NASA Contractor Report NASA/CR-2008-214746

Automated VAHIRR Product
Volume 1 (Main Report)

Joe H. Barrett III
NASA Applied Meteorology Unit
Kennedy Space Center, Florida

Juli Miller
Debbie Charnasky
Robert Gillen
ENSCO, Inc.

July 2008

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA Contractor Report NASA/CR-2008-214746

Automated VAHIRR Product
Volume 1 (Main Report)

Joe H. Barrett III
NASA Applied Meteorology Unit
Kennedy Space Center, Florida

Juli Miller
Debbie Charnasky
Robert Gillen
ENSCO, Inc.

July 2008

Acknowledgments

The authors thank the Airborne Field Mill II program team for providing data and answering questions about the VAHIRR radar product. In addition, the authors thank Dr. James Dye of the National Center for Atmospheric Research for providing a thorough and helpful review of an earlier draft of this report.

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

This report is also available in electronic form at

http://science.ksc.nasa.gov/amu/

Executive Summary

The Lightning Launch Commit Criteria (LLCC) are a set of rules used to avoid natural and rocket-triggered lightning strikes to space vehicles. The LLCC are maintained by the Lightning Advisory Panel (LAP). The previous LLCC were shown to be overly restrictive, potentially leading to costly launch delays and scrubs. The Volume-Averaged Height Integrated Radar Reflectivity (VAHIRR) algorithm and the updated LLCC for anvil clouds were developed using data collected by the Airborne Field Mill II (ABFM) program conducted in 2000/2001. The use of the VAHIRR product is expected to increase launch opportunities, while maintaining safety. Currently, a manual work-around is used to calculate VAHIRR with existing radar products. The AMU was tasked to develop an automated version of the VAHIRR product, to reduce the operational impact of the anvil cloud LLCC.

VAHIRR is defined as the product of the Volume-Averaged Radar Reflectivity and the Average Cloud Thickness within a Specified Volume. The Specified Volume is bounded in the horizontal and vertical planes, with perpendicular sides located 5.5 km north, east, south, and west of a point on the flight track, on the bottom by the 0^o C level, and on the top by the upper extent of all clouds. Any grid point within the Specified Volume is included in the average if and only if it has a radar reflectivity ≥ 0 dBZ.

The AMU automated VAHIRR software passed all test procedures except for the ABFM Comparison Test, which compared the AMU VAHIRR product with the ABFM VAHIRR product. Large differences between the two were found in the ABFM Comparison Test. Several possible sources of error in comparing the two products were identified and evaluated. The sources included differences in the calculation of cloud heights, Average Cloud Thickness, fractional coverage of non-negative reflectivity, latitude/longitude positions of the VAHIRR values, and differences in grid spacing.

Since the AMU was unable to completely determine the cause of the differences between the two products, the current version of the automated VAHIRR software will not be released for operational use.

Although the AMU work on the automated VAHIRR software did not result in an operational product, the lessons learned will be used on any future development and testing. Some recommendations for future work include the following:

- The AMU VAHIRR product is a 4-bit product, with only 16 data levels. A better choice would be to create an 8-bit product, with 256 data levels.
- Unless the LAP changes the definition of VAHIRR, the automated VAHIRR product should be implemented in the same way as the ABFM VAHIRR product. This includes calculating cloud heights, cloud bottoms, cloud tops, average cloud thickness, and vertical and horizontal grid spacing in the same way,
- The comparison between the automated VAHIRR product and the ABFM VAHIRR product should be limited to areas in which the fractional coverage of non-negative reflectivity is ≥ 10%. This is to reduce the uncertainty of the VAHIRR calculation in highly broken clouds or regions in which radar elevation scan gaps exist,
- In the ABFM VAHIRR product, VAHIRR values are only displayed along the aircraft’s track. In the AMU VAHIRR product, VAHIRR values are displayed for the entire radar coverage. Therefore, it would be useful to run the ABFM’s VAHIRR software for the entire radar coverage,
- Compare the automated VAHIRR product to manual calculations,
- A software program should be used to write out the intermediate values (such as sample size, average cloud bottom/top, average reflectivity, average cloud thickness, etc.) to output files, for later display in a spreadsheet application. This would make the testing procedures less manually-intensive, and
- The automated VAHIRR software should undergo an “operational” test in which the reliability of the software is tested with live incoming radar data.

This report consists of four volumes: main report, source code, installation guide, and test results.

Table of Contents – Volume 1

List of Figures – Volume 1

List of Tables – Volume 1

1. Introduction

Lightning Launch Commit Criteria (LLCC) are used for all U.S. government and commercial launches at government and civilian ranges (Willet et al. 1999). They are maintained by the Lightning Advisory Panel (LAP), a a team of scientists with lightning expertise. Space Shuttle Flight Rules (FR) incorporate provisions similar to the LLCC for all Shuttle landings. These FR and LLCC are designed to avoid natural and triggered lightning strikes to space vehicles, which can endanger the vehicle, payload, crew and general public. The previous LLCC for anvil cloud, meant to avoid triggered lightning, have been shown to be overly restrictive. They ensure safety, but falsely warn of danger and can lead to costly launch delays and scrubs. A new LLCC for anvil clouds, and an associated radar-derived quantity called the Volume Averaged Height Integrated Radar Reflectivity (VAHIRR) needed to evaluate the new LLCC, were developed using data collected by the Airborne Field Mill II (ABFM) research program managed by Kennedy Space Center (KSC) (Dye et al. 2006, 2007). Dr. Harry Koons of Aerospace Corporation conducted a risk analysis of the VAHIRR product. The results indicated that the LLCC based on the VAHIRR product would pose a negligible risk of flying through hazardous electric fields.

The Applied Meteorology Unit (AMU) was tasked to create an automated VAHIRR product for the Weather Surveillance Radar – 1988 Doppler (WSR-88D) system. Currently, a work-around is required to estimate VAHIRR values using existing radar products, such as Composite Reflectivity and reflectivity cross sections (Merceret et al., 2006). The work-around is manually intensive and gives conservative results. An automated VAHIRR product would give quicker and more accurate VAHIRR values.

1.1 VAHIRR Background

The ABFM program was conducted during June 2000 and May/June 2001 near KSC to develop improved and physically-based LLCC that would be safe but less restrictive than the previous LLCC (Dye et al., 2006, 2007). The project investigated the magnitude of electric fields inside thunderstorm anvils, and how they were related to the cloud microphysics and radar reflectivity. Airborne measurements were made by instruments mounted on a University of North Dakota Citation II jet aircraft. Radar coverage included the Patrick Air Force Base (PAFB) Weather Surveillance Radar 1974-C (WSR-74C) radar and the WSR-88D radar at Melbourne, FL (KMLB). Total lightning measurements were made with the KSC Lightning Detection and Ranging and the Air Force Cloud to Ground Lightning Surveillance System networks. More details on these sensors are found in Roeder *et al*. (2003).

When the radar reflectivity near the aircraft was less than 5 to 10 dBZ, the magnitude of the three-dimensional electric fields was less than 3 kilovolts/meter (kV/m). This value poses little threat for rocket-triggered lightning. A new radar-derived parameter called Volume Averaged Height Integrated Radar Reflectivity (VAHIRR) was evaluated. VAHIRR showed a trend of increasing values with increases in the electric field magnitude above 3 kV/m. An extreme value analysis of VAHIRR values ≤ 10 dBZ-km (equivalent to a 5 dBZ reflectivity average in a 2 km-thick anvil) showed that the probability of having an electric field magnitude larger than 3 kV/m was less than 1 in 10,000. The LLCC for anvil clouds was updated in 2005 to incorporate the VAHIRR parameter resulting from the ABFM program (Krider et al., 2006).

1.2 Formal Definition of VAHIRR

The following is a summary of the LAP's definition of VAHIRR.

VAHIRR (units of dBZ-km) is the product of the Volume-Averaged Radar Reflectivity and the Average Cloud Thickness within a Specified Volume relative to a point along the flight track of a space launch vehicle (Merceret et al., 2006). The Specified Volume is bounded in the horizontal and vertical planes, with perpendicular sides located 5.5 km north, east, south, and west of a point on the flight track, on the bottom by the 0° C level, and on the top by the upper extent of all clouds. The Volume-Averaged Radar Reflectivity is the arithmetic average (in dBZ) of the cloud radar reflectivity within the Specified Volume. Normally, a radar processor will report reflectivity values interpolated onto a regular, three-dimensional array of grid points. Any such grid point within the Specified Volume is included in the average if and only if it has a radar reflectivity ≥ 0 dBZ.

The Average Cloud Thickness is the altitude difference between the average top and the average base of all clouds within the Specified Volume. The cloud base is the higher of (1) the 0° C level and (2) the lowest extent in altitude of all cloud radar reflectivities 0 dBZ or greater. The cloud top is the highest extent of all cloud radar reflectivities 0 dBZ or greater. Allowance must be made for the vertical separation of grid points in calculating cloud thickness. The cloud base will be the altitude of the base grid point minus half of the grid-point vertical separation. Similarly, the cloud top will be the altitude of the top grid point plus half of the vertical separation. Thus, a cloud represented by only a single grid point having a radar reflectivity equal to or greater than 0 dBZ within the Specified Volume, would have an Average Cloud Thickness equal to the vertical grid-point separation in its vicinity.

The VAHIRR measurement must be made in the absence of significant attenuation by intervening storms or by water or ice on the radome. It is invalid at any point on the flight track that is within 20 km of reflectivities 35 dBZ or greater, at altitudes of 4 km above mean sea level (MSL), and at any point within 20 km of any type of lightning that occurred in the previous 5 minutes. The Specified Volume must not contain any portion of the cone-of-silence above the radar (Figure 1 shows the cone-of-silence), nor any portion of sectors that may have been blocked out for payload-safety reasons. A vertical limit is added to the cone-of-silence restriction to avoid invalidating the VAHIRR values everywhere. The individual grid-point reflectivities used to determine either the Volume-Averaged Radar Reflectivity or the Average Cloud Thickness must be from meteorological targets.

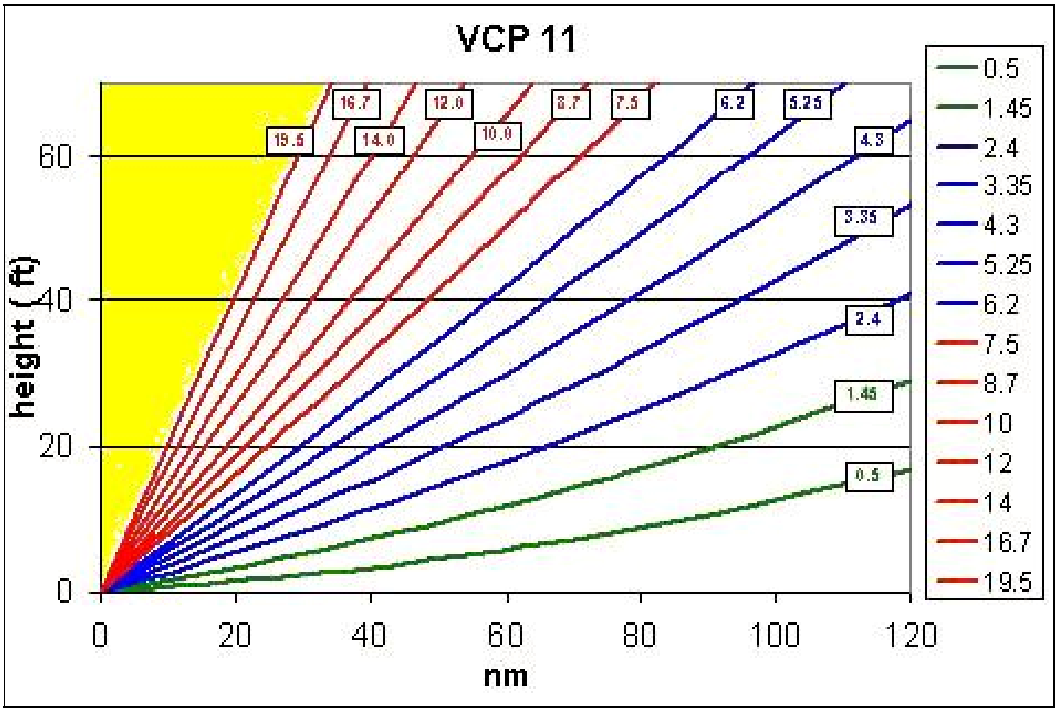


Figure 1. The volume scan pattern used in the ABFM program. The cone-of-silence region is shown in yellow. Distance from the radar is shown in n mi. The cone-of-silence is a cone-shaped region above the radar antenna, in which no pulses are transmitted or signals detected. The legend at right shows the elevation angle values. The figure is modified from Figure 5-1 in Federal Meteorological Handbook-11, Doppler Radar Meteorological Observations (FMH-11), Part C.

2. Automated VAHIRR product

The AMU developed the automated VAHIRR software from the LAP's definition of VAHIRR. Sections 2.1 through 2.7 describe how the software was developed.

2.1 Programming Environment

The National Weather Service and Mitretek Systems (now called Noblis, Inc.) created a software package called Common Operations and Development Environment (CODE) so users could develop and test new radar algorithms for the WSR-88D. A public version of CODE can be downloaded for free at http://www.weather.gov/code88d. The CODE tool set includes sample products, visualization tools and a Human Computer Interface (HCI) application to manage the radar products. A computer using the Linux operating system with CODE installed is referred to as an "ORPG-clone". An ORPG-clone is identical to the operational Open Radar

Product Generator (ORPG), except that it is not connected to a WSR-88D. The ORPG-clone was used by the AMU to create, debug, and execute the automated VAHIRR software and to display the results. The Build 8 version (corresponding to Build 8 of the operational ORPG) was used, since it was the current version during the course of this work.

The VAHIRR product requires access to all elevation angles of a WSR-88D volume scan in Level II format. In Level II data, the elevation angles are stored in Polar format so that reflectivity values are displayed with respect to an azimuth angle and distance from the radar. Since the automated VAHIRR product is a Cartesian product with a horizontal resolution of 1 km, the AMU used the CODE algorithms to convert from Polar to Cartesian systems as much as possible, but had to generate some unique conversions as well.

2.2 ORPG-clone System

The automated VAHIRR software is part of a suite of meteorological analysis algorithms that execute within the ORPG-clone system. The ORPG-clone ingests Level II data from the Radar Data Acquisition (RDA) and stores the final products in a centralized database file. The products are in Level III format and can be displayed with the CODEview Graphics (CVG) application. Products can also be displayed on subsystems such as the Advanced Weather Interactive Processing System (AWIPS) and Open Principal User Processor (OPUP). The following diagram shows the flow of data from the radar to the display system.

reflectivity,
WSR-88D-----*mean radial velocity,*---->**RDA**-----*level II data*---->**RPG**-----*level III data*---->**Display**
& spectrum width data

The format of Level II data is described in the document, "ICD for the RDA/RPG". The format of Level III data is described in "ICD for RPG to Class 1 User". Both documents are available for download at: http://www.osf.noaa.gov/ssb/cm/icd_downloads.asp.

2.3 Requirements for the Automated VAHIRR product

The automated VAHIRR product was built upon the following requirements, which are based on the definition of VAHIRR in the LLCC:

1. Calculate VAHIRR only for points where the lowest and highest elevation scans lie outside of any clouds. Reflectivity, if any, on the lowest elevation scan must be at or below the height of the 0° Celsius isotherm or negative. Reflectivity from the highest elevation scan, if any, must be negative. The highest elevation scan in a volume scan may not extend as far horizontally as the lowest elevation scan in the same volume, therefore lower elevation scans serve as the highest elevation over some points. Figure 2 shows a horizontal area where the lowest elevation scan is above the freezing level, when the freezing level is 15,000 ft MSL. The area to the right of the vertical purple line is disqualified for VAHIRR calculations. Figure 3 shows a horizontal area where the reflectivity on the highest elevation scan is non-negative, due to a cloud layer between 45,000 ft and 55,000 MSL. The area to the left of the vertical purple line is disqualified for VAHIRR calculations.

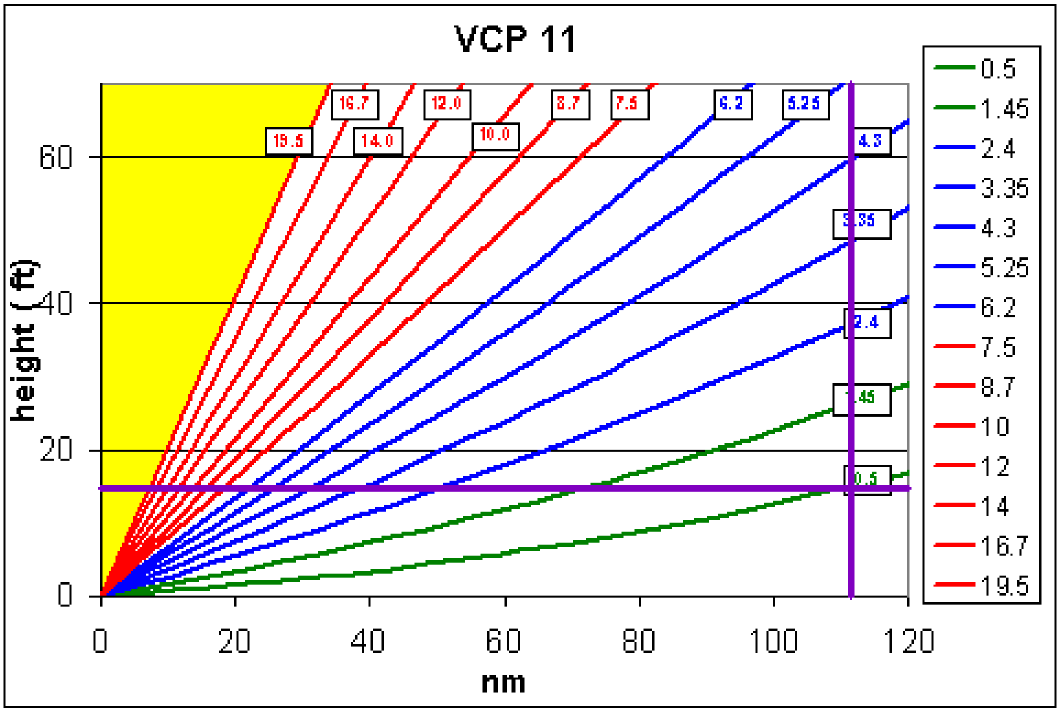


Figure 2. The area to the right of the vertical purple line is disqualified for VAHIRR calculations since the lowest elevation scan is above the freezing level of 15,000 ft MSL. The horizontal purple line depicts the freezing level.

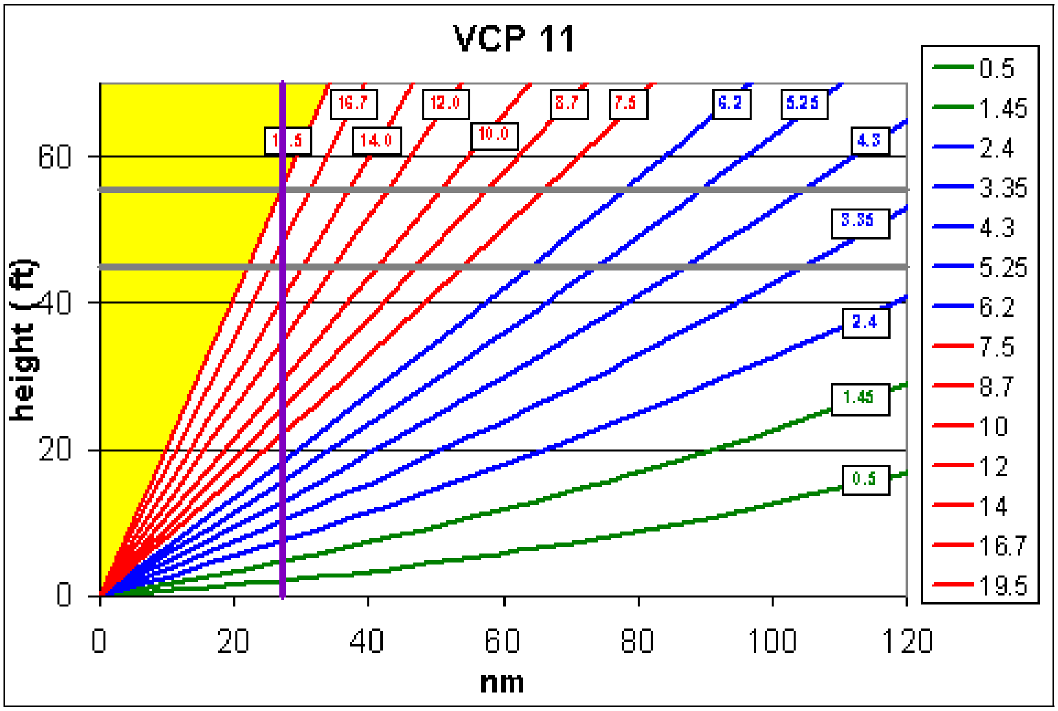


Figure 3. The area to the left of the vertical purple line is disqualified for VAHIRR calculations since the highest elevation scan contains non-negative reflectivity. The two horizontal gray lines depict a cloud layer with non-negative reflectivities between 45,000 ft and 55,000 ft MSL.

2. Calculate VAHIRR only for the radar volume outside of the cone-of-silence. A vertical cap is added to the cone-of-silence restriction to avoid invalidating the VAHIRR values everywhere. The cone-of-silence parameter in the product is defined as the highest height of interest in calculating VAHIRR. For example, if the cone-of-silence parameter is set to 15 km MSL, then a grid point will be disqualified if the highest elevation scan above the point lies at or below 15 km MSL. The default cone-of-silence value is set to 20 km, but can be adjusted by the radar operator. The top of the anvil cloud is preferred for the value, followed by the tropopause height, with the default 20 km MSL being the last option. Figure 4 shows the horizontal area where the highest elevation scan is at or below the cone-of-silence parameter, which is set at 55,000 ft MSL. The area to the left of the vertical purple line is disqualified for VAHIRR calculations.

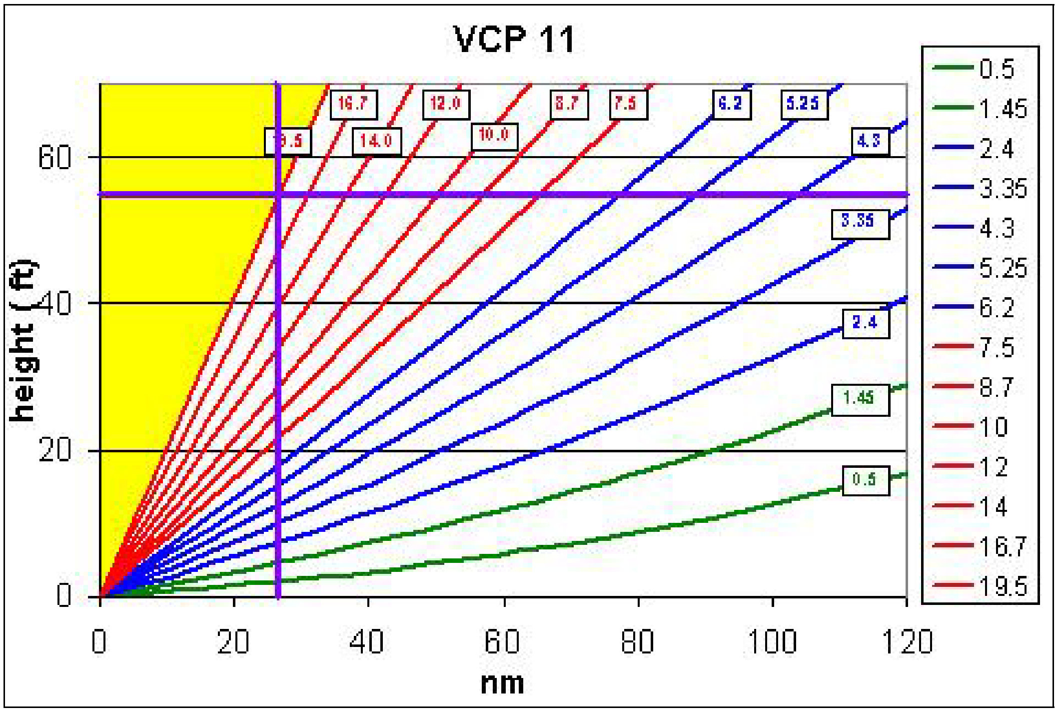


Figure 4. The area to the left of the vertical purple line is disqualified for VAHIRR calculations since the highest elevation scan is at or below the cone-of-silence parameter. The cone-of-silence parameter is set at 55,000 ft MSL.

3. Do not calculate VAHIRR for target points containing missing or invalid reflectivities,

4. Calculate VAHIRR only for target points that have a complete 11 x 11 km grid point set (121 gridpoints) that satisfies the conditions in the first three requirements according to the following equations:

$$\text{VAHIRR}_{\text{target point}} = \text{VolumeAveragedReflectivity}_{\text{target point}} \times \text{AverageCloud Thickness}_{\text{target point}}$$

where,

$$\text{VolumeAveragedReflectivity}_{\text{targetpoint}} = \frac{\sum_{\text{gridpoint}=1}^{121}\left(\text{Vertical Reflectivity Ave}_{\text{gridpoint}} \times \text{Vertical Sample Size}_{\text{gridpoint}}\right)}{\sum_{\text{gridpoint}=1}^{121}\text{Vertical SampleSize}_{\text{gridpoint}}}$$

$$\text{AverageCloud Thickness}_{\text{target point}} = \frac{\sum_{\text{gridpoint}=1}^{121} \text{Cloud Top Height}_{\text{gridpoint}}}{\text{Horizontal Sample Size}} - \frac{\sum_{\text{gridpoint}=1}^{121} \text{Cloud Bottom Height}_{\text{gridpoint}}}{\text{Horizontal Sample Size}}$$

and

$$\text{Vertical Reflectivity Ave}_{\text{gridpoint}} = \frac{\text{Sum of non-negative reflectivities from } 0^{\circ}\text{C to highest elevation scan}}{\text{Number of elevation scans at or above } 0^{\circ}\text{C with non-negative reflectivity}}$$

5. Average only non-negative reflectivities at or above the height of the 0° C isotherm,
6. Cloud top is the height of the highest non-negative reflectivity, vertically above a point,
7. Cloud bottom is the height of the 0° C isotherm if non-negative reflectivity occurs below the 0° C isotherm, vertically above a point. Otherwise it is the height of the first non-negative reflectivity at or above the 0° C isotherm,
8. Given the grid-point representation of a typical radar processor, make allowance for the vertical separation of grid points in calculating cloud thickness. The cloud bottom will be the altitude of the corresponding bottom grid point minus half of the grid-point vertical separation. Similarly, cloud top will be the altitude of the corresponding top grid point plus half of the vertical separation. Thus, a cloud represented by only a single grid point having a radar reflectivity ≥ 0 dBZ within the Specified Volume would have an average cloud thickness equal to the vertical grid point separation in its vicinity.

   Since the product uses Level II data as input, the elevation scans are not evenly spaced and the vertical spacing between adjacent elevation scans increases with distance from the radar. The cloud bottom is therefore calculated as the height of the elevation scan minus half the distance between the elevation scan and the adjacent elevation scan below it. The cloud top is calculated as the height of the elevation scan plus half the distance between the elevation scan and the adjacent elevation scan above it.
9. When calculating cloud height and thickness, the product must take into account the MSL elevation of the WSR-88D.
10. Output VAHIRR as dBZ-kilofeet (kft).

2.4 Two-Pass Algorithm to Calculate VAHIRR

In order to calculate the VAHIRR value at a target point, the volume must be defined by the five 1-km grid intervals in the horizontal plane north, south, east and west of the grid point of interest, i.e. 11 x 11 grid points in the x-y directions centered on the target point. The VAHIRR value is calculated by multiplying the average cloud thickness by the average non-negative radar reflectivity in the volume.

A two-pass algorithm is used. The cloud top and bottom, number of non-negative radar reflectivity values, and average radar reflectivity above each horizontal grid point is calculated on the first pass. For each horizontal grid point, the algorithm determines if it is "qualified" or not. There are three conditions that disqualify a grid point:

- The non-negative reflectivity on the lowest elevation scan is above the height of the 0° C isotherm. This is to prevent the product from underestimating the VAHIRR value,
- The reflectivity on the highest elevation scan is non-negative. This also prevents the product from underestimating the VAHIRR value, and
- The grid point is within the cone-of-silence. The product has a default cone-of-silence height of 20 km, but this can be modified by the radar operator. For example, if the cone-of-silence height is 20 km, every horizontal grid point in which the highest elevation scan is at or below 20 km is disqualified. Again, this is to prevent the product from underestimating the VAHIRR value.

The VAHIRR value for each horizontal grid point is calculated on the second pass by averaging the 11 x 11 grid point values calculated in the first pass. The entire 11 x 11 grid point set must be qualified in order to calculate the VAHIRR value. If all of the grid points have a negative volume averaged reflectivity, then no VAHIRR value is calculated.

2.5 Inputs to the Automated VAHIRR Product

All of the inputs listed below are obtained from the RDA except for the freezing level and cone-of-silence heights. The freezing level is obtained from the Hail Product. The cone-of-silence height is read from a configuration file.

Inputs to the product include:

- Height of 0^{o} C isotherm – freezing level height (kft) above MSL,
- COS – cone-of-silence height (kft) above ground level (AGL),
- Polar (Azimuth Angle) – azimuth position for a radial. Values range from 0^{o} to 360^{o} with an accuracy of 0.1^{o} or better. The beamwidth of the WSR-88D is 1.0^{o} ,
- Polar (Elevation Angle) – elevation angle in a radar volume scan. Values range from -5.0^{o} to 90.0^{o} with a precision of at least 0.1^{o},
- Polar (First Gate) – index of the first gate that contains available reflectivity data. Values range from 0 to 299.
- Polar (Number of Gates) – the number of gates with data. Values range from 0 to 299.
- Polar (Reflectivity) – radar base reflectivity data. Reflectivity values range from -32 dBZ to 90 dBZ with a precision of 0.5 dBZ.
- Range Beginning Surveillance – range (m) to the beginning of the first surveillance bin (i.e. first gate with available reflectivity data).
- Reflectivity Enabled Flag – the second bit of the message type that indicates whether reflectivity is enabled,
- Surveillance Bin Size – the bin size (m). The bin size is the horizontal length of a reflectivity value,
- Volume Scan Number – the number of the volume scan, ranging from 1 to 80.

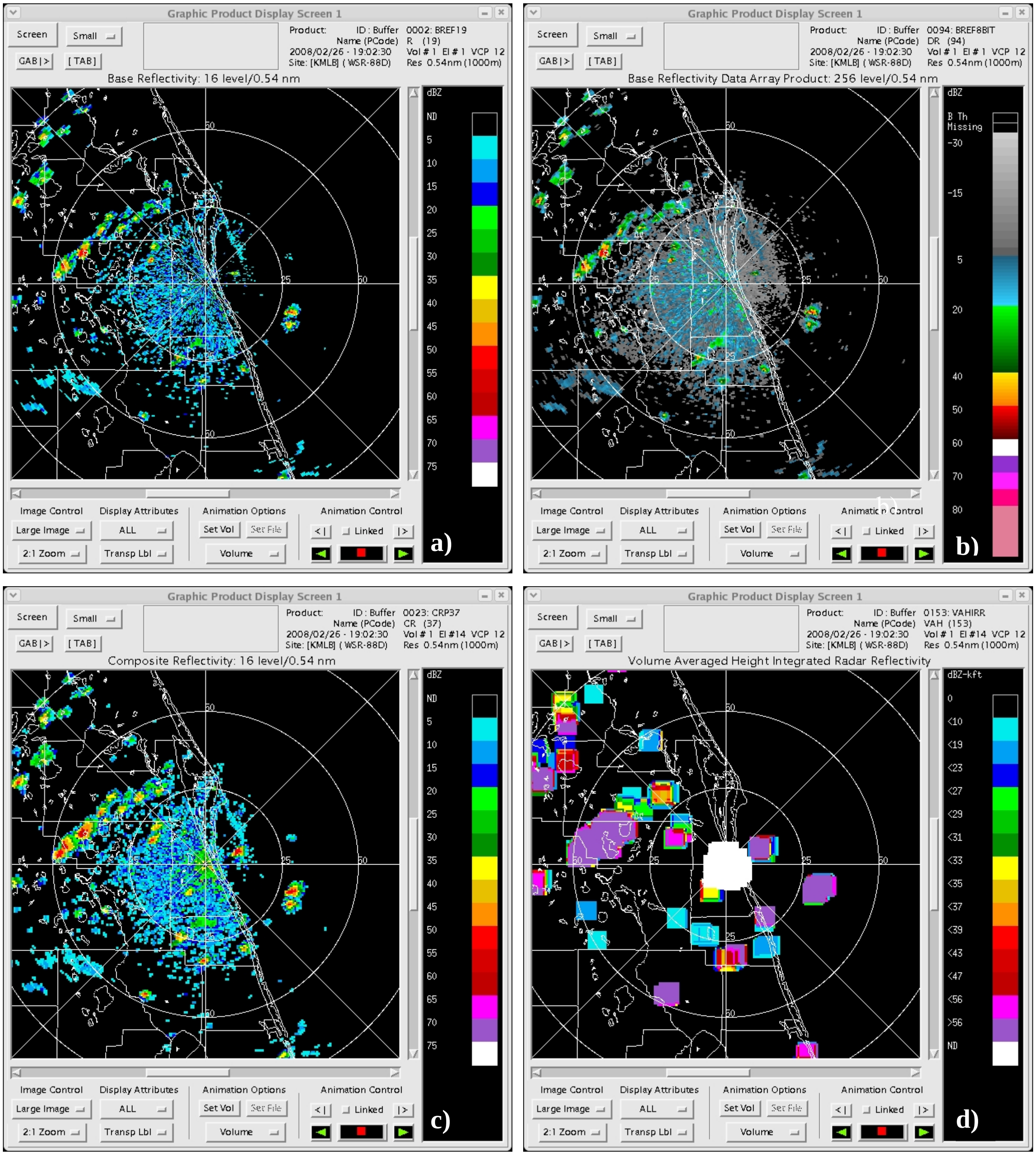
2.6 Output of the Automated VAHIRR Product

The automated VAHIRR software calculates a VAHIRR value for every square kilometer over the horizontal coverage area of a WSR-88D. Results are displayed in dBZ-kft and mapped to 16 color codes. The colors are encoded and output as a Level III product. The color codes are more closely spaced around 33 dBZ-kft to aid in determining whether the LAP's VAHIRR criteria of 33 dBZ-kft has been exceeded. The automated VAHIRR product uses the same display format as the 16-level 1-km Composite Reflectivity product. The colors associated with the VAHIRR values are shown in Table 1.

Figure 5 was generated using the ORPG-clone and shows an example of the Base Reflectivity (0.5^{o} elevation), Base Reflectivity Data Array (0.5^{o} elevation), Composite Reflectivity, and VAHIRR products from the KMLB WSR-88D at 1902 UTC on 26 February, 2008. A broken line of heavy showers or thunderstorms was to the west of KSC and Cape Canaveral Air Force Station (CCAFS). Isolated showers or thunderstorms were over the rest of east central Florida. Except for VAHIRR, the specifications of these products are available in Part C of the FMH-11 (http://www.ofcm.gov/fmh11/fmh11.htm). The Composite Reflectivity and VAHIRR products are "volume" products, meaning that all elevation scans are used to create them. The display values in the reflectivity products are in units of dBZ. The VAHIRR values are displayed in dBZ-kft instead of dBZ-km, since the Spaceflight Meteorology Group (SMG) and 45th Weather Squadron (45 WS), the primary users, usually work with English units. The Launch Weather Officers in the 45 WS were consulted during the development of the VAHIRR color scheme for the 16 data levels. The VAHIRR product uses the color black to represent a VAHIRR value of zero or for no computed VAHIRR. The color white represents disqualified grid points.

Table 1. Mapping of VAHIRR values to color levels.

VAHIRR in dBZ-km	**VAHIRR in dBZ-kft**	**VAHIRR level (color)**
-99.0 (out-of-bounds or disqualified)	-99.0 (out-of-bounds or disqualified)	15 (white)
-1.0 or 0.0	-1.0 or 0.0	0 (black)
0.0 < dBZ-km < 3.05	0.0 < dBZ-kft < 10.00	1 (light blue)
3.05 ≤ dBZ-km < 5.79	10.00 ≤ dBZ-kft < 19.00	2 (medium blue)
5.79 ≤ dBZ-km < 7.01	19.00 ≤ dBZ-kft < 23.00	3 (dark blue)
7.01 ≤ dBZ-km < 8.23	23.00 ≤ dBZ-kft < 27.00	4 (light green)
8.23 ≤ dBZ-km < 8.84	27.00 ≤ dBZ-kft < 29.00	5 (medium green)
8.84 ≤ dBZ-km < 9.45	29.00 ≤ dBZ-kft < 31.00	6 (dark green)
9.45 ≤ dBZ-km < 10.06	31.00 ≤ dBZ-kft < 33.00	7 (light yellow)
10.06 ≤ dBZ-km < 10.67	33.00 ≤ dBZ-kft < 35.00	8 (dark yellow)
10.67 ≤ dBZ-km < 11.28	35.00 ≤ dBZ-kft < 37.00	9 (orange)
11.28 ≤ dBZ-km < 11.89	37.00 ≤ dBZ-kft < 39.00	10 (light red)
11.89 ≤ dBZ-km < 13.11	39.00 ≤ dBZ-kft < 43.00	11 (medium red)
13.11 ≤ dBZ-km < 14.33	43.00 ≤ dBZ-kft < 47.00	12 (dark red)
14.33 ≤ dBZ-km < 17.07	47.00 ≤ dBZ-kft < 56.00	13 (fuschia)
17.07 ≤ dBZ-km	56.00 ≤ dBZ-kft	14 (purple)



Graphic Product Display Screen 1
Product: ID : Buffer 0002: BREF19
Name (PCode) R (19)
2008/02/26 - 19:02:30 Vol # 1 El # 1 VCP 12
Site: [KMLB] (WSR-88D) Res 0.54nm (1000m)
Base Reflectivity: 16 level/0.54 nm

Graphic Product Display Screen 1
Product: ID : Buffer 0094: BREF8BIT
Name (PCode) DR (94)
2008/02/26 - 19:02:30 Vol # 1 El # 1 VCP 12
Site: [KMLB] (WSR-88D) Res 0.54nm (1000m)
Base Reflectivity Data Array Product: 256 level/0.54 nm

Graphic Product Display Screen 1
Product: ID : Buffer 0023: CRP37
Name (PCode) CR (37)
2008/02/26 - 19:02:30 Vol # 1 El #14 VCP 12
Site: [KMLB] (WSR-88D) Res 0.54nm (1000m)
Composite Reflectivity: 16 level/0.54 nm

Graphic Product Display Screen 1
Product: ID : Buffer 0153: VAHIRR
Name (PCode) VAH (153)
2008/02/26 - 19:02:30 Vol # 1 El #14 VCP 12
Site: [KMLB] (WSR-88D) Res 0.54nm (1000m)
Volume Averaged Height Integrated Radar Reflectivity

Figure 5. Radar products from the WSR-88D KMLB volume scan at 1902 UTC 26 February, 2008: a) (upper left) shows the Base Reflectivity product at 0.5° elevation; b) (upper right) shows the Base Reflectivity Data Array product at 0.5° elevation; c) (lower left) shows the Composite Reflectivity product; and d) (lower right) shows the VAHIRR product.

2.7 Source Code for the Automated VAHIRR Product

The design of the source code is available in Appendix A, while the actual source code is in Volume 2.

3. Initial Test Procedure

After the initial development of the automated VAHIRR software, the AMU created an initial test procedure. The purpose of this test procedure was to verify the accuracy of the product using artificial or "canned" data. The test and its results are provided in Appendix B.

4. Installing and Running the Automated VAHIRR Software

After the initial test procedure was completed successfully at ENSCO's Cocoa Beach facility, the AMU installed an ORPG-clone in the AMU laboratory, located in Room 151 of the Morrell Operations Center (MOC, formally Range Operations and Control Center) at CCAFS. The ORPG-clone was configured so that the automated VAHIRR product could be generated in near real-time. The AMU assisted the 45 WS and SMG with the installation of ORPG-clones at their facilities. In order to standardize the installation of the ORPG-clone, the AMU wrote the VAHIRR Installation Guide, presented in Volume 3. The following process is used for generating a new product in near real-time:

1. The KMLB WSR-88D usually completes a new volume scan every 5 to 6 minutes. When a volume scan has completed, the Level II data are transmitted to several locations, including the Marshall Space Flight Center (MSFC). The radar data are stored by the MSFC Local Data Manager (LDM) server (software and documentation available at http://www.unidata.ucar.edu/software/ldm/).
2. Linux servers at the AMU and SMG request data from MSFC's LDM server. When new Level II data is received by a Linux server, it sends the data to an ORPG-clone. After the ORPG-clone receives the data, it produces new radar products, such as Composite Reflectivity and VAHIRR. Newly-created products are added to a centralized database file on the ORPG-clone. All of the products in the database can be viewed by the CVG application.

Based on the NASA IT Security Plan for the AMU, the AMU was required to disconnect the 45 WS ORPG-clone from the AMU LDM server. As a result, the 45 WS lost their source of real-time Level II data. The 45 WS is in the process of finding a new source. Since SMG's primary weather display system is AWIPS, the AMU and SMG created procedures to use the automated VAHIRR product in AWIPS. Because they have different versions of AWIPS, two separate procedures were created to view the product at SMG (Figure 6) and the AMU (Figure 7). Appendix C and Appendix D describe the procedures to display the product in AWIPS at SMG and the AMU, respectively.

The AMU also created the "AMU Trajectory Map Maker" software program that overlays launch and landing trajectories on the product in AWIPS. The installation and use of the program is described in Appendix E.

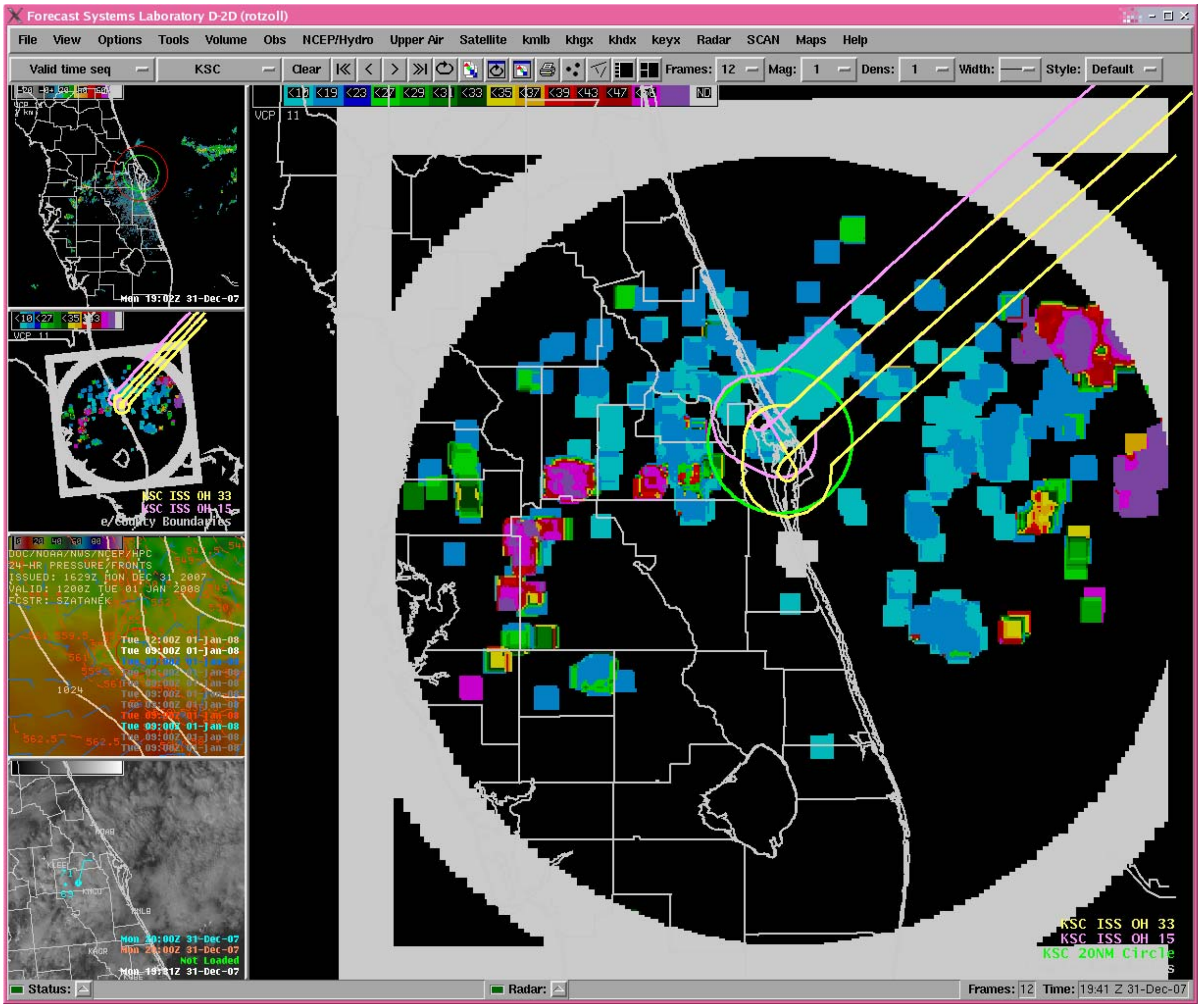
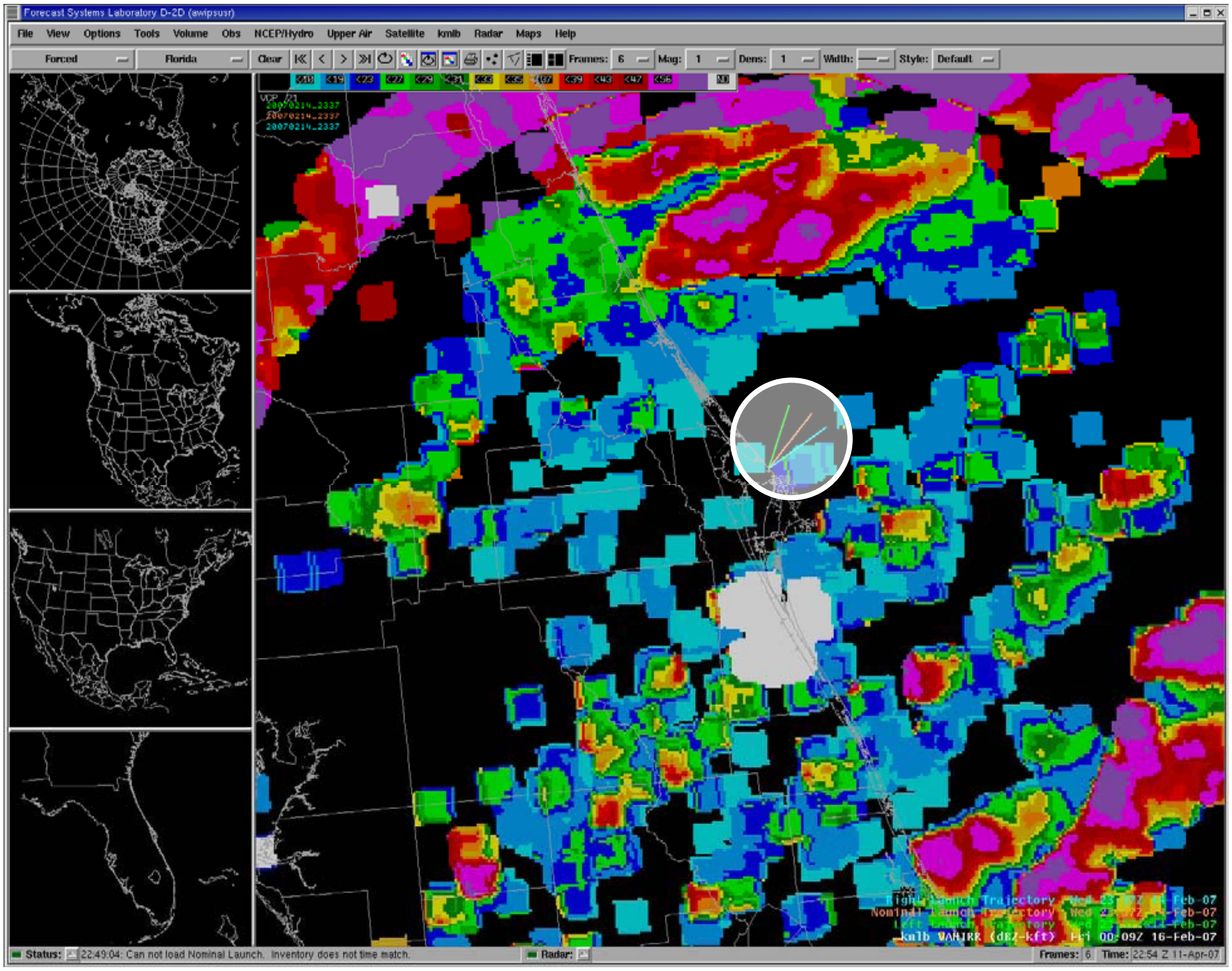


Figure 6. The automated VAHIRR product in the SMG AWIPS system, from a volume scan on 31 December 2007. The shuttle landing trajectory is overlaid on top of the image. The graphic was created with a software script written by SMG.



Status: 22:49:04: Can not load Nominal Launch. Inventory does not time match.

Figure 7. The automated VAHIRR product in the AMU AWIPS system from a volume scan on 16 February 2007. A launch trajectory graphic is overlaid over the image (inside white circle). The graphic was created with the AMU Trajectory Map Maker.

5. Factory Acceptance Test

The AMU was tasked to undertake much more rigorous testing than the Initial Test Procedure accomplished. Therefore, the VAHIRR Factory Acceptance Test (FAT) was written and carried out. The FAT was composed of five procedures:

- The Baseline Test was performed to demonstrate the accuracy of the automated VAHIRR software for a basic set of input data,
- The 0^{o} C Height Test was performed to demonstrate that the software produces the correct results when varying the 0^{o} C height,
- The Cone-of-Silence Test was performed to demonstrate that the software produces the correct results when varying the height of the cone-of-silence,
- The Reflectivity Average for Multiple Tilts Test was performed to demonstrate that the software handles multiple cloud layers correctly, and
- The ABFM Comparison Test was performed to compare the AMU VAHIRR product with the ABFM VAHIRR product.

All of the procedures passed, except for the ABFM Comparison Test. The results of the ABFM Comparison Test are discussed in Section 6 of this report. The contents of the FAT are presented in Volume 4.

6. ABFM Comparison Test

The AMU VAHIRR product was compared to the "Volume Integral 11x11_0" product from the ABFM program. The latter product is hereafter referred to as the ABFM VAHIRR product. Data from the ABFM program were used for the test and were downloaded from their website at http://abfm.ksc.nasa.gov. In order to have a large enough data set, data from multiple case study days were obtained. Only case study days without data quality issues or aircraft instrumentation problems were considered. The comparison only used data points in which the ABFM VAHIRR values were less than 56 dBZ-kft, since the highest range bin in the AMU VAHIRR product is ≥ 56 dBZ kft. This limited the comparison to target points with relatively low values of reflectivity, cloud thickness, and fractional coverage of non-negative reflectivity. In addition, areas with large gradients in VAHIRR value were mostly avoided.

In order to pass the test, all of the AMU VAHIRR values should have been within 2 data levels of the ABFM VAHIRR values. The mapping of the 16 data levels to color levels is shown in Table 1.

The test procedure for the ABFM comparison is described in Section 7 of Volume 4.

6.1 Initial Results of the ABFM Comparison Test

The initial results of the comparison showed large differences between the two products, with a significant positive bias in the AMU VAHIRR product (Table 2). This prompted the AMU to investigate the reasoning behind the large differences.

Table 2. Initial results of the ABFM Comparison Test. Failed values, in which the two products differed by more than two data levels, are highlighted in yellow.

Date	Flight track time (UTC)	Volume scan time (UTC)	Latitude	Longitude	ABFM VAHIRR (dBZ-kft)	AMU VAHIRR (dBZ-kft)
6/12/2000	163410	162955	29.070	-81.058	54.60	47 to < 56
6/12/2000	163420	162955	29.060	-81.063	43.63	≥ 56
6/12/2000	163430	162955	29.051	-81.070	34.91	≥ 56
6/12/2000	163440	162955	29.042	-81.073	30.27	≥ 56
6/12/2000	163450	162955	29.031	-81.071	29.94	≥ 56
6/12/2000	163520	163502	29.016	-81.044	35.53	≥ 56
6/12/2000	163530	163502	29.020	-81.032	34.68	≥ 56
6/12/2000	163540	163502	29.027	-81.024	35.16	≥ 56
6/12/2000	163550	163502	29.036	-81.019	34.98	≥ 56
6/12/2000	163600	163502	29.047	-81.020	35.51	≥ 56
6/12/2000	164150	164010	29.091	-81.071	35.33	≥ 56
6/12/2000	164200	164010	29.082	-81.066	29.92	≥ 56
6/12/2000	164210	164010	29.074	-81.057	31.20	≥ 56
6/12/2000	164220	164010	29.069	-81.046	36.84	≥ 56
6/12/2000	164230	164010	29.062	-81.037	42.43	≥ 56
6/12/2000	164750	164516	29.174	-81.210	15.17	37 to < 39
6/12/2000	164800	164516	29.164	-81.206	28.52	35 to < 37
6/12/2000	164810	164516	29.154	-81.201	29.60	35 to < 37
6/12/2000	164820	164516	29.144	-81.196	35.14	35 to < 37
6/12/2000	164830	164516	29.135	-81.191	35.32	39 to < 43
6/12/2000	164840	164516	29.125	-81.185	34.63	33 to < 35
6/12/2000	164850	164516	29.115	-81.180	30.87	29 to < 31
6/12/2000	164900	164516	29.106	-81.174	27.19	29 to < 31
6/12/2000	164910	164516	29.097	-81.168	22.39	31 to < 33
6/12/2000	164920	164516	29.091	-81.158	19.64	47 to < 56

6.2 VAHIRR-Derived Products

To investigate the differences between the two products in the initial results, the AMU gathered a much larger data set than the initial comparison, which only contained data from 6 June 2000. The larger data set includes multiple case days in the ABFM program (11-13 June 2000 and 5 June 2001). The dates and times of these data points can be requested from the AMU. To help determine the cause of the differences, four VAHIRR-derived products were created:

- VAHIRR Average Reflectivity
- VAHIRR Average Cloud Thickness
- VAHIRR Average Cloud Top, and
- VAHIRR Average Cloud Bottom

The derived products used intermediate values that are used in the final calculation of VAHIRR. The final VAHIRR value is the product of the VAHIRR Average Reflectivity and the Average Cloud Thickness. The Average Cloud Thickness is the difference of the Average Cloud Top and Average Cloud Bottom. Appendix F describes how to install the derived products onto the ORPG-clone. Table 3 displays the color scales for each of the VAHIRR derived products. Figure 8 compares the AMU VAHIRR product to the four derived products. The products show that large average reflectivity and thickness values are associated with large VAHIRR values.

Table 3. Mapping of VAHIRR derived products to color levels.

Display Color Level	Average Reflectivity	Average Cloud Bottom	Average Cloud Top	Average Cloud Thickness
15 (white)	Data out-of-bounds or disqualified	Data out-of-bounds or disqualified	Data out-of-bounds or disqualified	Data out-of-bounds or disqualified
0 (black)	dBZ ≤ 0	km ≤ 0	km ≤ 0	km ≤ 0
1 (light blue)	0 < dBZ ≤ 1	0 < km ≤ 4.5	0 < km < 5.0	0 < km ≤ 0.5
2 (medium blue)	1 < dBZ ≤ 2	4.5 < km ≤ 5.0	5.0 < km ≤ 5.5	0.5 < km ≤ 1.0
3 (dark blue)	2 < dBZ ≤ 3	5.0 < km ≤ 5.5	5.5 < km ≤ 6.0	1.0 < km ≤ 1.5
4 (light green)	3 < dBZ ≤ 4	5.5 < km ≤ 6.0	6.0 < km ≤ 6.5	1.5 < km ≤ 2.0
5 (medium green)	4 < dBZ ≤ 5	6.0 < km ≤ 6.5	6.5 < km ≤ 7.0	2.0 < km ≤ 2.5
6 (dark green)	5 < dBZ ≤ 6	6.5 < km ≤ 7.0	7.0 < km ≤ 7.5	2.5 < km ≤ 3.0
7 (light yellow)	6 < dBZ ≤ 7	7.0 < km ≤ 7.5	7.5 < km ≤ 8.0	3.0 < km ≤ 3.5
8 (dark yellow)	7 < dBZ ≤ 8	7.5 < km ≤ 8.0	8.0 < km ≤ 8.5	3.5 < km ≤ 4.0
9 (orange)	8 < dBZ ≤ 9	8.0 < km ≤ 8.5	8.5 < km ≤ 9.0	4.0 < km ≤ 4.5
10 (light red)	9 < dBZ < 10	8.5 < km ≤ 9.0	9.0 < km ≤ 9.5	4.5 < km ≤ 5.0
11 (medium red)	10 < dBZ ≤ 11	9.0 < km ≤ 9.5	9.5 < km ≤ 10.0	5.0 < km ≤ 5.5
12 (dark red)	11 < dBZ ≤ 12	9.5 < km ≤ 10.0	10.0 < km ≤ 10.5	5.5 < km ≤ 6.0
13 (fuschia)	12 < dBZ ≤ 13	10.0 < km ≤ 10.5	10.5 < km ≤ 11.0	6.0 < km ≤ 6.5
14 (purple)	13 < dBZ	10.5 < km	11.0 < km	6.5 < km

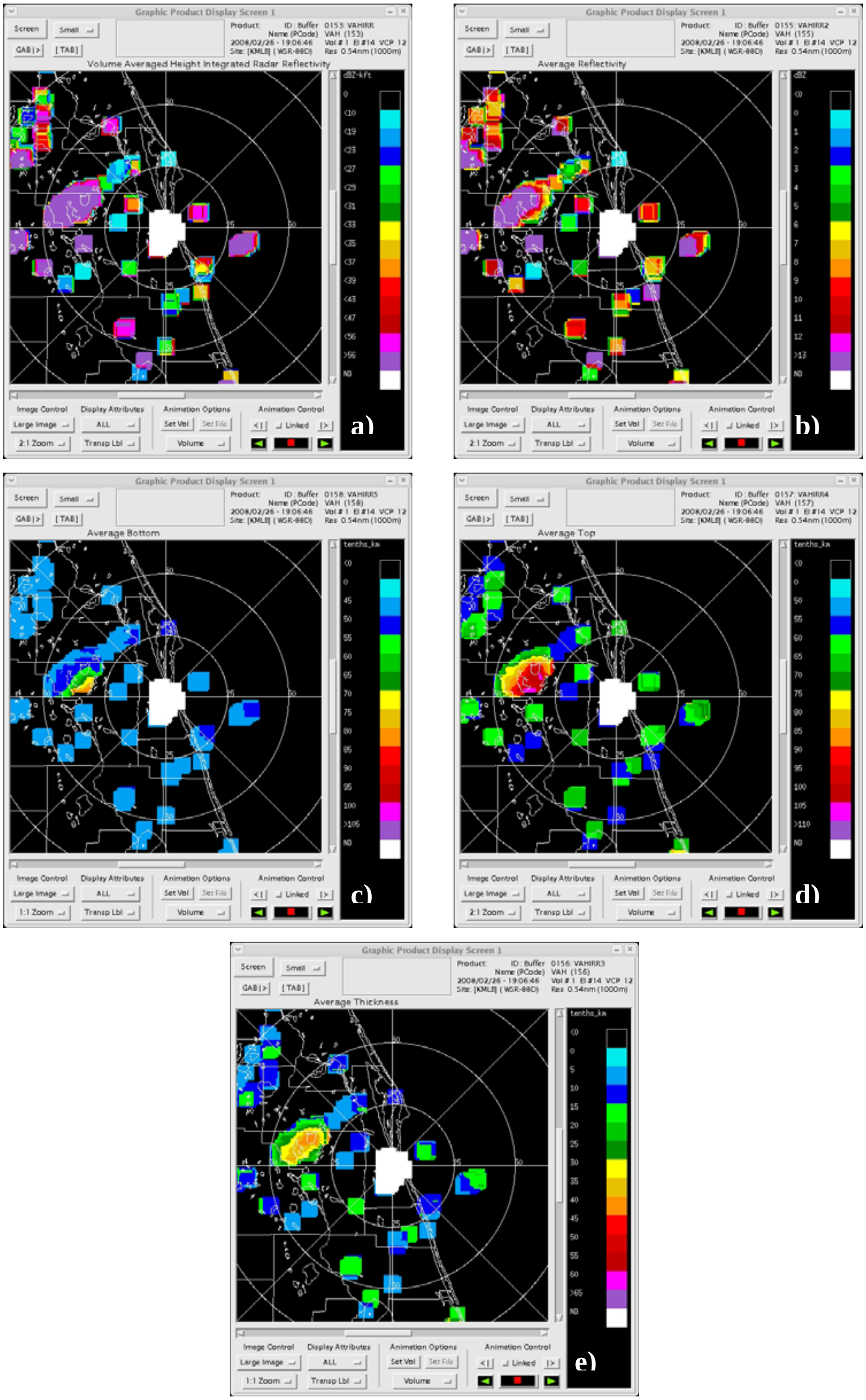


Figure 8. Radar products from the WSR-88 Melbourne volume scan at 1906 UTC 26 February 2008: a) (upper left) shows the VAHIRR product; b) (upper right) shows the VAHIRR Average Reflectivity product; c) (middle left) shows the VAHIRR Average Cloud Bottom product; d) (middle right) shows the VAHIRR Average Cloud Top product; and e) (lower left) shows the VAHIRR Average Cloud Thickness product.

6.3 Comparing the AMU and ABFM Products

In Section 6.3.1, the differences between the AMU and ABFM VAHIRR products are discussed. Section 6.3.2 compares the AMU and ABFM Average Reflectivity products, while Section 6.3.3 compares the AMU and ABFM Average Cloud Top, Bottom, and Thickness products. In Section 6.3.4, the AMU and ABFM VAHIRR products are compared with and without a minimum fractional coverage of non-negative reflectivity.

6.3.1 VAHIRR Product Comparison

Figure 9 shows a scatter plot of the ABFM VAHIRR product versus the AMU VAHIRR product. In the figure there is a large amount of scatter across the linear regression line. The low coefficient of determination value (R^2) of 0.5703 shows only a weak linear relationship between the two products. The average AMU VAHIRR value in the comparison test's data set was 43.92 dBZ-kft, while the average ABFM VAHIRR value was 33.09 dBZ-kft. This gives a positive bias in the AMU product of 33%.

The ABFM VAHIRR value was usually close to or equal to the product of the ABFM Average Reflectivity and the ABFM Average Cloud Thickness. Figure 10 compares the ABFM VAHIRR values to the product of the ABFM Average Reflectivity and ABFM Average Cloud Thickness. Except for one outlier, the values were the same, taking into account rounding errors. In the outlier, the ABFM VAHIRR value is 29.94 dBZ-kft, while the product of the AFBM Thickness and Average Reflectivity is 32.34 dBZ-kft. The ABFM average cloud base was 3.204 km, while the average cloud top was 5.002 km. Taking into effect vertical grid spacing and a freezing level of 5 km, the average cloud thickness should have been 1.002 km. However, the reported average cloud thickness was 1.057 km. An examination of this outlier by Dr. James Dye of the ABFM team revealed that there is an apparent error in the ABFM code when the cloud top is slightly above 5 km. The source of this error is unknown.

In the data set used in the comparison, the average ABFM VAHIRR value was 33.09 dBZ-kft, while the average ABFM VAHIRR value calculated from Average Reflectivity and Cloud Thickness was 36.69 dBZ-kft. This gives a positive bias of 11% in the calculated ABFM VAHIRR product, compared to the reported ABFM VAHIRR product. The reason behind the difference is unknown.

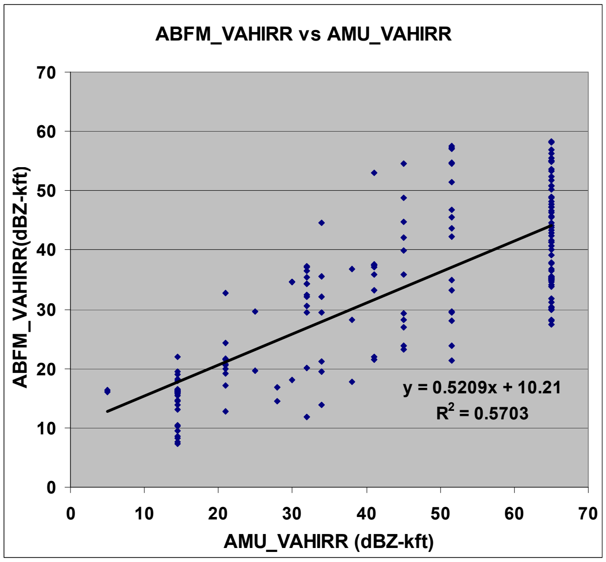


Figure 9. ABFM VAHIRR vs AMU VAHIRR. A lot of the data points in the AMU VAHIRR product are centered at 65 dBZ-kft because the highest data level is for values≥ 56 dBZ-kft. These values were estimated at 65 dBZ-kft to have enough data points to perform a valid comparison.

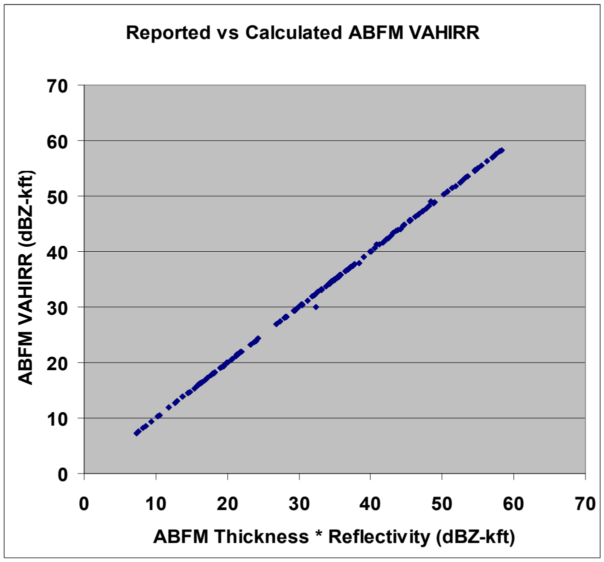


Figure 10. The reported ABFM VAHIRR values versus the product of the ABFM Average Cloud Thickness and Average Reflectivity.

6.3.2 Average Reflectivity Product Comparison

In the scatter plot for the Average Reflectivity products (Figure 11), there is much less scatter across the linear regression line and a stronger linear relationship ($R^2 = 0.8839$). The linear regression line shows nearly a one-to-one relationsip between the two products. The average AMU value was 5.73 dBZ and the average ABFM value was 5.19 dBZ, giving a positive bias in the AMU product of 10%.

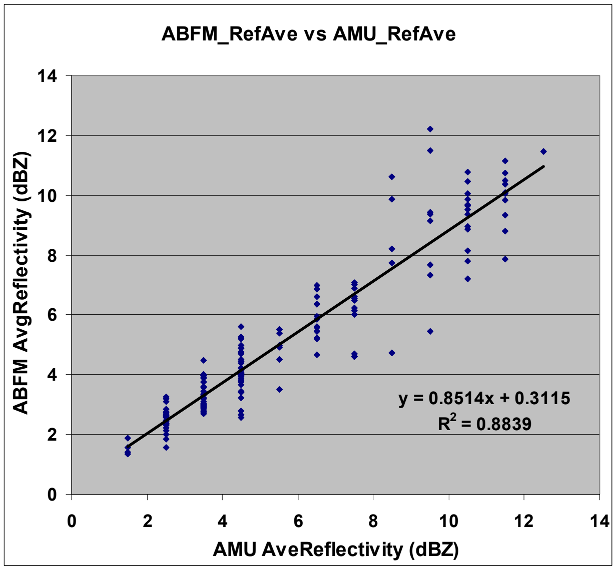


Figure 11. ABFM Average Reflectivity versus AMU Average Reflectivity.

6.3.3 Average Cloud Bottom, Top, and Thickness Product Comparison

The scatter plots for the Average Cloud Bottom and Average Cloud Top products (Figures 12 and 13) show that the linear relationship between the Average Cloud Bottom products is a little better than in the Average Cloud Top products ($R^2 = 0.8501$ and 0.6875, respectively). However, the scatter plot for Average Cloud Bottom shows a number of large outliers, while the scatter plot for Average Cloud Top shows only a few large outliers. The mean AMU Average Cloud Top was 8.26 km, while the mean ABFM Average Cloud Top was 6.53 km, giving a positive bias in the AMU product of 26%. The mean AMU Average Cloud Bottom was 6.34 km, while the mean ABFM Average Cloud Bottom was 5.42 km, giving a positive bias in the AMU product of 17%. Note that some of the AMU Average Cloud Top values were not used in the comparison because they were in the highest data level. In the highest data level, the AMU Average Cloud Top is greater than 11 km. The impact of not including AMU Average Cloud Tops above 11 km is discussed further in the last paragraph of this section. At least part of the reason for the AMU positive bias in Average Cloud Top and Bottom can be explained by how the cloud top/bottom is calculated differently by the AMU and ABFM VAHIRR products. The AMU product uses only non-negative reflectivity values at or above the freezing level, while the ABFM product uses all non-negative reflectivity values. If the ABFM average cloud base is below the freezing level (i.e. 5 km), it is set to 5 km. If the ABFM average cloud top is below 5 km, then the VAHIRR calculations are undefined. This is discussed further in Section 6.4.3.

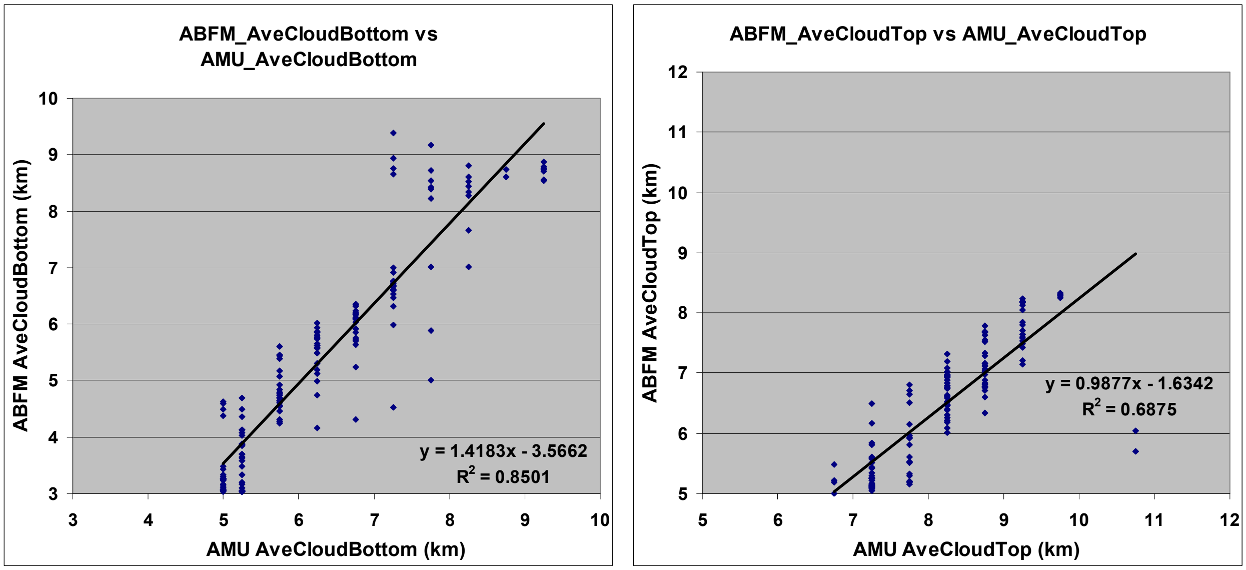


Figure 12. ABFM Average Cloud Bottom/Top versus AMU Average Cloud Bottom/Top.

All of the data points, including ones in which the AMU Average Cloud Top was greater than 11 km, were used in the comparison of Average Cloud Thickness. The scatter plot for the VAHIRR Average Cloud Thickness (Figure 13) shows a large amount of scatter across the linear regression line. The linear relationship between the two products is very weak, with R^2 = 0.1084. The mean AMU Average Cloud Thickness was 2.59 km and the mean ABFM Average Cloud Thickness was 2.10 km, giving a positive bias in the AMU product of 23%.

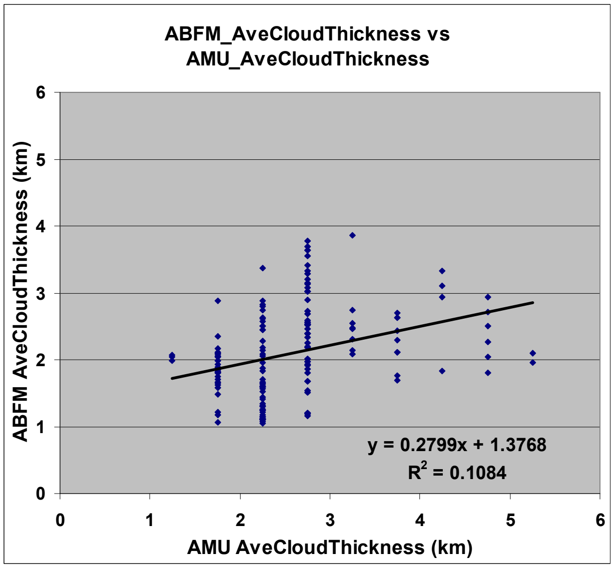


Figure 13. ABFM Average Cloud Thickness versus AMU Average Cloud Thickness.

Since the AMU Average Cloud Bottom and Average Cloud Top products have a positive bias of 17% and 26%, respectively, the positive bias in the AMU Average Cloud Thickness would be expected to be around 9%. However, in the ABFM VAHIRR product, the average cloud bottom and top are modified in order to calculate the average cloud thickness. In the ABFM program, the average cloud bottom was set to a minimum of 5 km, the estimated height of the freezing level. In order to take into effect the 1-km vertical grid spacing, 1 km was added to the average cloud thickness after subtracting the average cloud bottom from the average cloud top. The AMU's average cloud bottom and top have already been adjusted for the freezing level and vertical grid spacing.

Figure 14 compares the Average Cloud Bottom and Top in the AMU and ABFM products, after the ABFM cloud bottoms and tops have been modified to calculate the average cloud thickness. Figure 15 also compares the Average Cloud Bottom and Top products using line charts. In Figures 14 and 15, the ABFM Average Cloud Bottom is set to a minimum of 5 km, and 1 km is added to the ABFM Average Cloud Top in order to take into effect the 1-km vertical grid spacing. The scatter plots in Figure 14 show more of a one-to-one relationship between the AMU and ABFM Average Cloud Bottom/Top products in that the slopes of the linear regression lines are closer to 1 and the y-intercepts are closer to 0. The strength of the linear relationship in Average Cloud Top has not changed – the R^2 is still 0.6875. The linear relationship in Average Cloud Bottom weakened slightly – the R^2 decreased from 0.8501 to 0.7787. The mean ABFM Average Cloud Bottom has increased to 5.95 km, giving a positive bias in the AMU product of 7%. The mean ABFM Average Cloud Top has increased to 7.53 km, giving a positive bias in the AMU product of 10%. This would now imply a AMU positive bias in Average Cloud thickness around 3% (10% - 7%).

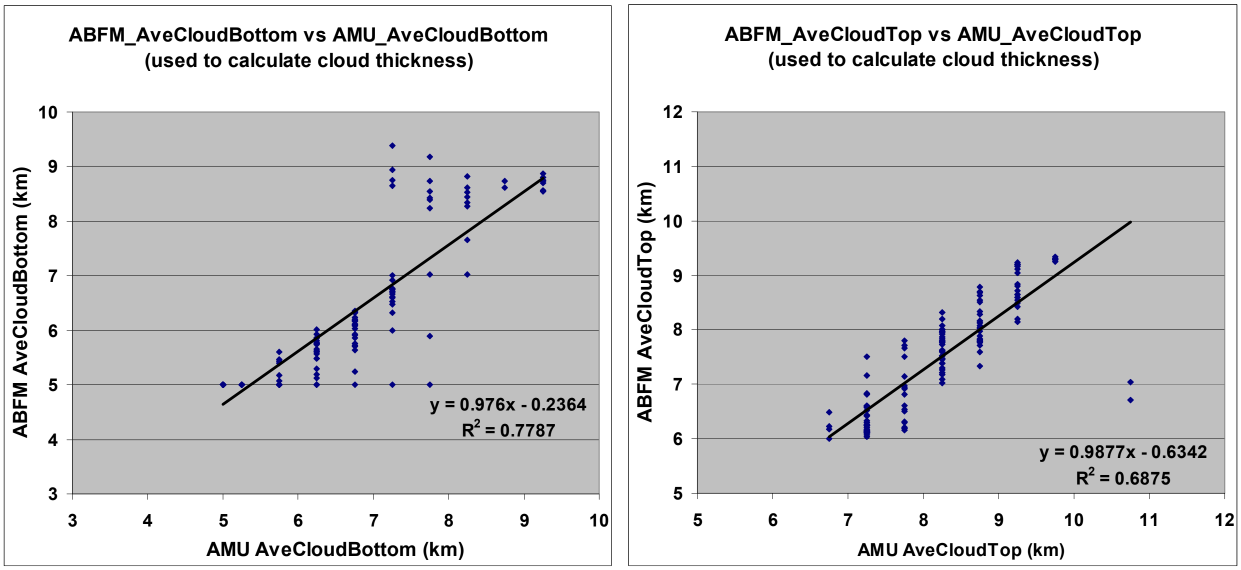


Figure 14. Scatter plots comparing the Average Cloud Bottom (left) and Average Cloud Top (right) used by the ABFM program and AMU to calculate Average Cloud Thickness. The ABFM Average Cloud Bottom is set to a minimum of 5 km, and 1 km is added to the ABFM Average Cloud Top to take into account the 1-km vertical grid spacing.

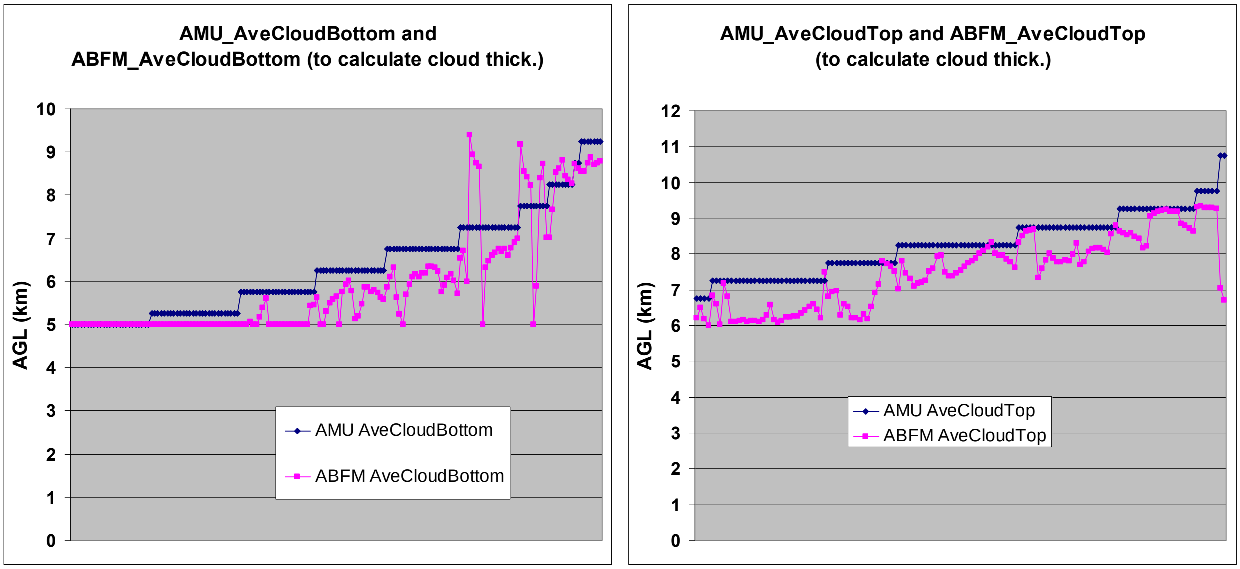


Figure 15. Line plots comparing the Average Cloud Bottom (left) and Average Cloud Top (right) used by the ABFM program and AMU to calculate Average Cloud Thickness. The ABFM Average Cloud Bottom is set to a minimum of 5 km, and 1 km is added to the ABFM Average Cloud Top to take into account the 1-km vertical grid spacing. The ABFM Average Cloud Bottom/Top are plotted as a function of increasing AMU Average Cloud Bottom/Top.

Figures 16 and 17 are scatter plots and line graphs that compare the Average Cloud Top in the AMU and ABFM products when data points with AMU values > 11 km are included. The AMU values > 11 km are estimated to be 11.5 km. The accuracy of this estimate is unknown. The mean AMU Average Cloud Top increases to 8.85 km and the ABFM Average Cloud Top increases to 8.05 km, giving a positive bias in the AMU product of 10%. This still gives an estimated AMU positive bias in Average Cloud Thickness of 3%, much less than the actual AMU positive bias in Average Cloud Thickness of 23% (second paragraph of this section). There are three possible explanations for the difference between the estimated and reported bias in Average Cloud Thickness:

- For AMU Average Cloud Top values > 11 km, the estimated value of 11.5 km may be too low. If the estimated value was instead 13.5 km (for example), the AMU positive bias in Average Cloud Top increases from 10% to 14%,
- The four AMU derived products are 16 data level products, which introduces measurement errors due to the ranges in values for each data level. This should be a small source of error, given the relatively large data set, and
- The ABFM Average Cloud Thickness may not be exactly equal to the equation (average cloud top – average cloud bottom) + 1 km, where the average cloud bottom is set to a minimum of 5 km. To verify this, the ABFM Average Cloud Bottom and Top values were converted to Average Cloud Thickness (after setting the Average Cloud Bottom to a minimum of 5 km and adding 1 km to the Average Cloud Top). Figure 18 shows there is very little difference between the reported ABFM Average Cloud Thickness and the ABFM Average Cloud Thickness calculated from the ABFM Average Cloud Bottom and Top.

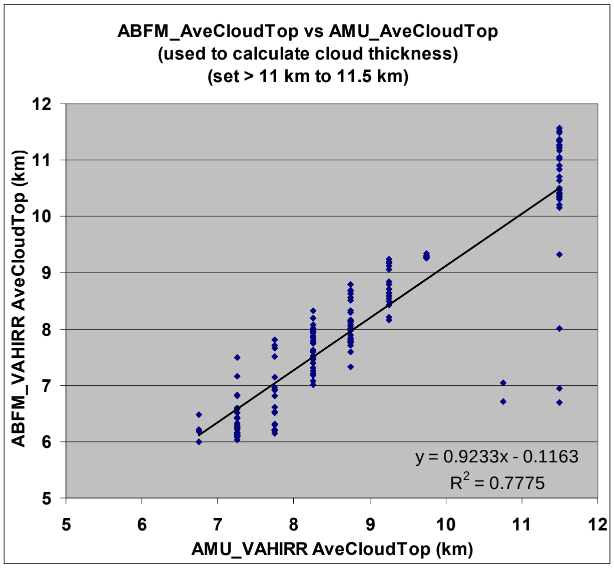


Figure 16. Scatter plot comparing the ABFM and AMU Average Cloud Tops used to calculate Average Cloud Thickness. The ABFM Average Cloud Bottom is set to a minimum of 5 km, and 1 km is added to the ABFM Average Cloud Top to take into account the 1-km vertical grid spacing. If the AMU Average Cloud Top is greater than 11 km, it is estimated to be 11.5 km.

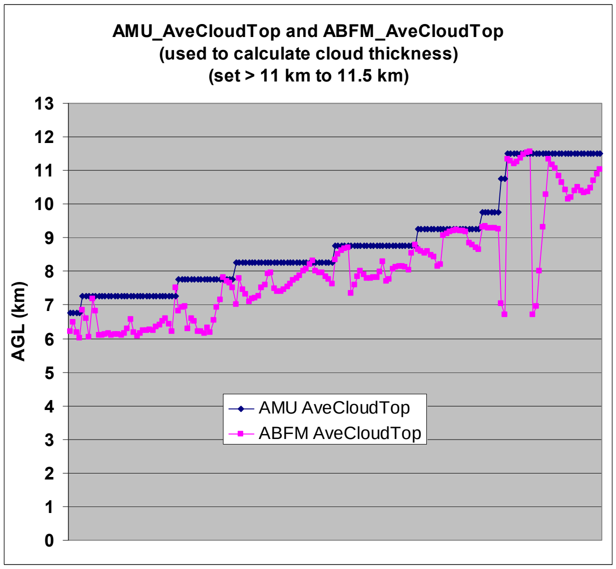


Figure 17. Same as Figure 16, only a line plot.

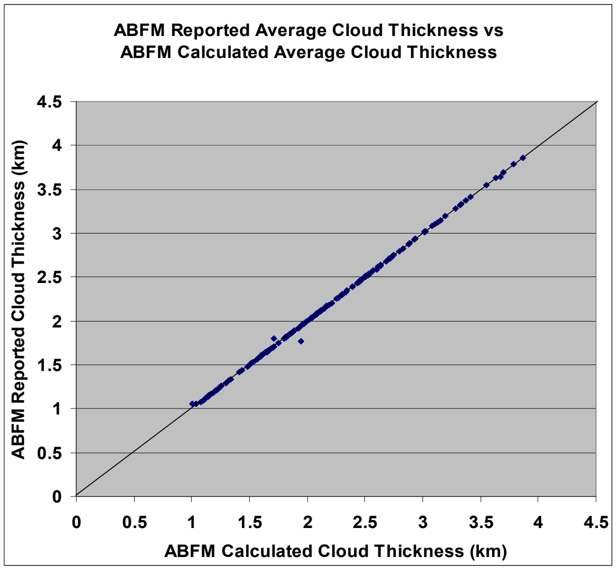


Figure 18. ABFM Average Cloud Thickness values versus Average Cloud Thickness calculated from the ABFM Average Cloud Bottom and Top. The Average Cloud Bottom is set to a minimum of 5 km and 1 km is added to the Average Cloud Top.

6.3.4 Minimum Fractional Coverage Comparison

If VAHIRR values are only included when the fractional coverage of non-negative reflectivity in the 11 x 11 grid point set is 5% or greater, then the linear relationships between the AMU and ABFM products improves a little. This could be due to better sampling of radar data. As an example, Figure 19 shows the comparison between the VAHIRR products with and without a minimum fractional coverage requirement of 5%.

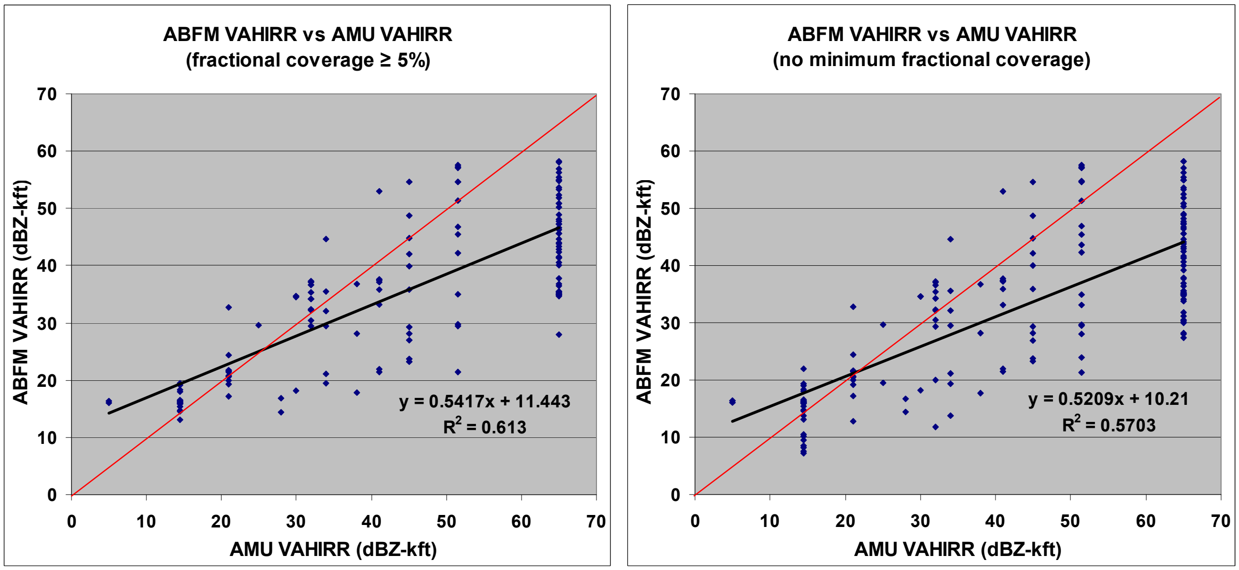


Figure 19. The AMU and ABFM VAHIRR products compared with (left) and without (right) a minimum fractional coverage of 5%. The linear regression line is displayed in black and the one-to-one relationship line is in red.

6.4 Sources of Error in the ABFM Comparison Test

Comparing the VAHIRR derived products did not resolve the differences between the AMU and ABFM products. Several possible sources of error were identified in the two products that could account for their differences:

- There may be errors in the latitude/longitude (lat/lon) positions of the ABFM aircraft or AMU VAHIRR values. This is discussed in Section 6.4.1.
- There may be errors in the calculation of cloud heights in the AMU product. This is discussed in Section 6.4.2.
- There are differences in how the AMU and ABFM products derive cloud bottoms and tops. This is discussed in Section 6.4.3.
- The differences in vertical grid spacing may be creating differences in the AMU and ABFM VAHIRR values. This is discussed in Section 6.4.4.

6.4.1 Possible Errors in the VAHIRR Lat/Lon Positions

The accuracy of the ABFM Comparison Test depends on the accuracy of the lat/lon positions in the two products. In the ABFM, the VAHIRR product was only calculated along the aircraft flight track, in order to compare the VAHIRR values and other radar-derived parameters to aircraft observations of cloud microphysics and electric field magnitude. The AMU VAHIRR product, on the other hand, displays VAHIRR values at a 1-km horizontal resolution across the entire radar coverage area. In order to perform the ABFM Comparison Test, the lat/lon of the aircraft and ABFM VAHIRR values at 10-second increments were recorded. The ABFM VAHIRR values were based on the most recent volume scan. The AMU and ABFM VAHIRR products were produced from Level II data, available from the National Climatic Data Center (http://www.ncdc.noaa.gov/nexradinv). Based on the same volume scan, AMU and ABFM VAHIRR values were compared at the same lat/lon locations. In the AMU VAHIRR product, the lat/lon was obtained from the D2D application in AWIPS.

The following three possible sources of error in the lat/lon positions were identified: position and time of the aircraft data (Section 6.4.1.1), lat/lon readout in AWIPS (Section 6.4.1.2), and the two products' conversion of radar data from Polar to Cartesian coordinates (Section 6.4.1.3). A discussion of two tests that were performed to compare the lat/lon differences in the two products is provided in Section 6.4.1.4.

6.4.1.1 Position and Time of the Aircraft Data

In order to do an accurate comparison of the products, the lat/lon and time data for the ABFM aircraft must be accurate. A VAHIRR value covers a horizontal area of 1 km^2. The precision of the recorded lat/lon in the ABFM program was in 10^{-3} degrees. At the lat/lon of the KMLB WSR-88D, a distance of 1 km is approximately 32 seconds of latitude and 37 seconds of longitude, or 0.009° latitude and 0.01° longitude (calculated from http://www.movable-type.co.uk/scripts/latlong.html). In other words, a distance of 1 km translates into approximately 0.01° lat/lon. The aircraft position was measured by an Applanix POS-AV strap-down gyro system with an integrated global positioning system (Dye et al. 2004). The reported response time of the position data is 10 milliseconds, the accuracy is 0.1 km, and the precision is 1 m. Therefore, the aircraft position data do not appear to be a significant source of error.

The aircraft and radar data in the ABFM program were recorded in 10-second intervals. The time of the data is important because it determines the most recent volume scan. The ABFM and AMU VAHIRR data must be compared using the same volume scan. The time accuracy of the ABFM data is unknown. However, it should not be a significant source of error since none of the data points in the ABFM Comparison Test were within 10 seconds of a volume scan time.

6.4.1.2 Lat/Lon Readout in AWIPS

The lat/lon of a location can be reported in AWIPS using the lat/lon readout in the D2D application. As an example, Figure 20 shows the lat/lon readout in a base reflectivity product. The precision of the lat/lon is in 10^{-2} degrees. The AMU could not determine how the AWIPS software calculates lat/lon, nor its accuracy. However, the precision of the readout is close to 1 km. This means that in AWIPS, the same lat/lon value could be reported by multiple adjacent VAHIRR values. Also, a VAHIRR value may have multiple lat/lon values across it, as is demonstrated in Figure 21. Although the lat/lon readout in AWIPS can create large errors in individual comparisons, it should not produce a significant bias in a large data set. As much as possible, the ABFM Comparison Test tried to avoid large horizontal gradients in VAHIRR values. Even when there was a weak horizontal gradient in VAHIRR values, there was a significant positive bias in the AMU VAHIRR product.

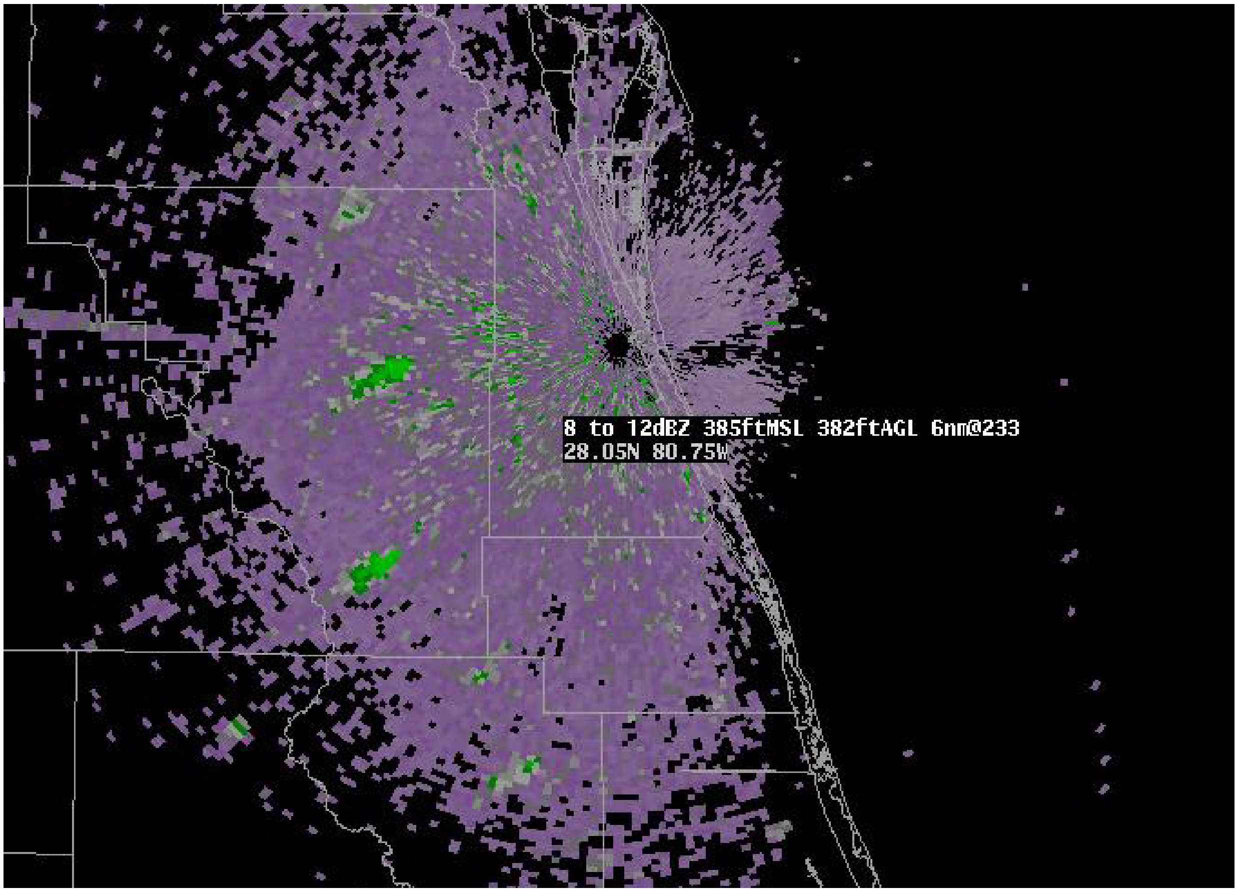


Figure 20. A lat/lon readout of 28.05° North/80.75° West is shown in AWIPS for the current cursor position on a base reflectivity radar product.

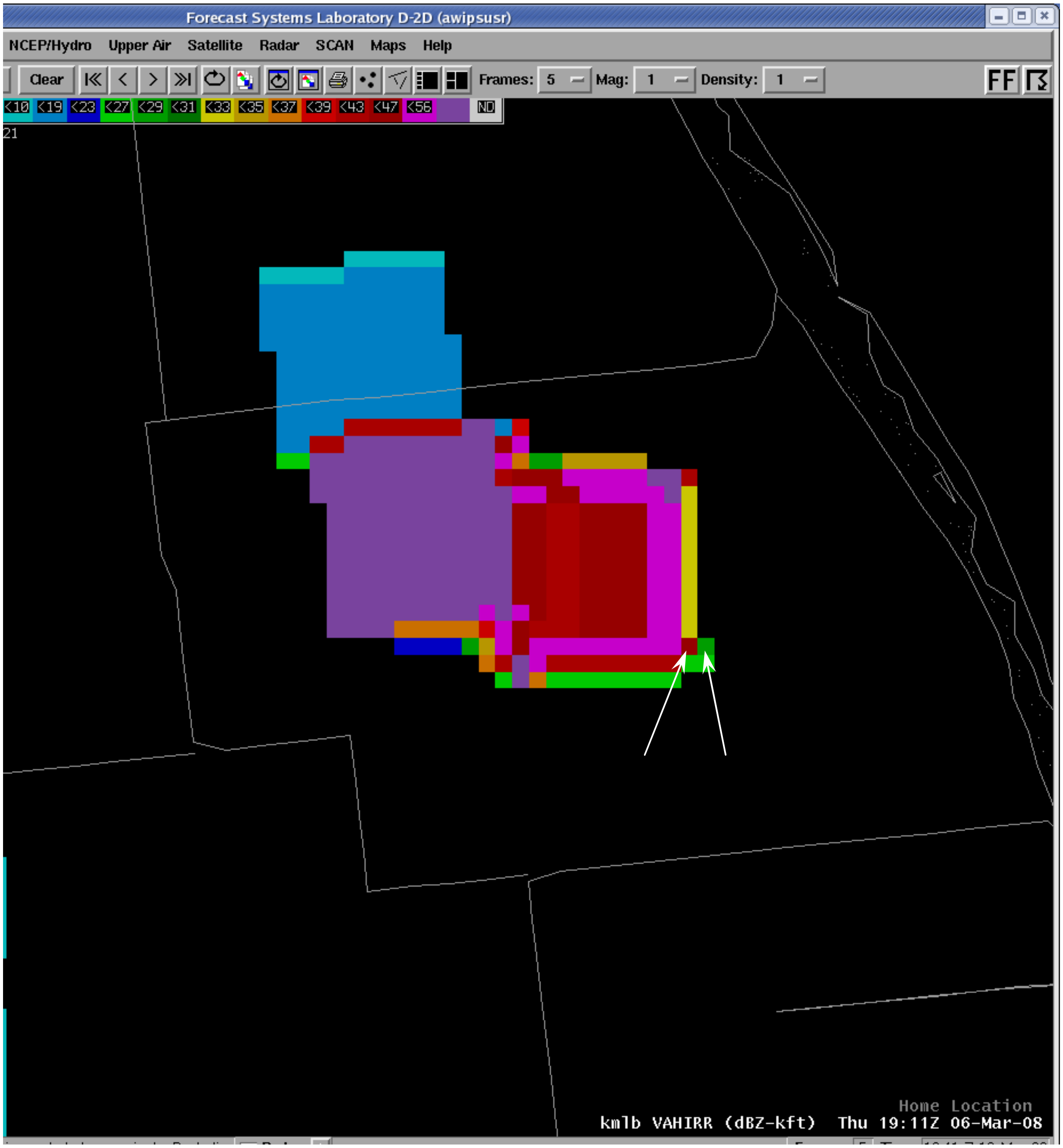


Figure 21. An AMU VAHIRR product from the KMLB WSR-88D radar volume scan at 1911 UTC on 6 March 2008. The left arrow points to a VAHIRR value (dark red) of 43-46 dBZ-kft, while the right arrow points to a VAHIRR value (medium green) of 27-28 dBZ-kft. In the left VAHIRR value, the latitude readout is a constant 27.67° North. However, the longitude ranges from 80.57° West to 80.56° West. In the right VAHIRR value, the latitude readout is also constant at 27.67° North, while the longitude ranges from 80.56° West to 80.55° West.

6.4.1.3 Conversion from Polar to Cartesian Coordinates

The AMU and ABFM VAHIRR products both had to convert the Level II data from Polar to Cartesian coordinates. In the ABFM, the radar data were converted to a Cartesian grid using the National Center for Atmospheric Research Mesoscale and Microscale Meteorology Division Sorted Position Radar INTerpolation (MMM-SPRINT) software (Dye et al. 2004). The SPRINT software can be downloaded at: http://box.mmm.ucar.edu/radarcode/; documentation is at: http://box.mmm.ucar.edu/pdas/Postscript/sprint_doc.ps.

The SPRINT software created a horizontal grid domain of 225 km x 225 km with 1-km grid spacing. The vertical grid spacing was also 1 km, and the data ranged from 1 to 20 km in altitude. The reflectivity at each 1-km grid point was determined using bilinear interpolation, in which the reflectivity measurements in polar coordinates were mapped to Cartesian coordinates. Bilinear interpolations are used to interpolate functions of two variables on a regular grid (http://en.wikipedia.org/wiki/Bilinear_interpolation). Linear interpolation is done in one direction (e.g. along the x-axis), then in the other direction (e.g. along the y-axis). If a grid point in polar coordinates was not within 0.2 km of a 1-km Cartesian grid point, then it was ignored. Since the KMLB WSR-88D and the PAFB WSR-74C radars were used in the ABFM, the PAFB radar was used as the origin of all radar data. The KMLB radar data were translated so that the origin for the gridded data was at the PAFB radar. The interpolation or rounding errors of the conversion from Polar to Cartesian coordinates have not been estimated.

Unlike the ABFM, the AMU did not convert from Polar to Cartesian coordinates prior to creating the VAHIRR product. The AMU automated VAHIRR software creates a two-dimensional array with 459 rows and columns, corresponding to a horizontal domain of 459 km x 459 km. When a new volume scan is read, each of the 459 x 459 VAHIRR values in the array is initialized to 0.0, along with the intermediate values used to calculate the final VAHIRR values. The VAHIRR software also initially assumes that each point in the array is “qualified”. Only qualified points can have a VAHIRR value calculated for them. Whenever one of the three disqualifying events (lowest elevation scan above freezing level, non-negative reflectivity in highest elevation scan, or in the cone-of-silence) is encountered, the point is disqualified. Whenever a final or intermediate AMU VAHIRR value needs to be calculated on the Cartesian grid, the radar value is translated from Polar to Cartesian coordinates using the ORPG-clone’s “RPGCS_azranelev_to_xy_u” built-in function described in Section 3 of Appendix A. The horizontal grid point spacing in Polar coordinates increases with distance from the radar due to beam spreading. On the other hand, the grid point spacing in Cartesian coordinates is constant. If the Polar grid point spacing is smaller than 1 km, then more than one Polar grid point may map to the same Cartesian grid box. This will occur more frequently closer to the radar, where the Polar grid point spacing is smaller. This should not create any problems because the sample size is included in the VAHIRR calculations. On the other hand, if the Polar grid spacing is larger than 1 km, then a Cartesian grid box may not have any Polar grid points mapped to it. The effect of the Polar grid spacing should be reduced due to the fact that the final VAHIRR value is the average of an 11 x 11 Cartesian grid point set.

6.4.1.4 VAHIRR Location Tests

In order to estimate possible lat/lon differences between the AMU and ABFM products, the AMU performed the following two VAHIRR location tests:

- Pick a particular feature in the radar reflectivity, then compare its lat/lon in the AMU and ABFM products. This test only verifies that the raw reflectivity data for the two products have the same position data. Table 4 shows the results, which used ABFM data from 12 June 2000. The position differences for all of the data points were less than 2 km. Therefore, this test showed that the lat/lon of the raw reflectivity data was accurate.
- Overlay the AMU VAHIRR product on top of the Composite Reflectivity product and the radar reflectivity for all of the elevation scans (from 0.5^{o} to 19.5^{o}). The VAHIRR values should be correlated with the reflectivity values at or above the freezing level. This test was performed for three volume scans:
  - 2055 UTC on 2 June 2001,
  - 2232 UTC on 4 June 2001, and
  - 1836 UTC on 5 June 2001.

  This test showed that the AMU VAHIRR product appears to be “off-center” by around 2 km, which is not considered to be a significant amount of error.

Table 4. Results of the first VAHIRR location test. The CAPPI Height is the height of the Constant Altitude Plan Position Indicator (CAPPI) reflectivity product in the ABFM. There is no CAPPI product in Build 8 of the ORPG-clone, so the nearest reflectivity product in height was used for the comparison. The AMU lat/lon values were obtained from AWIPS.

Time (UTC)	**ABFM Lat.**	**ABFM Lon.**	**AMU Lat.**	**AMU Lon.**	**CAPPI Height**	**AMU elevation angle of refl. product**	**AMU height of feature in refl. product**	**Distance between AMU and ABFM**	**Desc. Of the feature**
153000	29.032N	81.029W	29.04N	81.03W	4 km (13.1 kft)	1.5°	11.7 kft	0.9 km	center of a rain cell
153600	29.098N	81.004W	29.09N	81.02W	4 km (13.1 kft)	1.5°	12.3 kft	1.8 km	aircraft exiting out of cell
154050	29.057N	81.050W	29.07N	81.04W	4 km (13.1 kft)	1.5°	12.2 kft	1.7 km	center of a rain cell
154740	29.087N	81.014W	29.08N	81.03W	4 km (13.1 kft)	1.5°	12.3 kft	1.7 km	center of a rain cell
155300	29.079N	81.048W	29.09N	81.05W	4 km (13.1 kft)	1.5°	12.4 kft	1.2 km	center of a rain cell
160130	29.108N	81.019W	29.11N	81.03W	4 km (13.1 kft)	1.5°	12.7 kft	1.1 km	center of a rain cell

6.4.2 Possible Errors in AMU Cloud Height Calculations

To calculate the height of a cloud bottom or top in the AMU automated VAHIRR software, the ground range is first calculated from the radar elevation angle and the slant range. The ground range of a reflectivity gate is the distance from the radar along the ground, while the slant range is the distance from the radar along the radar beam. The ground range is later used to convert back to slant range, which is in turn is used to calculate gate height. For example, the formula to calculate cloud top is:

```
Height = Gate Height of the Highest Elevation Scan with non-negative reflectivity
Elevation Angle = Highest Elevation Scan with non-negative reflectivity
Upper Angle = Next Higher Elevation Angle
Upper Slant Range = Ground Range of Gate / cosine(Upper Angle * PI/180.0)
Upper Height = Gate Height of Upper Angle, calculated from Upper Angle and Upper Slant
               Range
Cloud Top = Height + (Upper Height – Height)/2
```

The calculation of the ground range does not take into account the curvature of the earth and standard radar propagation. For normal atmospheric conditions, the net effect of the earth's curvature and radar propagation will result in the radar beam increasing in height with distance at approximately 4/3 of the earth's radius (Section 3.4.1, Part B, FMH-11). This is illustrated in Figure 22. However, this value may be greater or less than 4/3, depending on the atmospheric profile of temperature and moisture. Because both the calculation of ground range from slant range, and calculation of slant range from ground range do not take into effect earth curvature and standard radar propagation, the errors may cancel each other out. The accuracy of the calculation of cloud bottom and top was not evaluated for the AMU VAHIRR product.

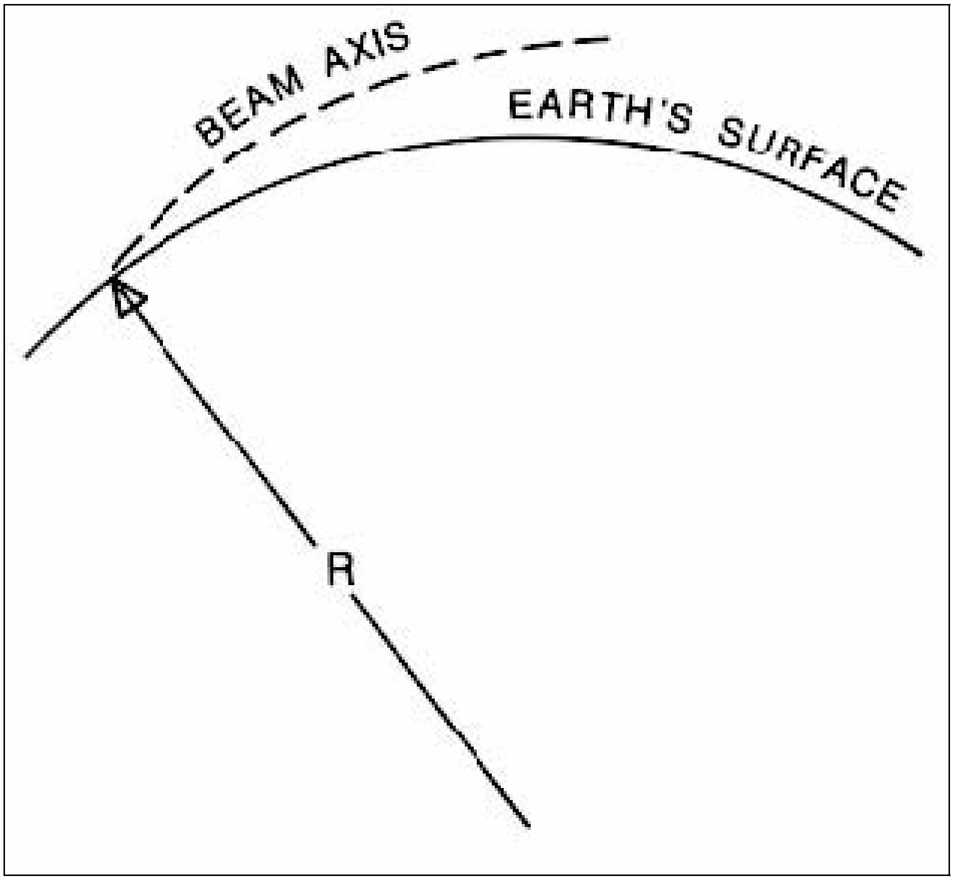


Figure 22. The radar beam axis due to the earth's curvature and standard atmospheric refraction. R is the radius of the earth. The figure is from FMH-11, Part B, Section 3.4.1.

6.4.3 Possible Errors in Deriving Average Cloud Bottom and Top

There are differences in how the AMU and ABFM derived the average cloud bottom and top for each VAHIRR value. The ABFM VAHIRR product used all non-negative reflectivity values to calculate the average cloud bottom and top, while the AMU VAHIRR product used only non-negative reflectivity values at or above the freezing level. This partly explains the positive bias in the average cloud bottom and top in the AMU product. In the definition of VAHIRR (see Section 1.2), the Volume-Averaged Radar Reflectivity and the Average Cloud Thickness are to be calculated from reflectivity values within the Specified Volume. The Specified Volume is defined as: "bounded in the horizontal and vertical planes, with perpendicular sides located 5.5 km north, east, south, and west of a point on the flight track, on the bottom by the 0° C level, and on the top by the upper extent of all clouds." Although the LAP developed VAHIRR from the ABFM data, the ABFM product and the LAP definition of VAHIRR are not identical, since the ABFM product included reflectivity values below the 0^o C level to calculate average cloud bottom and top. However, the LAP may be changing the VAHIRR definition to reflect the ABFM product (Dr. Francis Merceret, personal communication). The AMU and ABFM products calculated Average Reflectivity according to the definition in the LLCC governing the software development by only using non-negative reflectivity values at or above the freezing level.

6.4.4 Possible Errors due to Different Vertical Grid Spacing

The AMU and ABFM VAHIRR products both translated the radar data from Polar to Cartesian coordinates, although their implementations were different (see Section 6.4.1.3). The ABFM product used the SPRINT software to remap the radar data along the elevation scans to a constant 1-km vertical resolution. As an example, Figure 23 shows a vertical cross section of radar data from the ABFM program. Instead of using 1-km vertical grid spacing, the AMU product used the data along the elevation scans to calculate cloud tops, bottoms, and thicknesses. Since the vertical distance between elevation scans is variable, the vertical grid spacing in the AMU product is variable as shown in Figure 1. The vertical grid spacing also increases with distance from the radar. The current definition of VAHIRR does not state the required vertical grid spacing, so this becomes dependent on the implementation of the product.

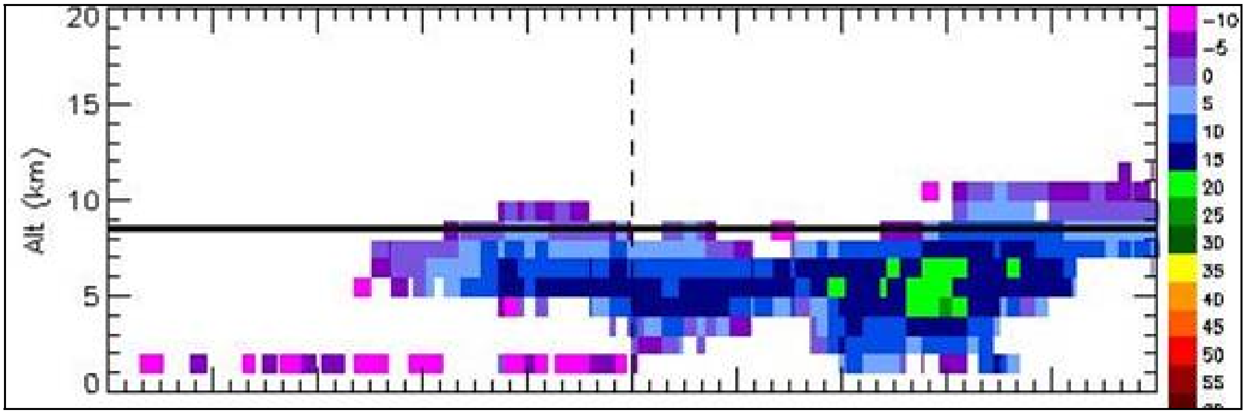


Figure 23. Vertical cross section of radar data from one of the ABFM case study days. The left-axis shows the altitude in km, while the right-axis shows reflectivity values in dBZ. From http://abfm.ksc.nasa.gov/.

If the distance between adjacent elevation scans is greater than 1 km, the AMU product may produce a positive bias in Average Cloud Thickness compared to the ABFM product. Conversely, if the distance between adjacent elevation scans is less than 1 km, the AMU product may produce a negative bias in Average Cloud Thickness. The AMU plotted the AMU-to-ABFM ratios for the VAHIRR and Average Cloud Thickness values to see if the ratios increased with distance from the radar. However, the data set in the ABFM Comparison Test did not support the belief that the ratios would increase with distance from the radar (figures not shown). The vertical grid spacing affects the calculation of cloud top and bottom, since the definition of VAHIRR requires that a half vertical grid space be added to the cloud top and subtracted from the cloud bottom. The AMU performed two tests to help determine how the vertical spacing affected the comparison between the AMU and ABFM products:

1. For clouds with a limited range of thicknesses (thicker clouds are preferred), calculate the ratios of Average Cloud Thickness, Average Reflectivity, and VAHIRR from the two products as a function of distance from the radar. Therefore, three variables were analyzed as a function of distance from the radar:

   $$\frac{\text{AMU Average Cloud Thickness}}{\text{ABFM Average Cloud Thickness}}$$

   $$\frac{\text{AMU Average Reflectivity}}{\text{ABFM Average Reflectivity}}\text{, and}$$

   $$\frac{\text{AMU VAHIRR}}{\text{ABFM VAHIRR}}$$

   For a fixed cloud thickness, the ratios of Average Cloud Thickness and VAHIRR should increase with distance from the radar due to radar beam spreading.

2. For clouds within a limited range of distances from the radar such that the vertical distance between elevation scans is significantly greater than 1 km, calculate the same three ratios as a function of Average Cloud Thickness. The ratios of Average Cloud Thickness and VAHIRR should decrease with increasing cloud thickness as long as the vertical radar beam spacing is significantly greater than 1 km. The reason is that there should be a greater positive bias in the AMU VAHIRR values for smaller cloud thicknesses.

6.4.4.1 First Vertical Grid Spacing Test

Keeping the Average Cloud Thickness relatively fixed, the ratios of Average Cloud Thickness, Average Reflectivity, and VAHIRR for the AMU and ABFM products were calculated as a function of distance from the radar. The test was performed for three thickness ranges: ABFM Average Cloud Thickness of 2 - 3 km and AMU

Average Cloud Thickness of 2.25 - 3.25 km and 3.25 - 4.25 km. The AMU expected that the ratios of Average Cloud Thickness and VAHIRR would increase with distance from the radar due to radar beam spreading. In the scatter plots shown in Figures 24-26, only VAHIRR values in which the fractional coverage of non-negative reflectivity was 5% or greater are included. However, including all VAHIRR values, regardless of coverage, usually had little effect on the linear relationships between the AMU and ABFM products. In addition, only ABFM values in which the observed average cloud bottom was 5 km or greater were included. Including all ABFM values regardless of observed average cloud bottom, usually had little effect on the linear relationships between the two products.

Figure 24 shows the Average Reflectivity ratios as a function of distance from the radar. As expected, the Average Reflectivity ratios changed little with distance from the radar. There are only a few large outliers in the plots.

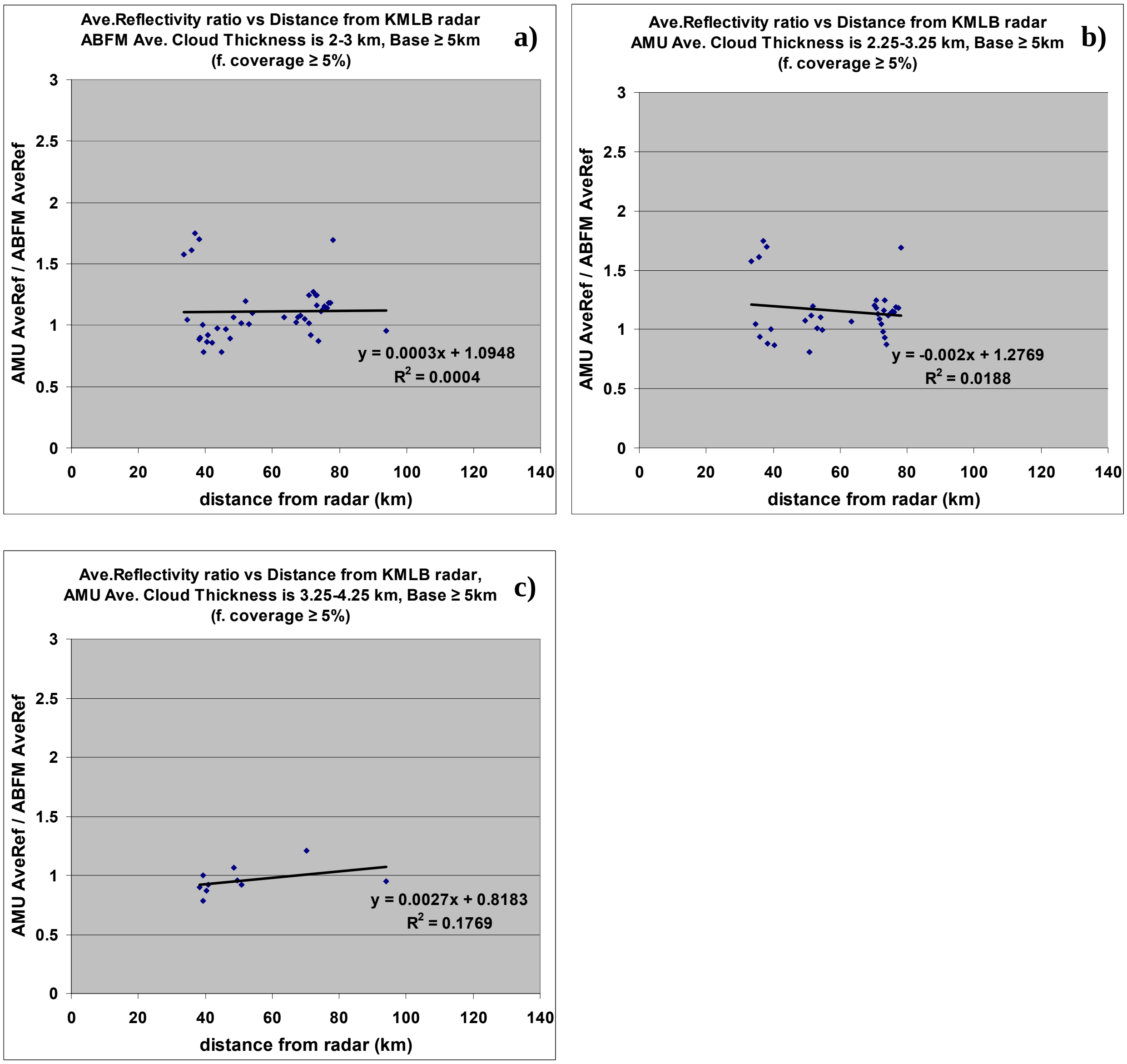


Figure 24. The Average Reflectivity ratios for the AMU and ABFM products as a function of distance from the radar: a) (upper-left) the ABFM Average Cloud Thickness is between 2 km and 3 km; b) (upper-right) the AMU Average Cloud Thickness is between 2.25 km and 3.25 km; and c) (lower-left) the AMU Average Cloud Thickness is between 3.25 km and 4.25 km.

Figure 25 shows the Average Cloud Thickness ratios as a function of distance from the radar. Although it was expected that the thickness ratio would increase with distance from the radar, this did not occur. In all three scatter plots, the linear regression line has a negative slope. The linear regression relationships in all three plots were weak. Comparing Figure 25b and c, the average thickness ratio with an AMU Average Cloud Thickness of 3.25 km - 4.25 km was larger than the ratio with an AMU Average Cloud Thickness of 2.25 km - 3.25 km. This is also contradictory to what was expected for the test results.

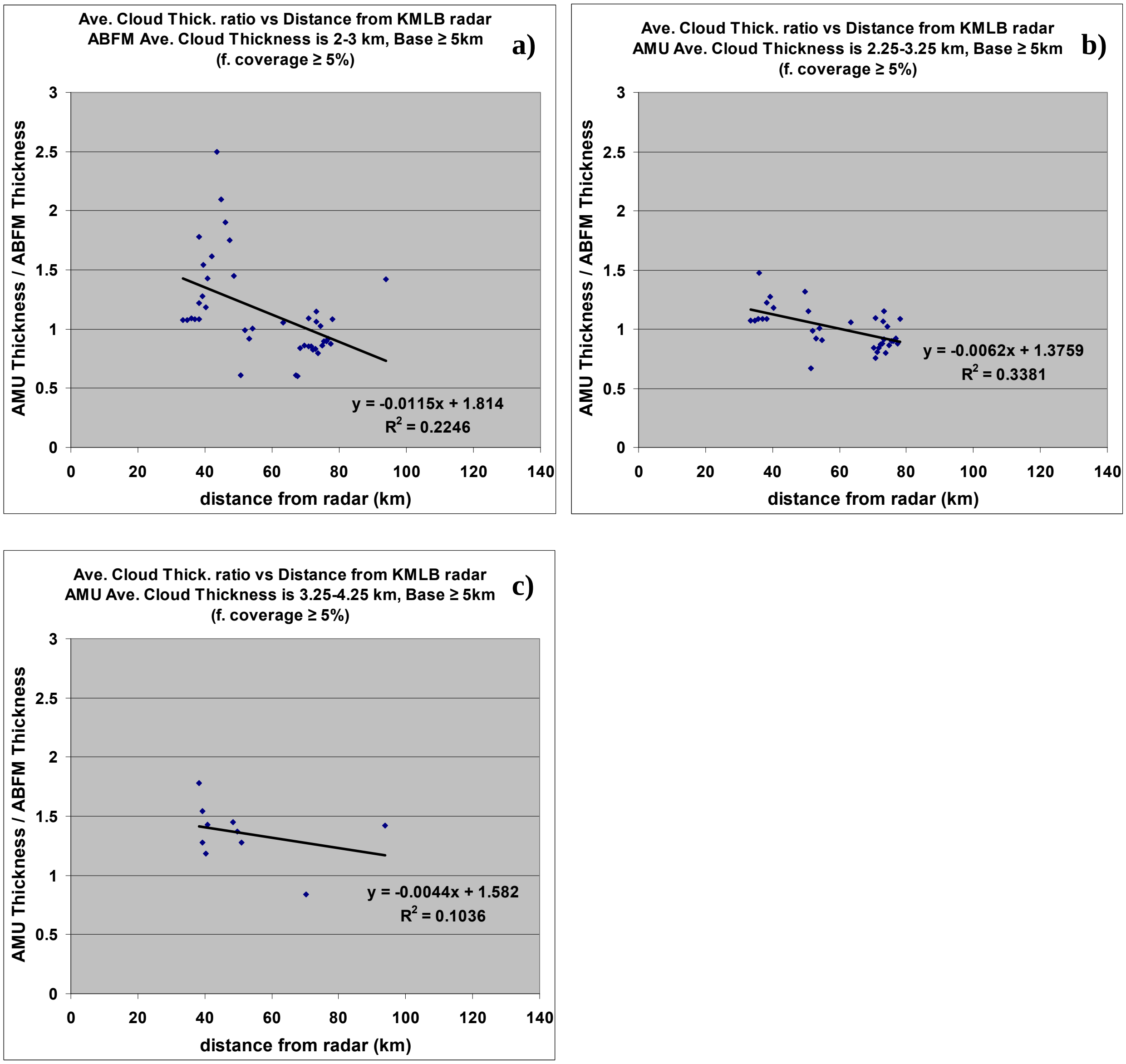


Figure 25. The Average Cloud Thickness ratios for the AMU and ABFM products as a function of distance from the radar: a) (upper-left) the ABFM Average Cloud Thickness is between 2 km and 3 km; b) (upper-right) the AMU Average Cloud Thickness is between 2.25 km and 3.25 km; and c) (lower-left), the AMU Average Cloud Thickness is between 3.25 km and 4.25 km.

Figure 26 shows the VAHIRR ratios as a function of distance from the radar. If a value was > 56 dBZ-kft, it was estimated to be 65 dBZ-kft. However, not including data points with AMU VAHIRR values > 56 dBZ-kft did not significantly affect the linear relationships. Although it was expected that the VAHIRR ratio would increase with distance from the radar, this also did not occur. In Figure 26a and b, the linear regression line has a negative slope. Figure 26c has a slight positive slope, however there are not many data points and the linear relationship is very weak. The linear regression relationships in Figure 26a and b are also weak.

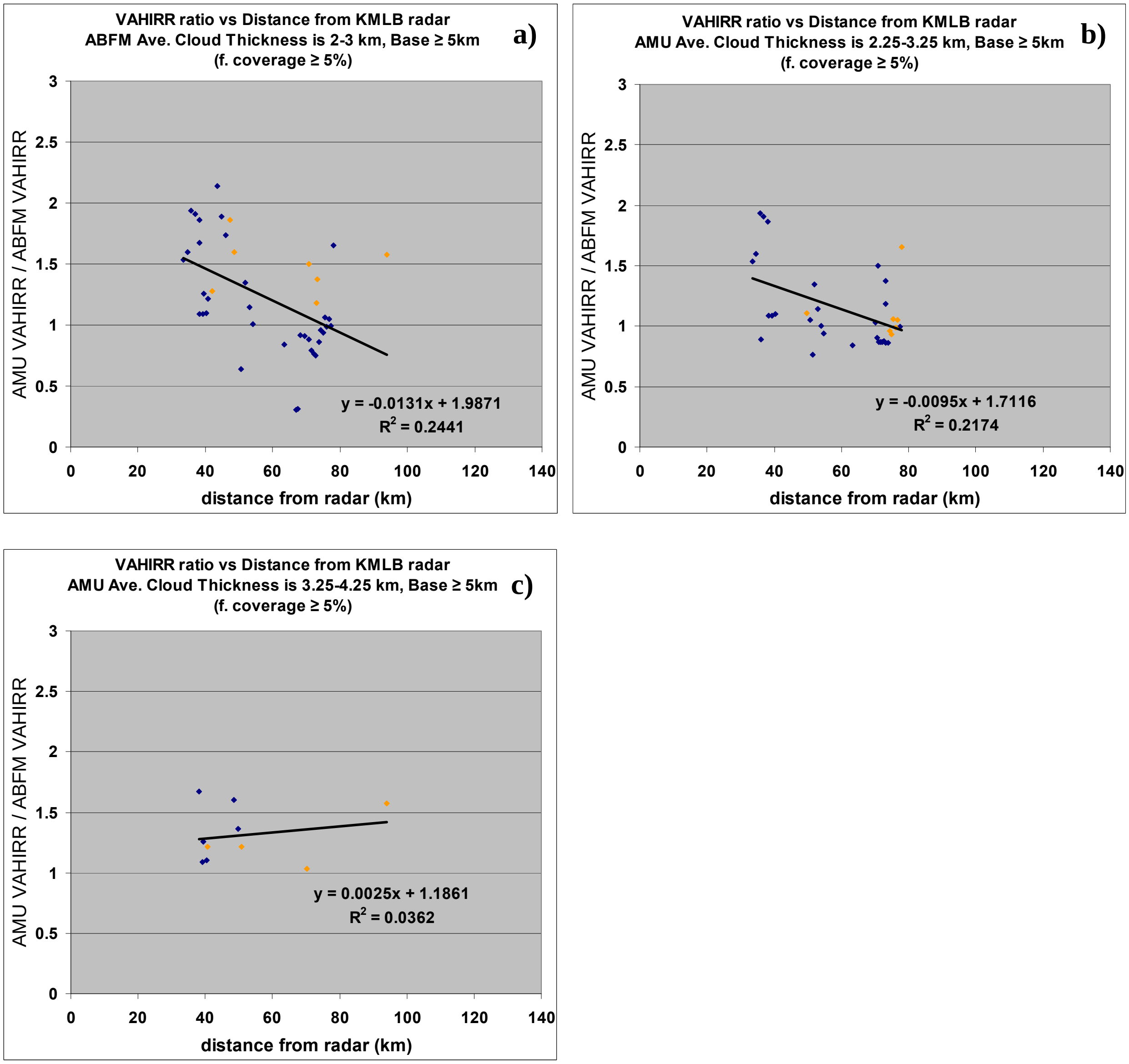


Figure 26. The Average Cloud Thickness ratios for the AMU and ABFM products as a function of distance from the radar: a) (upper-left) the ABFM Average Cloud Thickness is between 2 km and 3 km; b) (upper-right) the AMU Average Cloud Thickness is between 2.25 km and 3.25 km; and c) (lower-left) the AMU Average Cloud Thickness is between 3.25 km and 4.25 km. The AMU VAHIRR values greater than 56 dBZ-kft were estimated to be 65 dBZ-kft, and are displayed as orange dots.

6.4.4.2 Second Vertical Grid Spacing Test

For a limited range of distance from the radar, the ratios of Average Cloud Thickness, Average Reflectivity, and VAHIRR for the two products were calculated as a function of Average Cloud Thickness. The vertical distance between elevation scans should be significantly greater than 1 km, since this is the vertical grid spacing used by the ABFM. The AMU expected that the ratios of Average Cloud Thickness and VAHIRR would decrease with increasing Average Cloud Thickness, since there should be a greater positive bias in the AMU VAHIRR values for smaller cloud thicknesses. Just like the first test, only VAHIRR values in which the fractional coverage was 5% or greater were included. However, including all VAHIRR values usually had little effect on the linear relationships between the AMU and ABFM products. In addition, only ABFM values in which the observed average cloud bottom was 5 km or greater were included. Including all ABFM values, regardless of observed average cloud bottom, usually had little effect on the linear relationships between the two products.

The test was performed for three distance ranges from the radar: 35-45 km, 45-55 km, and 65-75 km. The AMU VAHIRR Average Cloud Tops greater than 11 km were estimated to be 11.5 km. The mean Average Cloud Bottom/Top for each distance was used to estimate the vertical spacing between elevation scans. Table 5 shows the vertical spacing between the elevation scans for the three distance ranges from the radar, assuming the radar is in VCP-11 mode (see Figure 1). The information contained in Figure 27 was used to estimate the height of the elevation scans at the different distances from the radar. As shown in the right-most column of Table 5, the average vertical distance between elevation scans was approximately 1.5 km.

Table 5. The vertical distance between adjacent elevation scans for three relatively fixed distances from the radar.

Range from Radar	**Midpoint of Range**	**Mean AMU Average Cloud Bottom**	**Mean AMU Average Cloud Top**	**Elevation Scans Surrounding Mean Average Cloud Bottom/Top**	**Average Vertical Distance between Adjacent Elevation Scans**
35 km – 45 km	40 km	7.6 km	10.9 km	10.0°, 12.0°, 14.0°, 16.7°	1.5 km
45 km – 55 km	50 km	7.1 km	10.0 km	7.5°, 8.7°, 10.0°, 12.0°	2.0 km
65 km – 75 km	70 km	6.5 km	8.7 km	4.3°, 5.25°, 6.2°, 7.5°	1.5 km

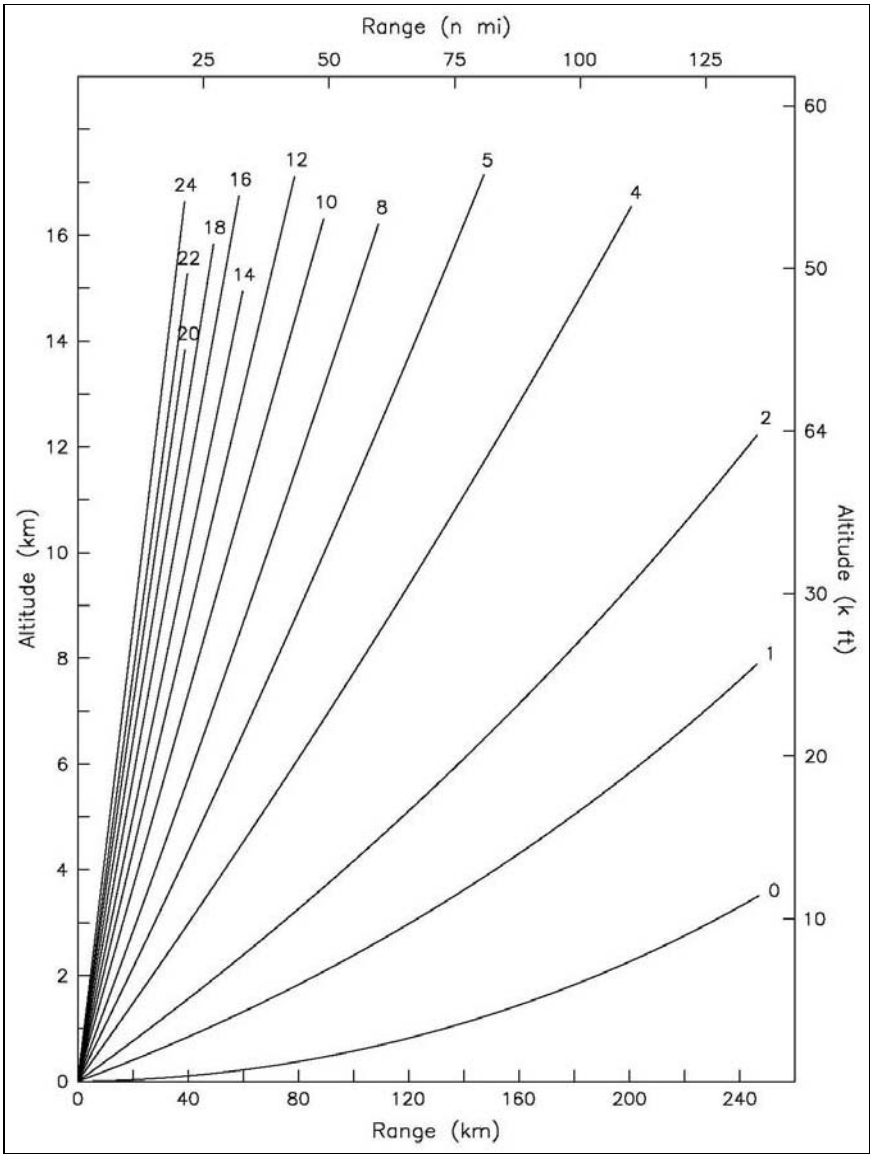


Figure 27. A Range-Radar Beam Altitude Nomogram showing the height of the radar beam as a function of distance from the radar and elevation scan angle. The figure is from FMH-11, Part B, Section 3.4.1.

Figures 28 and 29 show the Average Reflectivity ratios as a function of Average Cloud Thickness (only distance ranges of 35-45 km and 45-55 km are shown). As expected, the Average Reflectivity ratios generally changed little with Average Cloud Thickness, with only a few large outliers.

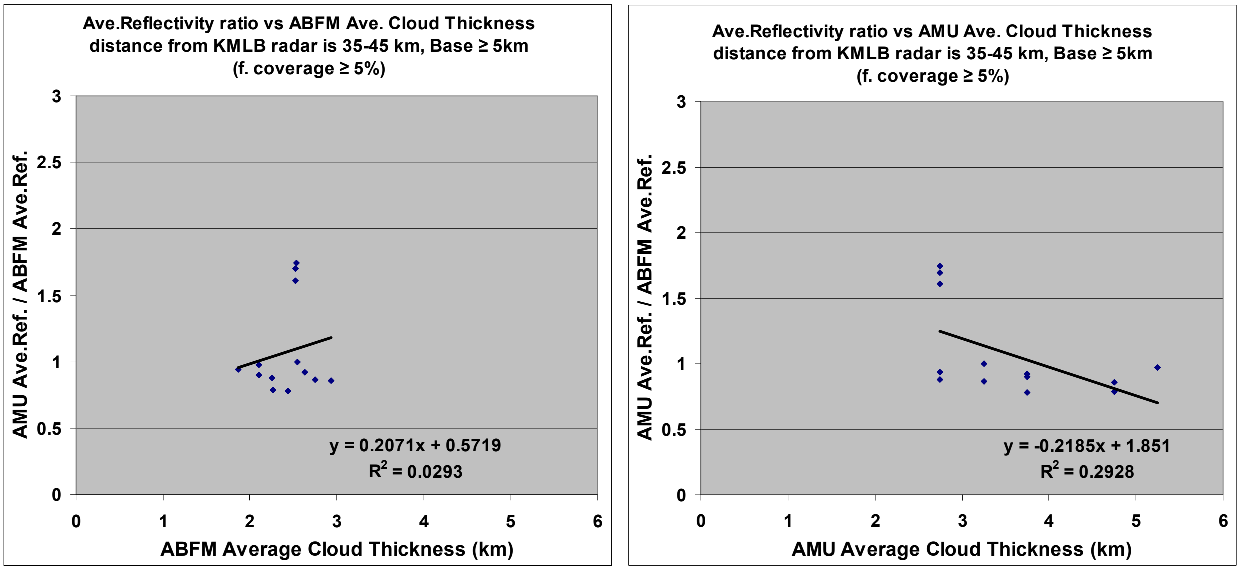


Figure 28. The Average Reflectivity ratio as a function of ABFM Average Cloud Thickness (left) and AMU Average Cloud Thickness (right). The data points are 35 - 45 km from the radar.

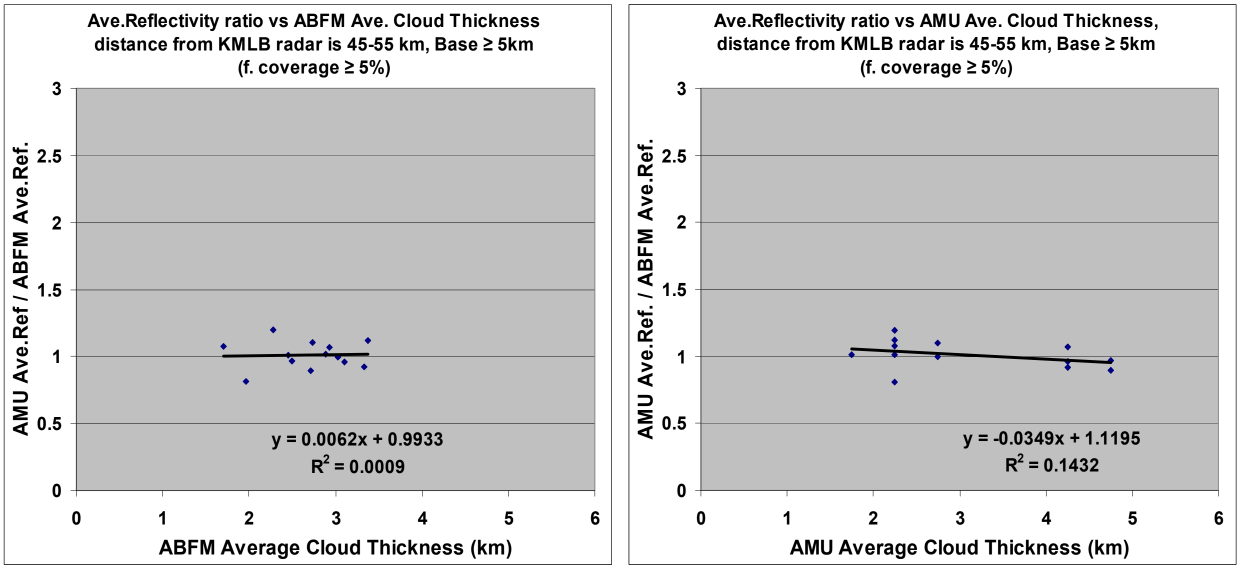


Figure 29. The Average Reflectivity ratio as a function of ABFM Average Cloud Thickness (left) and AMU Average Cloud Thickness (right). The data points are 45 - 55 km from the radar.

Figures 30 and 31 show the Average Cloud Thickness ratios as a function of Average Cloud Thickness (only distance ranges of 35-45 km and 45-55 km are shown). The scatter plots show that the Average Cloud Thickness ratios generally decreased with the ABFM Average Cloud Thickness and increased with the AMU Average Cloud Thickness. However, the linear relationships are weak. This can be explained in part because if the AMU Average Cloud Thickness is held constant, increasing the ABFM Average Cloud Thickness will cause the Thickness ratio to decrease. The results of the scatter plots do not appear to support the hypothesis that the Average Cloud Thickness ratio will decrease with increasing Average Cloud Thickness.

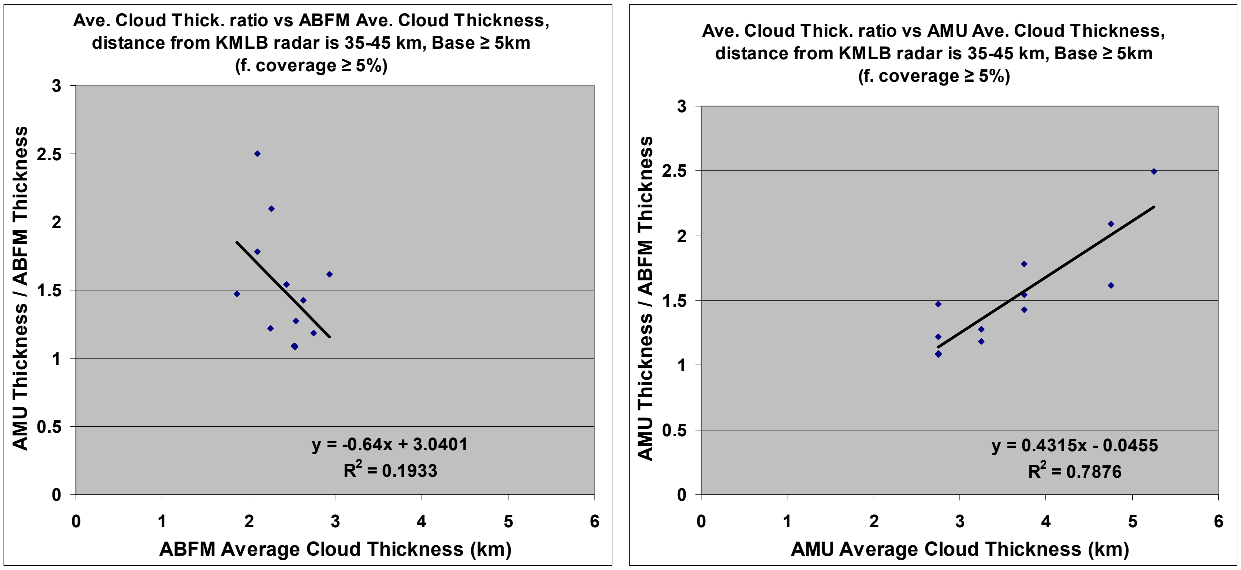


Figure 30. The Average Cloud Thickness ratio as a function of ABFM Average Cloud Thickness (left) and AMU Average Cloud Thickness (right). The data points are 35 km to 45 km from the radar.

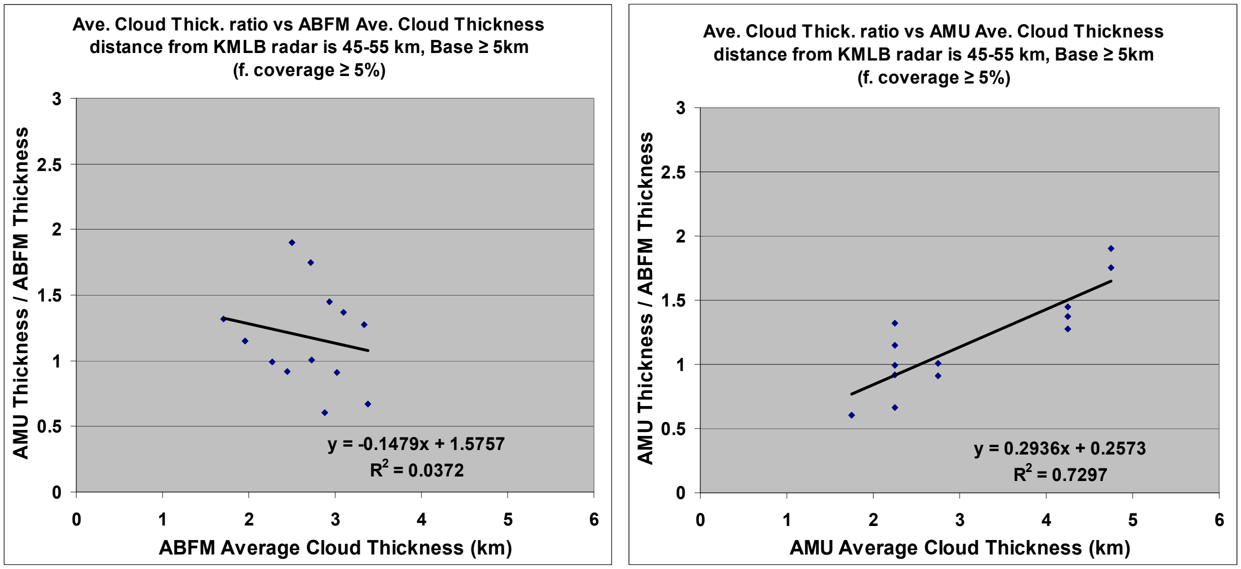


Figure 31. The Average Cloud Thickness ratio as a function of ABFM Average Cloud Thickness (left) and AMU Average Cloud Thickness (right). The data points are 45 km to 55 km from the radar.

Figures 32 and 33 show the VAHIRR ratios as a function of Average Cloud Thickness (only distance ranges of 35-45 km and 45-55 km are shown). The scatter plots for the VAHIRR ratios are very similar to that of the Average Cloud Thickness ratios. Once again, the ratios generally decreased with the ABFM Average Cloud Thickness and increased with the AMU Average Cloud Thickness. The results of the scatter plots therefore do not support the hypothesis that the VAHIRR ratio will decrease with increasing Average Cloud Thickness.

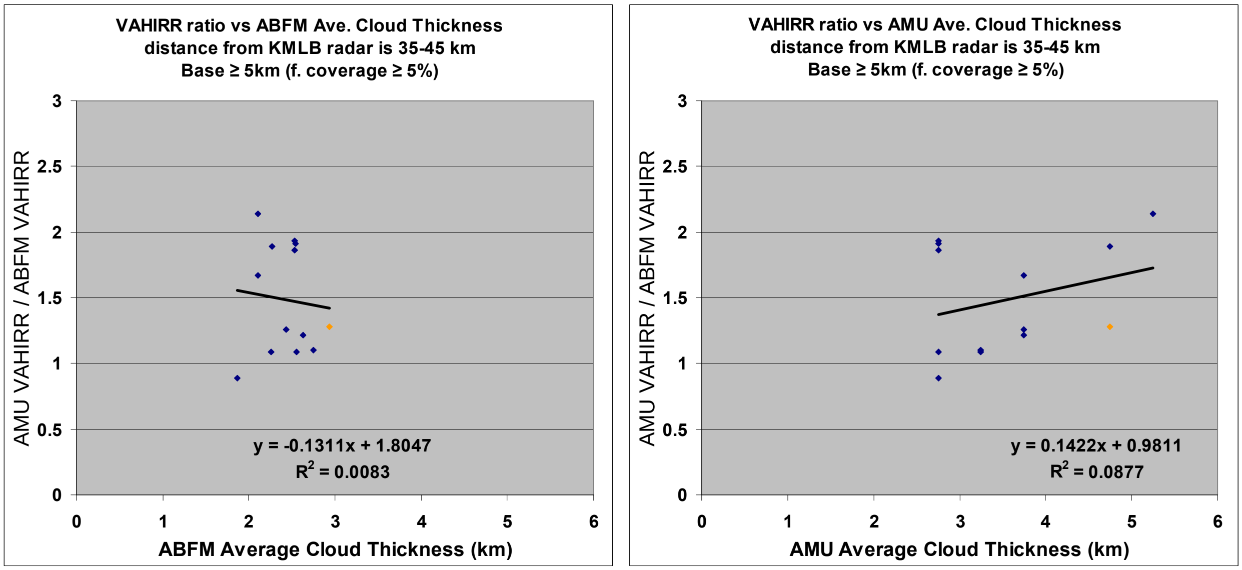


Figure 32. The VAHIRR ratio as a function of ABFM Average Cloud Thickness (left) and AMU Average Cloud Thickness (right). The data points are 35 km to 45 km from the radar. The AMU VAHIRR values > 56 dBZ-kft were estimated to be 65 dBZ-kft and are displayed in orange.

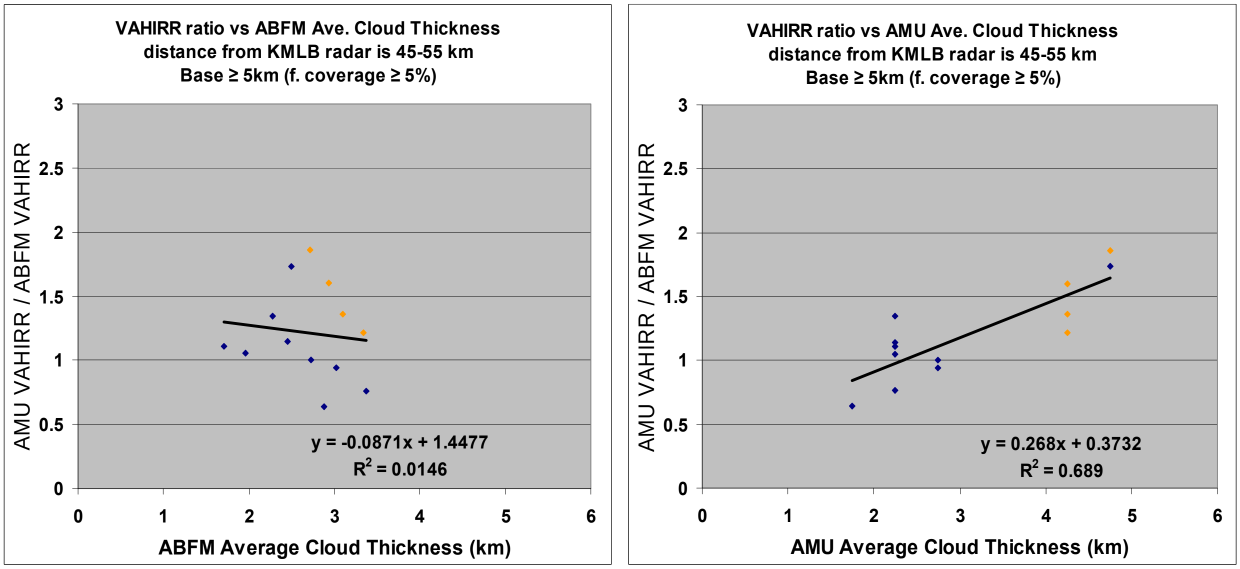


Figure 33. The VAHIRR ratio as a function of ABFM Average Cloud Thickness (left) and AMU Average Cloud Thickness (right). The data points are 45 km to 55 km from the radar. The AMU VAHIRR values > 56 dBZ-kft were estimated to be 65 dBZ-kft and are displayed in orange.

6.5 Conclusions of the ABFM Comparison Test

The AMU performed several tests to help determine the cause of the differences between the AMU and ABFM products found in the ABFM Comparison Test. The AMU determined several possible causes of the differences, but was not able to explain all of the differences. Therefore, the ABFM Comparison Test of the AMU automated VAHIRR software was considered a failure. Since the ABFM Comparison Test was considered vital in determining the accuracy of the product, the AMU automated VAHIRR software will not be released for operational use.

7. Summary

The LLCC are a set of rules used to avoid natural and rocket-triggered lightning strikes to space vehicles. The previous LLCC were shown to be overly restrictive, potentially leading to costly launch delays and scrubs. The VAHIRR parameter and the updated LLCC for anvil clouds were developed using data collected by the Airborne Field Mill II program conducted in 2000/2001. The use of the VAHIRR product is expected to lead to increased launch opportunities, while maintaining safety.

VAHIRR is defined as the product of the Volume-Averaged Radar Reflectivity and the Average Cloud Thickness within a Specified Volume relative to a point along the flight track of a space launch vehicle. The Specified Volume is bounded in the horizontal and vertical planes, with perpendicular sides located 5.5 km north, east, south, and west of a point on the flight track, on the bottom by the 0^o C level, and on the top by the upper extent of all clouds. The Volume-Averaged Radar Reflectivity is the arithmetic average (in dBZ) of the cloud radar reflectivity within the Specified Volume. Any grid point within the Specified Volume can only be included in the average if it has a radar reflectivity ≥ 0 dBZ. The Average Cloud Thickness is the altitude difference in km between the average top and the average base of the cloud within the Specified Volume. The cloud base to be averaged is the higher of (1) the 0^o C level and (2) the lowest extent in altitude of all cloud radar reflectivities 0 dBZ or greater. The cloud top to be averaged is the highest extent of all cloud radar reflectivities 0 dBZ or greater. Allowance must be made for the vertical separation of grid points. The cloud base at any horizontal position will be the altitude of the corresponding base grid point minus half of the grid-point vertical separation. Similarly, the cloud top at that horizontal position will be the altitude of the corresponding top grid point minus half of the grid-point vertical separation.

Currently, a manual work-around is used to calculate VAHIRR with existing radar products. The AMU was tasked to develop an automated version of the VAHIRR product for use on the WSR-88D, to reduce the operational impact of the anvil cloud LLCC. After the AMU completed development of the automated VAHIRR software, they tested it thoroughly for correctness and reliability. The software passed all the tests, except for the ABFM Comparison Test, which compared the AMU VAHIRR product to the ABFM VAHIRR product. The comparison test was over a limited range near the critical value of 33 dBZ-kft used in the LLCC. In the comparison test, data points were limited to ABFM VAHIRR values of 56 dBZ-kft or less. Large differences between the two products were found in the ABFM Comparison test. The following possible sources of error in comparing the two products were identified and evaluated:

- In the ABFM product, the calculation of VAHIRR from the Average Reflectivity and Average Cloud Thickness,
- The calculation of cloud heights and average cloud bottom/top in the two products,
- The calculation of Average Cloud Thickness from average cloud bottom and top in the two products,
- The effect of limiting the fractional coverage of non-negative reflectivity in the 11 x 11 grid point set for each VAHIRR value to 5% or greater,
- Errors in calculating lat/lon positions in the two products,
- Differences in the conversion from polar to Cartesian coordinates, and
- Differences in vertical grid spacing in the two products.

The AMU was unable to completely determine the cause of the differences between the two products. Therefore, the AMU automated VAHIRR software will not be released for operational use.

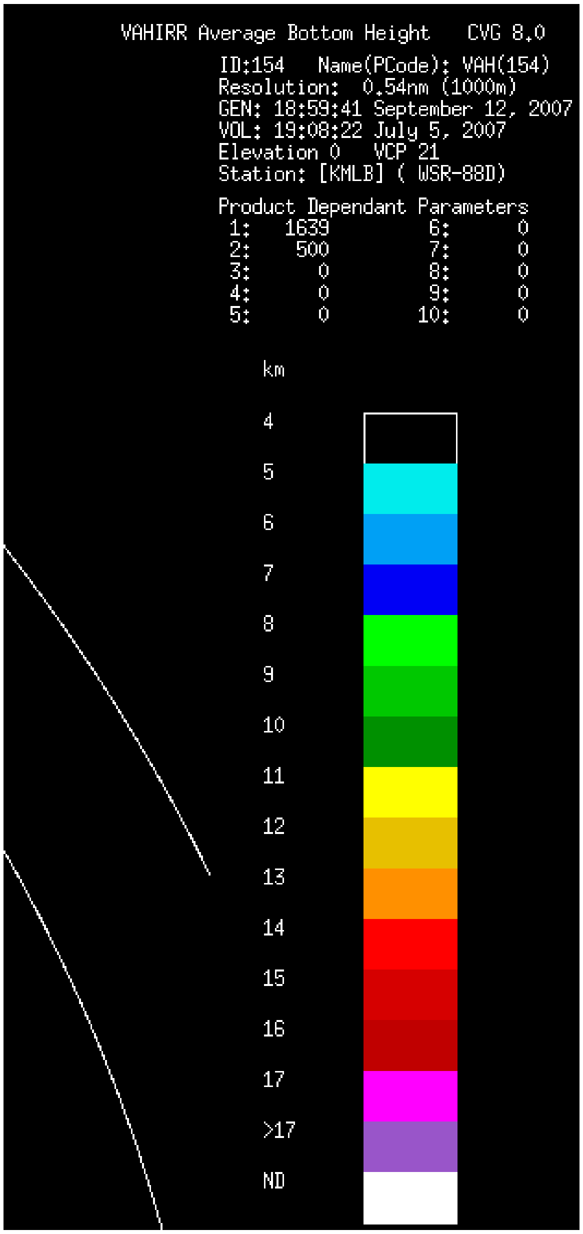
8. Future Work

Although the AMU work on the automated VAHIRR product did not result in an operational product, the lessons learned will be used on any future development and testing of an automated VAHIRR product. The results of the ABFM Comparison Test revealed that implementation differences may have resulted in large discrepancies between the AMU and ABFM VAHIRR products. Some recommendations for future development and testing of an automated VAHIRR product include the following:

- The AMU VAHIRR product is a 4-bit product. This limits the precision of the displayed product, since there are only 16 data levels. A better choice would be to create an 8-bit product, with 256 data levels.

In the current version, the disqualified data points are represented by data level 15 and displayed in white. There are three events that will disqualify a data point: lowest elevation scan is above the freezing level, the highest elevation scan has non-negative reflectivity, and the data point is inside the cone-of-silence. With an 8-bit product, these disqualifying events could be represented by three data levels, instead of one. Perhaps most importantly, it will be easier to compare an 8-bit product to the ABFM VAHIRR product;

- Unless the LAP changes the definition of VAHIRR, the automated VAHIRR product should be implemented in the same way as the ABFM product, so they result in the same values given the same input radar data. This includes calculating cloud heights, cloud bottoms, cloud tops, average cloud thickness, and vertical and horizontal grid spacing the same way;
- The comparison between the automated VAHIRR product and the ABFM VAHIRR product should be limited to areas in which the fractional coverage of non-negative reflectivity is ≥ 10%. This is to reduce the uncertainty of the VAHIRR calculation in highly broken clouds or regions in which radar elevation scan gaps exist;
- One of the difficulties in comparing the AMU and ABFM VAHIRR products was differences in how they were displayed. In the ABFM product, VAHIRR values were only displayed along the aircraft track. In the AMU product, VAHIRR values are displayed for the entire radar coverage area. Therefore, it would be helpful to run the ABFM VAHIRR source code for the entire radar coverage area;
- The automated VAHIRR product should be compared to manual calculations. This would be time-intensive, but would only require the manual calculation of a few data points;
- Comparing the two products in the ABFM Comparison Test was manually-intensive and possibly error-prone, even though four additional VAHIRR-derived products were created for testing purposes. The AMU created a software program that writes the intermediate values (e.g. sample size, average cloud bottom/top, average reflectivity, average cloud thickness, etc.) to binary files. The software reads the binary files and outputs the data in text format for display in a spreadsheet application. Although the software was not used in the testing procedures, it would be useful in future tests of the automated VAHIRR product;
- The automated VAHIRR product should undergo an "operational" test in which the reliability of the product is tested with live incoming radar data. This would verify that the product does not crash or perform incorrectly under various precipitation patterns. The AMU wrote a draft operational test plan, but did not perform it since the ABFM Comparison Test failed; and
- There are two user inputs to the automated VAHIRR product – the cone-of-silence height and the height of the freezing level. Since they affect the calculation of the VAHIRR values, it would be practical if they were displayed in the product. This can be done by modifying the product dependent parameters (PDP) in the ORPG-clone. Each PDP is stored as a signed 2-byte integer. Figure 34 shows an example of how the legend on the automated VAHIRR product would look if the freezing level was 16,390 ft MSL and the cone-of-silence height was 5000 ft AGL.



```
VAHIRR Average Bottom Height   CVG 8.0
             ID:154   Name(PCode): VAH(154)
             Resolution:  0.54nm (1000m)
             GEN: 18:59:41 September 12, 2007
             VOL: 19:08:22 July 5, 2007
             Elevation 0   VCP 21
             Station: [KMLB] ( WSR-88D)

             Product Dependant Parameters
              1:   1639        6:      0
              2:    500        7:      0
              3:      0        8:      0
              4:      0        9:      0
              5:      0       10:      0
```

Figure 34. On the ORPG-clone, the PDP in a VAHIRR product displays the freezing level height of 16,390 ft MSL as “1639” and the cone-of-silence height of 5000 ft AGL as “500”.

Appendix A
Design of the Source Code for AMU Automated VAHIRR product

The source code is discussed in three sections: Data Structures, Algorithm for AMU Automated VAHIRR Product, and Formulas. The first section describes the data structures that were used, such as arrays and constants. The second section uses pseudo-code to describe the source code. The third section includes several algorithms and equations that were used in the source code.

1) Data Structures

The following data structures were used in the VAHIRR source code and are described with the C programming language, since the source code was written in C. A "float" in C is a 4-byte floating point number. A "short" is a 2-byte signed integer number. An "unsigned" short is a 2-byte unsigned integer number. An "int" is a 4-byte signed integer number. A "struct" is a data structure used to hold multiple related variables. An array is an ordered collection of elements that have the same data type. The size (i.e. number of elements) of the array is declared in parentheses. Variables are in lower-case and constants are in upper-case.

Constants:

```
MAX_GATES = 230              /* max. number of gates */
MAX_AZIMUTHS = 360           /* max. number of azimuth angles */
MAX_ELEVATIONS = 20          /* max. number of elevation scans */
MAX_ROWS = 459               /* max. number of rows in Cartesian format */
MAX_COLUMNS = 459            /* max. number of columns in Cartesian format */
NO_DATA_FVALUE = -99.0       /* initial float value */
NO_DATA_IVALUE = -99         /* initial integer value */
```

Structures:

```
/* struct called "bin_type" representing one reflectivity value */
typedef struct
{
  float refl_dBZ; /* reflectivity in dBZ */
  float srg_km;   /* slant-range in km */
  float hght_km;  /* height AGL in km */
  float grg_km;   /* ground-range in km */
} bin_type;

/* struct called "radial_type" representing the reflectivity values for all gates
along one azimuth angle in one elevation scan */
typedef struct
{
  float az_angle_deg;      /* azimuth angle in degrees */
  bin_type bin[MAX_GATES]; /* array of bin_type structs, size=MAX_GATES */
} radial_type;

/* struct called "volume_type" representing the reflectivity values for one elevation
scan */
typedef struct
{
  float el_angle_deg;               /* elevation angle in degrees */
  radial_type radial[MAX_AZIMUTHS]; /* array of radial_type structs,
                                       size=MAX_AZIMUTHS */
  short first_gt;                   /* first gate with reflectivity data */
  short num_gts;                    /* number of gates with reflectivity data */
} volume_type;

/* struct called "vertical_info_type" representing the values for a horizontal grid
point after the first pass of the algorithm */
typedef struct
{
  float avg_refl_dBZ; /* ave. non-negative reflectivity at or above 0C for a
```

```
                            horizontal grid point */
  float sum_refl_dBZ; /* sum of non-negative reflectivity at or above 0C for a
                            horizontal grid point */
  int   samplesz;     /* number of elevation scans at or above 0C that have
                            non-negative reflectivity for a horizontal grid
                            point */
  float top_hght_km;  /* top of non-negative reflectivity at or above 0C for a
                            horizontal grid point */
  float bot_hght_km;  /* bottom of non-negative reflectivity at or above 0C for a
                            horizontal grid point */
  short meets_conds;  /* whether or not the horizontal grid point is
                            qualified to be included in VAHIRR calculation */
} vertical_info_type;

/* struct called "horizontal_info_type" representing the values for a horizontal grid
point after the second pass of the algorithm */
typedef struct
{
  float vahirr_dBZkm;                /* VAHIRR value in dBZ-km */
  float sum_weighted_refl_avgs_dBZ; /* sum of the average non-negative
                                           reflectivity values for the 11 x 11 km
                                           grid points */
  int sum_samplesizes;               /* sum of the sample sizes for the
                                       11 x 11 km grid points */
  float sum_top_hghts_km;            /* sum of top heights for the 11 x 11 km
                                           grid points */
  int top_hght_samplesz;             /* sum of the number of top heights for
                                       the 11 x 11 km grid points */
  float sum_bot_hghts_km;            /* sum of the bottom heights for the
                                       11 x 11 km grid points */
  int bot_hght_samplesz;             /* sum of the number of bottom heights
                                           for the 11 x 11 km grid points */
} horizontal_info_type;

/* array of volume_type structs, size=MAX_ELEVATIONS */
/* name of the array is “scan” and it holds reflectivity data for one volume scan */
volume_type scan[MAX_ELEVATIONS];

/* 2-D array of vertical_info_type structs, number of rows=MAX_ROWS,
number of columns=MAX_COLUMNS */
/* name of the array is “pt” and it holds reflectivity data for all of the horizontal
grid points after the first pass of the algorithm */
vertical_info_type pt[MAX_ROWS][MAX_COLUMNS];

/* 2-D array of horizontal_info_type structs, number of rows=MAX_ROWS,
number of columns=MAX_COLUMNS */
/* name of the array is “point” and it holds reflectivity data for all of the
horizontal grid points after the second pass of the algorithm */
horizontal_info_type point[MAX_ROWS][MAX_COLUMNS];
```

2) Algorithm for AMU Automated VAHIRR Product

Based on the actual source code, the following is the algorithm for the product:

```
Begin Algorithm

While (true)
```

1. Wait on input data

```
Wait until data from a new volume scan is collected
```

2. Initialize data structures and variables

```
Initialize most of volume scan data values to zero:
```

Source code snippet:

```
for (i = 0; i < MAX_ELEVATIONS; i++)
{
  scan[i].el_angle_deg = 0.0;
  scan[i].first_gt     = 0;
  scan[i].num_gts      = 0;

  for (j = 0; j < MAX_AZIMUTHS; j++)
  {
    scan[i].radial[j].az_angle_deg = NO_DATA_FVALUE;

    for (k = 0; k < MAX_GATES; k++)
    {
      scan[i].radial[j].bin[k].refl_dBZ = NO_DATA_FVALUE;
      scan[i].radial[j].bin[k].srg_km   = 0.0;
      scan[i].radial[j].bin[k].hght_km  = 0.0;
      scan[i].radial[j].bin[k].grg_km   = 0.0;
    }
  }
}

for (i = 0; i < MAX_ROWS; i++)
{
  for (j = 0; j < MAX_COLUMNS; j++)
  {
    pt[i][j].meets_conds  = TRUE;
    pt[i][j].top_hght_km  = 0.0;
    pt[i][j].bot_hght_km  = 0.0;
    pt[i][j].sum_refl_dBZ = 0.0;
    pt[i][j].samplesz     = 0;
    pt[i][j].avg_refl_dBZ = 0.0;
  }
}

for (i = 0; i < MAX_ROWS; i++)
{
  for (j = 0; j < MAX_COLUMNS; j++)
  {
    point[i][j].sum_weighted_refl_avgs_dBZ = 0.0;
    point[i][j].sum_samplesizes           = 0;
    point[i][j].sum_top_hghts_km          = 0.0;
    point[i][j].top_hght_samplesz         = 0;
    point[i][j].sum_bot_hghts_km          = 0.0;
    point[i][j].bot_hght_samplesz         = 0;
    point[i][j].vahirr_dBZkm              = NO_DATA_FVALUE
  }
}
```

3. Read COS height and height of freezing level

```
Read COS from the configuration text file
If COS cannot be read, is < 0.0, or is > maximum COS height (65.616 kft),
  Set COS to 65.616 kft
End If
Read RadarHeight in MSL (ft) from site adaptation data

Read 0Deg_height in MSL (kft) from Hail Product
If 0Deg_height >= 0.0 and <= Max_0Deg_height
  If 0Deg_height > RadarHeight
    Set 0Deg_height to 0Deg_height - RadarHeight
  Else
    Print error message to log
    Set 0Deg_height to 0.0
```

```
  End If
Else
  Print error message to log
  Abort VAHIRR product for this volume scan
End If
```

4. Read the latest volume scan

```
While volume scan data left to read
  Read one radial of data
  Read the radial’s elevation angle, first gate, number of gates, and volume
    number
  Read the radial’s azimuth angle
  For each gate in the radial
    Read the reflectivity value
    Calculate slant_range of gate from the gate number and gate size
    Calculate gate_height from elevation angle and slant range
    Calculate ground_range of gate from elevation angle and slant range
  End For
End While
```

5. Determine the index of the highest elevation scan in the volume

Note: The code assumes that elevation scans were received in increasing order, thus the last elevation index is the highest elevation scan.

```
Set elevation_index to 0
While elevation angle associated with elevation_index > 0
  Increment elevation_index
  If elevation_index == MAX_ELEVATIONS
    Break out of While loop
  End If
End While
If elevation_index > 0
  Return (elevation_index - 1)
Else
  Return (-1)
End If
```

6. Determine which grid points meets the conditions to calculate VAHIRR

Note: Three tests are performed to determine if a grid point should be disqualified from having VAHIRR calculated:
Lowest Elevation Test – Disqualify a grid point if the height of the lowest elevation scan is above the height of the freezing level.
Highest Elevation Test – Disqualify a grid point if the reflectivity value on the highest elevation scan is non-negative. The highest elevation scan means the highest visible scan above a point and not necessarily the highest elevation scan in the volume.
Cone of Silence Test – Disqualify a grid point if the height of the highest elevation scan is below or equal to the cone-of-silence height. The highest elevation scan in this test is the highest elevation scan in the volume.

```
Lowest Elevation Test
For each azimuth angle
  For each gate along the azimuth
    Find (Row, Column) of gate, from elevation angle, azimuth angle, and slant_range
    If gate_height > 0Deg_height
      Set pt[Row][Column].meets_conds to false
      If first time to set a flag to false
        Set grid_offset equal to Row
      End If
    End If
  End For
End For

Highest Elevation Test
```

```
Calculate farthest ground range of the lowest elevation scan
Set elevation_index to the highest elevation index
Set ground_range to 0.0
While ground_range < farthest ground range
  For highest elevation scan, find the first azimuth angle with data
  Find start_gate from the elevation_index, azimuth angle, and ground_range
  Find end_gate from the number of gates in the elevation scan
  For each azimuth angle
    For each gate in the azimuth angle
      Find (Row, Column) of gate, based on elevation angle, azimuth angle, slant_range
      If reflectivity of gate >= 0.0
        Set pt[Row][Column].meets_conds to false
      End If
    End For
  End For
  Set ground_range to range of end_gate for the current elevation_index
  Decrement elevation_index
End While

Cone of Silence Test
Calculate start gate index for first gate of highest elevation scan
Calculate end gate index for last gate of highest elevation scan
For each azimuth angle
  For each gate in the azimuth angle
    Find (Row, Column) of gate, based on elevation angle, azimuth angle, slant_range
    If height of gate <= COS
      Set pt[Row][Column].meets_conds to false
    End If
  End For
End For
```

7. Collect vertical data

```
For each elevation_index
  For each gate in the elevation scan
    For each azimuth angle in the elevation scan
      Find (Row, Column) of gate, from elevation angle, azimuth angle, slant_range
      If pt[Row][Column].meets_conds
        If reflectivity of gate >= 0.0
          If gate_height < 0Deg_height
            If cloud_bottom for pt[Row][Column] has not been set yet
              Set cloud_bottom for pt[Row][Column] to 0Deg_height
              Set cloud_top for pt[Row][Column] to 0Deg_height
            End If
          Else
            If cloud_bottom for pt[Row][Column] has not been set yet
              If not on the lowest elevation scan
                Set cloud bottom_for pt[Row][Column] to
                  (gate_height – half vertical distance to lower elevation scan)
              Else
                Set cloud_bottom to gate_height
              End If
            End If
            If not at the highest elevation scan
              Set cloud_top to
                (gate_height + half vertical distance to higher elevation scan)
            Else
              Set cloud_top to gate_height
            End If
            Increment sample size for pt[Row][Column]
            Add gate reflectivity to sum for pt[Row][Column]
            Calculate average reflectivity for pt[Row][Column]
          End If
        Else
```

```
          If reflectivity of gate == INVALID_DATA
            Set pt[Row][Column].meets_conds to false
          End If
        End If
      End If
    End For
  End For
End For
```

8. Calculate VAHIRR

Note #1: The weighted reflectivity average for a grid point is the average reflectivity in the column multiplied by the sample size. This gives points with a larger sample size (i.e. more elevation scans with non-negative reflectivity) more weight in the calculation of VAHIRR.

Note #2: When calculating intermediate and final values of VAHIRR, each reflectivity value in Polar coordinates (i.e. azimuth, range) is mapped to the closest Cartesian grid point (i.e. x,y). Multiple reflectivity values in Polar coordinates may map to the same Cartesian grid point. A Cartesian grid point may also have no reflectivity values in Polar coordinates mapped to it.

```
For Row from (Grid Offset+5) to (MAX_ROWS – (Grid Offset+5) – 1) by 1
  For column from (Grid Offset+5) to (MAX_ROWS – (Grid Offset+5) – 1) by 1
    If all of the 11 x 11 km grid points are qualified
      For all 11 rows
        For all 11 columns
          If sample size of pt[row][column] > 0
            Increment sample size of cloud bottoms for point[row][column]
            Add bottom height of pt[row][column] to sum of bottom heights for
              point[row][column]
            Increment sample size of cloud tops for point[row][column]
            Add top height of pt[row][column] to sum of top heights for
              point[row][column]
            Add sample size of pt[row][column] to sample size of
              point[row][column]
            Add weighted reflectivity average for pt[row][column] to sum of
              weighted reflectivity averages for point[row][column]
            If sum of sample sizes for point[row][column] > 0
              Calculate cloud reflectivity average
              Calculate average cloud top
              Calculate average cloud bottom
              Calculate average cloud thickness
              Calculate VAHIRR
            Else
              Set VAHIRR to -1.0
            End If
          End If
        End For
      End For
    End If
  End For
End For
```

9. Assemble final product

```
For Row from 0 to MAX_ROWS by 1
  For Column from 0 to MAX_COLUMNS by 1
    Convert from VAHIRR value to color code
  End For
End For
Write Product Description Block
Write color code to Raster Data Block
Write Raster Header Block
Write Product Symbology Block and Layer Header Block
Write Message Header Block
```

```
End While

End Algorithm
```

3) Formulas

1. Calculate POLAR (Slant Range)

Inputs: RANGE BEGINNING SURVEILLANCE, SURVEILLANCE BIN SIZE, Gate Number
Formula:

```
POLAR (Slant Range) = (RANGE BEGINNING SURVEILLANCE +
(SURVEILLANCE BIN SIZE * Gate Number) +
(SURVEILLANCE BIN SIZE / 2)) * 0.001
```

2. Calculate POLAR (Ground Range)

Note: the calculation does not take into effect the curvature of the earth and standard radar propagation. It would probably be more accurate if the azimuth and elevation were converted to Cartesian coordinates using the ORPG's RPGCS_azranelev_to_xy ORPG function, then use the distance formula: distance = $\mathrm{sqrt}(x^2 + y^2)$.

Inputs: POLAR(Elevation Angle), POLAR(Slant Range)
Formula:

```
IF POLAR (Elevation Angle) ≥ -90.0 AND POLAR (Elevation Angle) ≤ 90.0
  Elev_Radians = POLAR (Elevation Angle) * (PI / 180.0)
  POLAR (Ground Range) = (POLAR (Slant Range) * cosine(Elev_Radians))
END IF
```

3. Calculate Gate Height

Inputs: POLAR(Elevation Angle), POLAR(Slant Range)

Called a function from ORPG's function library:
RPGCS_height_u(srg_m, METERS, el_angle_deg, DEG, &hght_m, METERS)
Function definition:

```
int RPGCS_height_u( REAL range, int range_units, REAL elev,
    int elev_units, REAL *height, int height_units )
```

RETURN VALUE:	Returns -1 if the elevation angle parameter is out of range; returns -1 if any unit of measure parameter (**range_units**, **elev_units**, and **height_units**) has an invalid value; otherwise returns 0.

Description: Given the conical section slant range and elevation angle, the function calculates the height above the earth's surface using the input parameters **range_units**, **elev_units**, and **height_units**.

Function parameters:

range – The conical section slant range.
Elev - The conical section elevation angle.
Height – The height above the earth's surface calculated by the function.
Range_units – The unit of measure for the input slant range. Permitted values are **METERS**, **KFEET**, and **NMILES** as defined in **rpgcs_coordinates.h**.
elev_units – The unit of measure for the input elevation angle. Permitted values are **DEG** (degrees) and **DEG10** (degrees*10) as defined in **rpgcs_coordinates.h**. The **DEG10** unit is used by the ORPG in order to represent tenths of a degree with integer variables.
Height_units – The unit of measure for the calculated height. Permitted values are **METERS**, **KFEET**, and **NMILES** as defined in **rpgcs_coordinates.h**.

4. Calculate (Row,Column) coordinates of gate

Inputs: POLAR(Elevation Angle), POLAR(Azimuth Angle), POLAR(Slant Range)

Called a function from ORPG's function library:
RPGCS_azranelev_to_xy_u(srg_m, METERS, az_angle_deg, DEG, el_angle_deg, DEG, &x_m, &y_m, METERS)
Function definition:

```
int RPGCS_azranelev_to_xy_u( REAL range, int range_units, REAL azm,
    int azm_units, REAL elev, int elev_units, REAL *x, REAL *y,
    int xy_units )
```

RETURN VALUE:	Returns -1 if the elevation angle parameter is out of range; returns -1 if any unit of measure parameter (**range_units**, **azm_units**, **elev_units**, and **xy_units**) has an invalid value; otherwise returns 0.

Description: Given the conical section slant range, azimuth, and elevation angle, the function calculates the flat-plane Cartesian coordinates x and y using the input parameters **range_units**, **azm_units**, **elev_units**, and **xy_units**.

Function parameters:

range - The conical section slant range value.
azm - The conical section azimuth angle value.
elev - The conical section elevation angle value.
x - The calculated Cartesian x distance (positive is East).
y - The calculated Cartesian y distance (positive is North).
range_units - The unit of measure for the input slant range. Permitted values are **METERS**, **KFEET**, and **NMILES** as defined in **rpgcs_coordinates.h**.
azm_units - The unit of measure for the input azimuth angle. Permitted values are **DEG** (degrees) and **DEG10** (degrees*10) as defined in **rpgcs_coordinates.h**. The **DEG10** unit is used by the ORPG in order to represent tenths of a degree with integer variables.
elev_units - The unit of measure for the input elevation angle. Permitted values are **DEG** (degrees) and **DEG10** (degrees*10) as defined in **rpgcs_coordinates.h**. The **DEG10** unit is used by the ORPG in order to represent tenths of a degree with integer variables.
xy_units - The unit of measure for the calculated Cartesian coordinates. Permitted values are **METERS**, **KFEET**, and **NMILES** as defined in **rpgcs_coordinates.h**.

5. Calculate Azimuth Angle and Ground Range from Cartesian coordinates

Inputs: Row, Column

Called a function from ORPG's function library:
RPGCS_xy_to_azran_u(x_m, y_m, METERS, &grg_m, METERS, az_angle_deg)
Function definition:

```
int RPGCS_xy_to_azran_u( REAL x, REAL y, int xy_units, REAL *range,
    int range_units, REAL *azm )
```

RETURN VALUE:	Returns -1 if any unit of measure parameter (**xy_units** and **range_units**) has an invalid value, otherwise returns 0.

Description: Given Cartesian distance inputs x and y, the function calculates polar coordinate range and azimuth using the input parameters **xy_units** and **range_units**.

Function parameters:

x - The Cartesian x distance (positive is East).
y - The Cartesian y distance (positive is North).
range - The polar coordinate range value calculated by the function.
azm - The polar coordinate azimuth value calculated by the function.
xy_units - The unit of measure for the input Cartesian coordinates. Permitted values are **METERS**, **KFEET**, and

NMILES as defined in **rpgcs_coordinates.h**.

range_units - The unit of measure for the calculated polar coordinate range. Permitted values are **METERS**, **KFEET**, and **NMILES** as defined in **rpgcs_coordinates.h**.

6. Calculate Sum of Weighted Reflectivity Averages

Inputs: Reflectivity Average for each grid point, Sample Size for each grid point

Formula:

```
Sum of Weighted Reflectivity Averages =
```

$$\sum_{gridpoint=1}^{121} \texttt{Reflectivity Average}_{\texttt{gridpoint}} * \texttt{Sample Size}_{\texttt{gridpoint}}$$

7. Calculate Cloud Reflectivity Average

Inputs: Sum of Weighted Reflectivity Averages and Sum of Sample Sizes from 11 x 11 grid points

Formula:

```
Cloud Refl Average = Sum of Weighted Reflectivity Averages / Sample Size Sum
```

8. Calculate Cloud Top

Note: When converting from ground range to slant range, the curvature of the earth and standard radar propagation are not taken into effect. It may be more accurate if the azimuth and elevation was converted to Cartesian coordinates using the RPGCS_azranelev_to_xy ORPG function, then using the distance formula: distance = sqrt($x^2 + y^2$).

Inputs: Elevation Angle, Ground Range, Height

Formula:

```
Upper Angle = Next Higher Elevation Angle
Upper Slant Range = Ground Range / cosine(Upper Angle * PI/180.0)
Upper Height = Height of Upper Angle, calculated with the third formula
 (CALCULATE Gate Height)
Cloud Top = Height + (Upper Height - Height)/2
```

9. Calculate Cloud Bottom

Note #1: When converting from ground range to slant range, the curvature of the earth and standard radar propagation are not taken into effect. It may be more accurate if the azimuth and elevation was converted to Cartesian coordinates using the RPGCS_azranelev_to_xy ORPG function, then using the distance formula: distance = sqrt($x^2 + y^2$).

Note #2: If the cloud bottom is in the lowest elevation scan, then the cloud bottom must be equal to or below the freezing level. If the cloud bottom is in the lowest elevation scan and above the freezing level, the grid point is disqualified. This is because the lowest elevation scan must be at or below the freezing level to calculate VAHIRR. If the cloud bottom is lower than the freezing level, the product set the cloud bottom to the height of the freezing level.

Inputs: Elevation Angle, Elevation Index, Ground Range, Height

Formula:

```
If not at lowest elevation scan
  Lower Angle = Next Lower Elevation Angle
  Lower Slant Range = Ground Range / cosine(Lower Angle * PI/180.0)
  Lower Height = Height of Lower Angle, calculated with the third formula
   (CALCULATE Gate Height)
  Cloud Bottom = Height - (Height - Lower Height)/2
  If Cloud Bottom < 0Deg_height
    Set Cloud Bottom to 0Deg_height
  End If
Else
  Set Cloud Bottom to Height
```

```
End If
```

10. Calculate Average Cloud Thickness

Inputs: Average Top Height, Average Bottom Height
Formula:

```
Average Cloud Thickness = Average Top Height – Average Bottom Height
```

11. Calculate VAHIRR

Inputs: Cloud Reflectivity Average, Average Cloud Thickness
Formula:

```
VAHIRR = Cloud Reflectivity Average * Average Cloud Thickness
```

Appendix B

Initial Test Procedure

1) Test Plan

The VAHIRR task performs the following functionality: input data, interpret data positions, interrogate data for conditions, collect qualifying data, and calculate the final results. Using a single test scenario, each of these capabilities will be exercised and verified.

The test scenario shall,

a) Ingest test data into the VAHIRR product,

b) Stage unique reflectivity quantities in multiple quadrants, over varying sector sizes, and for full and partial sectors,

c) Include satisfied and failed conditions,

d) Utilize fixed reflectivity quantities to reveal increased results over distance from the radar, and

e) Display the final product.

2) VAHIRR Test Data Description

The following table shows the “canned” data used in the test procedure.

RADIALS	GATES	ELEVATION ANGLE(S) *	REFLECTIVITY encoded value	REFLECTIVITY dBZ equivalent
45.0 - 135.0 (inclusively)	0-130	0.5, 1.5, 2.4, 3.4	73	3.5
	131-229	0.5 – 3.4	62	-2.0
	all	4.5, 5.3, 6.2, 7.5, 8.7, 10.0, 12.0, 14.0 16.6, 19.5	65	-0.5
180.0 - 200.0 (inclusively)	all	0.5 – 3.4	74	4.0
	all	4.5 – 19.5	65	-0.5
256.0 - 260.0 (inclusively)	all	0.5 – 3.4	66	0.0
	0-61	4.5 – 19.5 **	66	0.0
	62-229	4.5 – 19.5	65	-0.5
322.0 - 353.0 (inclusively)	all	0.5 – 3.4	70	2.0
	all	4.5 – 19.5	65	-0.5
all other radials	all	0.5 – 3.4	66	0.0
	all	4.5 – 19.5	64	-1.0

***** Higher elevation angles are set to a negative value over all radials.

****** …except for a short gate range through radials 256.0° – 260.0° to show proper handling when the highest tilt(s) do not have negative reflectivity.

3) Pre-Test Setup

A task in the ORPG subsystem called pbd (process base data), pre-processes RDA base data and creates ORPG base data for consumption by ORPG products such as VAHIRR. A routine in this task (PD_move_data) has been modified according to the test data description above to create tailored Level II base data for testing the VAHIRR product:

```
#define TESTVAHIRR
#ifdef TESTVAHIRR
/*---OVERLAPPING QUADRANTS 1 AND 2-------------------------------------*/
if (45.0 <= rpg_hd->azimuth && rpg_hd->azimuth <= 135.0)
{
   if (rpg_hd->elevation < 4.0)
   {
      for (i = 0; i < 131; i++)   ref_dest_ptr[i + bin_start] = 73;/* 3.5*/
      for (i = 131; i < num; i++) ref_dest_ptr[i + bin_start] = 62;/*-2.0*/
   }
   else /* for highest elevations */
```

```
      for (i = 0; i < num; i++)   ref_dest_ptr[i + bin_start] = 65;/*-0.5*/
}
/*---QUADRANT 3---------------------------------------------------------*/
else if (180.0 <= rpg_hd->azimuth && rpg_hd->azimuth <= 200.0)
{
   if (rpg_hd->elevation < 4.0)
      for (i = 0; i < num; i++)   ref_dest_ptr[i + bin_start] = 74;/* 4.0*/
   else /* for highest elevations */
      for (i = 0; i < num; i++)   ref_dest_ptr[i + bin_start] = 65;/*-0.5*/
}
/*---QUADRANT 3 WHITE BAR-----------------------------------------------*/
else if (256.0 <= rpg_hd->azimuth && rpg_hd->azimuth <= 260.0)
{
   if (rpg_hd->elevation < 4.0)
      for (i = 0; i < num; i++)   ref_dest_ptr[i + bin_start] = 66;/* 0.0*/
   else /* for highest elevations */
   {
      for (i = 0; i < 62; i++)    ref_dest_ptr[i + bin_start] = 66;/* 0.0*/
      for (i = 62; i < num; i++)  ref_dest_ptr[i + bin_start] = 65;/*-0.5*/
   }
}
/*---QUADRANT 4--------------------------------------------------------*/
else if (322.0 <= rpg_hd->azimuth && rpg_hd->azimuth <= 353.0)
{
   if (rpg_hd->elevation < 4.0)
      for (i = 0; i < num; i++)   ref_dest_ptr[i + bin_start] = 70;/* 2.0*/
   else /* for highest elevations */
      for (i = 0; i < num; i++)   ref_dest_ptr[i + bin_start] = 65;/*-0.5*/
}
/*---REMAINING SPACE---------------------------------------------------*/
else
{
   if (rpg_hd->elevation < 4.0)
      for (i = 0; i < num; i++)  ref_dest_ptr[i + bin_start] = 66;/* 0.0*/
   else /* for highest elevations */
      for (i = 0; i < num; i++)  ref_dest_ptr[i + bin_start] = 64;/*-1.0*/
}
#else
for( i = 0; i < num; i++ )   /*---ORIGINAL CODE--------------------------*/
   ref_dest_ptr[i + bin_start] = (short) ref_src_ptr[i + first_surv_bin_off];
#endif
```

4) Test Procedure

The following steps will execute the VAHIRR product as an ORPG concurrent task, read a single Archive Level II data file, monitor data ingest, and display the final VAHIRR output product.

Date: ___________________________ **Time:** ______________________

___ 1) Log on the computer. Username: **codev6** Password: **test123**

___ 2) Open a terminal window.

___ 3) Enter: **mrpg –r –v startup**
The Manage RPG utility and its startup command invoke a number of concurrently running ORPG processes. The –r option recreates all non-persistent global data stores and –v sets the verbose level for the log-error messaging module.

___ 4) Notice will be given once ORPG startup has completed. Wait for completion.

___ 5) Open another terminal window.

___ 6) Enter: **hci**
The Human Computer Interface utility provides RPG status.

___ 7) The *RPG Control/Status* window opens.

___ 8) In the *RPG Control/Status* window, under *Applications*, select *Base Data Display*.

___ 9) The *RPG Base Data Display* window opens.

___ 10) Return to the “RPG startup” terminal window

___ 11) Enter: **play_a2 –s KMLB_1993:03:13:09:36:38.ar2.bz2 –n 1**
The play_a2 utility reads WSR-88D Level II data from compressed files on disk and ingests the data into the ORPG. The –s option provides the input file name and –n the number of the volumes to play.
Note: The environment variable AR2_DIR must be defined for normal operation of the play_a2 utility. The directory specified by $AR2_DIR is the default location of Archive II disk files to be used as input for the ORPG. Otherwise include the –d DIR option on the command line.

___ 12) As data is ingested, observe the *RPG Base Data Display* window.

___ 13) In the “play data” terminal window, notice will be given when the specified number of volumes has been reached. Wait for the notice.

___ 14) Open a third terminal window.

___ 15) Enter: **cvg**
The CODEview Graphics utility is an X Windows based tool used for decoding and displaying final products.

___ 16) The *CODEview Graphics 6.0* window will open.

___ 17) In the *CODEview Graphics 6.0* window, under the *Product Description* list, select *VAH Volume Averaged Height Integrated Radar Reflectivity* and then press the *Select Database Product* button.

___ 18) A temporary window will open.

___ 19) In this window, select the *OK* button.

___ 20) The *Screen 1* window will open displaying the VAHIRR output product.

___ 21) Review and discuss results.

VAHIRR results are in dBZ-kft
White represents out of bounds or failed conditions for calculating VAHIRR
Black represents VAHIRR is zero or reflectivity is negative/missing
Light Blue represents 0 < VAHIRR < 10
Medium Blue represents 10 <= VAHIRR < 19
…
Purple represents VAHIRR >= 56

The VAHIRR sector in the first and second quadrants has reflectivity of 3.5 dBZ for the first 130 gates. The VAHIRR results change over distance from the radar because the cloud thickness increases while average reflectivity remains a constant. The irregularity in the “rings” is a consequence of input radial angle precision and whole degree algorithm processing. Sometimes two input radials represent the same degree to the VAHIRR algorithm, thus data is lost for the next radial degree. When data is collected vertically and there is a missing radial, a lower cloud top height may occur, resulting in a thinner cloud thickness, and therefore a smaller VAHIRR result.

The VAHIRR sector in the third quadrant has reflectivity of 4.0 dBZ for all 230 gates. The same situation arises – a constant average reflectivity as cloud thickness increases. The white bar in the third quadrant is where reflectivity on the highest elevation scan(s) is non-negative (0.0 dBZ) for the first 61 gates, therefore violating the condition that reflectivity on the highest elevation scan must be negative.

The VAHIRR sector in the fourth quadrant has reflectivity of 2.0 dBZ for all 230 gates. Since the reflectivity is lower, the VAHIRR results are lower on the color scale. The small black triangles near the four corners of the display are an insignificant consequence of representing spherical data in a two-dimensional array.

___ 22) Close the *Screen 1* and *CODEview Graphics 6.0* windows.

___ 23) Close the *RPG Base Data Display* and *RPG Control/Status* windows.

___ 24) Return to the “play data” terminal window.

___ 25) Enter: **mrpg shutdown**

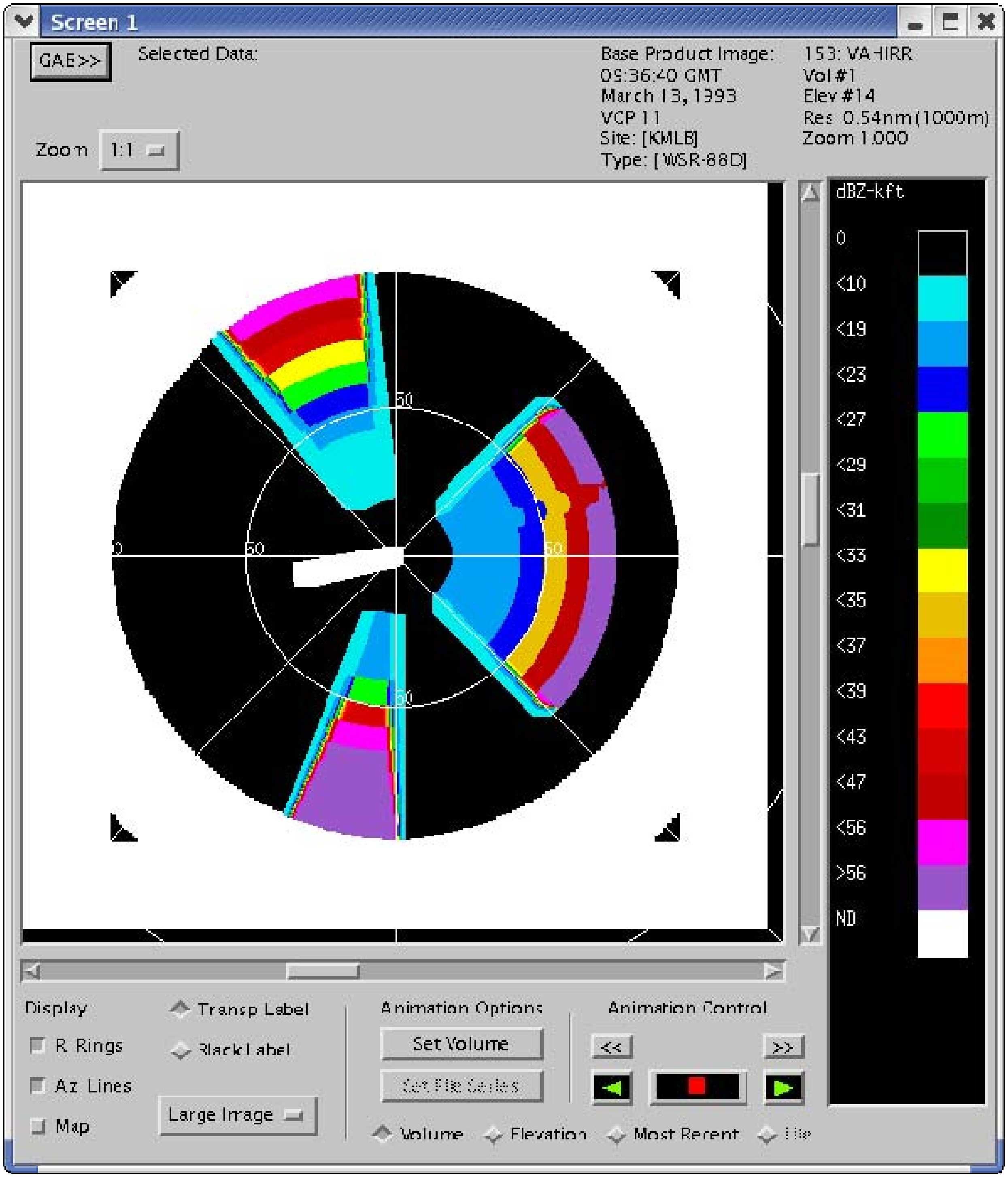
___ 26) Enter: **mrpg cleanup**

___ 27) Close all terminal windows.

___ 28) Log off the computer.

5) Expected Results

Figure 35 shows the expected VAHIRR product.



Selected Data:

Zoom 1:1

Base Product Image:
09:36:40 GMT
March 13, 1993
VCP 11
Site: [KMLB]
Type: [WSR-88D]

153: VAHIRR
Vol #1
Elev #14
Res 0.54nm(1000m)
Zoom 1.000

Figure 35. VAHIRR product produced from the Initial Test Procedure.

Appendix C

AWIPS Localization instructions for the automated VAHIRR product at SMG

1. Add the following line to **`/awips/fxa/data/radarInfo.txt`** (the 76 is an AWIPS identifier that has not been used in the radarInfo.txt file):

```
|153 | 16 | 0 | 1.0 | 230 | VA | VAHIRR (VA)    | Raster |  |  |  |  | Z |  |  |  |  | 76
```

2. Create the **`radarDepictKeys.template`** file in **`/data/fxa/customFiles`**. It should contain the following line (1112 in the last column is a custom color enhancement that SMG made for the VAHIRR product):

```
50476 | 16 | 50476 |  | 0 | 0 |@@@@ VAHIRR (dBZ-kft) |@@@@ VAHIRR | 8 | 0 | 1 | 1112
```

The radar depict keys for VAHIRR were created using the instructions in the header of the radarDepictKeys.template file. Edit the **`radarDepictKeys.template`** file in **`/data/fxa/nationalData`** or **`/data/fxa/customFiles`**.

A custom color map containing the VAHIRR color enhancement is in the **`customColorMaps.txt`** file. A copy of the file can be requested from the AMU or SMG. The **`customColorMaps.txt`** file is in CDL format. Create a NetCDF file from it with the **`ncgen`** utility. Rename the NetCDF file to **`customColorMaps.nc`** and place it in the **`/data/fxa/workFiles/customColorMaps.nc`** directory.

3. Create the **`radarDataKeys.template`** file in **`/data/fxa/customFiles`**. It should contain the following line:

```
50476 | @@@@/@@@@Ras464km |76|0| 1 | 16 | 0 |radar/@@@@/VAHIRR/layer0/res1/level16 | | |
```

The radar data keys for VAHIRR were created using the instructions in the header of the radarDataKeys.template file. A copy of the file can be requested from the AMU or SMG. Edit the **`radarDataKeys.template`** file in **`/data/fxa/nationalData`** or **`/data/fxa/customFiles`**.

4. Create the **`radarProductButtonInfo.template`** file in **`/data/fxa/customFiles`**. It should contain the following line:

```
50476 | @ | VAHIRR  | @@@@ VAHIRR  | 0 |
```

A copy of the file can be requested from the AMU or SMG. Edit the **`radarProductButtonInfo.template`** file in **`/data/fxa/nationalData`** or **`/data/fxa/customFiles`**.

5. Create the **`radarDataMenus.template`** file in **`/data/fxa/customFiles`**. It should contain the following line:

```
productButton: 50476 # VAHIRR
```

A copy of the file can be requested from the AMU or SMG. Edit the **`radarDataMenus.template`** file in **`/data/fxa/nationalData`** or **`/data/fxa/customFiles`**.

6. Run a –radar and –tables localization on each workstation in the KSC localization. Run a –dirs and –radar localization on the AWIPS servers which run under a MLB localization. MLB is the ingest site for the Spaceflight Meteorology Group, while KSC is one of the display localizations. See if the **`/data/fxa/radar/kmlb/VAHIRR/layer0/res1/level16`** directory exists and that the awipsusr and fxa accounts have read and write permission to it. If not, create the directory. The VAHIRR button will show up at the very bottom of the kmlb radar menu. Do not define a **`radarImageStyleInfo.template`** file for VAHIRR, since it will put the data into 5 dBZ-kft bins.

7. If D2D “hangs” while trying to view VAHIRR products, remove any copies of the `radarImageStyleInfo.txt` or `radarImageStyleInfo.template` files from the `/awips/fxa/data/` and `/awips/fxa/data/localizationDataSets/LLL` directories, where LLL is the localization identifier.

8. Create a `$HOME/cronLogs` directory and a `$HOME/cronScripts` directory. Request the `sendvahirr.exp` script from the AMU or SMG. Place the script into the `$HOME/cronScripts` directory on the ORPG-clone.

9. Request the `smgScour.tcl` script from the AMU or SMG. Place the script into the `$HOME/cronScripts` directory on the ORPG-clone. The `smgScour.tcl` script deletes old files on the ORPG-clone.

10. Request the `orpgVahirrScour.cfg` file from the AMU or SMG. Place the file in the `$HOME/cronScripts` directory on the ORPG-clone. This is a configuration file read by `smgScour.tcl`. It specifies the directories, file pattern matches, and number of hours to keep products.

11. Add the following two lines to the ORPG account’s crontable (enter `crontab –e` at a command prompt):

```
55 * * * * /home/orpg/cronScripts/smgScour.tcl
/home/orpg/cronScripts/orpgVahirrScour.cfg > /home/orpg/cronLogs/smgScour.crontxt 2>&1

* * * * * csh -c '/home/orpg/cronScripts/sendvahirr.exp' >
/home/orpg/cronLogs/sendvahirr.crontxt 2>&1
```

The two entries will run the `sendvahirr.exp` script once a minute and run `smgScour.tcl` once an hour. The `sendvahirr.exp` script extracts the latest VAHIRR product from the ORPG-clone’s product database and inserts the VAHIRR product (naming convention is `SFMG-VAHIRR.yyyymmdd_hhmm`) into the LDM product queue with the `pqinsert` utility. The `smgScour.tcl` script deletes old LDM and log data on the ORPG-clone.

Verify that the crontable entries are correct, by entering `crontab –l` at a command prompt.

12. Request the `ldmd.conf` file from the AMU or SMG. Note – for security reasons the IP addresses in the file have been changed to symbols. Place the file in LDAD. This file should contain the following line (A.B.C.D is the IP address of the LDAD server):

```
request  EXP  “^SFMG-.*”  A.B.C.D
```

13. Request the `pqact.conf` file from the AMU or SMG. Place the file in LDAD. The file should contain the following line:

```
EXP  ^SFMG-(.*)  FILE  -overwrite  /data/Incoming/\1
```

14. Request the `LDADinfo.txt` file from the AMU or SMG. Place the file in the `/data/fxa/LDAD/data` directory on the DS1 machine. This file tells LDAD what to do. The file should contain the following line:

```
VAHIRR    |    |    |    |    |    |preProcessVAHIRR.pl    |
```

15. Request the `preProcessVAHIRR.pl` file from the AMU or SMG. Place the file in the `/awips/fxa/ldad/bin` directory on the DS1 machine. This is a pre-processor script that files the VAHIRR product into the proper directory (i.e. `/data/fxa/radar/kmlb/VAHIRR/layer0/res1/level16/`) and then triggers the notification server with the `testFileNotify` utility.

16. The SMG added the following two lines to the `ldmd.conf` file on their “arps” workstation (A.B.C.D is the IP address of the ORPG-clone):

```
request EXP “^SFMG-VAHIRR” A.B.C.D
allow EXP ^(A.B.C.D)
```

This **`ldmd.conf`** file affects all of the outgoing MIDDS products. If products are sent directly from the ORPG-clone to LDAD, the request line above is not needed. Contact SMG for a copy of the **`ldmd.conf`** file on their arps workstation.

Appendix D

AWIPS Localization instructions for the automated VAHIRR product at AMU

These instructions include the four VAHIRR-derived products that were created for testing purposes: VAHREF (VAHIRR Volume-Averaged Reflectivity), VAHTHI (VAHIRR Average Cloud Thickness), VAHTOP (VAHIRR Average Cloud Top), and VAHBOT (VAHIRR Average Cloud Bottom).

1. Add the following lines to **`/awips/fxa/data/radarInfo.txt`** (the 76 is an AWIPS identifier that has not been used in the radarInfo.txt file):

```
|153 | 16 | 0 | 1.0 | 230 | VA | VAHIRR (VA)    | Raster |  |  |  |  | Z |  |  |  |  | 76
|155 | 16 | 0 | 1.0 | 230 | VA | VAHREF (VAREF) | Raster |  |  |  |  | Z |  |  |  |  | 77
|156 | 16 | 0 | 1.0 | 230 | VA | VAHTHI (VATHI) | Raster |  |  |  |  | Z |  |  |  |  | 78
|157 | 16 | 0 | 1.0 | 230 | VA | VAHTOP (VATOP) | Raster |  |  |  |  | Z |  |  |  |  | 79
|158 | 16 | 0 | 1.0 | 230 | VA | VAHBOT (VABOT) | Raster |  |  |  |  | Z |  |  |  |  | 80
```

2. Create the **`radarDepictKeys.template`** file in **`/data/fxa/customFiles`**. It should contain the following lines:

```
50476 | 16 | 50476 |  | 0 | 0 |@@@@ VAHIRR (dBZ-kft)   |@@@@ VAHIRR | 8 | 0 | 1 | 43
50477 | 16 | 50477 |  | 0 | 0 |@@@@ VAHREF (dBZ)       |@@@@ VAHREF | 8 | 0 | 1 | 43
50478 | 16 | 50478 |  | 0 | 0 |@@@@ VAHTHI (tenths-km) |@@@@ VAHTHI | 8 | 0 | 1 | 43
50479 | 16 | 50479 |  | 0 | 0 |@@@@ VAHTOP (tenths-km) |@@@@ VAHTOP | 8 | 0 | 1 | 43
50480 | 16 | 50480 |  | 0 | 0 |@@@@ VAHBOT (tenths-km) |@@@@ VAHBOT | 8 | 0 | 1 | 43
```

The radar depict keys for VAHIRR were created using the instructions in the header of the radarDepictKeys.template file. Edit the **`radarDepictKeys.template`** file in **`/data/fxa/nationalData`** or **`/data/fxa/customFiles`**.

3. Create the **radarDataKeys.template** file in **/data/fxa/customFiles**. It should contain the following lines:

```
50476 | @@@@/@@@@Ras464km |76|0| 1 | 16 | 0 |radar/@@@@/VAHIRR/layer0/res1/level16 | | |
50477 | @@@@/@@@@Ras464km |77|0| 1 | 16 | 0 |radar/@@@@/VAHREF                     | | |
50478 | @@@@/@@@@Ras464km |78|0| 1 | 16 | 0 |radar/@@@@/VAHTHI                     | | |
50479 | @@@@/@@@@Ras464km |79|0| 1 | 16 | 0 |radar/@@@@/VAHTOP                     | | |
50480 | @@@@/@@@@Ras464km |80|0| 1 | 16 | 0 |radar/@@@@/VAHBOT                     | | |
```

The radar data keys for VAHIRR were created using the instructions in the header of the radarDataKeys.template file. Edit the **`radarDataKeys.template`** file in **`/data/fxa/nationalData`** or **`/data/fxa/customFiles`**.

4. Create the **`radarProductButtonInfo.template`** file in **`/data/fxa/customFiles`**. It should contain the following lines:

```
50476 | @ | VAHIRR  | @@@@ VAHIRR  | 0 |
50477 | @ | VAHREF  | @@@@ VAHREF  | 0 |
50478 | @ | VAHTHI  | @@@@ VAHTHI  | 0 |
50479 | @ | VAHTOP  | @@@@ VAHTOP  | 0 |
50480 | @ | VAHBOT  | @@@@ VAHBOT  | 0 |
```

Edit the **`radarProductButtonInfo.template`** file in **`/data/fxa/nationalData`** or **`/data/fxa/customFiles`**.

5. Create the **`radarDataMenus.template`** file in **`/data/fxa/customFiles`**. It should contain the following lines:

```
productButton: 50476 # VAHIRR
productButton: 50477 # VAHREF
productButton: 50478 # VAHTHI
productButton: 50479 # VAHTOP
productButton: 50480 # VAHBOT
```

Edit the `radarDataMenus.template` file in `/data/fxa/nationalData` or `/data/fxa/customFiles`.

6. Create the following five directories inside `/data/fxa/radar/kmlb/`: `VAHIRR/layer0/res1/level16`, `VAHTOP`, `VAHBOT`, `VAHTHI`, and `VAHREF`.

The directories should be owned by the fxa account, with the wxman group. The owner, group, and others should have read and write permissions to the directories, because they will be accessed remotely via NFS.

Shutdown D2D. Run a full localization on the AWIPS workstation as user fxa (e.g. `/awips/fxa/data/localization/scripts/mainScript.csh UAL`). Restart D2D. The VAHIRR product buttons should show up in the radar menu.

Do not define a `radarImageStyleInfo.template` file for VAHIRR, since it will put the data into 5 dBZ-kft bins.

7. If D2D “hangs” while trying to view VAHIRR products, remove any copies of the `radarImageStyleInfo.txt` or `radarImageStyleInfo.template` files from the `/awips/fxa/data/` and `/awips/fxa/data/localizationDataSets/LLL` directories, where LLL is the localization identifier.

8. Mount the ORPG-clone to the pseudo-AWIPS server (amu-awipssvr). The following entry needs to be in the `/etc/fstab` file (assuming that the IP address of amu-awipssvr is 10.7.101.201):

```
10.7.101.201:/data/fxa    /data/fxa    nfs    defaults    1 1
```

View the `/etc/mtab` file to view which partitions and NFS directories are currently mounted. The **df** command also shows the current partitions and NFS directories. The /etc/mtab file should have the following entry:

```
10.7.101.201:/data/fxa /data/fxa nfs rw,addr=10.7.101.201 0 0
```

The VAHIRR directories should also be accessible from the ORPG-clone’s command line (e.g. `/data/fxa/radar/kmlb/VAHIRR/layer0/res1/level16/`).

If the ORPG-clone is not properly mounted to the pseudo-AWIPS server, then `su` to root on the ORPG-clone. At a command prompt, enter `mount –a`.

9. Create a `$HOME/cronLogs` directory and a `$HOME/scripts` directory. Request the `sendvahirr_v2.exp` script from the AMU. Place the script into the `$HOME/scripts` directory on the ORPG-clone.

10. Start the `sendvahirr_v2.exp` script. Every 30 seconds, the script extracts the latest VAHIRR product from the ORPG-clone’s product database, then copies it to the pseudo-AWIPS server so that it can be viewed with the AWIPS D2D application. A local copy of the VAHIRR product is stored on the ORPG-clone in the `$HOME/data/153` directory.

Appendix E
AMU Trajectory Map Maker application

1) Description of the application

The AMU created an AWIPS local application, the AMU Trajectory Map Maker, that plots the ground track of space launches and shuttle landings. The plots can then be overlaid on the VAHIRR product to assist with the analysis of radar data along a launch or landing trajectory. For each trajectory, the application reads an input text file in Deorbit Opportunity Map (DOP) file format or Launch format.

In the DOP file format, each line contains either a lat/lon pair or a blank line (east and north are positive). Sequential lines of latitude/longitude pairs are treated as a single linked vector. A blank line causes a new linked vector to start.

In the Launch file format, three separate text files are used for each launch – for the left and right edges of the expected trajectory and the center of the expected trajectory. The center of the expected trajectory is known as the nominal track. Each line in a file contains three decimal numbers – the first is the altitude or time into the flight, the second is the latitude, and the third is the longitude.

The Tcl/Tk source code file (**`AMUmaptool.tcl`**), associated helper script (**`dgmfmt.csh`**), and two example files in DOP (**`nomipdata.txt`**) and Launch (**`nomipbcd.txt`**) format, can be requested from the AMU.

2) How the application works

Based on the input text file, the application creates a output file in the Denver AWIPS Risk Reduction and Requirements Evaluation (DARE) Graphics Metafile (DGM) format. The DGM file format is not well-known and was created specifically for AWIPS. The DGM files are binary files, containing a series of commands stored in two byte integers. (GSD 2006). The DGM filenames must be in the format YYYYMMDD_hhmm. For example, 20070927_1520 would be a DGM file that was created at 1520 UTC on 27 September 2007.

There are three DGM drawing commands and the rest are state commands that stay in effect until they are changed. Each DGM command is represented by a unique "opcode" in hexadecimal format. Absolute coordinates normally are in minutes of latitude or longitude (east and south are positive). For example, 30 degrees north latitude is encoded as -1800 (-60*30), and 80 degrees west longitude is encoded as -4800 (-60*80). The most useful commands are:

a) <u>Draw Linked Vectors:</u>
Opcode = 0100
Format:
N (number of points beyond start point)
X(0) (starting point x-coordinate)
Y(0) (starting point y-coordinate)
X(1)
Y(1)
:
X(N) (ending point x-coordinate)
Y(N) (ending point y-coordinate)

b) <u>Draw Unlinked Vectors:</u>
Opcode = 0200
Format:
N (number of pairs of endpoints)
X1(1) (the pair number is in parentheses)
Y1(1)
X2(1)
Y2(1)
:
X1(N)

Y1(N)
X2(N)
Y2(N)

c) Draw Text:
Opcode = 0300
Format:
N (number of characters)
X location
Y location
C(2) C(1) (characters are packed two per instruction)
C(4) C(3)
:
C(N) C(N-1) (For even N. For odd N, use 00 C(N).)

d) Set Text Centering:
Opcode = 08hv
Format:
(h = 0 for left, h = 1 for middle, h = 2 for right, v = 0 for top, v = 1 for middle, v = 2 for bottom)
Top left is the default.

e) Set Text Orientation:
Opcode = 09ur
Format:
u is the update direction for placing new characters and r is the rotation of individual characters with respect to the update direction; 00 is the default
0 = normal, 1 = 90 deg clockwise, 2 = upside down, 3 = 90 deg counterclockwise

f) Select Absolute Addressing:
Opcode = 0ctt
Format:
When tt = FF, then frame addressing is in effect. Otherwise, coordinates refer to a location in the frame and items stay at the same location on the visible

g) Select Relative Addressing:
Opcode = 0Dtt
Format:
X origin
Y origin

This command starts a mode where coordinates refer to an offset from the origin point. The origin point is mapped according to frame addressing (tt = FF) or normal addressing (tt = 00) rules. All following coordinates are offsets in pixels from the origin point.

h) Set Frame Size:
Opcode = 0Fgn (where 2^n = size of frame)
Format:
Central longitude (only used when g = 2)
Central latitude (only used when g = 2)

If this command is used, it must occur before any instruction which contains coordinate information. The default n value is 09 (512 frame size with standard integer Cartesian coordinates). n can range from 8 to D (8 to 13 decimal).

When g = 0, absolute coordinates in the graphic refer to an integer location. When g = 1, absolute coordinates are in minutes of longitude and latitude; east and south are positive. When g = 2, a central longitude and latitude is given in minutes, and absolute coordinates refer to an offset from the central location in seconds of longitude and latitude.

3) Installing the application in AWIPS

a) Copy the **`AMUmaptool.tcl`** and **`dgmfmt.csh`** files to the **`/awips/fxa/bin`** directory.

b) Create two directories to hold the input text files:
  - **`/awips/fxa/awipsusr/AMUMapMaker/DOP`**
  - **`/awips/fxa/awipsusr/AMUMapMaker/Launch`**

c) Create directories to hold the output DGM files:
  - **`/data/fxa/dgm/map/left`**
  - **`/data/fxa/dgm/map/nominal`**
  - **`/data/fxa/dgm/map/right`**
  - **`/data/fxa/dgm/map/landing`**

d) Add the tool to the **`$FXA_HOME/data/appInfo.txt`** localization file.

e) Add an application product button to the **`dataMenus.txt`** localization file.

f) In the **`backgroundMenus.txt`** localization file, add a submenu called "Trajectory Maps". Inside the submenu add four product buttons: "Left Track", "Nominal Track", "Right Track", and "Shuttle Landing".

g) Add the three product buttons to the **`productButtonInfo.txt`** localization file.

h) Add data keys for the four DGM products to the **`dataInfo.manual`** localization file.

i) Add depict keys for the four DGM products to the **`depictInfo.manual`** localization file.

j) Run a full localization with the localization script. Restart the D2D application.

4) Using the application

a) The application can be started from an AWIPS D2D menu or directly from the command line. When it is started, a graphical user interface (GUI) is displayed.

b) Select the DOP File Format button to display the input filenames in DOP format (Figure 36). Select the Launch File Format button to display the input filenames in Launch format (Figure 37).

c) Select the Map Type – Landing, Left, Nominal, or Right.

d) The Label Trajectory option has not been implemented.

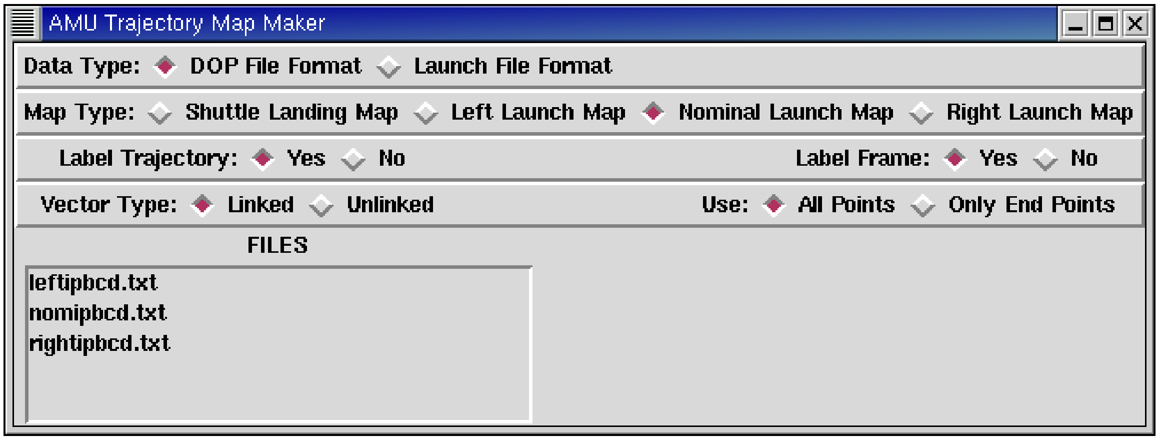
e) The Label Frame option writes the DGM file creation date/time in the upper left corner of the D2D display frame.

f) The Vector Type determines whether the points in the input file are linked or unlinked.

g) The "All Points" option uses all of the latitude/longitude points in the input file (Figure 38), while the "Only End Points" option uses only the beginning and ending latitude/longitude points in the input file (Figure 39). Using the All Points option may cause the trajectory to look jagged, because the points are limited in resolution to minutes of latitude/longitude.

h) Select one of the filenames in the FILES listbox to create the DGM file containing the trajectory plot. A pop-up dialog box will then be displayed with the input filename. Click on the OK button to create the DGM file.

i) Go into the Maps menu in D2D to display the newly-created plot.


AMU Trajectory Map Maker

Data Type: ● DOP File Format ○ Launch File Format

Map Type: ○ Shuttle Landing Map ○ Left Launch Map ● Nominal Launch Map ○ Right Launch Map

Label Trajectory: ● Yes ○ No

Label Frame: ● Yes ○ No

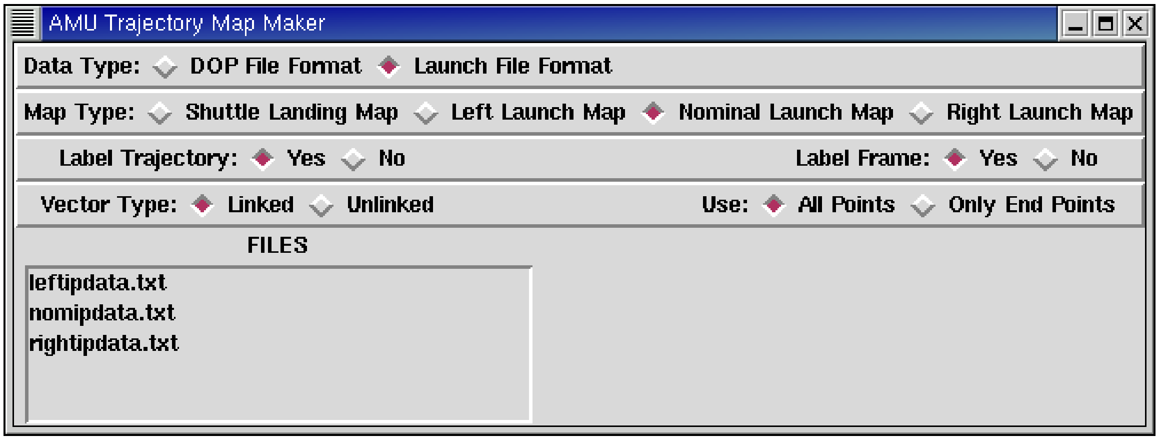
Vector Type: ● Linked ○ Unlinked

Use: ● All Points ○ Only End Points

FILES

leftipbcd.txt
nomipbcd.txt
rightipbcd.txt

Figure 36. The AMU Trajectory Map Maker displaying the files in DOP format.


AMU Trajectory Map Maker

Data Type: ○ DOP File Format ● Launch File Format

Map Type: ○ Shuttle Landing Map ○ Left Launch Map ● Nominal Launch Map ○ Right Launch Map

Label Trajectory: ● Yes ○ No

Label Frame: ● Yes ○ No

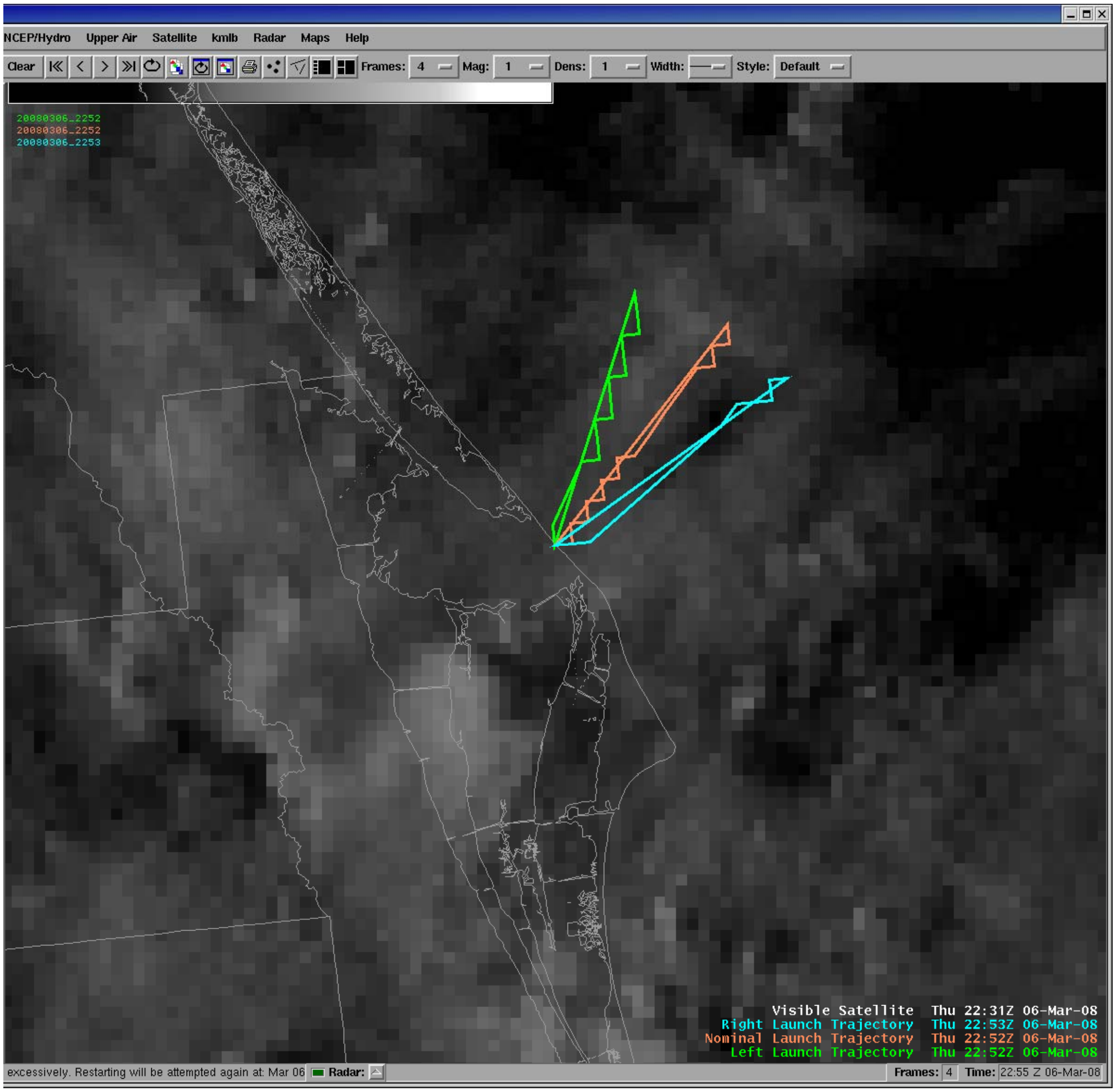
Vector Type: ● Linked ○ Unlinked

Use: ● All Points ○ Only End Points

FILES

leftipdata.txt
nomipdata.txt
rightipdata.txt

Figure 37. The AMU Trajectory Map Maker displaying the files in Launch format.



Frames: 4
Mag: 1
Dens: 1
Width:
Style: Default

20080306_2252
20080306_2252
20080306_2253

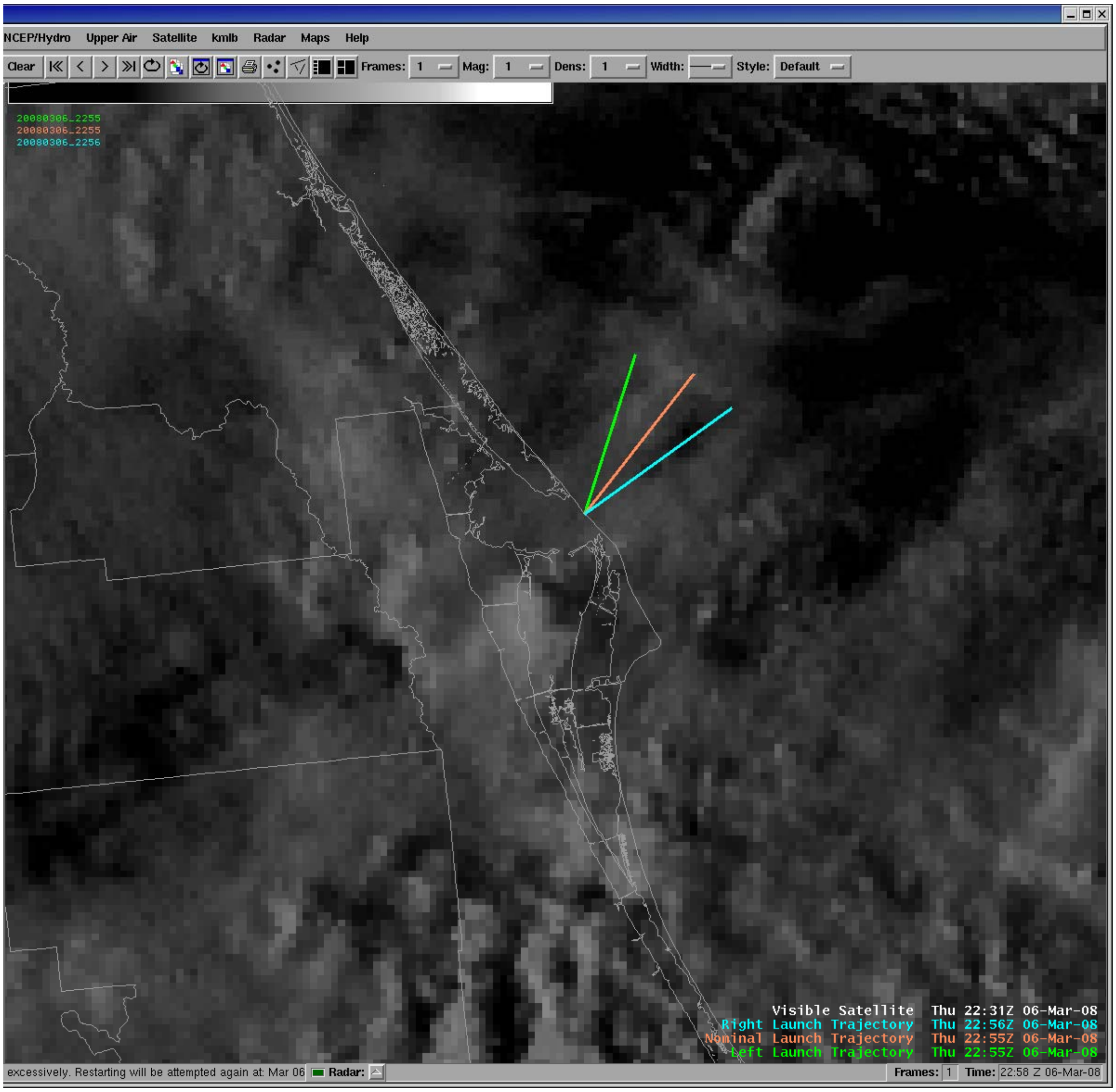
Visible Satellite Thu 22:31Z 06-Mar-08
Right Launch Trajectory Thu 22:53Z 06-Mar-08
Nominal Launch Trajectory Thu 22:52Z 06-Mar-08
Left Launch Trajectory Thu 22:52Z 06-Mar-08

excessively. Restarting will be attempted again at: Mar 06

Radar:

Frames: 4
Time: 22:55 Z 06-Mar-08

Figure 38. A plot of launch trajectories created by the AMU Trajectory Map Maker, using the All Points option.



Frames: 1 Mag: 1 Dens: 1 Width: Style: Default

20080306_2255
20080306_2255
20080306_2256

Visible Satellite Thu 22:31Z 06-Mar-08
Right Launch Trajectory Thu 22:56Z 06-Mar-08
Nominal Launch Trajectory Thu 22:55Z 06-Mar-08
Left Launch Trajectory Thu 22:55Z 06-Mar-08

excessively. Restarting will be attempted again at: Mar 06

Frames: 1 Time: 22:58 Z 06-Mar-08

Figure 39. A plot of launch trajectories created by the AMU Trajectory Map Maker, using the Only End Points option.

Appendix F

Installing the AMU VAHIRR derived products

1) Purpose of the derived products

To help interpret the results of the ABFM Comparison Test, four VAHIRR-derived products were created:

- VAHIRR Average Reflectivity
- VAHIRR Average Cloud Thickness
- VAHIRR Average Cloud Top, and
- VAHIRR Average Cloud Bottom

The derived products used intermediate values that are used in the final calculation of VAHIRR. The source code and installation files for the derived products can be requested from the AMU.

2) Creating and installing the VAHIRR Average Reflectivity product

a. Create a new directory for the new product: **`/home/codev8/src/cpc305/tsk009`**. Copy the following files from the directory that holds the VAHIRR products (**`/home/codev8/src/cpc305/tsk007`**) to the new directory: **`format_product.c`**, **`input_parms.c`**, **`makefile`**, **`vahirr_driver.c`**, **`vahirr.h`**, and **`vahirr.mak`**.

b. In the **`/home/codev8/src/cpc305/tsk009`** directory, rename the files to **`format_product2.c`**, **`input_parms2.c`**, **`vahirr_driver2.c`**, **`vahirr2.h`**, and **`vahirr2.mak`**. Do not rename the **`makefile`** file.

c. Open the **`vahirr2.h`** file for editing. Change the #ifndef and #define statements at the beginning of the file to:

```
#ifndef VAHIRR2_H

#define VAHIRR2_H
```

Next, change the value of the PCODE and VAHIRR constants to 155. Save and close the **`vahirr2.h`** file.

d. Open the **`input_parms2.c`** file for editing. In the second #include statement in the file, change the header filename to “vahirr2.h”. Save and close the **`input_parms2.c`** file.

e. Open the **`vahirr2.mak`** file for editing. Rename the filenames for the SRCS and TARGET variables. Here is a snippet of the file:

```
SRCS    =    vahirr_driver2.c \

             format_product2.c \

             input_parms2.c

TARGET = vahirr_prod2
```

Save and close the **`vahirr2.mak`** file.

f. Open the **`makefile`** file for editing. Change the value of the BINMAKEFILES variable to “vahirr2.mak”. Save and close the **`makefile`** file.

g. Open the **`vahirr_driver2.c`** file for editing. In the “include files” section, change the filename in the last #include statement to “vahirr2.h”. In the Provide_results() function, change the nested for loop to the following:

```
for (i = 0; i < MAX_ROWS; i++)
{
  for (j = 0; j < MAX_COLUMNS; j++)
  {
    if (point[i][j].vahirr_dBZkm <= -2.0)
      vahirr_color_idx[i][j] = convert_vahirr(-99.0);
    else if (point[i][j].sum_samplesizes > 0)
      vahirr_color_idx[i][j] =
convert_vahirr(point[i][j].sum_weighted_refl_avgs_dBZ/point[i][j].sum_samplesizes);
    else
      vahirr_color_idx[i][j] = convert_vahirr(0.0);

     if (HIDE) create_test_pattern();

     if (HIDE) fprintf(stderr,"(%3d,%3d)%2d ",i,j,vahirr_color_idx[i][j]);
  }
}
```

In the convert_vahirr(float vahirr_avg) function, change the if statement to the following:

```
if (vahirr_avg < -1.0)
{
  c15++;
  result = 15; /* 15 white - out of bounds or conditions failed for calculating vahirr */
}
else if (vahirr_avg <= 0.0)
{
  c0++;
  result = 0; /* 0 black - reflectivity in volume is negative or below snr threshold or
            vahirr calculated to zero */
}
else if (vahirr_avg <= 1.0)
{
  c1++;
  result = 1;
}
else if (vahirr_avg <= 2.0)
{
  c2++;
  result = 2;
}
else if (vahirr_avg <= 3.0)
{
  c3++;
  result = 3;
}
else if (vahirr_avg <= 4.0)
{
  c4++;
  result = 4;
}
else if (vahirr_avg <= 5.0)
{
  c5++;
  result = 5;
}
else if (vahirr_avg <= 6.0)
{
  c6++;
  result = 6;
}
else if (vahirr_avg <= 7.0)
{
  c7++;
  result = 7;
}
else if (vahirr_avg <= 8.0)
{
  c8++;
  result = 8;
}
else if (vahirr_avg <= 9.0)
```

```
{
  c9++;
  result = 9;
}
else if (vahirr_avg <= 10.0)
{
  c10++;
  result = 10;
}
else if (vahirr_avg <= 11.0)
{
  c11++;
  result = 11;
}
else if (vahirr_avg <= 12.0)
{
  c12++;
  result = 12;
}
else if (vahirr_avg <= 13.0)
{
  c13++;
  result = 13;
}
else /* (vahirr_avg > 13.0) */
{
  c14++;
  result = 14; /* purple - vahirr_avg greater than some maximum */
}
```

Save and close the **`vahirr_driver2.c`** file.

h. Open the **`format_product2.c`** file for editing. In the last #include statement, change the header filename to “vahirr2.h”. In the add_to_product_header function, change the statements in the “/* enter threshold values */” section to the following:

```
hdr->level_1  = (short)0x0400; /* 0 meaning missing(-999) or negative refl or vahirr of
                                  zero */
hdr->level_2  = (short)0x0000; /* 0x04="<" symbol, <= 1.0 */
hdr->level_3  = (short)0x0001; /* <= 2.0 */
hdr->level_4  = (short)0x0002; /* <= 3.0 */
hdr->level_5  = (short)0x0003; /* <= 4.0 */
hdr->level_6  = (short)0x0004; /* <= 5.0 */
hdr->level_7  = (short)0x0005; /* <= 6.0 */
hdr->level_8  = (short)0x0006; /* <= 7.0 */
hdr->level_9  = (short)0x0007; /* <= 8.0 */
hdr->level_10 = (short)0x0008; /* <= 9.0 */
hdr->level_11 = (short)0x0009; /* <= 10.0 */
hdr->level_12 = (short)0x000A; /* <= 11.0 */
hdr->level_13 = (short)0x000B; /* <= 12.0 */
hdr->level_14 = (short)0x000C; /* <= 13.0 */
hdr->level_15 = (short)0x080D; /* > 13.0 */
hdr->level_16 = (short)0x8002; /* 0x8002="ND" or no data, meaning out of bounds or failed
                                  conds in our case */
```

Save and close the **`format_product2.c`** file.

i. Go to the directory created in step a (**`cd /home/codev8/src/cpc305/tsk009`**). Enter **`make all`** at the command prompt. Verify that no error messages are output. Enter **`make install`** at the command prompt. Verify that no error messages are output.

j. Go to the **`$HOME/cfg`** directory. Open the **`product_generation_tables`** file for editing. Add an entry to the “Default_prod_gen” table, keeping the first column in numerical order. The entry should look like the following:

```
155    6    1    120    UNU    UNU    UNU    UNU    UNU    UNU
```

Save and close the **`product_generation_tables`** file.

k. Go to the **`$HOME/cfg/extensions`** directory. Open the **`product_tables.sample_snippet`** file for editing. Add an entry to the “Prod_attr_table” table. The entry should look like the following:

```
Product {
    prod_id                 155    VAHIRR2
    prod_code               155
    gen_task                30509  vahirr_prod2
    wx_modes                7
    disabled                0
    n_priority              4
    priority_list           89 89 89 89
    n_dep_prods             1
#   dependent products:     REFLDATA
    dep_prods_list          79
    desc            "VAHIRR Average Reflectivity"
    type                    0
    alert                   0
    warehoused              0
    path                    base/vahirr2.lb
    lb_n_msgs               10
    max_size                96
       params
                             2 -20 3599 0 10 "Elevation" "Degrees"
    }
```

Save and close the **`product_tables.sample_snippet`** file.

l. Open the **`task_table.sample_snippet`** file in the **`$HOME/cfg/extensions`** directory. Add an entry to the “Task_attr_table” table. The entry should look like the following:

```
Task {
    id              30509
    name            vahirr_prod2
#   in:             REFLDATA
#   out:            RASTOR PRODUCT
    input_data      79
    output_data     155
    desc            "Calculate VAHIRR2"
    args
                    0 "-v"
}
```

Add a “vahirr_prod2” entry to the “Operational_processes” section. Save and close the **`task_table.sample_snippet`** file.

3) Creating and installing the VAHIRR Average Cloud Thickness product

a. Create a new directory for the new product: **`/home/codev8/src/cpc305/tsk010`**. See step a of Section 2 of this appendix, for which files to copy to the new directory.

b. In the new directory, rename the files to **`format_product3.c`**, **`input_parms3.c`**, **`vahirr_driver3.c`**, **`vahirr3.h`**, and **`vahirr3.mak`**. Do not rename the makefile file.

c. Perform step c of Section 2 (in this appendix), except replace the “2” with a “3” and the “155” with a “156”.

d. Perform steps d-f of Section 2 (in this appendix), except replace the “2” with a “3”.

e. Open the **`vahirr_driver3.c`** file for editing. In the “include files” section, change the filename in the last #include statement to “vahirr3.h”. In the Provide_results() function, change the nested for loop to the following:

```
for (i = 0; i < MAX_ROWS; i++)
```

```
{
  for (j = 0; j < MAX_COLUMNS; j++)
  {
    if (point[i][j].vahirr_dBZkm <= -2.0)
      vahirr_color_idx[i][j] = convert_vahirr(-99.0);
    else if (point[i][j].top_hght_samplesz > 0 && point[i][j].bot_hght_samplesz > 0)
    {
      avg_top = point[i][j].sum_top_hghts_km / point[i][j].top_hght_samplesz;
      avg_bot = point[i][j].sum_bot_hghts_km / point[i][j].bot_hght_samplesz;
      vahirr_color_idx[i][j] = convert_vahirr(avg_top - avg_bot);
    }
    else
      vahirr_color_idx[i][j] = convert_vahirr(0.0);

     if (HIDE) create_test_pattern();

     if (HIDE) fprintf(stderr,"(%3d,%3d)%2d ",i,j,vahirr_color_idx[i][j]);
  }
}
```

In the convert_vahirr(float vahirr_thick) function, change the if statement to reflect the data levels in the VAHIRR Average Cloud Thickness product – see Table 3. Save and close the **`vahirr_driver3.c`** file.

f. Open the **`format_product3.c`** file for editing. In the last #include statement, change the header filename to "vahirr3.h". In the add_to_product_header function, change the statements in the "/* enter threshold values */" section to reflect the data levels in the VAHIRR Average Cloud Thickness product – see Table 3. Save and close the **`format_product3.c`** file.

g. Perform step i of Section 2 (of this appendix), except replace "tsk009" with "tsk010".

h. Perform step j of Section 2 (of this appendix), except replace "155" with "156".

i. Perform step k of Section 2 (of this appendix), except replace "155" with "156", "2" with "3", "30509" with "30510", and "Reflectivity" with "Thickness".

j. Perform step l of Section 2 (of this appendix), except replace "155" with "156", "2" with "3", and "30509" with "30510".

4) Creating and installing the VAHIRR Average Cloud Top product

a. Create a new directory for the new product: **`/home/codev8/src/cpc305/tsk011`**. See step a of Section 2 (of this appendix) on which files to copy to the new directory.

b. In the new directory, rename the files to **`format_product4.c`**, **`input_parms4.c`**, **`vahirr_driver4.c`**, **`vahirr4.h`**, and **`vahirr4.mak`**. Do not rename the makefile file.

c. Perform step c of Section 2 (of this appendix), except replace the "2" with a "4" and the "155" with a "157".

d. Perform steps d-f of Section 2 (of this appendix), except replace the "2" with a "4".

e. Open the **`vahirr_driver4.c`** file for editing. In the "include files" section, change the filename in the last #include statement to "vahirr4.h". In the Provide_results() function, change the nested for loop to the following:

```
for (i = 0; i < MAX_ROWS; i++)
{
  for (j = 0; j < MAX_COLUMNS; j++)
  {
    if (point[i][j].vahirr_dBZkm <= -2.0)
      vahirr_color_idx[i][j] = convert_vahirr(-99.0);
    else if (point[i][j].top_hght_samplesz > 0)
    {
      avg_top = point[i][j].sum_top_hghts_km / point[i][j].top_hght_samplesz;
      vahirr_color_idx[i][j] = convert_vahirr(avg_top);
    }
```

```
    else
      vahirr_color_idx[i][j] = convert_vahirr(0.0);

    if (HIDE) create_test_pattern();

    if (HIDE) fprintf(stderr,"(%3d,%3d)%2d ",i,j,vahirr_color_idx[i][j]);
    }
}
```

In the convert_vahirr(float vahirr_thick) function, change the if statement to reflect the data levels in the VAHIRR Average Cloud Top product – see Table 3. Save and close the **`vahirr_driver4.c`** file.

f. Open the **`format_product4.c`** file for editing. In the last #include statement, change the header filename to "vahirr4.h". In the add_to_product_header function, change the statements in the "/* enter threshold values */" section to reflect the data levels in the VAHIRR Average Cloud Top product – see Table 3. Save and close the **`format_product4.c`** file.

g. Perform step i of Section 2 (of this appendix), except replace "tsk009" with "tsk011".

h. Perform step j of Section 2 (of this appendix), except replace "155" with "157".

i. Perform step k of Section 2 (of this appendix), except replace "155" with "157", "2" with "4", "30509" with "30511", and "Reflectivity" with "Top".

j. Perform step l of Section 2 (of this appendix), except replace "155" with "157", "2" with "4", and "30509" with "30511".

5) Creating and installing the VAHIRR Average Cloud Bottom product

a. Create a new directory for the new product: **`/home/codev8/src/cpc305/tsk012`**. See step a of Section 2 (of this appendix) on which files to copy to the new directory.

b. In the new directory, rename the files to **`format_product5.c`**, **`input_parms5.c`**, **`vahirr_driver5.c`**, **`vahirr5.h`**, and **`vahirr5.mak`**. Do not rename the makefile file.

c. Perform step c of Section 2 (of this appendix), except replace the "2" with a "5" and the "155" with a "158".

d. Perform steps d-f of Section 2 (of this appendix), except replace the "2" with a "5".

e. Open the **`vahirr_driver5.c`** file for editing. In the "include files" section, change the filename in the last #include statement to "vahirr5.h". In the Provide_results() function, change the nested for loop to the following:

```
for (i = 0; i < MAX_ROWS; i++)
{
  for (j = 0; j < MAX_COLUMNS; j++)
  {
    if (point[i][j].vahirr_dBZkm <= -2.0)
      vahirr_color_idx[i][j] = convert_vahirr(-99.0);
    else if (point[i][j].bot_hght_samplesz > 0)
    {
      avg_bottom = point[i][j].sum_bot_hghts_km / point[i][j].bot_hght_samplesz;
      vahirr_color_idx[i][j] = convert_vahirr(avg_bottom);
    }
    else
      vahirr_color_idx[i][j] = convert_vahirr(0.0);

      if (HIDE) create_test_pattern();

      if (HIDE) fprintf(stderr,"(%3d,%3d)%2d ",i,j,vahirr_color_idx[i][j]);
  }
}
```

In the convert_vahirr(float vahirr_thick) function, change the if statement to reflect the data levels in the VAHIRR Average Cloud Bottom product – see Table 3. Save and close the **`vahirr_driver5.c`** file.

f. Open the **`format_product5.c`** file for editing. In the last #include statement, change the header filename to “vahirr5.h”. In the add_to_product_header function, change the statements in the “/* enter threshold values */” section to reflect the data levels in the VAHIRR Average Cloud Bottom product – see Table 3. Save and close the **`format_product5.c`** file.

g. Perform step i of Section 2 (of this appendix), except replace “tsk009” with “tsk012”.

h. Perform step j of Section 2 (of this appendix), except replace “155” with “158”.

i. Perform step k of Section 2 (of this appendix), except replace “155” with “158”, “2” with “5”, “30509” with “30512”, and “Reflectivity” with “Bottom”.

j. Perform step l of Section 2 (of this appendix), except replace “155” with “158”, “2” with “5”, and “30509” with “30512”.

References

Dye, J.E., S. Lewis, M.G. Bateman, D. M. Mach, F. J. Merceret, J. G. Ward and C. A. Grainger, 2004: Final Report on the Airborne Field Mill Project (ABFM) 2000-2001 Field Campaign, NASA Technical Memorandum NASA/TM-2004-211534, November, 2004, 132 pp.

Dye, J. E., M. G. Bateman, D. M. Mach, C. A. Grainger, H. J. Christian, H. C. Koons, E. P. Krider, F. J. Merceret, and J. C. Willet, 2006: The Scientific Basis for a Radar-Based Lightning Launch Commit Criterion for Anvil Cloud. Preprints, *12th Conf. on Aviation, Range and Aerospace Meteorology,* Atlanta, GA, Amer. Meteor. Soc., 4 pp. [Available online at: http://ams.confex.com/ams/pdfpapers/100563.pdf]

Dye, J. E., M. G. Bateman, H. J. Christian, E. Defer, C. A. Grainger, W. D. Hall, E. P. Krider, S. A. Lewis, D. M. Mach, F. J. Merceret, and J. C. Willet, and P. T. Willis, 2007: Electric fields, cloud microphysics and reflectivity in anvils of Florida thunderstorms, *J. Geophys. Res*., **112**, D11215, doi10.1029/2006JD007550.

Krider, E. P., H. J. Christian, J. E. Dye, H. C. Koons, J. Madura, F. Merceret, W. D. Rust, R. L. Walterscheid, and J. C. Willett, 2006: Natural and Triggered Lightning Launch Commit Criteria. Preprints, *12th Conf. on Aviation, Range and Aerospace Meteorology,* Atlanta, GA, Amer. Meteor. Soc., 5 pp. [Available online at: http://ams.confex.com/ams/pdfpapers/104060.pdf]

Merceret, F. J., M. McAleenan, T. M. McNamara, J. W. Weems, and W. P. Roeder, 2006: Implementing the VAHIRR Launch Commit Criteria Using Existing Radar Products. Preprints, *12th Conf. on Aviation, Range and Aerospace Meteorology*, Atlanta, GA, Amer. Meteor. Soc., 8 pp. [Available online at: http://ams.confex.com/ams/pdfpapers/98076.pdf]

NOAA Earth System Research Laboratory, Global Systems Division. *DARE Graphics Metafile (DGM) Instruction Format,* 2006. Unpublished. [Request from NOAA Earth System Research Laboratory, Global Systems Division, Information Systems Branch, Joe Wakefield, joseph.s.wakefield@noaa.gov]

Office of the Federal Coordinator for Meteorological Services and Supporting Research, December 2005: Federal Meteorological Handbook No. 11 Doppler Radar Meteorological Observations, Part B, Doppler Radar Theory and Meteorology, Washington, D.C., 219 pp.

Office of the Federal Coordinator for Meteorological Services and Supporting Research, April 2006: Federal Meteorological Handbook No. 11 Doppler Radar Meteorological Observations, Part C, WSR-88D Products and Algorithms, Washington, D.C., 390 pp.

Roeder, W. P., D. L. Hajek, F. C. Flinn, G. A. Maul, and M. E. Fitzpatrick Meteorological And Oceanic Instrumentation At Spaceport Florida–Opportunities For Coastal Research, *Fifth Conference on Coastal Atmospheric and Oceanic Prediction and Processes*, 6-8 Aug 2003, Seattle, WA, 132-137, 6 pp. [Available online at: http://ams.confex.com/ams/pdfpapers/64065.pdf]

Roeder, W. P., and T. M. McNamara, 2006: A Survey of the Lightning Launch Commit Criteria, *2nd Conf. on Meteorological Applications of Lightning Data*, 29 Jan-2 Feb 2006, Atlanta, GA, 16 pp. [Available online at: http://ams.confex.com/ams/pdfpapers/103803.pdf]

Willett, J. C., H. C. Koons, E. P. Krider, R. L. Walterscheid, and W. D. Rust, 1999: Natural and Triggered Lightning Launch Commit Criteria (LLCC), Aerospace Report A923563, Aerospace Corp., El Segundo, CA, 23 pp.

List of Acronyms

Term	**Description**
45 WS	45th Weather Squadron
ABFM	Airborne Field Mill
AGL	Above Ground Level
AMU	Applied Meteorology Unit
AWIPS	Advanced Weather Interactive Processing System
CAPPI	Constant Altitude Plan Position Indicator
CCAFS	Cape Canaveral Air Force Station
CODE	Common Operations and Development Environment
CVG	CODEview Graphics
DARE	Denver AWIPS Risk Reduction and Requirements Evaluation
DGM	DARE Graphics Metafile
DOP	Deorbit Opportunity Map
FAT	Factory Acceptance Test
FMH	Federal Meteorological Handbook
FR	Flight Rules
GUI	Graphical User Interface
KMLB	Melbourne, FL
KSC	Kennedy Space Center
LAP	Lightning Advisory Panel
LDM	Local Data Manager
LLCC	Lightning Launch Commit Criteria
MSFC	Marshall Space Flight Center
MSL	Mean Sea Level
NWS	National Weather Service
ORPG	Open Radar Product Generator
PAFB	Patrick Air Force Base
PDP	product dependent parameters
PUP	Principal User Processor
RDA	Radar Data Acquisition
SMG	Spaceflight Meteorology Group
Tcl/Tk	Tool Command Language/Tool Kit
VAHIRR	Volume-Averaged Height Integrated Radar Reflectivity
VCP	Volume Coverage Pattern
WSR-74C	Weather Surveillance Radar 1974-C
WSR-88D	Weather Surveillance Radar 1988-Doppler

NOTICE