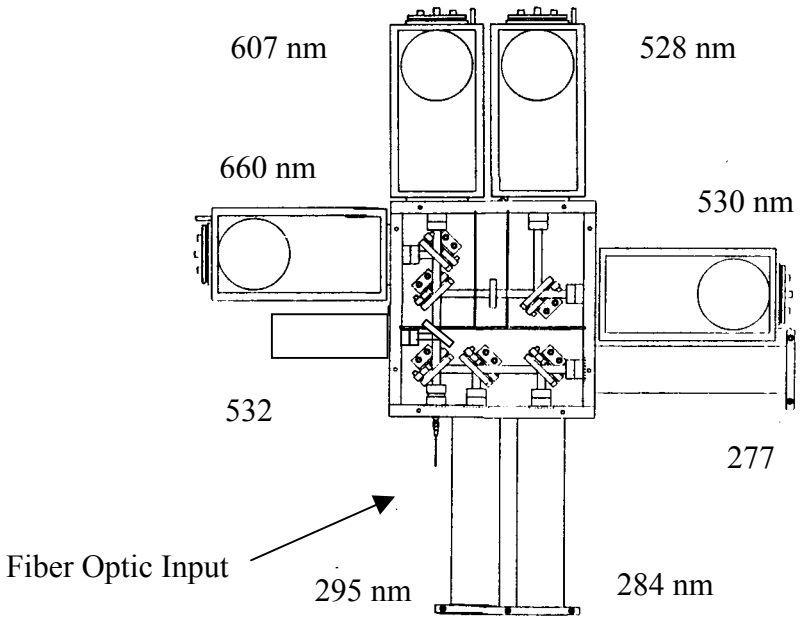


Figure 2.3. LAPS photon detector block diagram [14]

The electrical signal is stored as the photon counts for each altitude bin for each channel by a DSP board and the signal is retrieved by a computer on the console. The DSP board integrates the return signals for one minute intervals in 75 m range bins.

2.3 Measurement of Water Vapor

The LAPS instrument can obtain water vapor profiles from the atmosphere from the 607nm and 660nm channels . The instrument is ideally suited to measure the water vapor mixing ratio, defined as the ratio of number density of water vapor molecules in the atmosphere to the density of ambient air molecules. Nitrogen is a constant portion of dry air in the atmosphere and it is used to describe the density of ambient air. The water vapor mixing ratio in the atmosphere is determined by analyzing the lidar return signal.

The power of a signal measured by a monostatic lidar at a given wavelength is given by the lidar equation [16],

$$P(\lambda_R, z) = E_T(\lambda_T)\xi_T(\lambda_T)\xi_R(\lambda_R)\frac{c\tau}{2}\frac{A}{z^2}\beta(\lambda_T, \lambda_R)\exp\left[-\int_0^z [\alpha(\lambda_T, z') + \alpha(\lambda_R, z')]dz'\right] \quad [2.1]$$

where,

z	is the altitude of the volume element where the return signal is scattered,
λ_T	is the wavelength of the laser light transmitted,
λ_R	is the wavelength of the laser light received,
$E_T(\lambda_T)$	is the light energy per laser pulse transmitted at wavelength λ_T,
$\xi_T(\lambda_T)$	is the net optical efficiency at wavelength λ_T of all transmitting devices,
$\xi_R(\lambda_R)$	is the net optical efficiency at wavelength λ_R of all receiving devices,
c	is the speed of light,
τ	is the time duration of the laser pulse,
A	is the area of the receiving telescope,
$\beta(\lambda_T,\lambda_R)$	is the back scattering cross section of the volume scattering element for the laser wavelength λ_T at Raman shifted wavelength λ_R,
$\alpha(\lambda,z')$	is the extinction coefficient at wavelength λ at range z'.

The received signals of the N_2 and H_2O channels are used to calculate the water vapor mixing ratio. The water vapor mixing ratio for the ultraviolet channel is obtained from the following relationship [15],

$$W(z) = K_{cal} * \frac{S_{H2O}(z)}{S_{N2}(z)} \left(\frac{S_{O2}(z)}{S_{N2}(z)} \right)^{\frac{\sigma_{H2O} - \sigma_{N2}}{\sigma_{N2} - \sigma_{O2}}} \qquad [2.2]$$

where,

S_{O2} is the received signal from the vibrational Raman shift of O_2 at 277 nm.
S_{H2O} is the received signal from the vibrational Raman shift of H_2O at 660 and 295 nm,
S_{N2} is the received signal from the vibrational Raman shift of N_2 at 607 and 284 nm,
σ_x is the is the Raman cross-section of x at the laser wavelength
K_{cal} is a calibration constant.
σ_x can be calculated from standard tables given in [16].

K_{cal} is obtained by calibrating the lidar data with that obtained from radiosondes. Equation (2.2) takes into account the ozone absorption in the solar blind region of the spectrum in the troposphere for the 284nm and 295 nm signals. In the visible band, the signal suffers attenuation from the aerosol backscatter and accordingly, provision is made for it by computing the water vapor mixing ratio obtained from the 607 nm and 660 nm channel as,

$$W(z) = K * \frac{S_{H2O}(Z)}{S_{N2}(Z)} \exp(\sigma_{H2O} - \sigma_{N2}) . \qquad [2.3]$$

Where, $K = K_{cal} * \exp($

2.4 Summary

Penn State University's LAPS Lidar utilizes Raman scattering theory to determine the properties and atmospheric constituents useful for air pollution remote sensing. In particular, we have seen that the two water vapor channels in the visible and ultraviolet give us information about the water vapor mixing ratio for both night and day measurements. This water vapor mixing ratio data will be used by the wavelet and image processing algorithms to be discussed in the following chapters for automatic determination of boundary layer depth.

Chapter 3

Determination of Boundary Layer Depth Using Wavelet Techniques

A capability to routinely determine the boundary layer depth leads to several interesting applications. Thickness of ABL describes the mixing zone where gaseous emissions and air particles are distributed. A qualitative view of the evolution of the boundary layer can be obtained from the visual perception provided by a time sequence of water vapor profiles from backscatter profiles obtained by lidar and radar respectively. However, a measurement of the actual depth of the mixing layer is especially useful for the aerospace industry and for environmental monitoring. Weather prediction depends on various numerical weather prediction models (NWP), such as RAMS (Regional Atmospheric Modeling System) and MM5 (Mesoscale Meteorological Model). The MM5 model describes several meteorological features, including wind speed and direction, air temperatures, and the height of the atmospheric boundary layer [17]. Providing information about one of the outputs of the model, such as the boundary layer depth, as an input to the model will help in the adaptive scaling of the model parameters for a better prediction. The ABL depth is also one of the critical parameters needed as input for atmospheric dispersion models. The accuracy of these models depends on the accuracy of the input data. Lidar provides very high resolution data on the atmospheric parameters. Hence, a good ABL depth algorithm obtained from measurements of lidar will find applications in the atmospheric community. The previous work in the field of ABL depth determination had concentrated on wavelet analysis. Section 3.1 discusses the principles behind the estimation of ABL depth using these previous techniques. Previous published

work has utilized radar as a remote sensing instrument for estimation of ABL depth, but here we will address the correlation between radar backscatter profiles and lidar water vapor profiles. Raman lidar profiles of water vapor have been chosen to estimate ABL depth in this thesis. Section 3.1 will discuss the approach for determination of ABL depth from radar [18]. Section 3.2 will discuss some of the definitions in wavelet analysis. Section 3.3 discusses application of wavelet analysis to an algorithm for ABL depth [18]. These discussions in Chapter 3 lay the foundation of prior work developed by [18] before introducing our new approach in Chapter 4.

3.1 ABL Depth Estimation Principles

Possible methods to compute ABL depth could be developed from measurements obtained using tethersonde, radiosonde, wind profiler, RASS and LIDAR. The usually sharp gradients in aerosol concentration and specific humidity through the Entrainment Zone (EZ) at the top of the boundary layer provide signatures which define the boundary layer depth z_i. Results of z_i measurements from wind profilers have been reported [19, 20]. Investigations have compared results of radiosonde measurements of the PBL depth with those obtained from wind profilers [21]. Also, lidar measurements of aerosol backscatter have been used to describe ABL depth measurement [22, 23]. These investigations reported a good correlation in the gradient height, z_i, between the aerosol backscatter measured from lidar and moisture measured with aircraft. Thus, lidar backscatter can retrieve information on ABL depth based on gradients in aerosol or moisture concentration at the top of the ABL. Wind profilers measure the backscatter, which depends on moisture gradients, and a good correlation has been reported between

lidar backscatter and wind profiler backscatter [24, 25]. Our task of estimating the boundary layer depth will primarily focus on using the Penn State University's LAPS Raman lidar to measure the profiles of water vapor mixing ratio. The LAPS lidar is ideally suited to describe the thickness of the ABL using both the 607 nm, 530 nm, and 284 nm channels for extinction profiles and the ratio of 660 nm to 607 nm for water vapor profiles. The LAPS lidar also has the provision to measure profiles of the direct backscatter using the Pcount system. Hence, each of these 5 profiles can be used for the estimation of ABL height z_i.

A measurement of z_i is also an indicator of the region of the entrainment zone (EZ). In theory, the EZ is defined as the region in which the mean buoyancy flux is negative [26]. The buoyancy flux is given by,

$$\text{Buoyancy flux } (z_i) = E[\, w'(z)\theta'(z)\,], \qquad [3.1]$$

where $w(z)'$ is the turbulent component of the vertical wind velocity at altitude z,
$\theta(z)'$ is the turbulent component of the potential temperature at altitude z, and
E[ ] is the expectation operator averaging over measurement recorded at different times.

As flux profiles at the top of the boundary layer are quite difficult to measure , the EZ has also been defined as the region in which virtual potential temperature, moisture and/or aerosol profiles have strong gradients. In remote sensing, EZ is defined as the region in which more than 5% , but less than 100%, of the air in a horizontal or temporal transect has free troposphere characteristics, such as low aerosol concentration (lower backscatter) and/or low humidity [27, 28].

3.2 Theory of the measurement of ABL Height from a Wind Profiler

The backscattered power from a wind profiler is given by the radar equation [24]

$$P_R = P_T \frac{\eta G^2 \lambda^2 \theta^2 h}{1024\pi^2 \ln(2r^2)}, \qquad [3.2]$$

where η = radar reflectivity per unit volume,
P_T = transmitted peak power,
P_R = wind profiler backscattered power,
G = antenna gain,
λ = radar wavelength,
r = range to the backscatter volume,
θ = beam width,
h = pulse length = 2 x range resolution.

The wind profiler radar reflectivity η is proportional to the refractive index structure constant C_n^2 [25],

$$\eta = 0.38 C_n^2 \lambda^{-1/3} . \qquad [3.3]$$

The average value of C_n^2 is strongly dependent on gradients in the water vapor mixing ratio, q, in a turbulent boundary layer. The relationship between q and specific humidity is given in Appendix A. The average value of C_n^2 within the pulse volume is given by [24],

$$C_n^2 = \frac{\left\langle [n(r+\delta) - n(r)]^2 \right\rangle}{|\delta|^{2/3}}, \qquad [3.4]$$

where < > is a spatial average over the radar volume,
δ is a spatial offset,
n is the radio refractive index.

The value of δ is taken as $\lambda/2$ because wind profilers measure returns from Bragg scattering. When the radar wavelength, λ, projected onto a surface matches a periodic structure on the surface, a resonance effect is obtained causing a strong backscatter. This effect is known as Bragg scattering. In the atmosphere, the variations in refractivity of clear air caused by density gradients can result in Bragg scattering. We observe that P_R is directly proportional to C_n^2, which is directly proportional to the square of the gradient of the refractive index. The refractive index, n, at radar wavelengths is directly proportional to the water vapor mixing ratio, q,

$$n = 1 + \left[\frac{776P}{T}\left(1 + \frac{7.73q}{T}\right)\right] \times 10^{-6}. \qquad [3.5]$$

where P is the pressure in kPa and T is the temperature in K [24]. From the above equations, we see that large values of C_n^2, and therefore P_R, are due to refractive index gradients which correspond to water vapor gradients.

The wind profiler backscatter intensity exhibits a peak that is due to the gradient in water vapor between the boundary layer and the free troposphere. This peak can be used to identify boundary layer depth, z_i. The peak in the reflectivity grows throughout the morning hours and becomes less well defined in the afternoon. An exampler diurnal SNR profile from the radar on 07/14/1999 during the NEOPS campaign is shown in Figure 3.1. As seen from Figure 3.1, there exists a gradient in the backscattered power at the top of the boundary layer. Determination of boundary layer depth is therefore a problem of identifying the gradient location. If the depth is needed for every profile, then wavelets are ideally suited for this application as they focus on a small time resolution.

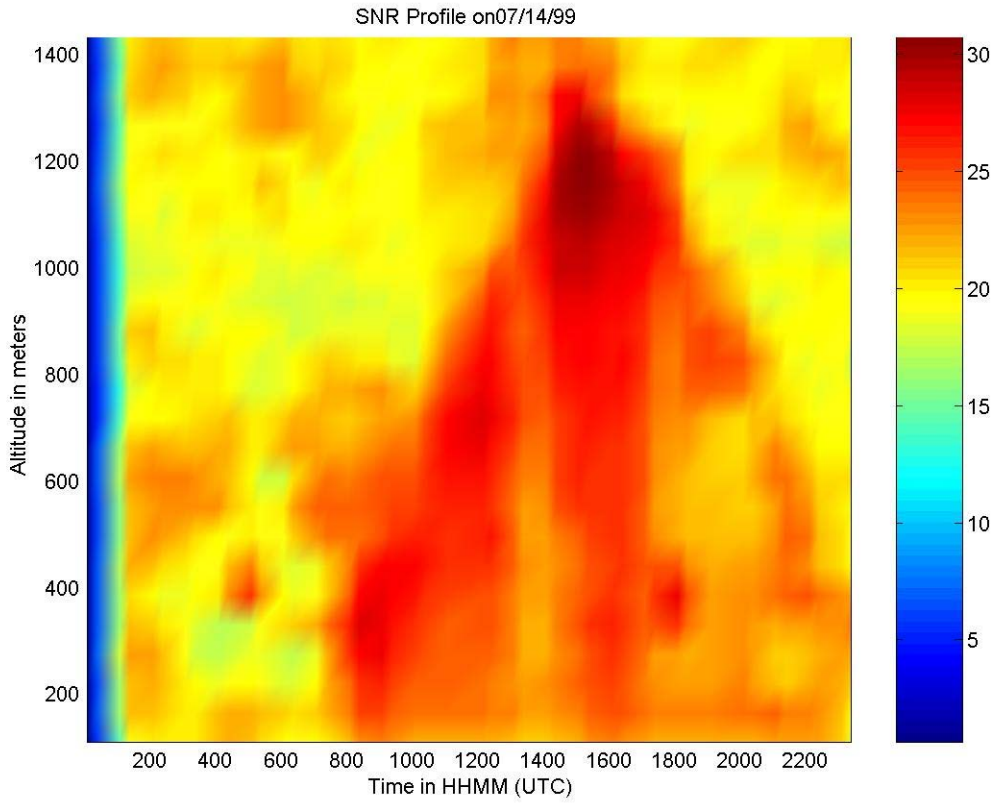


Figure 3.1. Plot of diurnal variation of radar SNR for a 24 hour period on 07/14/1999 during the NEOPS campaign. Radar SNR is directly proportional to the radar reflectivity from Equation 3.2.

Wavelet transforms are capable of providing the time and frequency information simultaneously, hence giving a joint time-frequency representation of the signal. It is especially suitable for geophysical signals which exhibit either non-stationarity or quasi-stationarity. The wavelet transform uses a basis set which is best able to obtain the desired features from the signal. In the case of the altitude profiles of water vapor, a basis set, such as the Haar wavelet which has steep transitions, can be useful to retrieve information about the gradient in the signal at the top of the boundary layer. The Haar wavelet definition is given in the following section.

3.3 Wavelet Transform Definitions and Insights

In this section we estimate the boundary layer depth obtained using a Haar wavelet transform as given in [18]. The Haar function is defined as [29],

$$h\left(\frac{z-b}{a}\right) = \begin{cases} -1 : b - \frac{a}{2} \le z \le b \\ +1 : b \le z \le b + \frac{a}{2} \\ 0 : elsewhere \end{cases} . \qquad [3.6]$$

where z represents the vertical extent in this application, a and b describe the dilation and translation of the function, respectively. In simple words, the dilation is an indication of the spatial scale of the "event", or the altitude resolution required, and the translation gives a measure of the altitude at which the "event" of interest occurs.

The convolution or localized transform is given by,

$$W_f(a,b) = \frac{1}{a}\int_{z_b}^{z_f} f(z)h\left(\frac{z-b}{a}\right)dz . \qquad [3.7]$$

where z_b and z_f are the bottom and final altitudes in the lidar backscatter profile and f(z) is the lidar backscatter signal as a function of altitude, z . While standard texts use $a^{-1/2}$ as a norm (magnitude) preserving normalizing factor, the advantage of the a^{-1} normalization is that sharp transitions can be determined easily [30].

The function $W_f(a,b)$, also known as the covariance transform is a measure of the similarity between the basis set, h, and the function of interest, f. In general, the covariance between functions, f and h, is defined as :

$$\mathrm{cov}_{f,h}(a,b) = \frac{1}{a}\int_{b-\frac{a}{2}}^{b+\frac{a}{2}} f(x)h\left(\frac{x-b}{a}\right)dx\,. \qquad [3.8]$$

The coefficient of correlation between two functions is given by,

$$r_{f,h}(a,b) = \frac{\mathrm{cov}_{f,h}(a,b)}{\left[\mathrm{cov}_{f,f}(a,b)\,\mathrm{cov}_{h,h}(a,b)\right]^{1/2}} \qquad [3.9]$$

For the Haar function considered, $\mathrm{cov}_{h,h}(a,b) = 1$. Noting that $\mathrm{cov}_{f,h}(a,b)$ is the same as $W_f(a,b)$, the above equation can be written as,

$$W_f(a,b) = r_{f,h}(a,b)[\mathrm{cov}_{f,f}(a,b)]^{1/2} \qquad [3.10]$$

In the above equation $r_{f,h}(a,b)$ is a measure of the similarity in shape between f and h in the neighborhood of b. While positive values of $r_{f,h}(a,b)$ indicate in-phase similarity, negative values indicate out of phase similarities. As $r_{f,h}(a,b)$ is just a measure of similarity, it will equally weight signals of arbitrary amplitude but of similar shape. This implies that it cannot distinguish between low variance noise and actual signal representing "an event". On the other hand, $\mathrm{cov}_{f,f}(a,b)$ is a measure of the variance of the signal in the neighbourhood of b and provides a means of distinguishing between "noise" signals (lower variance) and "event signals" (higher variance). Hence, $W_f(a,b)$ is a measure of both the similarity between the basis function, h, and the signal, f, and accounts for the variance of the signal in the neighborhood of b. It should be noted that the covariance transform at low dilations corresponds to finding the features in a very small altitude range and hence will give information about the noise in the region rather

than the signal. A very large value of the dilation, such that the entire altitude profile falls within one half of the Haar wavelet, will simply result in averaging the signal. Hence, a measure for choosing the right value of the dilation constant needs to be found. The appropriate dilation can be chosen by examining the variance of the covariance transform evaluated over all translations [30]. Accordingly, we can utilize

$$D^2(a) = \int_{z_b}^{z_f} \left[W_f(a,b)\right]^2 db\,, \qquad [3.11]$$

which is similar to the power spectrum and it of the portion of variance of f(z) at each dilation a. However, this provides no information about the location of the variance since we have integrated over b. $D^2(a)$ is also referred to as wavelet variance [30]. The maximum in the wavelet variance occurs at the maximum possible dilation. Applying the aforementioned principles, an algorithm has been developed for estimation of ABL depth [18].

3.4 Algorithm for ABL Depth Estimation Using Wavelet Transform Approach

In this section, we will describe a technique to estimate ABL depth using the principles enunciated in Section 3.3.

1) Calculate the covariance transform $W_f(a,b)$ and $D^2(a)$ for each vertical profile.

2) Search for the maximum $D^2(a)$, i.e find,

$$a_{max} = \arg_{max}\,(D^2(a)) \qquad [3.12]$$

3) Search $W_f(a_{max},b)$ for local maxima* and minima* in the translation domain, i.e, find,

$$b_{max} = \arg_{max}\,(W_f(a_{max},b)) \qquad [3.13(a)]$$

$$b_{min} = \arg_{min}(W_f(a_{max}, b)) \qquad [3.13(b)]$$

4) The boundary layer depth for the particular profile will be given by,

$$z_i = b_{max} . \qquad [3.14]$$

In some cases, a visual inspection may reveal that other processes in the atmosphere, or artifacts in the data, may cause the value of b_{max} to not give the right value of the boundary layer depth. A solution to this problem is found by arranging the descending order of local maxima values of the covariance transform and their corresponding translation to determine the boundary layer depth [18], i.e.,

$$b_{max}^{(1)} = \arg_{max}(W_f(a_{max}, b)) \quad \forall\ b \qquad [3.15]$$

Then, we recursively compute for i = 2 to NLmax in steps of 1,

$$b_{max}^{(i)} = \arg_{max}(W_f(a_{max}, b)) , b \neq b_{max}^{j} , j \neq 1,..i\text{-}1. \qquad [3.16]$$

where NLmax is the number of local maxima for a particular vertical profile. The set of possible ABL depths for a particular vertical profile, say the k^{th} vertical profile will be given by the vector,

$$\mathbf{z}_i^k = [\, b_{max}^{1,k}\ b_{max}^{2,k} \ldots\ldots b_{max}^{NL\,max,k} \,] . \qquad [3.17]$$

5) If a time series is being considered, the set of all possible boundary layer depths will form a matrix,

$$\mathbf{Z_i} = [\, \mathbf{z_i^1} \;\; \mathbf{z_i^2} \;\; \ldots \;\; \mathbf{Z_i^{NLmax}} \,] \,. \qquad [3.18]$$

In Sections 3.1 and 3.2, we argued for the application of the ABL depth estimation using data from LAPS Raman lidar. Accordingly, we have applied the wavelet transform approach on data obtained from LAPS lidar by finding only the first local maxima and results from the approach are shown in Figure 3.2. Including a higher number of local maxima and minima, we obtain information about other layers as depicted in Figure 3.3. We observe that while the former technique could only trace one residual layer, the latter approach can trace both the residual layer and the mixed layer as well as the thin cloud entering the mixed layer.

3.5 Summary

The wavelet technique is able to detect the various layers in the atmosphere. We used Equation 3.13a to compute the local maxima for an altitude profile at a particular time, and a time series was obtained by computing the local maxima for all profiles in time steps. The time series of gradient detected from the lidar water vapor profiles only traces the strongest gradients which may or may not be continuous.

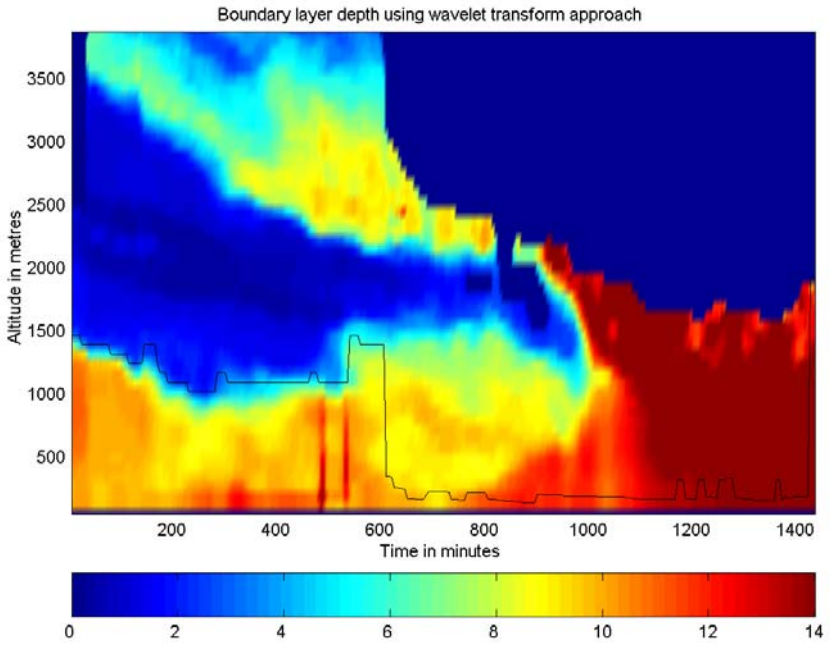


Figure 3.2 Wavelet Transform Technique applied to determine the ABL height for 08/22/98. The black line shows the ABL height estimated from the technique, where only global maxima and minima were taken into consideration.

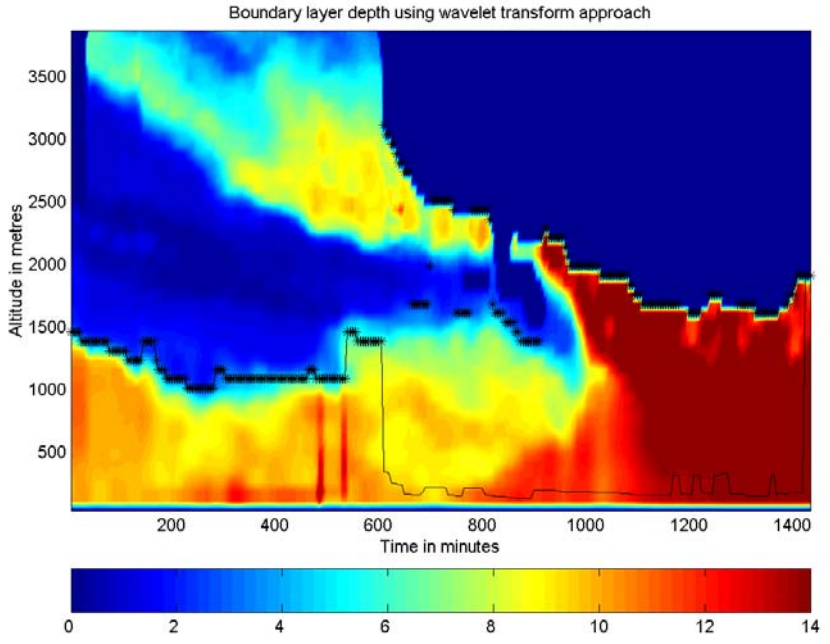


Figure 3.3 Wavelet Transform technique with up to 4 local maxima and minima per profile being used. The black asterisk indicates the values found from the non-global maxima and minima.

The analysis demonstrates that the global maxima traces the night time residual layer and then slips down to the surface layer without tracing the convective boundary layer. Although the successive local maxima trace the CBL, the approach does not produce a continuous trace of the residual layer and/or the CBL.

Chapter 4

Image Processing Techniques to Determine ABL Height

Introduction

The previous chapter discussed the wavelet technique to determine the gradients in water vapor concentration as a tracer of ABL height. The wavelet technique analyzes each profile and identifies the regions of gradients. Apart from the disadvantages of not being able to detect al gradients in the profile and its inability to connect the gradients to trace a layer, it suffers from one more disadvantage, the general shape of the gradient needs to be known. The choice of Haar wavelet in the previous chapter was linked to the shape of the gradient. When the change in the measured parameter at the top of the boundary layer does not follow the sharp gradient of the Haar wavelet, we are required to choose an intelligent wavelet basis. This requires individual effort and can become cumbersome to a person without a background in wavelets. While the ABL depth calculation routine can incorporate different wavelet basis functions and the operator needs only to choose one of them, the real world data always contains cases which may not fit any of the wavelet basis. In order to obviate this possibility, an image processing routine has been developed to extract boundary layer depth. The motivation for using image processing techniques comes from the recognition that lidar and radar data provide a two dimensional signal in space and time. An extensive background of work on edge/gradient detection already exists for two dimensional images. Therefore, edge detection techniques are developed and applied to determine ABL height in this thesis. A search has not located any published work in the literature on using these techniques for

ABL height measurement. The estimation of ABL depth using Image Processing techniques involves 4 steps :

1. Preprocessing the data to enhance the edges,
2. Edge detection using gradient techniques,
3. Edge linking,
4. Edge following.

4.1 Preprocessing the data

Preprocessing is used to remove noise in the image. Lidar data provides signals corresponding to a particular point in time and space which can be treated as a pixel of an image. Digital image noise usually appears in the high frequencies of the spectrum and hence a low pass filter is used to eliminate the effects of noise. But, in edge detection problems, care should be taken to ensure that edges, which are high frequency signals, are retained. A median filter fulfils this task efficiently [31]. Median filter belongs to a class of filters known as order-statistic filters in which the order of pixels is used for filtering. Applying a median filter involves replacing the gray level of each pixel by the median of the pixels in its neighborhood. A neighborhood of pixels is obtained by considering pixels in x and y directions on both sides of the pixel in consideration. This method is particularly effective when the noise pattern consists of strong, spike like components and the characteristic to be preserved is the edge sharpness. This approach achieves noise reduction without blurring [32].

Let **G**(i,j) be the grayscale value of a pixel in the image and consider a 3 x 3 neighborhood. The value of the pixel after median filtering will be :

$$\mathbf{G}(i,j)' = \text{median}\{\mathbf{G}(i-1,j-1), \mathbf{G}(i-1,j), \mathbf{G}(i-1,j+1), \mathbf{G}(i,j-1), \mathbf{G}(i,j), \mathbf{G}(i,j+1), \mathbf{G}(i+1,j+1), \mathbf{G}(i+1,j), \mathbf{G}(i+1,j+1)\}.$$

$$= \text{Middle value}(\text{sort}\{\mathbf{G}(i-1,j-1), \mathbf{G}(i-1,j), \mathbf{G}(i-1,j+1G), \mathbf{G}(i,j-1), \mathbf{G}(i,j), \mathbf{G}(i,j+1), \mathbf{G}(i+1,j+1), \mathbf{G}(i+1,j), \mathbf{G}(i+1,j+1)\}). \quad [4.1]$$

Another common procedure is to move a "mask" through the image and change the pixel values. A "mask", say '**M**' which is also referred to as a kernel, is a matrix of values such that the number of rows and columns are odd. The element corresponding to the center element of the masking matrix is the value corresponding to the pixel in consideration. The pixel and its neighborhood is multiplied by the elements of the mask and the new value of the pixel is the summation of these products. As seen from the equation below, "masking" is the convolution of a two dimensional signal with the filter also known as mask or kernel.

$$\mathbf{G}'(i,j) = \sum_{k=-2}^{k=+2}\sum_{l=-2}^{l=+2} G(i-k, j-l) * M(k,l) \quad [4.2]$$

A mask used for edge detection is shown in Figure 4.1. The large value of the center pixel and the gradual

1	**1**	**1**	**1**	**1**
1	**2**	**2**	**2**	**1**
1	**2**	**17**	**2**	**1**
1	**2**	**2**	**2**	**1**
1	**1**	**1**	**1**	**1**

Figure 4.1 Edge Filter Mask

Some of the features of the median filter are:

1) Nonlinearity and hence

$$\text{median}\{x(m) + y(m)\} \neq \text{median}\{x(m)\} + \text{median}\{y(m)\},$$

2) Useful for removing isolated lines or pixels while preserving spatial resolutions,

3) Performs well on images containing binary noise but performs poorly when the noise is Gaussian,

4) Poor performance when the number of noise pixels in the window is greater than or half the number of pixels in the window.

In some cases, the histogram of the image after edge detection revealed binary noise. In our case, the noise in the images is Gaussian and not the impulse noise that would be reduced by median filtering. Hence preprocessing the image by a median filter would not be beneficial. However it was observed that after edge detection, the main characteristic noise in the image was binary. Hence, better results were obtained by applying a median filter after the edge detected image was formed. The edge detected and median filtered image was then used as the input to a tracking algorithm whose edge retaining features were found to be improved.

4.2 Edge Detection

An edge is the boundary between two regions with relatively distinct gray level properties [33]. The basic approach to edge detection is the computation of a local derivative operator. From Figure 4.2, we can conclude that the magnitude of the derivative of the gray-scale profile will help in detecting the presence of an edge and the

magnitude of the second derivative will give information about the region which the edge is delineating.

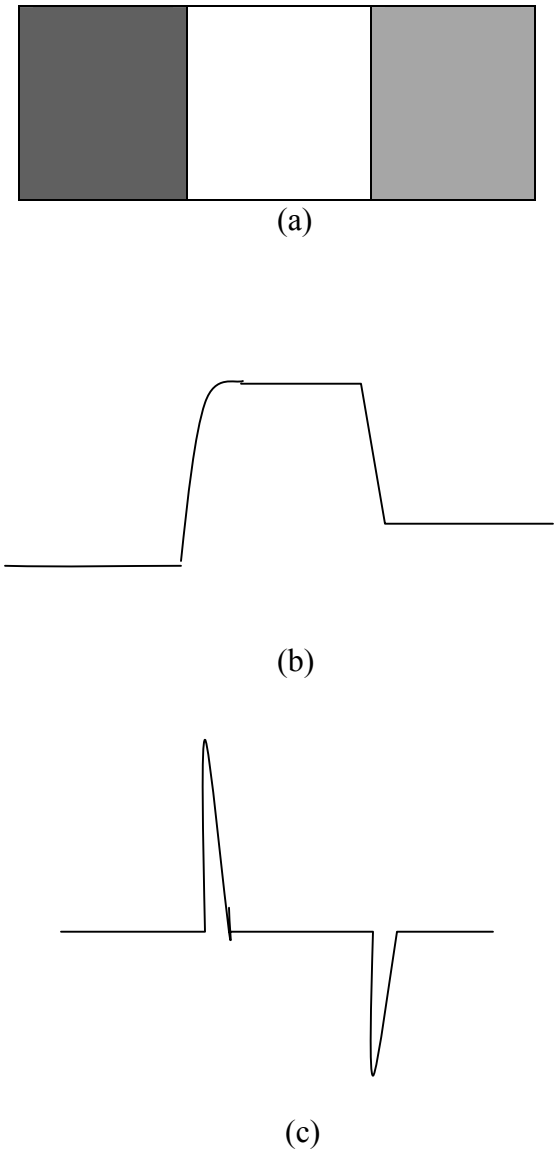


Figure 4.2 : A gray scale image (a) Its intensity profile (b) First Derivative (c) Second Derivative

Derivatives are implemented in digital form using masks. Some of the common operators or masks are Sobel, Prewitt, Laplacian, Laplacian of Gaussian and Derivative of Gaussian . Examples of these masks are given in Figure 4.3 [32].

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

1	1	1
0	0	0
-1	-1	-1

(c)

0	-1	0
-1	4	-1
0	-1	0

(d)

0	0	3	2	2	2	3	0	0
0	2	3	5	5	5	3	2	0
3	3	5	3	0	5	3	3	3
2	5	3	-12	-23	-12	3	5	2
2	5	0	-23	-40	-23	0	5	2
2	5	3	-12	-23	-12	3	5	2
3	3	5	3	0	5	3	3	3
0	2	3	5	5	5	3	2	0
0	0	3	2	2	2	3	0	0

(e)

Figure 4.3 Some common masks used (a) Sobel Vertical gradient detector (b) Sobel horizontal gradient detector (c) Prewitt vertical (d) Laplacian. (e) Laplacian of Gaussian for standard deviation = 1.4

4.3 Laplacian Mask

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore used for edge detection. The Laplacian is frequently applied to an image that has first been smoothed with an operator approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise, and hence the two variants will be described together here. The operator normally takes a single graylevel image as input and produces another graylevel image as output. The Laplacian *L(x,y)* of an image with pixel intensity values *I(x,y)* is given by,

$$\mathbf{L(x,y)} = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2}, \qquad [4.3]$$

where I(x,y) is the Intensity of the gray scale pixel [32].

Since the input image is represented as a set of discrete pixels, we need to find a discrete convolution kernel that can approximate the second derivatives. Using one of these kernels, the Laplacian can be calculated using Equation 4.2. These kernels approximate a second derivative measurement on the image and they are very sensitive to noise. Hence, the Laplacian cannot be used in isolation. Moreover, the Laplacian operation produces double edges and is unable to detect edge direction. To counter this, the image is often Gaussian smoothed before applying the Laplacian filter [32]. A more general use of the Laplacian is in finding the location of edges using its zero crossings property. This concept is based on convolving an image with the Laplacian of a 2-D Gaussian function. A two dimensional Gaussian function is of the form,

$$h(x,y) = \frac{1}{\left(\sqrt{2\pi}\right)^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad [4.4]$$

where σ is the standard deviation. In order to obtain the Laplacian of Gaussian (LoG), we can change variables to polar form for better manipulation. If $r^2 = x^2 + y^2$, the Laplacian of h (i.e, the second derivative of h with respect to r) is,

$$\nabla^2 h = \left(\frac{r^2 - \sigma^2}{2\pi\sigma^4}\right)\exp\left(-\frac{r^2}{2\sigma^2}\right) . \quad [4.5]$$

Substituting $r^2 = x^2 + y^2$ in the above equation, we get, the 2-D LoG function centered on zero and with Gaussian standard deviation σ of the form

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \quad [4.6]$$

This pre-processing step reduces the high frequency noise components prior to the differentiation step.

The advantages of using the Laplacian of Gaussian are:

- This method usually requires far fewer arithmetic operations because the Gaussian and the Laplacian kernels are usually much smaller than the image.
- The LoG ('Laplacian of Gaussian') kernel can be precalculated in advance so only one convolution needs to be performed at run-time on the image.

Figure 4.4 shows a plot of the LoG function in three dimensions. Results from applying the LoG mask and the edge detection to the image of 08/22/1998 is shown in Figure 4.5.

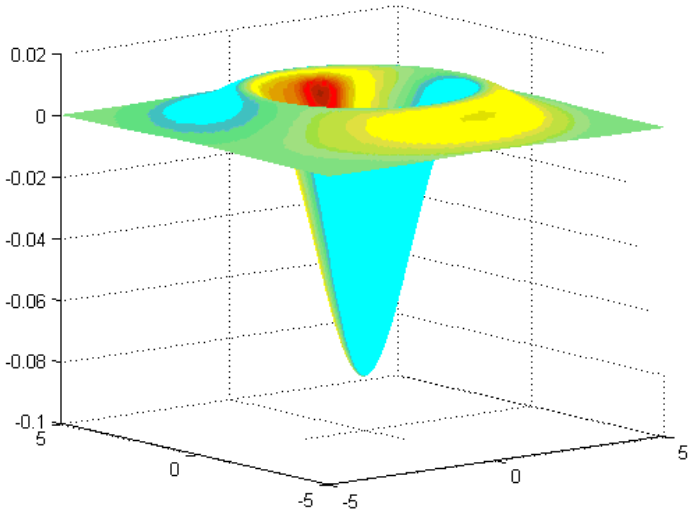
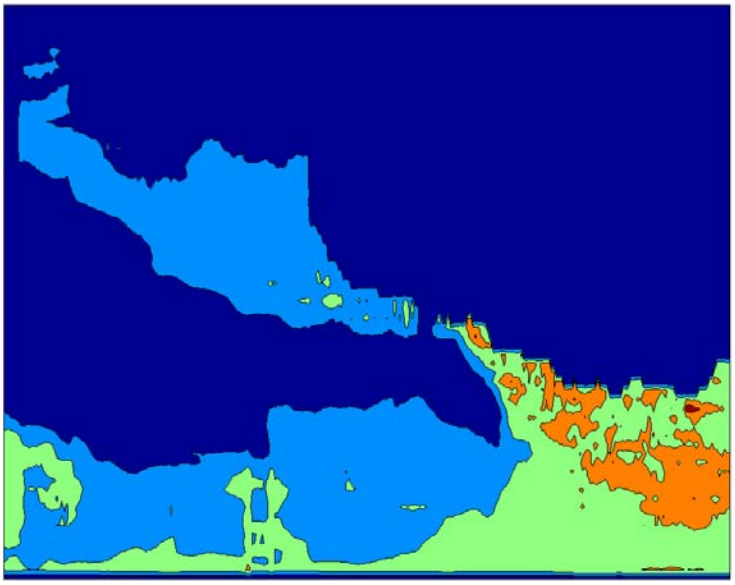
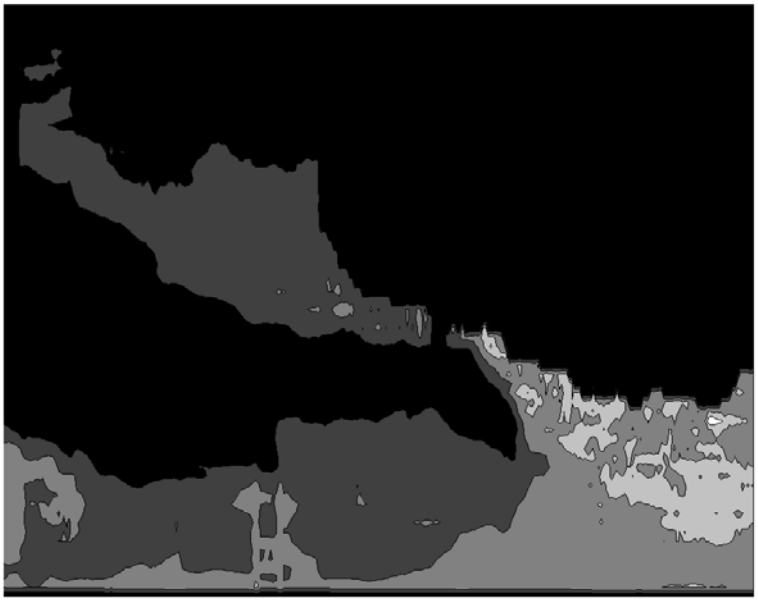


Figure 4.4 A 3-dimensional view of the Laplacian of Gaussian function for variance =1.4.

Other filters such as Canny, Prewitt and Sobel were also tried and their results are shown in Figure 4.6(a)–(e). One of the parameters which controls the edge features is the variance used for the mask. It was found that in order to reduce the noise and only detect the prominent edges, the variance of the filter needs to be minimized. A lower variance for the filter implies less tolerance to the outliers in the image, which in our case happen to be treated as noise. Among the Canny, Sobel, Prewitt, Roberts Cross and Laplacian of Gaussian (Log) filters, the LoG filter was able to obtain the image with the least noise,see Fig.4.5(e). We can obtain a similar output with improved noise performance from non-LoG filters,but at the expense of losing some edges pertaining to the main features of interest.

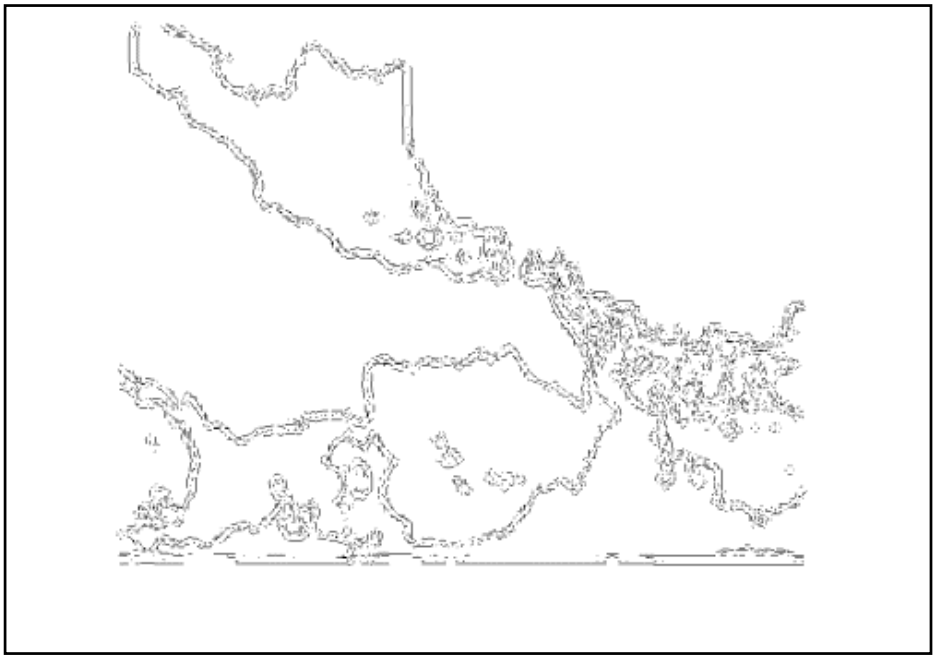


(a)

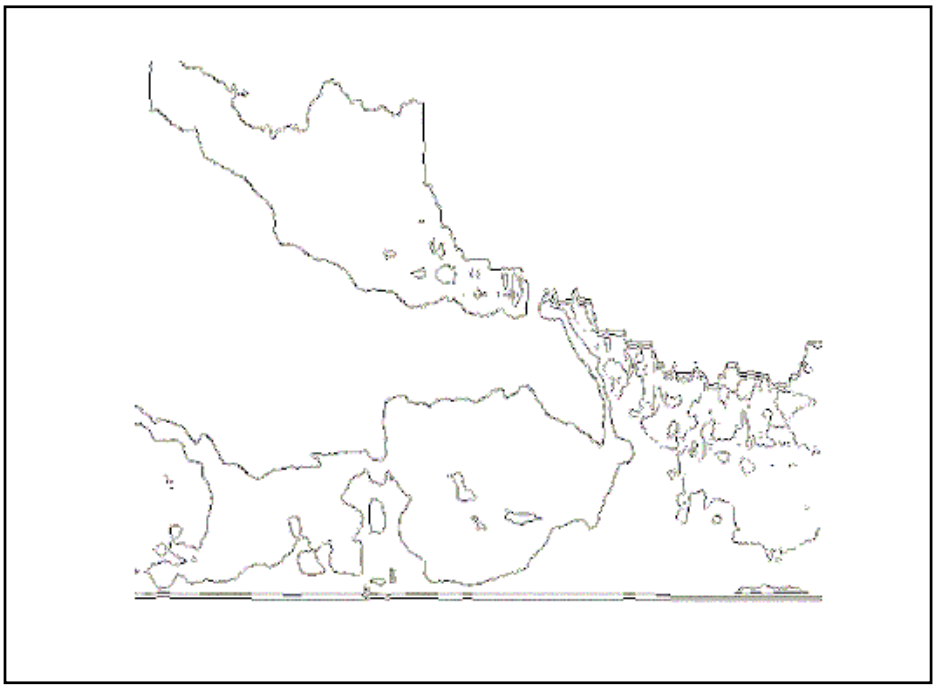


(b)

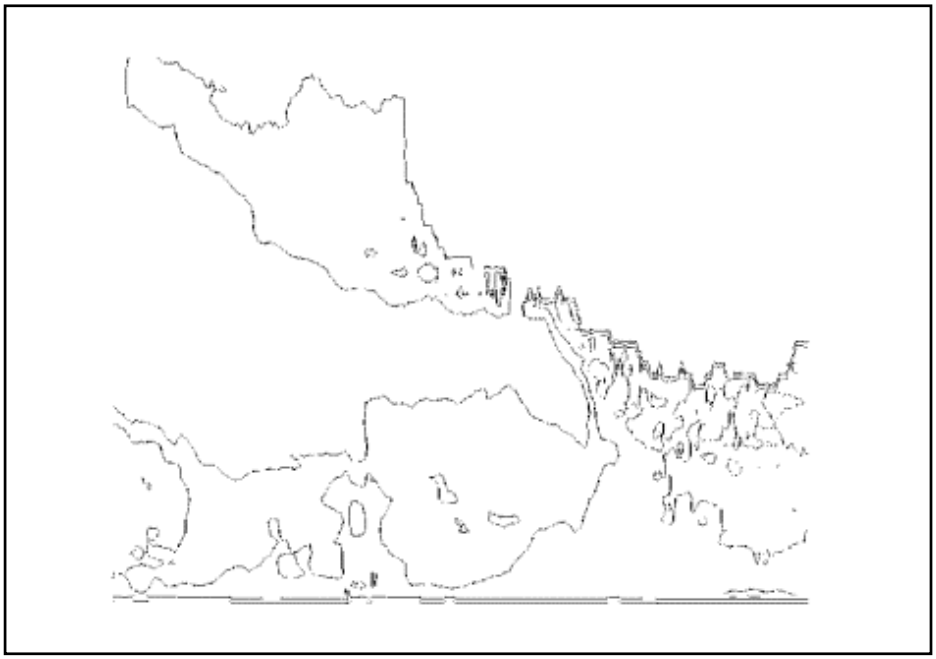
Figure 4.5. Contour plots of the data shown in Figure 3.2 (a) Color contour (b) Gray contour



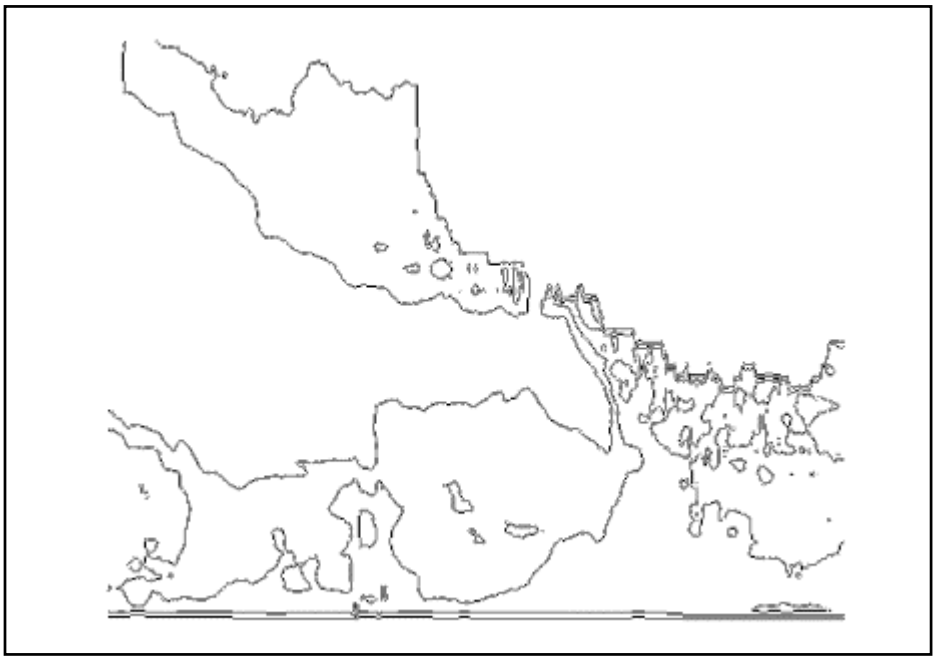
(a)



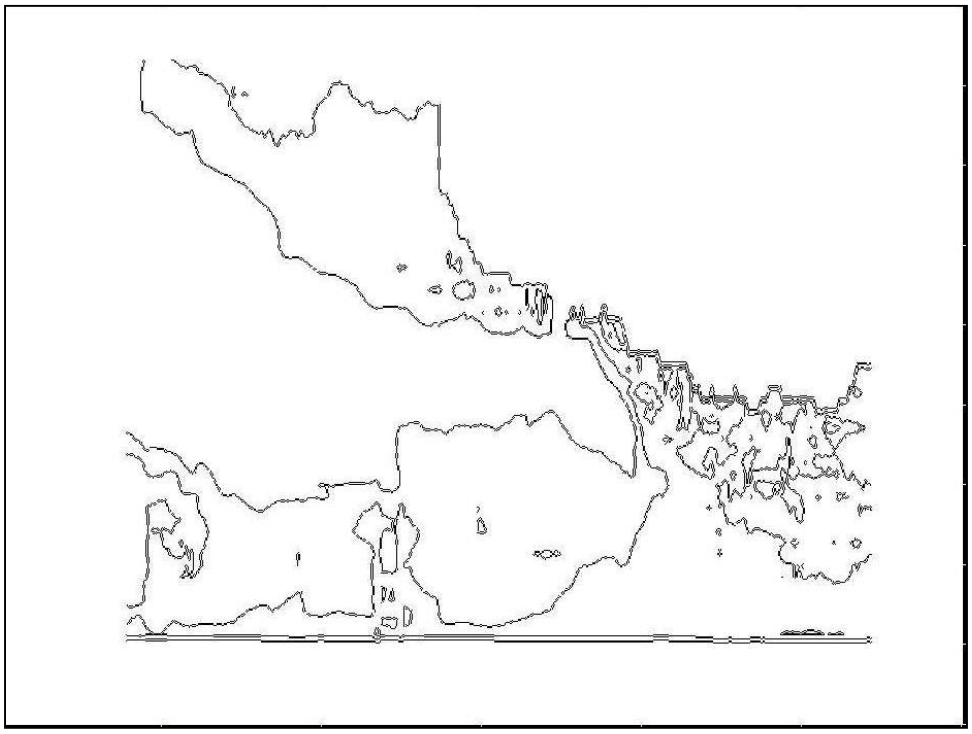
(b)



(c)



(d)



(e)

Figure 4.6. Edge Detection of the gray contour plot of Figure 4.4 (b) by various operators (a) Canny, (b) Prewitt, (c) Sobel both directions, (d) Roberts cross operator, (e) Laplacian of Gaussian (LoG).

A comparison of the result of the various edge detection techniques on the example dataset gives us some important insights. All the filters discussed earlier were able to obtain edge features from the image. The Canny filter of Figure 4.6 (a) gave thick boundaries which is not highly desirable. The Prewitt, bidirectional Sobel, and Roberts' cross operator of Figures 4.6 (b,c,d) were able to detect most of the gradients in the image, including those that did not fall in the purview of ABL boundaries such as the features found in the bottom right portion of the image which corresponds to the intense convective mixing during daytime. In contrast, these small scale features which are not part of the ABL boundaries have been eliminated with the LoG filter, see Figure 4.6(e). The small scale features will contribute as noise pixels to the edge following algorithm

and hence they were removed by fixing the variance of the LoG filter to a small value of 0.8. The other filters are also capable of giving the same result with respect to removal of minor features with a decrease in their variance, except that some of the edge information of the ABL boundaries is also lost in the process. Hence, we decided to use the LoG filter for further applications.

Comparison of Figure 4.6 (a-e) with Figure 3.3 shows that while the Haar wavelet basis function could only detect some of the edges with no continuity in the edges, all of the discussed edge filters provided continuous edges. The better performance of the edge detection compared to the wavelet transform technique is due to the 2-D spatial information processed by the edge filters compared to 1-D information used by the wavelet transform. The 2-D spatial filters exploit the spatial correlation of the atmospheric scalar variables. The weights associated with surrounding pixels and the spatial width of the filter determine the detection ability of an edge.

4.4 Edge Following

After edge detection and post processing by a median filter, we are left with a binary image. A visual display of the image will clearly show the different layers in the atmosphere for that time sequence. Often, we may need to know the actual altitude of a particular layer. This would involve tracking the different points to form a path and find N such paths. The edge information is arranged in the image matrix as binary information, 1 corresponding to the presence of an edge at a particular point in space and time and 0 otherwise. This arrangement of the edges on the image can be viewed as a graph with the 1's corresponding to nodes. Thus the problem of tracking k different layers in the atmosphere corresponds to choosing k of the N possible paths in the graph.

Extensive work has been done in the fields of Graph Theory and Operations Research to address these issues [34]. Here we choose a particular technique known as "Bellman's Principle of Optimality."

Bellman's optimality principle states that "***A policy is optimal if, at a stated period, whatever the preceding decisions may have been, the decisions still to be taken constitute an optimal policy when the result of the previous decisions is included. Thus an optimal policy must contain only optimal sub policies*** [35]**.**" In other words, successive local optima give rise to global optima [36].

4.4 Dynamic Programming Algorithm for Edge Following

In order to apply Bellman's principle to determine the k optimal paths, we develop an algorithm given below:

1) Initialize a matrix, to be called Distance Matrix and hereafter denoted as **D** with an arbitrary large value. This matrix will contain the accumulated distance between a node in consideration and its nearest neighbor. The dimensions of the D matrix will be H x T, the number of rows corresponding to the number of altitude bins and the columns corresponding the number of steps in time. For example, using a 2 hour dataset with 30 minutes integration and 5 minute steps, Nminsteps will be 120 /5 = 24 .

2) Initialize another matrix, called the Index Matrix, denoted **I,** with all elements indexed to the lowest altitude to be measured. For convenience sake, we can work with range bin numbers instead of altitudes and hence the number of range bins will be from 1 to (Max Height – Ground Altitude)/75. The upper limit is rounded off to the lowest integer. Hence, **I** can be initialized with 1 for all elements. The **I** matrix will give the

range bin index of the nearest neighbor. The dimensions of this matrix will be the same as D^m.

Our goal is to find N different paths through the nodes (pixels) and choose K of them as representing the various processes in the atmosphere.

3) Reinitialize the first column of **D** to 0 to make the reference distance starting from the first time instant as 0.

4) Advance to the next time step and consider a node. Find the set of all possible previous neighbors. This set will include the nodes of the previous time step, that by definition are edge points (1's). Let **v(**x**)** be a vector of valid edges corresponding to the particular time/column denoted by 'x' (the x-axis represents time whereas the y axis represents altitude). The elements of **v**(x) are

$$\mathbf{v}(i,x) = \begin{cases} 1 & \text{if an edge is present in the } i^{th} \text{ row of the } x^{th} \text{ column } 1 \le i \le H \\ 0 & \text{otherwise} \end{cases}$$

5) Find the nearest previous neighboring node (denoted by n) from the previous time index and store it as a valid previous index in the matrix I in the row and column of the Index matrix corresponding to the current node.

$$\mathbf{1} = [1\ 1\ldots..1]^T \quad H \times 1 \qquad [4.7a]$$

$$n(x,y) = \min(\arg_{\min} \{ | \mathbf{1} - \mathbf{v}(x\text{-}1) | \}) \qquad [4.7b]$$

$$\mathbf{I}(x,y) = n(x,y); \qquad [4.7c]$$

If the previous time index did not have any node (remember that edge detection is not perfect and usually leaves gaps between successive points), we can choose an option to search upto 'p' previous neighbors. The value of 'p' is arbitrarily chosen and depends on a subjective evaluation of the image. In general, n(x,y) should then be replaced by

$$n(x,y) = \arg_{\min} \{| \mathbf{y} - \arg(\mathbf{v}(x-1))|\} \quad [4.7d]$$

$$\text{where } \mathbf{y} = y\mathbf{*1;} \quad [4.7e]$$

If the image contains regions where the lidar data was not available, 'p' should be less than the width of the unavailable time period measured in time steps. In this thesis, we did not use the p previous neighbor concept and used only the immediate previous neighbor for our results.

6) Compute the Euclidean distance from this valid node to the current node and evaluate the sum of this distance and the accumulated distance to the previous node,

$$\mathbf{D}(x,y) = \mathbf{D}(x-1, \mathbf{I}(x,y)) + \sqrt{1 + (y - I(x,y))^2} \; . \quad [4.8]$$

In the general case of p previous neighbors, [4.8] can be written as,

$$\mathbf{D^p}(x,y) = \mathbf{D^p}(x-1, \mathbf{I}(x,y)) + \sqrt{p^2 + (y - I(x,y))^2} \; . \quad [4.9]$$

7) Repeat steps (5)and (6) for every node at a particular time index.

8) Repeat (7) for all time steps.

This completes the forward distance computation. Path tracking has to be done backwards.

9) Sort the accumulated distance of the last time step stored in the last column of **D** and choose the m^{th} smallest value for the m^{th} path. Let its altitude be denoted as $\mathbf{B^m}(m,T)$.

$$\mathbf{B^m}(m,T) = \arg_{min,m}(\mathbf{D}(:,T)) \; . \qquad [4.10]$$

Find the m^{th} best previous neighbor by

$$\mathbf{B^m}(m,x-1) = \mathbf{I}(\mathbf{B}^m(m,x), x) \qquad [4.11]$$

10) Compute Equation (4.11) for x decreasing from T-1 to 2 in steps of -1.

11) Choose K of these N different paths by inspection.

12) Convert the altitude bins to altitude using the relation

$$\text{altitude} = (37.5 + 8) + (\text{altitude\_bin} - 1)*75 \qquad [4.12]$$

Here, altitude_bin is the altitude bin. The altitude is referred to the center of the bin. The ground altitude of 8m and bin size of 75 m for the LAPS lidar has been taken into account.

13) This procedure should result in the high resolution boundary layer information. We have found the boundary information from the image and the result is up-sampled in time. Hence, we need to downsample/decimate the boundary layer information. As the

number of time steps in the image is usually not an integer multiple of the time steps we obtained from the lidar data, we need to decimate using polyphase implementation.

14) Repeat (1) to (12) for m varying from 2 to N.

Results and Discussion

The application of the dynamic programming algorithm to the example data set is shown in Figure 4.7. Only the 1st best path was computed and plotted as black asterisk. The boundary layer depth is plotted as a time series profile in Figure 4.8. It is observed that even the first iteration of the algorithm which computes the path with the shortest distance is able to give a continuous trace of the residual layer and the mixed layer.

Figures 4.9, 4.10 and 4.11 show the tracing of the residual layer at night time during the NEOPS campaign of 1999. In Figure 4.9, there are two gaps when the lidar data was unavailable and one period near 12:30 UTC because of a cloud feature. As a result, we see no profile information corresponding to those time periods. Such data outages are typical in a practical scenario when working with lidar or radars. The edge following algorithm described earlier utilized the 1st nearest neighbor variant which requires edges to be detected within one pixel value in order to obtain a continuous profile. Inspite of this requirement of the nearest neighbor at a pixel in an adjacent column, the algorithm was able to trace the layer due to provison being made for such data outages by assigning the pixel value of the lowest range bin to be a valid edge in the traced path in the event that no other edges are detected at that particular time step. Two points nearest the edge of the data outages should be discarded in the routne application of the analysis. Figures 4.10 and 4.11 show traces of the residual layer at night.

Figures. 4.12 and 4.13 are cases of multiple layers being tracked by the image processing algorithm. In Figure 4.12, we observe the mixed layer decaying to form a nocturnal boundary layer from around 1500 m to 750 m from 00:00 UTC to 04:00 UTC which corresponds to 8:00 PM to 12:00 AM local time. Sunset occurs around 8:00 PM in the month of July at Philadelphia, the site where the data was obtained. The sudden change in the upper altitude data range from 01:00 UTC in Figure 4.11 at 01:00 UTC and 01:20 UTC in Figure 4.12 is due to the availability of the visible signals after dark. The visible channel has a higher power output compared to the ultraviolet channel and can therefore obtain return signals from higher altitudes. We also observe a trapped layer above the nocturnal boundary layer. This residual layer may contain the effluents which are trapped when the buoyant convection disappears at night, also trapping the water vapor content from the previous day's mixed layer. This phenomenon is seen again in Figure 4.13, where a trapped layer is observed between 700 and 1000m above ground. The study of these trapped layers, their growth and chemical composition is of importance in air quality studies. In contrast to the image processing technique, the wavelet technique detected only steep gradients in the image in Figure 4.14. The gradients detected for a particular maxima/minima jumped from one layer to another and thereby did not yield any continuity information. The 2-D spatial filters utilize the local information in altitude and time to arrive at a correct estimate of the local gradient while the wavelet algorithm utilizes only altitude information. The profiles are correlated in time due to the continuous nature of the ABL boundaries. The gradient in a profile will be at an altitude close to a similar gradient at time steps preceding and succeeding the current observation. The 2-D spatial filters make use of this time correlation information

and a proper choice of the mask helps in giving appropriate weight to the available information. This availability of information of another dimension gives the image processing algorithm an advantage over the wavelet algorithm in gradient detection of scalar profiles in remote sensing.

Figure 4.15 shows a tracing of the residual layer during the night of July 30-31 1999 between 10:44 p.m and 1:50 a.m local time. The decay in the residual layer can be attributed to night time subsidence. When a high pressure system is formed in the atmosphere, a wind component which revolves around the high pressure vortex is formed. The vector sum of this wind component with the downward vertical wind velocity during night results in a wind vector at slant angles to the horizontal. This resultant wind vector creates a water vapor flux (the mean of the product of the turbulent quantities of water vapor and vertical wind velocity) which is pushed towards the ground. This results in the decrease in the height of residual layer as observed in Figure 4.15.

Figure 4.16 shows the image processing algorithm tracing the residual layer while Figure 4.17 traces the residual layer. We observe that the trace is quite accurately following the residual layer.

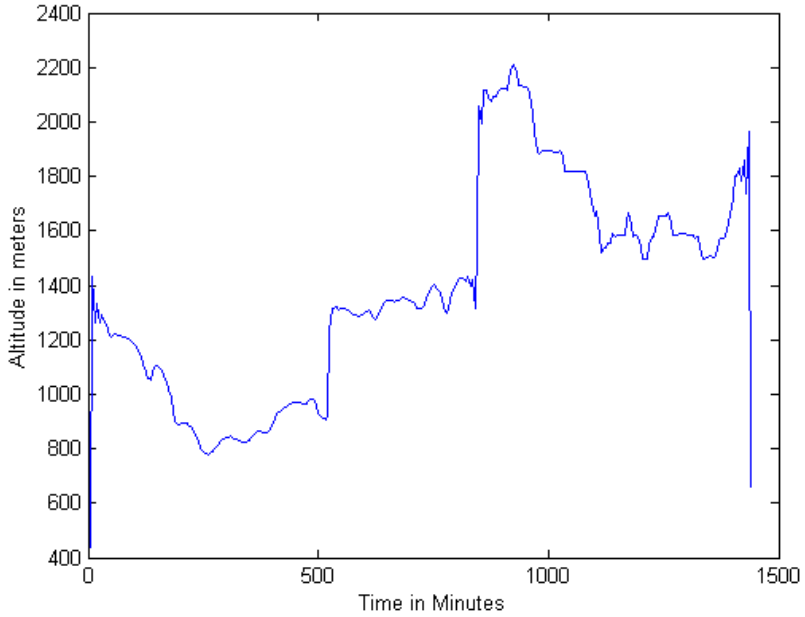


Figure 4.7. The boundary layer depth estimated by the first best path of the proposed algorithm for the lidar data of 08/26/1998 00:00 to 08/27/98 00:00 UTC.

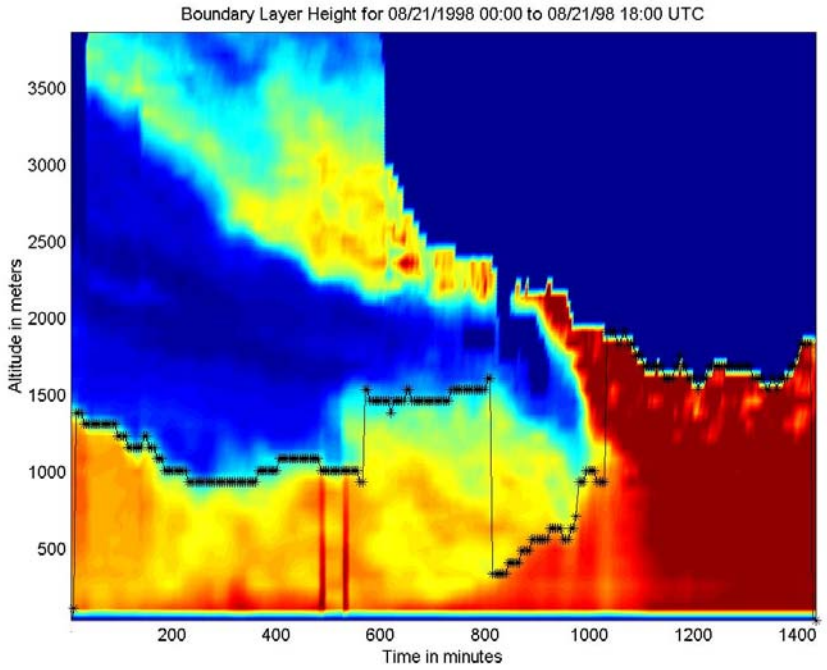


Figure 4.8. ABL depth estimated from time sequence of water vapor for August 21 1998, 05:55 – 11:55 UTC.

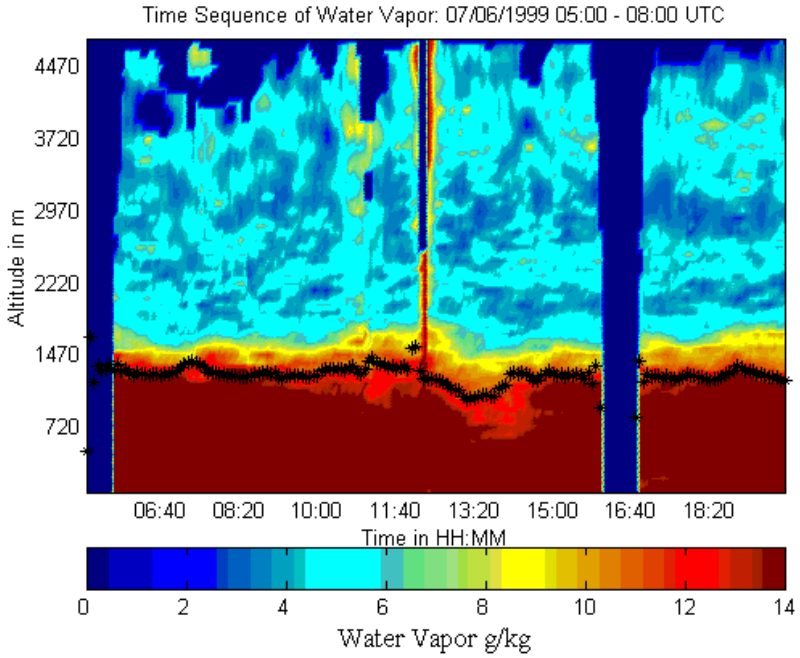


Figure 4.9. ABL depth estimated from time sequence of water vapor for July 06 1999, 05:55 – 11:55 UTC.

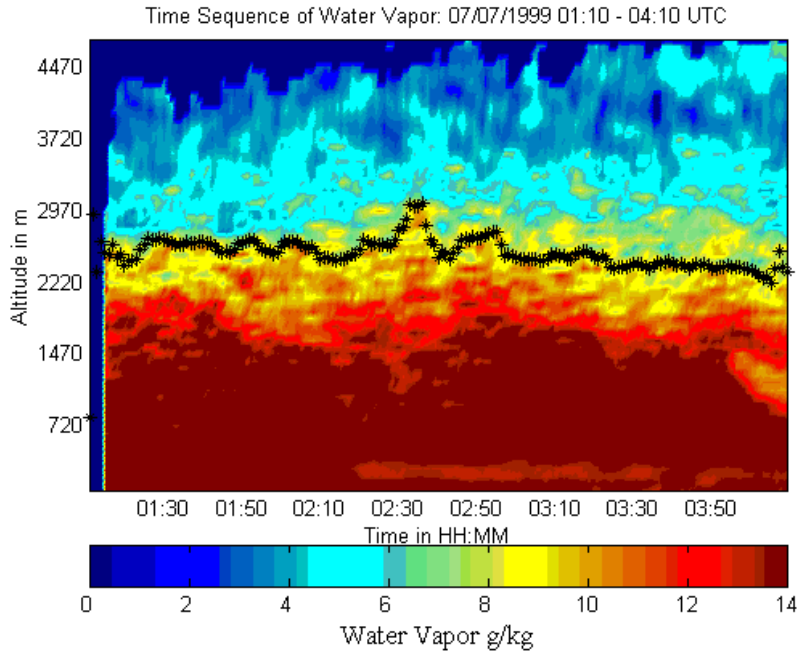


Figure 4.10. ABL depth estimated from time sequence of water vapor for July 07 1999, 05:55 – 11:55 UTC.

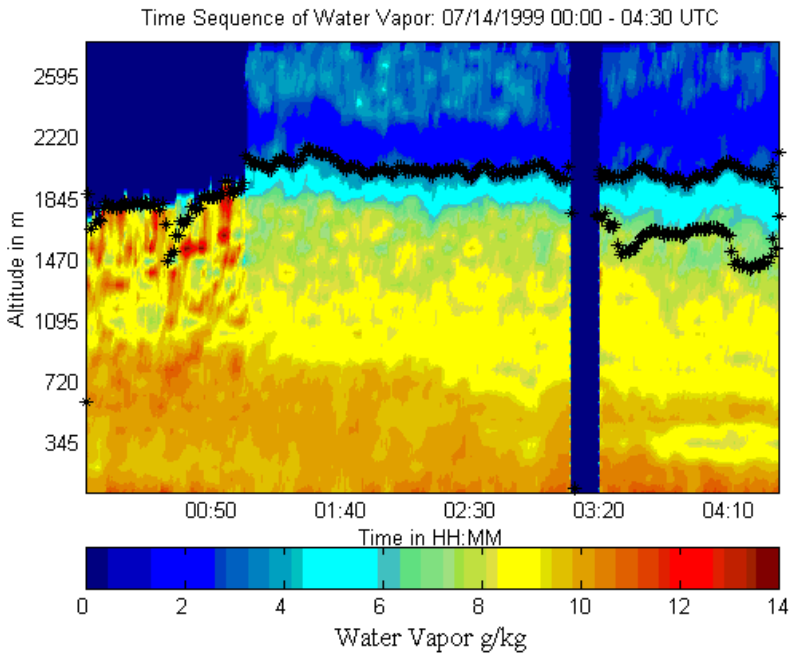


Figure 4.11. ABL depth estimated from time sequence of water vapor for July 14 1999, 00:00 – 04:30 UTC.

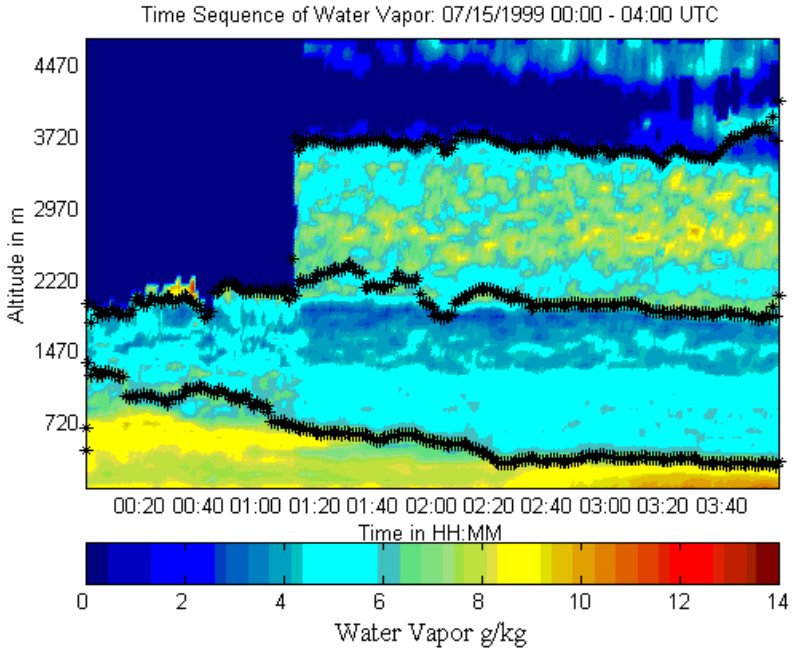


Figure 4.12. ABL depth estimated from time sequence of water vapor for July 15 1999, 00:00 – 04:00 UTC.

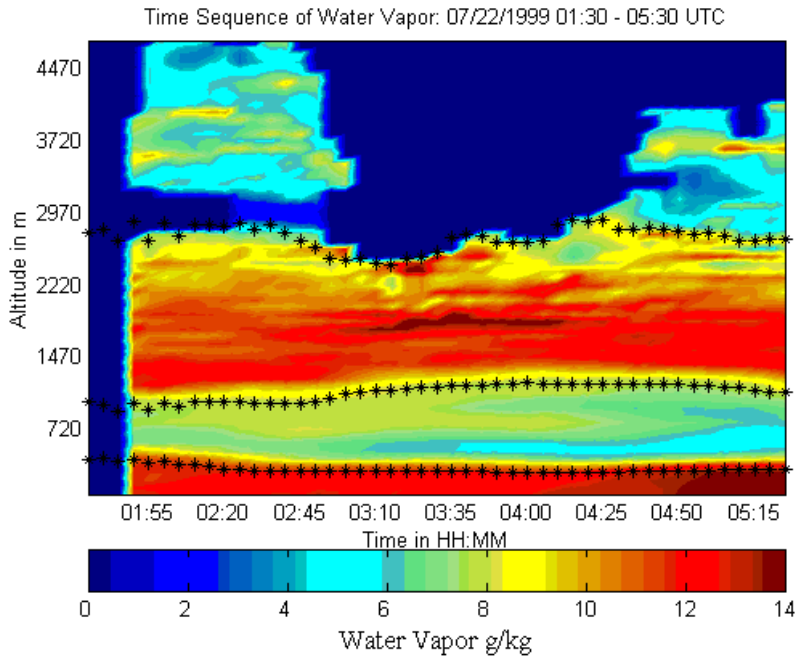


Figure 4.13. ABL depth estimated from time sequence of water vapor for July 22 1999, 05:00 – 08:00 UTC.

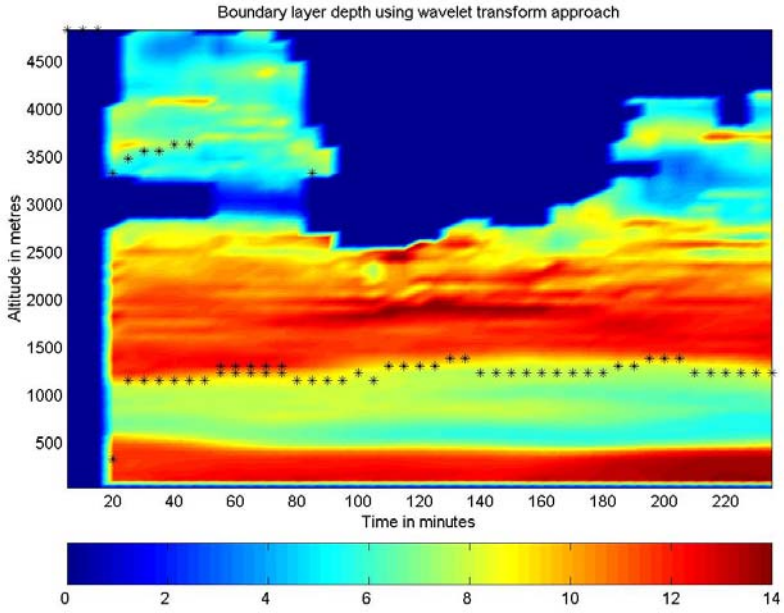


Figure 4.14. ABL depth estimated from time sequence of water vapor July 22 1999, 05:00 – 08:00 UTC using wavelet approach.

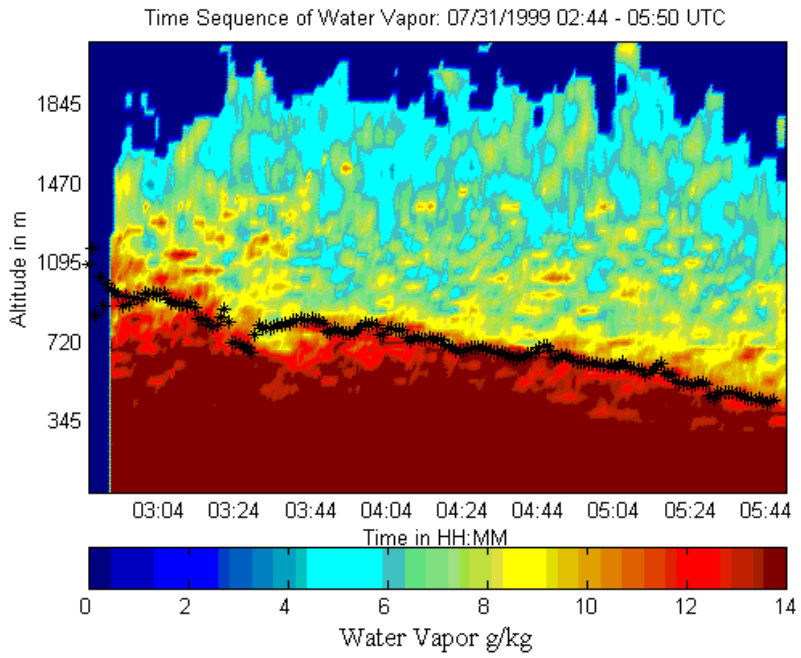


Figure 4.15. ABL depth estimated from time sequence of water vapor for July 31 1999, 02:44 – 05:50 UTC.

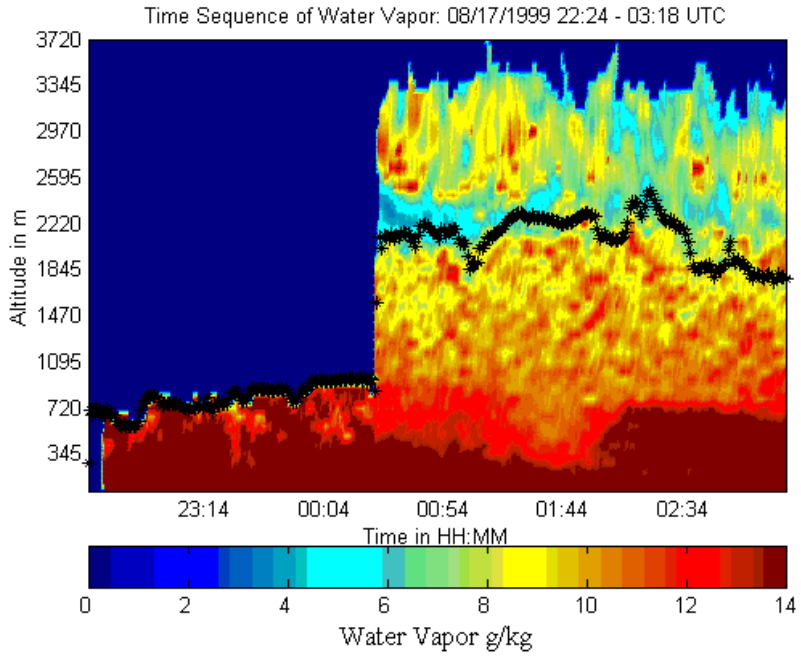


Figure 4.16. ABL depth estimated from time sequence of water vapor for August 17 1999, 22:24 – 03:18 UTC.

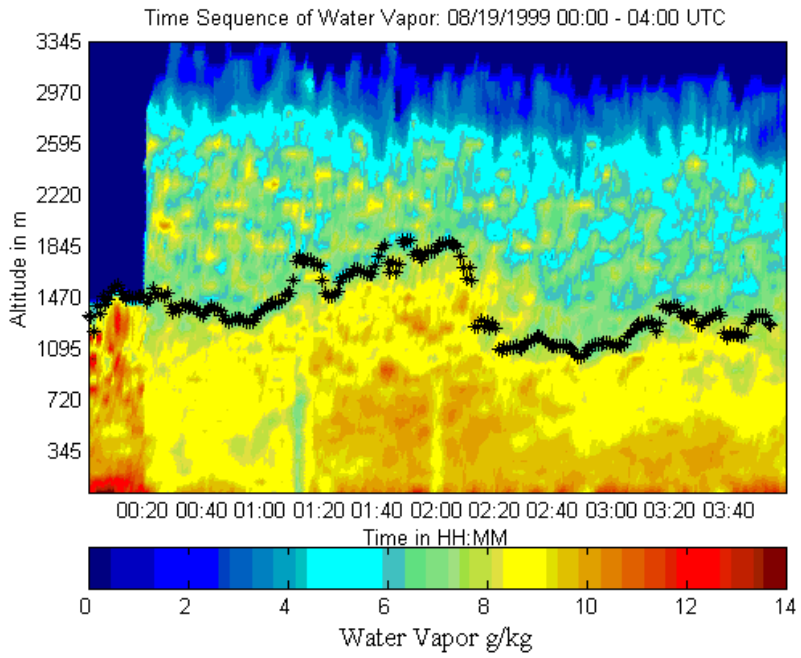


Figure 4.17. ABL depth estimated from time sequence of water vapor for August 19 1999, 00:00 – 04:00 UTC.

Chapter 5

Conclusions and Future Work

In this thesis, we demonstrated the utility of Raman lidar to measure the depth of the Atmospheric Boundary Layer (ABL) by making use of its ability to measure the water vapor profiles in the atmosphere which are a tracer of the dynamics of the ABL. The wavelet technique used to estimate the ABL depth was implemented and results from LAPS Raman lidar were shown. The wavelet technique operates on each profile (variation of a scalar parameter with height), and in order to obtain a time series of the ABL depth, the algorithm has to be run on profiles at successive time steps. The algorithm was shown to estimate gradients in the water vapor profile which corresponded to the boundary of layers in the atmosphere. It was shown that the main drawback of this algorithm was the absence of continuity of the different transition levels estimated from the algorithm. Thus, in the presence of multiple atmospheric layers, the algorithm is incapable of giving an estimate of the altitude of the various atmospheric layers. A constant wavelet basis was used for each profile. This assumed that all the transitions in all the profiles would obey the shape of the wavelet basis. This assumption prevented the detection of transitions which had a gradient different from that of the basis function. Hence, it is not good for a general case without some user interference in the choice of the wavelet basis function.

The image processing technique operates on the entire two dimensional time-altitude matrix of the scalar parameter (say, water vapor). The algorithm treated the problem of gradient detection and tracing of various layers in the atmosphere as equivalent to the approach of image edge detection, followed by edge following. By

choosing an appropriate variance, the edge detection was found to detect all the required gradients. Thus, the 2-D spatial filter exploits the correlated feature of the various layers in time and space in the ABL. The additional information of the adjacent profiles gives the image processing algorithm the advantage over the wavelet algorithm which operates on a single profile only. The testing of variety of edge detection operators ensures finding best approach to define that the various layers in the atmosphere. A unique feature of this algorithm is that it uses Bellman's optimality criterion for tracing the various layers. This edge following algorithm can also be used by the wavelet algorithm for linking the various transition points detected by that algorithm.

Future Work

The edge following algorithm used in this thesis considered only the next neighbor, although consideration of the p^{th} neighbor was also alluded. Future efforts could improve the analysis by using the next neighbor or the p^{th} neighbor technique. Median filtering was used to retain edge features while removing noisy pixels from the edge detected image prior to edge following. There exists in the published literature a plethora of non-linear and rank ordered filters whose applicability to the edge enhancement in lidar water vapor time sequence has not been studied here. Future work can come up with an objective tool to determine the choice of pre and post processing techniques for accurate ABL depth estimation.

This new tool to trace the various layers has the potential for improving the accuracy of numerical weather modeling, and for the incorporation of Raman lidar as a widespread tool in the meteorological community. Numerical Weather Prediction (NWP)

models, such as MM5 and RAMS can utilize the ABL depth information from the output of the algorithm for assimilation and improving their parameters, so that the weather forecast from these models is more accurate. Combined with other measurements, a Wiener filter can be used for the adaptive variation of the model parameters to minimize the error in model output. The image processing technique will be especially useful to the air quality community to obtain measurement of the thickness of the mixing layer and for tracing the transport of air pollutants.

References

[1] Garratt, J.R., *The Atmospheric Boundary Layer*, Cambridge Atmospheric and Space Science Series, 1992.

[2] Nieuwstadt, F.T.M., and P.G.Duynkerke, "Turbulence in the Atmospheric Boundary Layer," Atmospheric Research, Vol. 40, pp. 111-42, 1996.

[3] Wyngaard, J.C., N. Seaman, S.J. Kimmel, M. Otte, X. Di and K.E. Gilbert, "Concepts, Observations, and Simulations of Refractive Index Turbulence in the Lower Atmosphere," Radio Science," Vol. 36, No. 4, pp 643-669, 2000.

[4] Halliday, D., R. Resnick, and J. Walker, *Fundamentals of Physics*, Wiley, 6th edition, 2001.

[5] Prandtl, L., *Essentials of Fluid Dynamics : With Applications to Hydraulics, Aeronautics, Meterology, and Other Subjects. Authorized Translation,* Hafner Publications, 1952, From the original, "Ueber Flussigheitsbewegung Bei Sehr Kleiner Reibung." Verhandlung des III. Internationalen Mathematiker Kongresses, Heidelberg, 1904 (Leipzig, 1905).

[6] Stull, R.B., *An Introduction to Boundary Layer Meteorology*, Kluwer Academic Publishers, 1997.

[7] Azad, Ram.S., *The Atmospheric Boundary Layer for Engineers*, Kluwer Academic Publishers, 1993.

[8] Stephens, G. K., G. G. Campbell, and T.H.Vonder Haar, "Earth Radiation Budgets," J.Geophysics Research, Vol. 86, pp. 9739 – 9760, 1981.

[9] London, J., and T. Sasimori, "Radiative Energy Budget of The Atmosphere," Space Res.11, pp. 639-649, 1957.

[10] Gill, A.E., *Atmospheric-Ocean Dynamics*, Academic Press, Inc., 1982.

[11] List, R.J., *Smithsonian Meteorological Tables*, 6th ed, 1951.

[12] Philbrick, C.R., and K. Mulik, "Application of Raman Lidar to Air Quality Measurements," Proceedings of the SPIE Conference on Laser Radar Technology and Applications V – Vol. 4035, pp. 22-33, 2000.

[13] Slick, C.T., *Lower Tropospheric Temperature Measurement Scheme for an Advanced Lidar Atmospheric Profiling System*, MS Thesis, Pennsylvania State University, May 2002.

[14] Philbrick, C.R, and Daniel B. Lysak, Jr, "Lidar Measurements of Meteorological Properties and Profiles of RF Refractivity," Proceedings of the 1996 Battlespace Atmospherics Conference, pp. 595-609, 1996.

[15] Esposito, C.T., *Applications and Analysis of Raman Lidar Techniques for Measurements of Ozone and Water Vapor in the Troposphere*, MS Thesis, Pennsylvania State University, 2000.

[16] Measures, R.M., *Laser Remote Sensing: Fundamentals and Applications*, John Wiley and Sons, 1990.

[17] Grell, G.A., J. Dudhia and D.R. Stauffer, *A Description of the Fifth Generation Penn State/NCAR Mesoscale model (MM5)*, National Center for Atmospheric Research, Boulder, Colorado, 1994.

[18] Davis, K.J., "An Objective Method for Deriving Atmospheric Structure from Airborne Lidar Observations," American Meteorological Society, pp. 1455 – 1466, 2000.

[19] Angevine, W.M., "Entrainment Results with Advection and Case Studies from the Flatland Boundary Layer Experiments," J.Geophysical Research, Vol. 104, pp. 30947 – 30963, 1999.

[20] Angevine, W.M., A.W. Grimsdell, J.M. Warnock, W.C. Clark, and A.C. Delany, "The Flatland Boundary Layer Experiments," Bulletin of the American Meteorological Society, Vol. 34, pp. 357-375, 1998.

[21] Angevine, W.M., A.B. White, and S.K. Avery, "Boundary Layer Depth and Entrainment Zone Characterization with a Boundary Layer Profiler," Boundary Layer Meteorology, Vol. 68, pp. 375-385, 1994.

[22] Piironen, A.K., and E.W. Eloranta, "Convective Boundary Layer Mean Depths and Cloud Geometrical Properties Obtained from Volume Imaging Lidar Data," J.Geophysical Research., Vol. 100, pp. 25 569 – 25 576, 1995.

[23] Eloranta, E.W., and E.K. Forest, "Volume-Imaging Lidar Observations of The Convective Structure Surrounding the Flight Path of a Flux-Measuring Aircraft," J. Geophysical Research., Vol. 97, pp. 18 383 – 18 393, 1992.

[24] Cohn, S.A., and W.M. Angevine, "Boundary Layer Height and Entrainment Zone Thickness Measured by Lidars and Wind Profiling Radars," Journal of Applied Meteorology, Vol. 39, N0. 8, pp. 1233-1247, 2000.

[25] White, A.B., C.W. Fairall, D.W. Thomson, "Radar Observations of Humidity Variability in and Above the Marine Atmospheric Boundary Layer," J. Atmospheric and Oceanic Technology, Vol. 8 (5), pp. 639-658, 1991.

[26] Driedonks, A.G.M., and H. Tennekes, "Entrainment Effects in the Well Mixed Atmospheric Boundary Layer," Boundary Layer Meteorology, Vol. 30, pp. 75 – 105, 1984.

[27] Boers, R., and R.L. Coulter, "Lidar Observations of Mixed Layer Dynamics: Tests of Parameterized Entrainment Models of Mixed Layer Growth Rate, " Journal of Climate and Applied Meteorology, Vol. 23, pp. 247-266, 1984.

[28] Crum, T.D., and R.B. Stull, "Field Measurements of the Amount of Surface Layer Air Versus Height in the Entrainment Zone," J. Atmospheric. Science, Vol. 44, 2743 – 2753, 1987.

[29] Mallat, S., *A Wavelet Tour of Signal Processing,* Academic Press, 1998.

[30] Gamage, N., and C. Hagelberg, "Detection and Analysis of Microfronts and Associated Coherent Events Using Localized Transforms," J. Atmospheric Science, Vol. 50. pp. 750 – 756, 1993.

[31] Pitas. I, *Digital Image Processing Algorithms and Applications*, Wiley Interscience, 2000.

[32] Gonzalez, R.C., and R.E. Woods, *Digital Image Processing*, Addison Wesley, 1992.

[33] D. Marr, and E. Hildreth, "*Theory of edge detection*," Proc. of Royal Soc. London B. 207, pp.187-217, 1980

[34] Kaufmann. A., *Graphs, Dynamic Programming and Finite Games*, Academic Press, 1967.

[35] Bellman.R., *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1957.

[36] E. Minieka., *Optimization Algorithms for Networks and Graphs*, Marcel Drekker, New York, 1978.

Appendix A

Relationship between Water Vapor Mixing Ratio and Specific Humidity

The relations between specific humidity and water vapor mixing ratio is given below.

Vapor Pressure is defined as the pressure exerted by water vapor molecules in the atmosphere is denoted as e. The common atmospheric pressure p is the sum of air pressure and vapor pressure.

Mixing Ratio (w) is defined as the ratio of the mass of the water vapor to the mass of the dry air,

$$w = \frac{\text{Mass of Water Vapor}}{\text{Mass of Dry Air}} . \qquad [A2.1]$$

As mixing ratio is a ratio of 2 quantities with the same dimension , it is a dimensionless quantity.But, as the mass of water vapor in the atmosphere is far less than the mass of dry air, mixing ratio is commonly expressed in gm/Kg. In terms of vapor pressure, w can be expressed as,

$$w = \varepsilon \frac{e}{p - e} . \qquad [A2.2]$$

where ε = Ratio of the molecular weight of water vapor to dry air.
≈ 0.622

The specific humidity is defined as the ratio of the mass of the water vapor to the mass of the total air,

$$q = \frac{\text{Mass of Water Vapor}}{\text{Mass of Dry Air} + \text{Mass of Water Vapor}} . \qquad [A2.3]$$

$$= \frac{w}{1 + w} . \qquad [A2.4]$$

Appendix B

MATLAB Programs

```
**********************************************************************
%Waveletmanylayers.m
%Calculates Boundary layer height by Wavelet technique.
%Preprocessing by median filter to enhance gradients
%and Postprocessing by mean filter to remove spikes.
%Uses haar wavelet .Refer paper by K.J.Davis, "An Objective Method for
%Deriving Atmospheric Structure from airborne Lidar Observations",
%Journal of Atmospheric and Oceanic Technology, Nov.2000., AMS
%Does both global and 1st local maxima calculation compared to blayerheight3.m

clear all
close all
load variables19997221;
[row,col]=size(validmixingratio);
vmr=validmixingratio;


    for a= 1:1:floor(row/2)
          %basic filter definition.No zero padding - haar wavelet
          haar=[1*ones(1,a) -1*ones(1,a)];
          for b=max(a,4):row - a
           %Filter with length adjusted to translation 'b' such that zeros have been padded
           %before and after the translation up to a length equal to the max altitude or the
           %variable row, i.e the length of the filter is = row.

            haarfilter=[zeros(1,b-a) haar zeros(1,row - b - a)];

            %Create a matrix of the filter such that each col contains the basic haarfilter

            haarfiltermatrix=haarfilter(:)*ones(1,col);

            %Multiply the haar wavelet  for various times which is in the variable
            %haarfiltermatrix with the validmixingratio time sequence profile .
             % We will get a matrix of the size of the validmixingratio.
            %Integrate along the heights (rows) to get xd which is

            xd(b,:)=(1/a)*trapz(haarfiltermatrix.*validmixingratio,1);

      end %End computation in b - translation axis
         %Compute the covariance transform
         %For every value of dilation 'a' , integrate the square of wavelet filtered time
         % sequence profile , xd with translation b as the independent variable over all
```

```
          %altitudes.This will yield a covariance transform wchich depends on
          %scale 'a' for various times.hence dimension of d2 is (a,col);

          d2(a,:)=trapz(xd.^2,1);

matrix1=sign(diff(xd));
%Create a matrix which has 1 if it is higher than the earlier element in the column
%i.e , matrix1(i,j) =1 iff matrix(i,j) > matrix1(i-1,j)
%       matrix1(i,j) =0 iff matrix(i,j) = matrix1(i-1,j)
%       matrix1(i,j) =-1 iff matrix(i,j) < matrix1(i-1,j)

matrix1=[zeros(1,size(matrix1,2));matrix1];%Pad the matrix with zeros in the first row.
matrix2=[flipud(sign(diff(flipud(xd))));zeros(1,size(matrix1,2))];
%A matrix same as matrix1 with differences being
%between row1 - row2 instead of row2 - row1

 matrix3=matrix1 + matrix2;
%This matrix will have 2 wherever there is a local maxima and lesser values at other
%points.

matrix3(find(matrix3 ~= 2)) = 0;
%Replace all elements which are not 2 , i.e those that do not represent local
%maxima with 0.

matrix3=matrix3.*xd;
%Multiply matrix with xd so that the  value of the wavelet transform at local maxima can
%be known for future computation to know at what value of translation b is the max of
%the tranform achieved.

[matrix3 allindexmax]=sort(matrix3,1);
maxrow=size(allindexmax,1);%Get the last row index
indexmax(a,:) = allindexmax(maxrow,:);
%This gives the postition of the highest local maxima only

end %End computing in a - dilation axis.

%Compute the Global Max and 2nd local maxima of d2
      [val,amaxindex]=max(d2,[],1);
      amaxindex2 = computelocalmaxima(d2,1,2);


      %Now compute the local max of d2
      for i=1:col
         blayerheight(i)=indexmax(amaxindex(i),i);
         blayerheight2(i) = indexmax(amaxindex2(i),i);
      end
```

```
   %Spike suppression/smoothing
   %This is for spike smoothing using mean filter
   for i=5:col -3
      if (blayerheight(i) > blayerheight(i-1) + 8) |(blayerheight(i) > blayerheight(i+1) + 8)
         blayerheight(i) = mean(blayerheight(i-3) : blayerheight(i+3))/7;
      end
   end

altitude=[37.5:75:(row-1)*75 + 37.5];
time=[1:col]*5;
figure(3),pcolor(time,altitude,validmixingratio),shading
interp,h1=get(gca,'CLim'),set(gca,'CLim',[0 14]),colorbar('horiz')
%figure(1),pcolor(validmixingratio),shading interp,h1=get(gca,'CLim'),set(gca,'CLim',[0
14]),colorbar('horiz')
blayerheight = 75*[blayerheight + 1*ones(1,col)] -37.5*ones(1,col);
blayerheight2 = 75*[blayerheight2 + 1*ones(1,col)] -37.5*ones(1,col);
hold on, plot(time,blayerheight,'b*')
hold on,plot(time,blayerheight2,'b*')
xlabel('Time in minutes')
ylabel('Altitude in metres')
title('Boundary layer depth using wavelet transform approach')


**************************************************************************

**************************************************************************

%blayerheight5.m
%Detects Boundary layer gradients by Edge Detection techniques.
%Output of the program is stored as variable file to be used by bellmanmanylayers2.m
%Operates on the matrix instead of image obtained.
%refer Digital Image Processing by Gonzales and Woods
%Preprocessing by Median Filter to enhance edges.

clear all
close all
load variables98;
 [row,col]=size(validmixingratio);

%Edge Detection of Entire image using matlab edge function use LoG mask

%Simple edge detection using contourf function
contour(validmixingratio); %colormap('gray');
set(gca,'CLim',[0 14]);
set(gca,'Visible','off');
```

```
print -dtiff test;
i=imread('test.tif');
ig=rgb2gray(i);
e=edge(ig,'log',0,0.5);
imshow(e);
[row,col]=find(e > 0);
ee=e(min(row)+1:max(row)-1,min(col)+1:max(col)-1);
%Remove the background which has no points
%remove edges which are due to clipping due to altitude or signal loss
[row,col]=find(ee > 0);
ee(1:min(row)+5,:)=0;
%count= min(row) + 5;
%save edgevariables ee validmixingratio ig  plotData plotDataStddev plotStartTime
plotEndTime;
save edgevariables ee validmixingratio ig  plotData plotDataStddev ;


 [r,c]=size(ee);
altitude=[37.5:75:(r-1)*75 + 37.5];
time=[1:c]*5;
set(gca,'Visible','on');

**************************************************************************



**************************************************************************

%Bellmanmanylayers2.m
%Implements Bellman's Optimality Principle for edge following.
%Assumes 5 minute steps.
%Accepts variables from output of blayerheight5.m

clear all;close all;
load('edgevariables');
ee=im2double(flipud(ee));
ee=ee(15:size(ee,1),:);

count = 0;
D=inf*ones(size(ee));
D(:,1)=ee(:,1);%matrix of distance
I=zeros(size(ee));%matrix of indices of previous column
flag = 0;
num_layers=40;

for i=2:size(ee,2)
```

```
    index_present=find(ee(:,i) ~= 0);
    if isempty(index_present)
       continue
    end
    flag=0;
    index_previous=find(ee(:,i-1) ~= 0);
    if isempty(index_previous)
       flag=1;
    end

    for j=1:length(index_present)
       if flag == 0
      [val,index]=min(abs(index_present(j)*ones(size(index_previous)) - index_previous));
      I(index_present(j),i)=index_previous(index);
      D(index_present(j),i)=D(index_previous(index),i-1) + sqrt(1 + (index_present(j) -
index_previous(index)).^2);

       else
          D(index_present(j),i)=1;
       end

    end
end

[dummydistance,blayerindices] = sort(D(:,size(ee,2)));
for times=1:num_layers
    if I(blayerindices(times),size(ee,2)) == 0
       break
    end
    blayer(size(ee,2)) = I(blayerindices(times),size(ee,2));
    for i=size(ee,2)-1:-1:2
          blayer(i) = I(max(blayer(i+1),1),i);
    end
    yaxis=[1:size(ee,1)]*75 - 37.5*ones(1,size(ee,1)) + (count + 10)*ones(1,size(ee,1));
    num_mins = size(ee,2)*5;
    xaxis=[1:num_mins/size(ee,2):num_mins];
    allblayers(times,:)=blayer;
    clear blayer;
 end %End of times loop

 %Now convert height in pixel to height in altitude indices of validmixingratio
[matrixrows,matrixcols] = size(validmixingratio);
[imagerows,imagecols] = size(ee);
altituderatio = matrixrows/imagerows;
allblayers=allblayers*altituderatio;
timeratio = imagecols/matrixcols;
```

```
inttimeratio = round(timeratio*1000);
for i = 1: size(allblayers,1)
   newallblayers(i,:) = resample(allblayers(i,:),1000,inttimeratio);

end
ground_altitude = 10;
altitude = [1:size(validmixingratio,1)]*37.5 +
ground_altitude*ones(1,size(validmixingratio,1));

%Now start plotting
minStep=datenum(2000,1,1,1,2,0)-datenum(2000,1,1,1,1,0);   % the actual number for
one minute is best found this way
step = 1 %Assumes 5 minute steps
datestr(plotStartTime, 15) ' - ' datestr(plotEndTime, 15) ' UTC'];
xaxis = [1:size(validmixingratio,2)]*5;
%Y axis labeling
yaxiscoefficient = [1:size(validmixingratio,1)]*75 -
29.5*ones(1,size(validmixingratio,1));
yaxis = yaxis(2:length(yaxis));
maxAlt = 5000;
remainder = mod(maxAlt,1000);
loweredAlt = maxAlt - remainder;
htindexmultiple = (1000/maxAlt)*size(validmixingratio,1) ;

pcolor(validmixingratio(2:size(validmixingratio,1),:)),set(gca,'CLim',[0 14]),shading
interp
hold on, plot(newallblayers(1,:),'k*')
%pcolor(validmixingratio),set(gca,'CLim',[0 14]),shading interp,hold on,
plot(newallblayers(2,:),'k*')
hold on, plot(newallblayers(2,:),'k*')
hold on, plot(newallblayers(3,:),'k*')
%hold on, plot(newallblayers(15,:),'k*')
h=get(gca,'XTickLabel');
set(gca,'XTickLabel',datestr( minStep*step.*str2num(h)+plotStartTime,15));
h=get(gca,'YTicklabel');
set(gca,'YTickLabel',num2str(str2num(h)*75 - 30*ones(size(h,1),1)));%75 - (75/2 + 8 =
45.5) = 30
name=['Time Sequence of Water Vapor: ' datestr(plotStartTime, 23) ' '
datestr(plotStartTime, 15) ' - ' datestr(plotEndTime, 15) ' UTC'];
title(name)
xlabel('Time in HH:MM')
ylabel('Altitude in m')
barhan=colorbar('horiz');
axes(barhan);
h=xlabel('Water Vapor g/kg');
set(h,'FontName','Times New Roman','FontSize',12)
```

```
**********************************************************************


**********************************************************************
%-------------------------------------------------------
%  Program:       Reads and outputs WV Data
%  Programmed By: Steve Esposito
%
%                       Program changed on 09\13\98 - Steven Esposito
%                     Changes for clipping the data at the top of the plot.
%                  This file reads the data from the files from the LAPS_DSP
%                   program and outputs three arrays containing the altitude,
%             signal and error.  Use program in conjunction with tsplotDataclip.m
%
%
%                       changed 09/18/98 to pick off name from file list
%
%        Changed 10/6/98 to accomodate for gui inputs - run plotDataer.m first for
%   graphic user interface
%
%
%           A. A. 5/25/2000 Changed to make the program form the file list in the
%                                                  system temp directory for reading from read-only dirs.
%  A. A. 7/3/2000 Modified to add footer of information to plots
%  A. A. 7/20/2000 Added hanning filter to smooth data if it goes below the
%                                                                        cutoff value.
%
%  Sriram. N. Kizhakkemadam 05 May 2002. Changed it to store variables required
%             to store variables required for blayerheight5
%  Corey Slick  03 Mar 2001 Changed to make compatible with new gui (mainproc.m)
%             which lets the user process temperature, ozone and water vapor
%             from the same interface.
%   Alex Achey  7/18/2001 Changed to use new processed data reading function
%             Also introduced a new plotting method in the code which plots
%             only valid points and will not force plotting of low altitudes
%             or skip valid high altitudes which have a high error point before them.
%   AA 7/24/01  Modified to do automatic saving of plots as specified in the GUI
%-------------------------------------------------------


function tsplotData_new

  %  Time sequence UV plotDataer vapor data program accepting files from Laps_Dsp
                                        %  program.
```

```
clear all;
set(0,'ShowHiddenHandles','on')

% Getting the inputs from the Graphical user interface screen

figureH = findobj('Tag','mainproc');

HH = findobj(figureH,'Tag','maxWatContVal');
maxval = str2num(get(HH,'String'));
HH = findobj(figureH,'Tag','minWatContVal');
minval = str2num(get(HH,'String'));
HH = findobj(figureH,'Tag','maxAlt');
maxalt = str2num(get(HH,'String'));
HH = findobj(figureH,'Tag','IntTime');
int = str2num(get(HH,'String'));
HH = findobj(figureH,'Tag','StepTime');
step = str2num(get(HH,'String'));
 HH = findobj(figureH,'Tag','TimeZone');
 TZ = (get(HH,'value'));

HH = findobj(figureH,'Tag','WatSigClip');
sig_clip = str2num(get(HH,'String'));

HH = findobj(figureH,'Tag','WatAlaska');
unit = (get(HH,'value'));

HH = findobj(figureH,'Tag','WatSmooth');
smooth = (get(HH,'value'));

HH = findobj(figureH,'Tag','AutoSavePlots');
autosaveplots = (get(HH,'value'));
HH = findobj(figureH,'Tag','PlotSavePath');
PlotSavePath = (get(HH,'String'));

HH = findobj(figureH,'Tag','edit_start_yr');
start_year = str2num((get(HH,'String')));
HH = findobj(figureH,'Tag','edit_start_mo');
start_month  = str2num((get(HH,'String')));
HH = findobj(figureH,'Tag','edit_start_day');
start_day  = str2num((get(HH,'String')));
HH = findobj(figureH,'Tag','edit_start_hr');
start_hour  = str2num((get(HH,'String')));
HH = findobj(figureH,'Tag','edit_start_min');
start_min  = str2num((get(HH,'String')));
HH = findobj(figureH,'Tag','edit_plot_length_hrs');
plot_time_len = str2num((get(HH,'String')));
```

```
HH = findobj(figureH,'Tag','edit_base_path');
data_files_path = (get(HH,'String'));


% maximum width for the hanning filter if we are filtering the data
maxFiltWidth=5;

switch TZ
   case 1
     TZ=['UTC'];
   case 2
     TZ=['AST'];
   case 3
     TZ=['PDT'];
   case 4
     TZ=['PST'];
   case 5
     TZ=['EDT'];
   case 6
     TZ=['EST'];
   end


   %%%%
   % establish the starting and ending times of the plot and of the data files needed.  They
are different because
   % the proc file named for, say, 6:29 is the integration from 6:00 to 6:29 at 30 min
integration.
   % so, the 'filesStartTime' is the ending time of the integration period in the first file
   plotStartTime=datenum(start_year, start_month, start_day, start_hour, start_min, 0);
   filesStartTime=datenum(start_year, start_month, start_day, start_hour, start_min+int,
0);
   plotEndTime=plotStartTime+datenum(0,0,0,plot_time_len,0,0);
   filesEndTime=filesStartTime+datenum(0,0,0,plot_time_len,0,0);

   %%%%
   % Read in the files
   disp(['looking from ' datestr(filesStartTime,0) ' to ' datestr(filesEndTime,0) ' in '
data_files_path]);

   [procDataSeriesArray,
success]=GetProcDataSeries(data_files_path,filesStartTime,filesEndTime);
   size(procDataSeriesArray)
   if(success == 0)
      disp('no files found to plot!');
```

```
      break;
   end;

   %%%%
   % from the array of minutes structures, create an array of plotting data.  Include
"blank" times where needed
   % by skipping places where there's no data
   maxAltBinIdx=max( find(procDataSeriesArray(1).Altitude(:) <= maxalt ) );    %what's
the highest bin to read?
   minStep=datenum(2000,1,1,1,2,0)-datenum(2000,1,1,1,1,0);   % the actual number for
one minute is best found this way
   numPlotCols=floor( (filesEndTime-filesStartTime) /(step * minStep) )+1;  % how
many collumns should there be total
   %init Alt to alt of any processed data period...they're all the same
   for (place = 1 : numPlotCols )
      Alt(:,place)=procDataSeriesArray(1).Altitude(1:maxAltBinIdx)';
   end;
   %init other arrays to 0's for blank times
   plotData=zeros(maxAltBinIdx,numPlotCols);
   plotDataStddev=zeros(maxAltBinIdx,numPlotCols);

   %place the periods of data into the arrays as appropriate
   for (currMin = 1 :size(procDataSeriesArray,2) )
     place=floor( (procDataSeriesArray(currMin).dateNumValueEnd-filesStartTime) /
(step * minStep) )+1; % +1 since 1 based arrays
     Alt(:,place)=procDataSeriesArray(currMin).Altitude(1:maxAltBinIdx)';
     plotData(:,place)=procDataSeriesArray(currMin).CombWV(1:maxAltBinIdx)';

plotDataStddev(:,place)=procDataSeriesArray(currMin).CombWVSdev(1:maxAltBinIdx
)';
   end;
   %%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                              %smoothing using a hanning filter
                                    if smooth==1;
     disp('doing smoothing');
                                          clear a;
     a=size(Alt);
     % preset this here so the top and bottom that the loop excludes due to filter width
will get filled
     newplotData=plotData;
     newplotDataSigma=plotDataStddev;
     filtLevels=zeros(a(2));
```

```
                                                                              for minute=1:(a(2))
       for altBin=ceil(maxFiltWidth/2):(a(1)-1)  % start above filter width
         % if the error is above our clip value, keep trying larger filters until we hit our
limit
         % or we're too close to the top of the image
         filtWidth=3;

         while( ( plotDataStddev(altBin,minute) > sig_clip*plotData(altBin,minute) ) & ...
                               (maxFiltWidth >= filtWidth) & ...
           (altBin <= a(1)-floor(filtWidth/2) ) )

                 %set the filter alt level for this minute if it's not already set
                                   if(filtLevels(minute) == 0)
                                                       filtLevels(minute) = Alt(altBin,minute);
                                               end;

          filt=hanning(filtWidth)/sum(hanning(filtWidth));  %normalized hanning filter
numbers
                                       newWVLevel = 0;
           newWVSigma = 0;
           for( i=1:filtWidth)
             newWVLevel = newWVLevel + (plotData(altBin-ceil(filtWidth/2)+i,minute)
* filt(i)) ;
             newWVSigma = newWVSigma + ( filt(i) * plotDataStddev(altBin-
ceil(filtWidth/2)+i,minute) ) ^ 2; % from propogation of errors
                                               end;
           newWVSigma = newWVSigma ^ 0.5;              % from propogation of errors

           filtWidth=filtWidth+2;

           % update the new filtered data set
           newplotData(altBin,minute) = newWVLevel;
           newplotDataSigma(altBin,minute) = newWVSigma;
         end; % end of while loop
       end % end of for altBin
     end   %end of for minute

     % copy the filtered data into the working set
     plotData=newplotData;
     plotDataStddev=newplotDataSigma;
                                                end
   % done with smoothing

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
   hPlotFigure = figure;
   %  create an overall axes to position text in, then the plot axes
   hTextAxes=axes('Position',[0 0 1 1],'Visible','off');
   axes('Position',[.1 .15 .82 .74]);
   a=size(Alt);

   % select which plotting routine
                                                if(1==0)
      for (j=1:size(plotData,2) - 1)
         %%%%
         % is there valid info for this column?
         if (    ~isempty( find(plotData(:,j) ~=-99 & plotDataStddev(:,j) ~= 0 ) ) |...
               ~isempty( find(plotData(:,j) ~=  0 & plotDataStddev(:,j) ~= 0 ) )  )
            for i=6:maxAltBinIdx
               %%%%
               % if we hit the top of valid data, then plot this column.  We have to stop if
               % the next minute is 0 err at this altitude because that's the right hand side
               % of our 'square'
               if ( plotDataStddev(i,j)>(sig_clip*plotData(i,j)) | plotDataStddev(i,j)==0 |
plotDataStddev(i,j+1)==0 | i==maxAltBinIdx);
                  % only plot if the next minute has any valid info in it( i.e. non-missing)
otherwise
                  % we will get a 'rampdown' to the 0's of the next (missing) minute due to
interpolation
                  if(plotDataStddev(1,j+1)~=0 )
                     pcolor( [j j+1],[Alt(1:(i-1),j) Alt(1:(i-1),j)], [plotData(1:(i-1),j)
plotData(1:(i-1),j+1)]);
                     hold on;
                  end;
                  break;  %exit the loop for this column(minute)
               end;
            end;
         end;
      end;

                                                else
      for (j=1:size(plotData,2) - 1)
         for i=2:maxAltBinIdx-1
            if(   1==validDataPt(plotData(i,j),   plotDataStddev(i,j),   sig_clip*plotData(i,j))
& ...
                  1==validDataPt(plotData(i,j+1), plotDataStddev(i,j+1),
sig_clip*plotData(i,j+1))& ...
```

```
                1==validDataPt(plotData(i+1,j), plotDataStddev(i+1,j),
sig_clip*plotData(i+1,j))& ...
                1==validDataPt(plotData(i+1,j+1), plotDataStddev(i+1,j+1),
sig_clip*plotData(i+1,j+1)) )
              pcolor( ([j j+1]),[Alt(i-1:i,j) Alt(i-1:i,j+1)], [plotData(i-1:i,j) plotData(i-
1:i,j+1)]);
              validmixingratio(i,j)=plotData(i,j);
               %disp('hi')
              hold on;
            end;
          end;
       end;
                                          end;


filename=strcat(num2str(start_year),num2str(start_month),num2str(start_day),num2str(st
art_hour));
  filename = strcat('K:\sriramthesis\sriram\variables',filename,'.mat');

save(filename,'plotData','plotDataStddev','validmixingratio','plotStartTime','plotEndTime'
,'maxalt');
  cd('i:\sriram\homermatlabfiles\matlabfiles')
 %   save 'K:\sriramthesis\sriram\variables.mat' plotData plotDataStddev
validmixingratio;
   axis([1 a(2) Alt(1,1) Alt(a(1),1)])

   shading interp;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
   %set the contour limits
   h1=get(gca,'CLim');
   if( minval == -99)
     minval = h1(1);
   end;
   if( maxval == -99)
     maxval = h1(2);
   end;
   set(gca,'CLim',[minval  maxval]);

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


   barhan=colorbar('horiz');
```

```
    axhan=gca;

    name=['Time Sequence of Water Vapor: ' datestr(plotStartTime, 23) ' '
datestr(plotStartTime, 15) ' - ' datestr(plotEndTime, 15) ' UTC'];
    h=title(name);

    set(h,'FontName','Times New Roman','FontSize',14)
    h=xlabel('Time');
    set(h,'FontName','Times New Roman','FontSize',10)
    h=ylabel('Altitude (m)');
    set(h,'FontName','Times New Roman','FontSize',10)
    axes(barhan);
    h=xlabel('Water Vapor g/kg');
    set(h,'FontName','Times New Roman','FontSize',12)
    axes(axhan);

    h=get(gca,'XTickLabel');
    set(gca,'XTickLabel',datestr( minStep*step.*str2num(h)+plotStartTime,15 ));

    %%%%%
    % set up footer of paramaters
    set(hPlotFigure,'CurrentAxes',hTextAxes);

    plotParamString=['signal cliped at error: ' 177 num2str(sig_clip*100) '%'];
    h=text(0.05 ,0.03,plotParamString);
    set(h,'FontName','Times New Roman','FontSize',8)

    plotParamString=['plotted on: ' dateStr(now,2)];
    h=text(0.05 ,0.06,plotParamString);
    set(h,'FontName','Times New Roman','FontSize',8)

    if( mod(int,2) == 1)
       plotParamString=['integration period: ' num2str(int*2) 'm'];
    else
        plotParamString=['integration period: ' num2str(int*2+1) 'm'];
    end;
    h=text(0.35 ,0.03,plotParamString);
    set(h,'FontName','Times New Roman','FontSize',8)

    plotParamString=['step size: ' num2str(step) 'm'];
    h=text(0.35 ,0.06,plotParamString);
    set(h,'FontName','Times New Roman','FontSize',8)

          if smooth==1;
               plotParamString=['using hanning filter, max width: ' num2str(maxFiltWidth)]
               h=text(0.65 ,0.02,plotParamString);
```

```
            set(h,'fontname','timesnewroman','fontsize',8)
        end;

    % done with footer
    %%%%%%%%

    % save plot if we're supposed to
    if(autosaveplots)
        [ySt moSt daSt hrSt minSt secSt]=datevec(plotStartTime);
        [yEnd moEnd daEnd hrEnd minEnd secEnd]=datevec(plotEndTime);
        plotfilename =
sprintf('%s\\%02d%02d%02d%02d%02d_%02d%02d_wat.tif',PlotSavePath,mod(ySt,10
0), moSt, daSt, hrSt, minSt, hrEnd, minEnd);
        eval(['print -dtiff ' plotfilename])
    end;

    setptr(1,'arrow');
    setptr(figureH,'arrow');


%%%%
% little function to determine if a data point is valid or not
function [isValid]=validDataPt(value, stdev, cutoffStddev)
    if( (value==-99 & stdev==0) | (value==0 & stdev==0) | (cutoffStddev < stdev) )
        isValid=0;
    else
        isValid=1;
    end;
```

****