DEPARTMENT OF COMMERCE
UNITED STATES OF AMERICA

NOAA Technical Memorandum ERL ESG-3

PROFS' DOPPLER RADAR DATA PROCESSING SYSTEM, PART I: DESIGN AND IMPLEMENTATION WITH APPLICATION TO NEXRAD ALGORITHMS

Program for Regional Observing and Forecasting Services
Boulder, Colorado
September 1983

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / Environmental Research Laboratories

NOAA Technical Memorandum ERL ESG-3

PROFS' DOPPLER RADAR DATA PROCESSING SYSTEM, PART I: DESIGN AND IMPLEMENTATION WITH APPLICATION TO NEXRAD ALGORITHMS

Ron L. Alberty
F. James Holitza
Margot H. Ackley
John R. Smart
Robert C. Lipschutz
William F. Roberts
Kenneth J. Leap

Program for Regional Observing and Forecasting Services

Environmental Sciences Group
Boulder, Colorado
September 1983

UNITED STATES DEPARTMENT OF COMMERCE

Malcolm Baldrige, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

John V. Byrne, Administrator

Environmental Research Laboratories

Vernon E. Derr
Acting Director

NOTICE

CONTENTS

Page

Page

FIGURES

TABLES

PROFS' DOPPLER RADAR DATA PROCESSING SYSTEM,

PART I: DESIGN AND IMPLEMENTATION WITH APPLICATION TO NEXRAD ALGORITHMS

Ron L. Alberty, F. James Holitza, Margot H. Ackley, John R. Smart

Robert C. Lipschutz, William F. Roberts, Kenneth J. Leap

ABSTRACT. The Program for Regional Observing and Forecasting Services (PROFS) has designed and implemented a processing and display system for Doppler radar data. This system processes a set of meteorological algorithms developed from algorithm descriptions provided by the Next Generation Weather Radar (NEXRAD) Joint System Program Office (JSPO).

The methodology used to develop this system is discussed, and details of the computer hardware and software are given. Hardware characteristics of the radar and the implementation strategy are reviewed. For each algorithm, there is a detailed discussion of its meteorological purpose along with a summary of testing procedures and on-site evaluation.

1. INTRODUCTION

The mission of the Program for Regional Observing and Forecasting Services (PROFS) is to improve local weather services through introduction of new technologies into weather service operations. The first 4-year phase of PROFS, ending in October 1983, addressed two important tasks. The first was to acquire, interactively manipulate, and display conventional meteorological data. This included existing satellite data, conventional National Weather Service (NWS) radar data, and data from a mesoscale surface network. The purpose of this first task was to improve the forecaster's ability to give timely and accurate severe weather warnings. The second task was to prepare for the incorporation of experimental data sets. These included VAS satellite data, Doppler radar data, and surface-based microwave temperature, moisture, and wind profile data. The purpose of this second task was to further enhance the tools available to the forecaster (Reynolds, 1983; PROFS, 1980; Beran and MacDonald, 1981). In fiscal year 1984, PROFS will enter its second phase. One major objective of this second phase is to obtain in real time the complete set of Doppler data for each volume scan.

It was mutual interests of PROFS and the Next Generation Weather Radar (NEXRAD) Program that led to the effort reported here (Alberty, 1982). The NEXRAD Joint System Program Office (JSPO) is located in the National Weather

Service (NWS) and includes personnel from the NWS, the Air Weather Service (AWS), and the Federal Aviation Administration (FAA). Its mission is to manage the specification, procurement, and installation of the next generation of weather radars to be used by the participating agencies (NEXRAD, 1983d). To accomplish this, JSPO must not only develop complete hardware specifications but also designate software which will generate understandable products from voluminous amounts of raw Doppler data. The principal source of the processes, or algorithms, which generate these products is the atmospheric science research community. It is the responsibility of JSPO to determine which products are needed and to fully define the algorithms which produce these products in such a way that they can be programmed. The JSPO chose a method for specifying these algorithms defined as Algorithm Enunciation Language (AEL) which is a structured language or pseudocode (NEXRAD, 1983b).

PROFS agreed to code and test a subset of the chosen algorithms which generate products from Doppler data (Roberts et al., 1983). These algorithms are designed to (1) identify storms and their characteristics, and predict their motions, (2) detect mesocyclones, and (3) produce an analysis of the velocity field at a specified distance from the radar. It was agreed that this code would be tested on Doppler data acquired by PROFS using the National Center for Atmospheric Research (NCAR) CP-2 10-cm Doppler radar. Because one principal objective was to test the feasibility of writing executable code from the AEL statements of the algorithms, it was determined that interaction between PROFS and NEXRAD would be controlled and documented. PROFS accomplished this by formally submitting 51 questions regarding the algorithms through a computer conference link.

It was also agreed that NEXRAD would receive two deliverables from PROFS: a final report and an on-site evaluation. The on-site evaluation was conducted in June and consisted of five 1½-day presentations and demonstrations. Participating were NEXRAD employees, representatives from National Weather Service, Air Weather Service, and FAA, as well as source scientists and NEXRAD contractors. This document completes the initial agreement.

2. ALGORITHM DESCRIPTIONS

2.1 Overview

NEXRAD JSPO had identified 15 algorithms as candidates for the initial implementation of the system. PROFS was contracted to code and exercise seven of these. An eighth algorithm, Storm Segments, was later determined necessary as a preprocessing step for the Storm Processing subsystem. All algorithms identified for the NEXRAD project fall into two categories: Storm and Non-Storm. Of the eight algorithms coded by PROFS, only one, the Velocity-Azimuth Display, was from the Non-Storm category.

Six of the algorithms coded by PROFS make up the Storm Processing subsystem and are subsequently referred to as the Storm Sequence. These algorithms use principally reflectivity data; they are

Storm Segments
Storm Centroids
Storm Tracking
Storm Forecast
Storm Structure
Hail.

The remaining two algorithms rely on Doppler velocity data; they are

Velocity-Azimuth Display (VAD)
Mesocyclone Detection.

The Storm Forecast algorithm was later re-named Storm Position Forecast.

PROFS coded all eight algorithms from specifications provided by the individual algorithm descriptions furnished by NEXRAD. Copies of these descriptions appear in the NEXRAD Algorithm Report (1983c). The versions described in that report are either exactly those coded or slightly improved versions. For reader convenience and reference, the complete Storm Centroids Algorithm Description has been included in Appendix A. In addition to coding the eight algorithms, PROFS designed and wrote software to generate special image displays of the Doppler data used to exercise the algorithms. These displays subsequently played an important role when individual algorithm output product results were compared with the input Doppler image data.

2.2 Storm Sequence - Introduction

Storm Sequence is a set of six algorithms which make up the Storm Processing Subsystem. When executed in the proper sequence, these algorithms detect and characterize storm cells, and track and predict movement of identified cells.

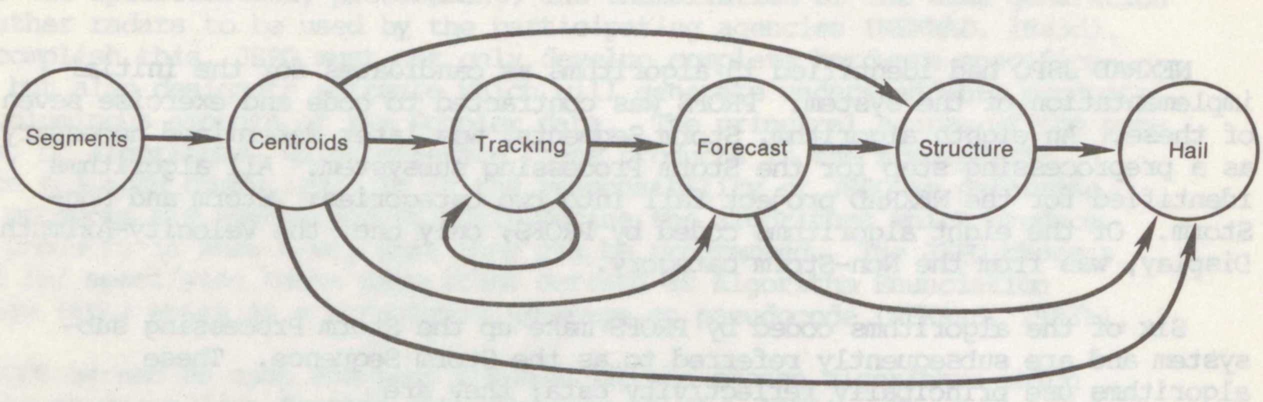


Figure 1. Data flow relationship of the six algorithms making up the Storm Sequence.

Table 1. Summary of Processes and Output Products of the Six Algorithms Making Up the Storm Sequence

Algorithm	Processes On	Produces
Segments	Radials	Segments
Centroids	Azimuths θ Elevations Φ	Components, storms
Tracking	Present/past times	Track positions
Forecast	Past/present positions	Future positions, speed, direction
Structure	Components, storms, speed, direction	Structure characteristics
Hail	Structure characteristics, speed, direction	Hail probability

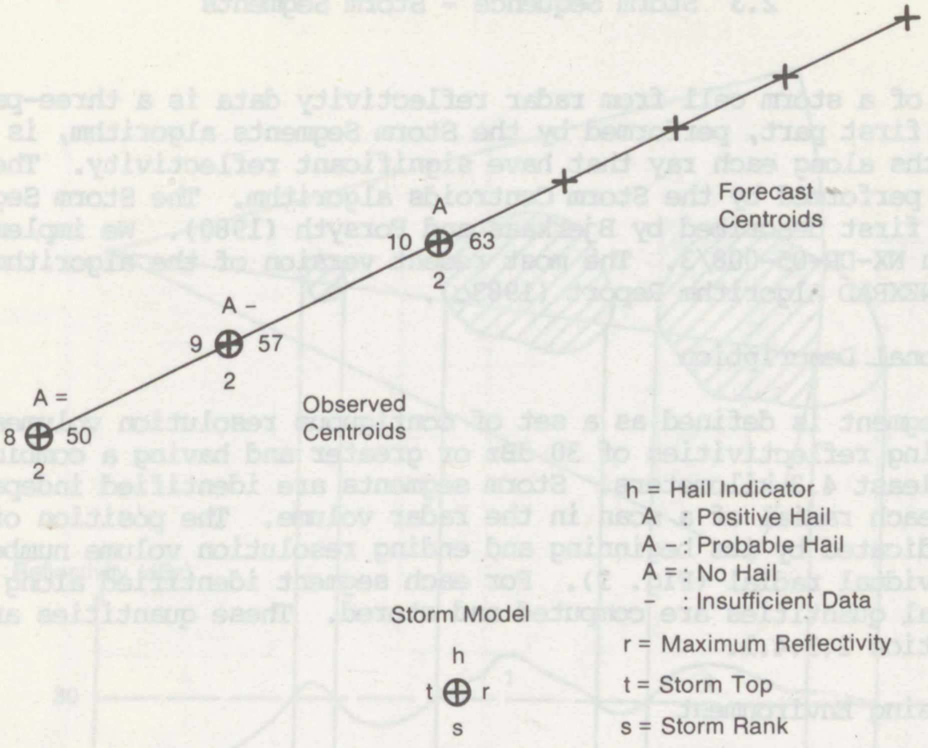


Figure 2. Example of a composite graphic display used for the algorithms of the Storm Sequence.

The algorithms are interrelated in that input data to certain algorithms are output product data from one or more other algorithms. Figure 1 shows the data flow relationship of the Storm Sequence. Because of this data flow dependency, these six algorithms are often treated as a single subsystem. However, each algorithm will be described as an individual entity in the following sections. Table 1 is included to aid understanding of data progression through the six algorithms and their interdependency.

Although each algorithm description discusses its individual output products and displays, Figure 2 is presented to show a typical composite output display used throughout the Storm Sequence. The individual observed centroid positions are determined by Storm Segments and Storm Centroids algorithms, and their placements relative to time are determined by Storm Tracking. The forecast centroid positions are determined by the Storm Position Forecast algorithm. The numbers surrounding the observed centroids are information determined by Storm Centroids and Storm Structure, and the letters by the Hail algorithm.

For a more detailed description of the methodology used in the Storm Sequence algorithms, see Bjerkaas and Forsyth (1980) and Boak et al. (1977).

2.3 Storm Sequence - Storm Segments

Detection of a storm cell from radar reflectivity data is a three-part process. The first part, performed by the Storm Segments algorithm, is to identify lengths along each ray that have significant reflectivity. The other two parts are performed by the Storm Centroids algorithm. The Storm Segments algorithm was first described by Bjerkaas and Forsyth (1980). We implemented NEXRAD version NX-DR-05-008/3. The most recent version of the algorithm is found in the NEXRAD Algorithm Report (1983c).

2.3.1 Functional Description

A storm segment is defined as a set of contiguous resolution volumes along a radial, having reflectivities of 30 dBz or greater and having a combined length of at least 4.2 kilometers. Storm segments are identified independently along each radial of a scan in the radar volume. The position of each segment is indicated by its beginning and ending resolution volume number along an individual radial (Fig. 3). For each segment identified along a radial, several quantities are computed and stored. These quantities are listed in Section 2.3.4.1.

2.3.2 Processing Environment

The Storm Segments algorithm is the interface between the radar data stored in computer disk files and the other algorithms of the Storm Sequence. Because of this, the reflectivity, velocity, and the coherent estimate, R(tau), are acquired for each ray processed. The software can process volume scans with minimum resolution of 0.5° beam width (720 rays/sweep) and with as many as 20 elevation angles. Once the entire volume scan has been processed to identify storm segments, the software then calculates a number of quantities for each segment and writes this information to the Storm Segments product file. This information is used by the Storm Centroids algorithm to complete the process of locating the storm centroid positions.

2.3.3 Input Requirements

The software for Storm Segments requires two basic types of input: Doppler data and processing parameters.

2.3.3.1 Doppler Data. The radar data include Doppler velocity in meters/second, the normalized coherent power (R(tau)), and reflectivity in dBz. The first two quantities are acquired and stored by this software because they are needed by other algorithms of the Storm Sequence.

2.3.3.2 Processing Parameters. Two parameters are required from the Parameter File: the significant reflectivity threshold and the segment length criterion. A listing of their values is given in the Storm Segments section (SEG) of the Parameter File (see Appendix J).

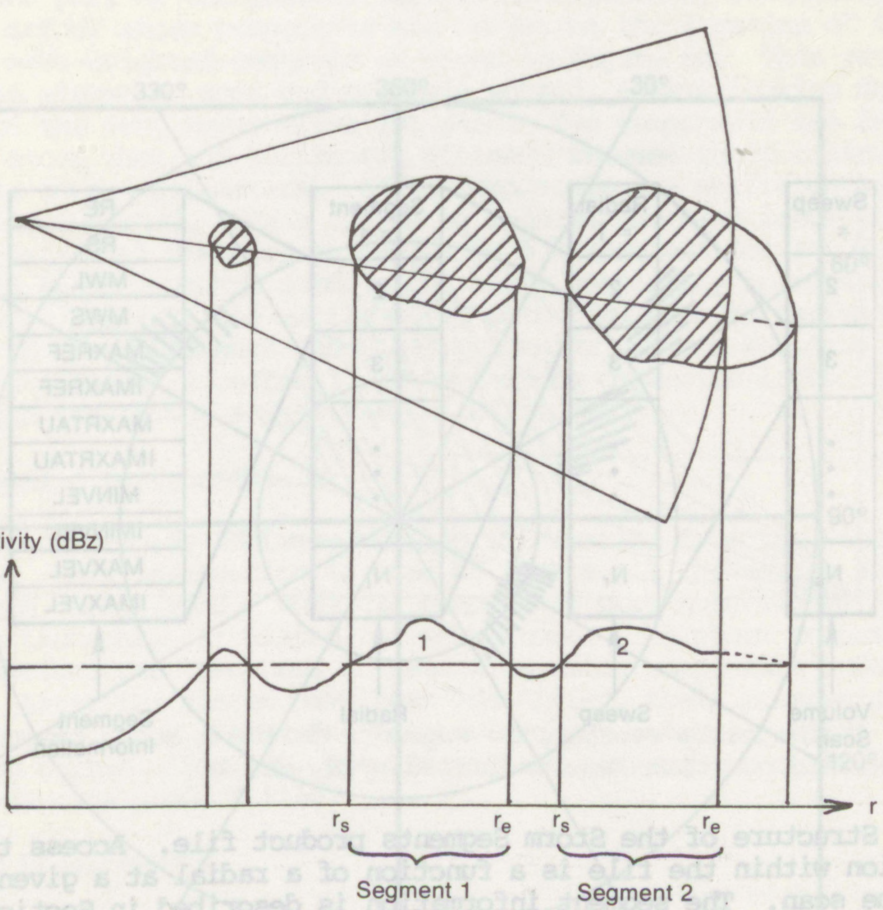


Figure 3. Along an individual radial are three lengths of 30 dBz or greater reflectivity. Only two of these meet criteria and qualify as segments as indicated by the starting (r_s) and ending (r_e) resolution volumes.

2.3.4 Output Information

Output from the Storm Segments algorithm is stored in the Storm Segments Product File in a compressed format as shown in Figure 4. The file can have a variable number of records depending on the number of segments identified in the volume. Each record has a fixed length and contains information for only one segment. This file structure allows easy access to the quantities stored for each segment.

2.3.4.1 Products. Each record of the output file contains information for one segment. Those quantities stored include the beginning and ending resolution volume number; mass-weighted length; mass-weighted length squared; and maxima and minima of reflectivity, velocity, and R(tau).

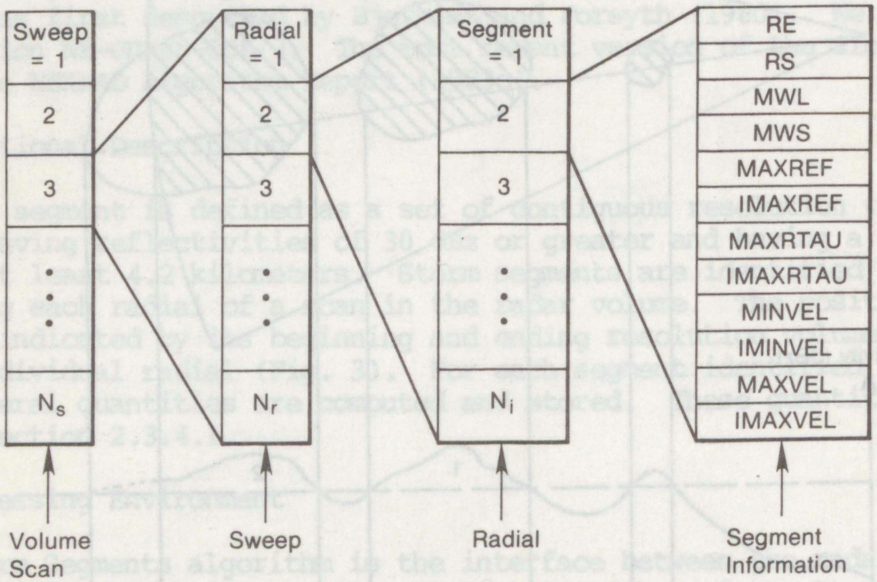


Figure 4. Structure of the Storm Segments product file. Access to segment information within the file is a function of a radial at a given sweep in the volume scan. The segment information is described in Section 2.3.4.1.

2.3.4.2 Displays. The segments that have been identified on each scan in the volume can be displayed on a graphics monitor by the Storm Segments display software (Fig. 5). This product was not specified by the algorithm description as a required output; however, it was valuable in implementing and testing the software.

2.4 Storm Sequence - Storm Centroids

The Storm Centroids algorithm completes the process of storm identification in locating the position of the storm's center of mass. The algorithm orders the storms by intensity using their mass-weighted volume, and stores the information for use by the remaining algorithms of the Storm Sequence. The Storm Centroids algorithm was first implemented by Bjerkaas and Forsyth (1980). We implemented NEXRAD version NX-DR-03-005/19. The most current version can be found in the NEXRAD Algorithm Report (1983c).

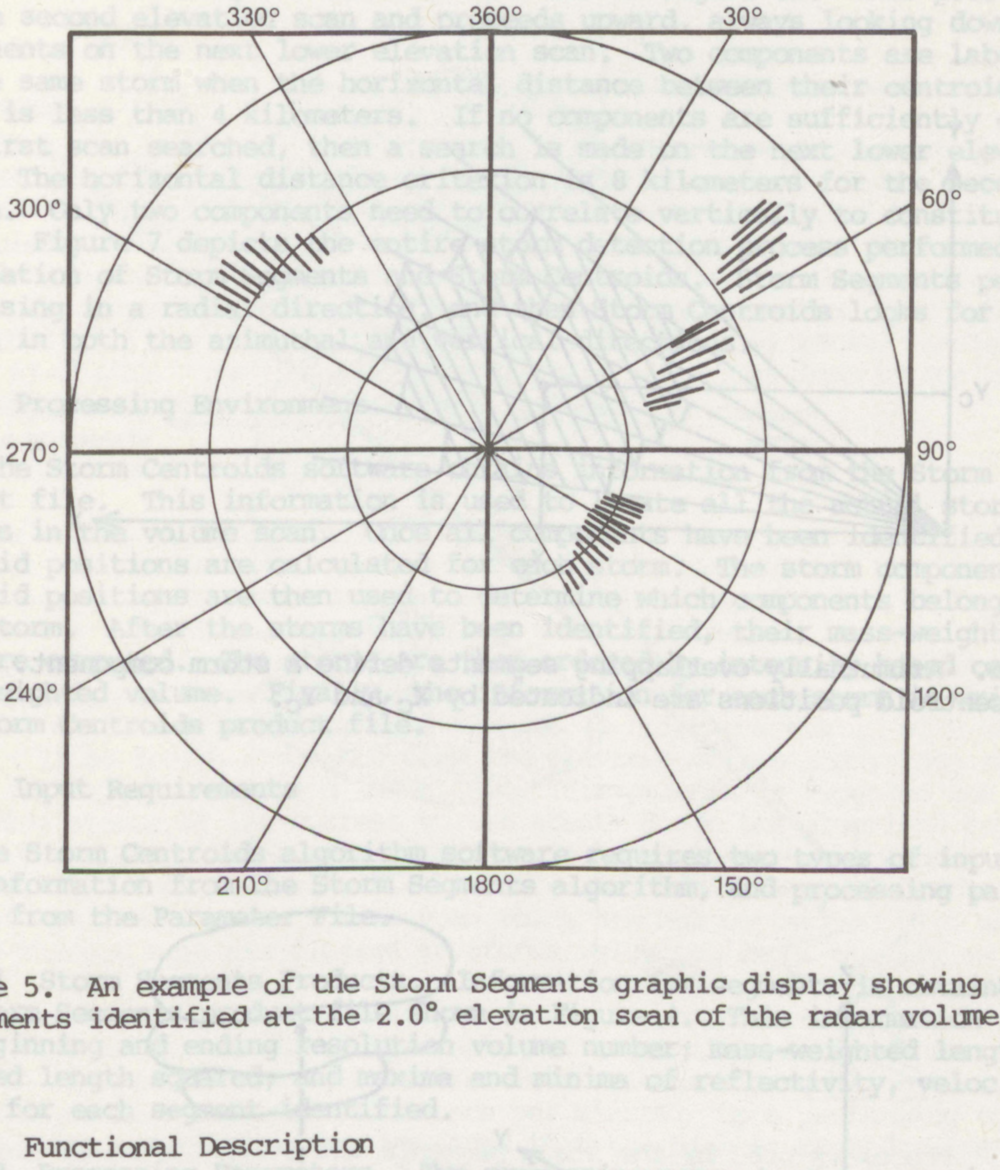


Figure 5. An example of the Storm Segments graphic display showing the segments identified at the 2.0° elevation scan of the radar volume.

2.4.1 Functional Description

The determination of a storm centroid is a two-part process. In the first part, storm segments are correlated in the azimuthal direction and identified as storm components. This process begins with the search for overlapping segments from two adjacent radials. If any two segments overlap by 1.5 kilometers or more, they are labeled part of the same "possible" storm component. Possible storm components having a sufficient number of storm segments are labeled "actual" storm components. The mass-weighted centroid position is computed for each actual storm component (Fig. 6a). Storm components are determined one elevation scan at a time.

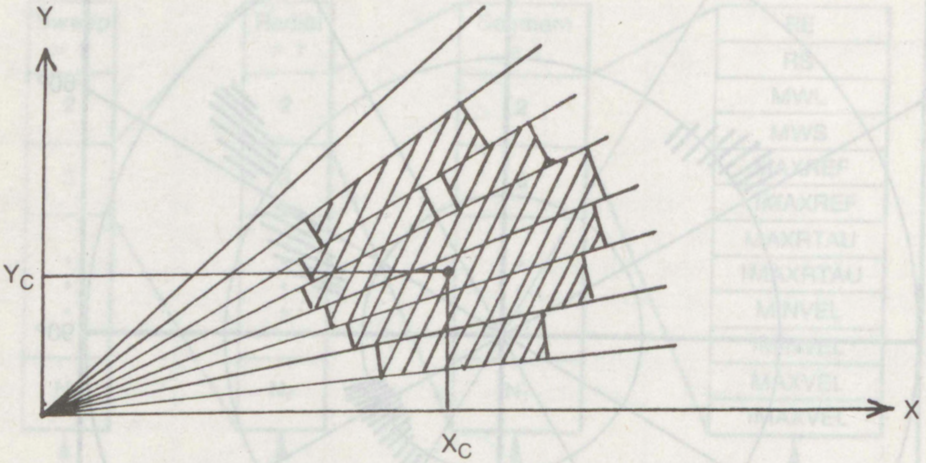


Figure 6a. Azimuthally overlapping segments define a storm component. The X- and Y-centroid positions are indicated by X_C and Y_C.

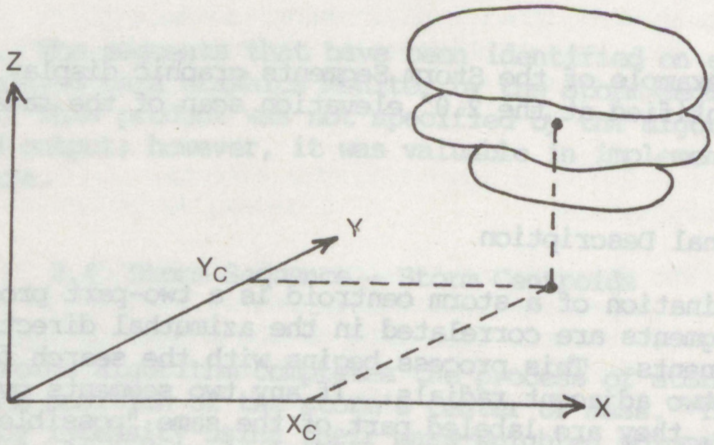


Figure 6b. Vertically overlapping storm components define a storm. The X- and Y-centroid positions are indicated by X_C and Y_C.

The second part of the process identifies a storm by vertically correlating actual storm components and computing the location of the three-dimensional mass-weighted centroid as shown in Figure 6b. This process begins at the second elevation scan and proceeds upward, always looking down at components on the next lower elevation scan. Two components are labeled part of the same storm when the horizontal distance between their centroid positions is less than 4 kilometers. If no components are sufficiently close on the first scan searched, then a search is made on the next lower elevation scan. The horizontal distance criterion is 8 kilometers for the second search. Only two components need to correlate vertically to constitute a storm. Figure 7 depicts the entire storm detection process performed by the combination of Storm Segments and Storm Centroids. Storm Segments performs processing in a radial direction, and then Storm Centroids looks for potential storms in both the azimuthal and vertical directions.

2.4.2 Processing Environment

The Storm Centroids software obtains information from the Storm Segments product file. This information is used to locate all the actual storm components in the volume scan. Once all components have been identified, the centroid positions are calculated for each storm. The storm component centroid positions are then used to determine which components belong to the same storm. After the storms have been identified, their mass-weighted volumes are computed. The storms are then ordered by intensity based on their mass-weighted volume. Finally, the information for each storm is written to the Storm Centroids product file.

2.4.3 Input Requirements

The Storm Centroids algorithm software requires two types of input: product information from the Storm Segments algorithm, and processing parameter values from the Parameter File.

2.4.3.1 Storm Segments Products. Information for segments is obtained from the Storm Segments product file shown in Figure 4. This information includes the beginning and ending resolution volume number; mass-weighted length; mass-weighted length squared; and maxima and minima of reflectivity, velocity, and R(tau) for each segment identified.

2.4.3.2 Processing Parameters. Two processing parameters are required inputs for the Storm Centroids algorithm software. These quantities are obtained from the Storm Centroid section (CEN) of the Parameter File (see Appendix J) and are the minimum overlap range for segments on adjacent azimuths and the maximum horizontal distance between centroids on successive elevation scans.

2.4.4 Output Information

The algorithm software produces one product file for each volume scan processed. Because of the interdependence of the Storm Sequence algorithms, some additional information to allow processing by the remaining algorithms is included in the file. This information includes an array indicating which storm components belong to a particular storm, and an array which contains the elevation scan angle on which each component is found.

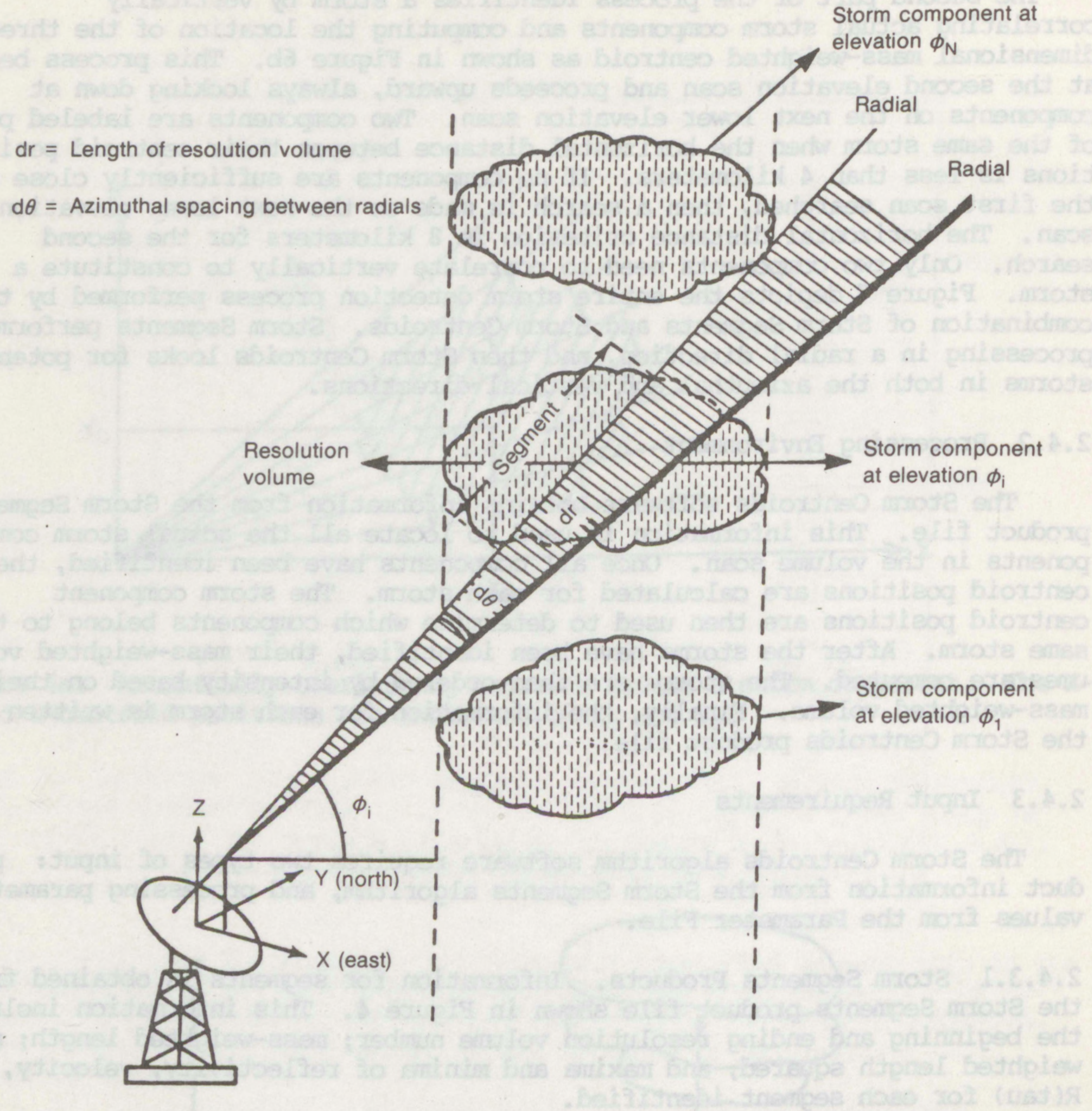


Figure 7. Processes used by the Storm Segments and Storm Centroids algorithms for the detection of storms.

2.4.4.1 Products. The complete set of products from the algorithm is available in tabular form through the Storm Centroids product display software. This information includes storm X- and Y-centroid positions, the storm rank, mass-weighted volume, number of components in the storm, the elevation angles where the components are found, component X- and Y-centroid positions, component areas, component heights above ground level, and component maxima of reflectivity, velocity, and R(tau).

2.4.4.2 Displays. The Storm Centroids software displays the Centroids product information on a graphics monitor as shown in Figure 2. An indicator marks the centroid position of the storm, and its numerical rank is written below this.

2.5 Storm Sequence - Storm Tracking

The Storm Tracking algorithm is the third step of the Storm Sequence. Its purpose is to monitor the movement of storm cells identified by Storm Centroids. For each volume scan, Storm Tracking correlates X- and Y-centroid positions to centroid positions of storms identified in the previous volume scan. The algorithm employs a tracking technique described by Bjerkaas and Forsyth (1980) and Boak et al. (1977). We implemented NEXRAD version NX-DR-03-004/13. The current Storm Tracking algorithm description may be found in the NEXRAD Algorithm Report (1983c).

2.5.1 Functional Description

The method used for correlating storms over time is relatively simple. For each volume scan processed, Storm Centroids has ordered the isolated storms according to their decreasing rank. Starting with the storm of highest rank in the current volume scan, Storm Tracking searches in order of decreasing rank through the storms of the previous volume scan. The first storm encountered, whose centroid lies within a square box centered on the centroid position of the current storm, is flagged as being correlated with the current storm. If no storms are found from the previous volume scan, then the current storm is flagged as being a new storm. This process is repeated for the remaining storms of the current volume scan. Since the algorithm requires unique correlation, storms from the previous volume scan which have already been correlated are not eligible for correlation with another storm. Finally, storms from the previous volume scan which are not correlated with any storm of the current scan are flagged as storms which no longer qualify as storms.

Figure 8 is an example of a volume scan in which three storms are identified. Each storm determines the center of a correlation box whose size is a function of the time interval between volume scans and is determined assuming a maximum storm speed of 4 kilometers per minute. Upon performing the correlation process for this case, we found that the highest ranked current storm was correlated with the second highest of the previous volume scan, the second highest of the current volume scan with the highest of the previous, the third highest of the current volume scan was determined to be a new storm, and the third highest of the previous volume scan had disappeared.

2.5.2 Processing Environment

When Storm Tracking is executed for a volume scan, Storm Segments and Storm Centroids must be executed first. Furthermore, two sequentially processed volume scans are required in order for the algorithm to establish any storm tracks. If only one scan is available (as at the beginning of a data set) Storm Tracking simply notes this and labels all storms as new ones. As

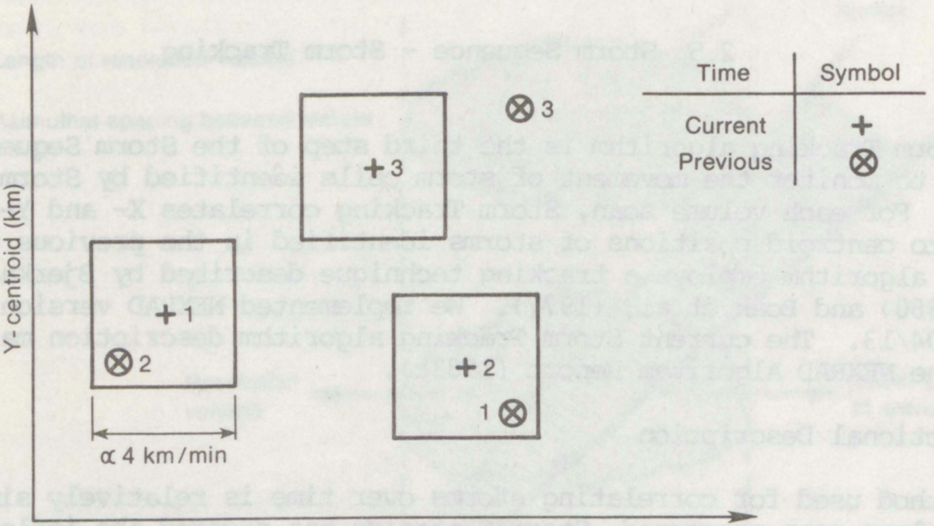


Figure 8. Volume scan containing three storms (+n). Each storm defines the center of a correlation box whose side is proportional to a storm movement speed of 4 kilometers per minute. Two storms, (⊗1, ⊗2), from the previous volume scan correlated with the current volume scan.

subsequent volume scans become available from Storm Centroids, the algorithm combines the Centroids output with previous Tracking information. After storm tracks are identified and catalogued, this information is available to Storm Position Forecast, the next algorithm in the Storm Sequence. Tracking information is updated as new volume scan data are made available from Storm Centroids. If the time interval between successive volume scans is too great (typically larger than 15 minutes), then a new tracking sequence is started.

2.5.3 Input Requirements

Storm Tracking requires three types of input: Storm Centroid product information, Storm Tracking product information, and processing parameters.

2.5.3.1 Storm Centroid Products. For the volume scan being processed, Storm Centroids provides centroid information for the storms it has identified. Information consists of the number of storms identified, their respective X- and Y-centroid positions as projected to the ground, and a number indicating the rank of each storm. In addition, a storm time is also provided. This time is the same for all storms of a given volume scan and is the time associated with the beginning of the volume scan.

2.5.3.2 Storm Tracking Products. Tracking product information associated with the previous volume scan is also needed. This information consists of a catalogue of the positions and associated times of all the storms for which tracks have been determined. After the current storm positions are correlated with the previous tracks, this current information is used to update the Tracking product information.

2.5.3.3 Processing Parameters. The present version of Storm Tracking allows for three processing parameters. The primary parameter is the correlation speed (kilometers per minute). This value determines the size of the correlation box and is directly related to the speed of the storm's average movement. Some storms may not be tracked or tracks may be misidentified if the box size is not suitable. The value used for our processing was 2 kilometers per minute, which resulted in a box size of twice that value. The two remaining parameters are of secondary importance. One determines the maximum time gap between successive volume scans before a new tracking sequence is initiated. A value of 16 minutes was used. The other gives the maximum time period for which an individual storm is tracked. The value for this parameter was 3 hours. All tracking information older than this was automatically aged out of the tracking information catalogue. Current values for these processing parameters are given in the Storm Tracking section (TRK) of the Parameter File listed in Appendix J.

2.5.4 Output Information

2.5.4.1 Products. Each time Storm Tracking processes a volume scan of data, the output product file generated is used by Storm Position Forecast and is also used later when Storm Tracking processes the next volume scan. The product file, mainly a catalogue of all storms tracked to date, includes information on the number of storms per volume scan, how the storms are correlated back in time, their ranks, centroid positions, and associated times.

2.5.4.2 Displays. Besides providing input to the Storm Tracking and Storm Position Forecast algorithms, the product file is utilized to display the results of Storm Tracking to the user. The displays are of two types: graphic and tabular. For the graphic display, the file provides the information needed to plot the centroid positions of storms over time. The information is usually combined with output information from the other algorithms in the Storm Sequence and presented as a composite display. An example of this display is given in Figure 2.

The tabular display gives information relevant only to Storm Tracking. Two display options are available. One gives a synopsis of the catalogue's bookkeeping information regarding each storm's relative rank over time. This display is used primarily for diagnostic purposes. The second allows the user to specify a particular storm. The centroid positions and corresponding times of the storm are then presented in a tabular form. Positions are ordered from current time back to the beginning of the storm's identified track.

2.6 Storm Sequence - Storm Position Forecast

The fourth step in the Storm Sequence is the Storm Position Forecast algorithm. Its primary function is to determine future centroid positions of a given storm based on its current and previous positions. The method of forecasting the position is based on a technique presented in Bjerkaas and Forsyth (1980) and Boak et al. (1977). We implemented NEXRAD version NX-DR-03-008/10. The current Storm Position Forecast algorithm description may be found in the NEXRAD Algorithm Report (1983c).

2.6.1 Functional Description

For each storm of the volume scan, new centroid positions are determined from the calculated speed and direction of that storm. The speed and direction are first determined by performing an unweighted linear least squares fit to the current position and available previous positions of the storm. Two fits are performed independently using first the X-centroid positions versus time and then the Y-centroid positions versus time. Figure 9 illustrates an example of a fit of two storms using the X-centroid positions. A similar fit would also be done using the Y-centroid positions.

2.6.2 Processing Environment

To determine future positions of a storm, a minimum of two locations of the storm must be known. These locations can be obtained after Storm Tracking has processed at least two sequential volume scans. However, for completeness' sake, if only one position of a storm is available, the forecast positions are arbitrarily made equal to the one known position. For a given storm, Storm Position Forecast uses all the storm positional information provided by Storm Tracking back for a specified maximum length of time. For the version we implemented, no storm positions older than 1 hour were used.

2.6.3 Input Requirements

Storm Position Forecast requires two types of input: Storm Tracking product information, and processing parameters.

2.6.3.1 Storm Tracking Products. For each storm of the current volume scan, the Storm Tracking product file contains all the centroid positional information of the storm's movement. This information consists of the X- and Y-centroid values, associated time, and storm rank. These centroid values and times are utilized in the calculation of the linear least square fit for each identified storm of the volume scan.

2.6.3.2 Processing Parameters. Storm Position Forecast uses two basic processing parameters. The first parameter defines the maximum time period of data used in generating a forecast, normally 60 minutes. Forecasts using data older than 1 hour are considered unreliable. The second parameter is the forecast interval. This interval is actually specified by giving the number of forecast periods desired and their respective time increments from the current time. The algorithm is normally executed with four forecast periods

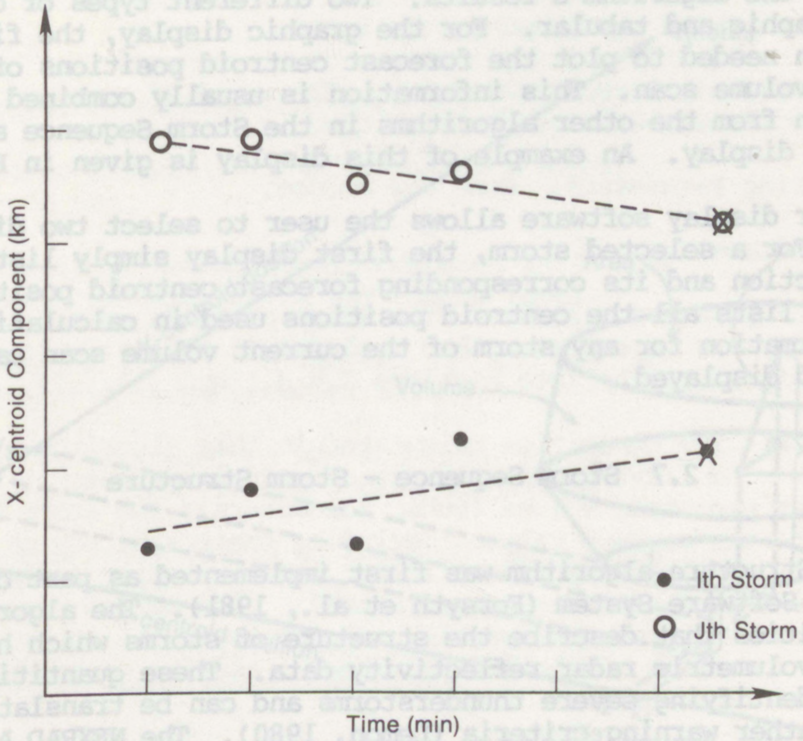


Figure 9. Example of the forecast of the X-positions determined from each of four known storm centroid positions.

at 15, 30, 45 and 60 minutes from current time. These processing parameters are given in the Storm Position Forecast section (FST) of the Parameter File listed in Appendix J.

2.6.4 Output Information

2.6.4.1 Products. After each volume scan is processed, Storm Position Forecast produces an output product file. For each storm identified in the volume scan, the file contains the storm's calculated speed and direction and a table of the storm's current and previous centroid positions used in the linear least squares fit. If a storm's direction cannot be properly calculated because the speed is zero, then the file will also contain a flag indicating that the direction is undefined and has been arbitrarily set to zero. This Storm Position Forecast output product file will later be used as input to Storm Structure and Hail, the two remaining algorithms of the Storm Sequence.

2.6.4.2 Displays. In addition to providing input to the Storm Structure and Hail algorithms, the Storm Position Forecast output product file is also used for displaying the algorithm's results. Two different types of displays are available: graphic and tabular. For the graphic display, the file provides the information needed to plot the forecast centroid positions of each storm for the given volume scan. This information is usually combined with the output information from the other algorithms in the Storm Sequence and presented as a composite display. An example of this display is given in Figure 2.

The tabular display software allows the user to select two different types of displays. For a selected storm, the first display simply lists the storm's speed and direction and its corresponding forecast centroid positions. The second display lists all the centroid positions used in calculating a storm's fit. The information for any storm of the current volume scan can be selected by the user and displayed.

2.7 Storm Sequence - Storm Structure

The Storm Structure algorithm was first implemented as part of the Modular Radar Analysis Software System (Forsyth et al., 1981). The algorithm computes quantities that describe the structure of storms which have been identified in volumetric radar reflectivity data. These quantities are valuable for identifying severe thunderstorms and can be translated directly into severe weather warning criteria (Lemon, 1980). The NEXRAD Algorithm Report (1983c) contains the most current version of the Storm Structure algorithm description. We implemented NEXRAD version NX-DR-03-009/35.

2.7.1 Functional Description

The Storm Structure algorithm processes those storms that have been identified by the Storm Centroids algorithm. Each storm is processed individually, beginning with the most intense and finishing with the least intense storm identified in a volume scan. When a storm is processed, two arrays are accessed. This first array gives the number of components that belong to that storm and the second array indicates which storm components belong to the storm being processed. These two features allow easy access to the information for the components of each storm. This information is used to calculate several quantities that describe the structure of a storm (see Fig. 10). The complete set of structure quantities appears in Section 2.7.4.1.

2.7.2 Processing Environment

The Storm Structure algorithm requires output from the Storm Centroids and Storm Position Forecast algorithms, therefore, it cannot execute until these algorithms have produced their product files. The algorithm does not access the radar data as all the necessary information is obtained from the Storm Centroids product file.

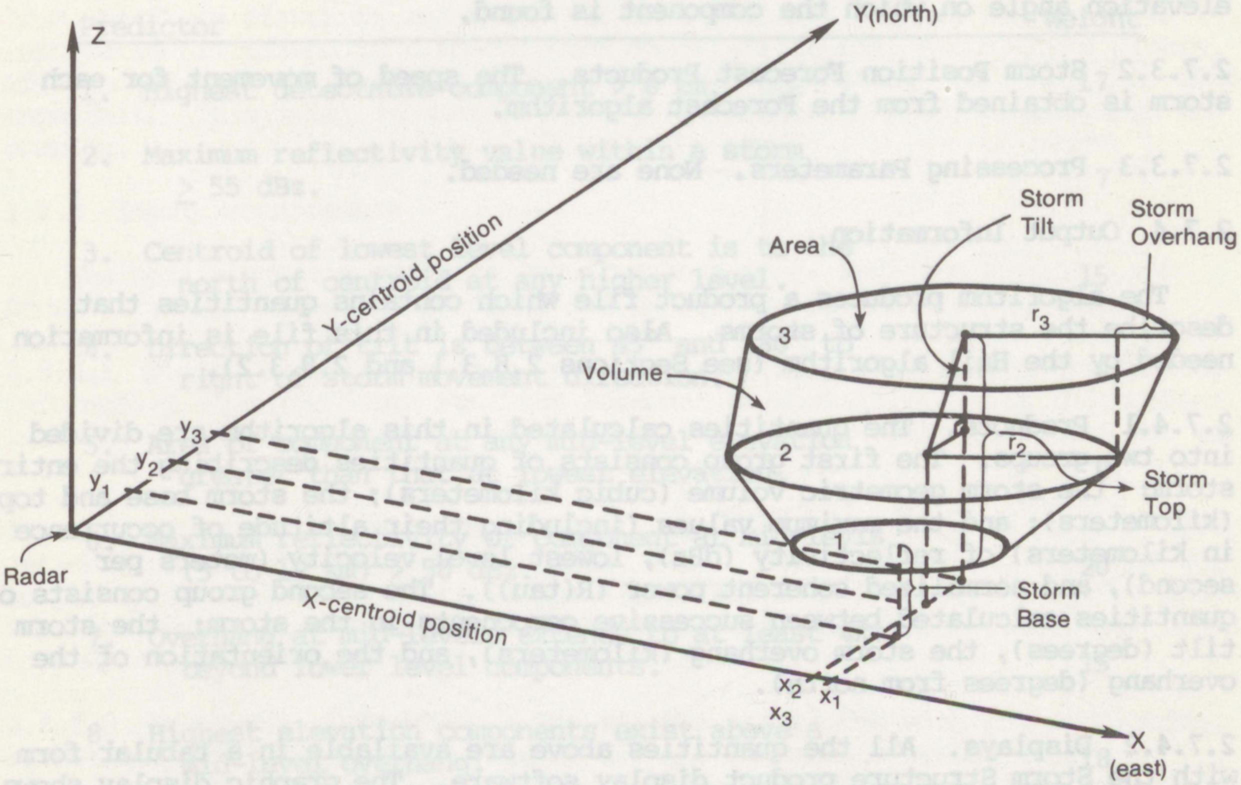


Figure 10. A three-dimensional diagram of a storm composed of three vertically correlated storm components. Shown are some of the quantities calculated by the Storm Structure algorithm. The components appear circular here since the radius of a storm component is computed from the relation, Area $= \pi r^2$.

After all information has been read from the Storm Centroid and Storm Position Forecast product files, this information is passed to the module that executes the Storm Structure algorithm. After a storm is processed, all quantities that have been computed, along with additional information needed by the Hail algorithm, are written to the Storm Structure product file.

2.7.3 Input Requirements

The Storm Structure algorithm requires information contained in the Storm Centroids and Storm Position Forecast product files. No processing parameters are needed.

2.7.3.1 Storm Centroids Products. The Centroids algorithm supplies the number of storms identified, their storm centroid positions, the number of components for each storm, and component information. The component information includes the X- and Y-centroid position, the component area, and the elevation angle on which the component is found.

2.7.3.2 Storm Position Forecast Products. The speed of movement for each storm is obtained from the Forecast algorithm.

2.7.3.3 Processing Parameters. None are needed.

2.7.4 Output Information

The algorithm produces a product file which contains quantities that describe the structure of storms. Also included in this file is information needed by the Hail algorithm (see Sections 2.8.3.1 and 2.8.3.2).

2.7.4.1 Products. The quantities calculated in this algorithm are divided into two groups. The first group consists of quantities describing the entire storm: the storm geometric volume (cubic kilometers); the storm base and top (kilometers); and the maximum values (including their altitude of occurrence in kilometers) of reflectivity (dBz), lowest level velocity (meters per second), and normalized coherent power (R(tau)). The second group consists of quantities calculated between successive components in the storm: the storm tilt (degrees), the storm overhang (kilometers), and the orientation of the overhang (degrees from north).

2.7.4.2 Displays. All the quantities above are available in a tabular form with the Storm Structure product display software. The graphic display shows only two of the structure quantities. These are the 30-dBz storm top and the storm maximum reflectivity. An example of this graphic display is shown in Figure 2.

2.8 Storm Sequence - Hail

The three-dimensional thunderstorm structure, as seen in radar reflectivity data, is valuable in identifying storms that produce hail (Lemon, 1978). Based on this concept, an algorithm has been developed in which hailstorms can be automatically identified (Petrocchi, 1982). It was first implemented as part of the Modular Radar Analysis Software System (Forsyth et al., 1981). The algorithm examines the structure of storms to determine whether the storms are positively hail-producing, probably hail-producing, or not hail-producing. The version of this algorithm in the NEXRAD Algorithm Report (1983c) is the same version we implemented for this project.

2.8.1 Functional Description

The Hail algorithm uses eight weighted predictors to determine the likelihood that a storm will produce hail. The predictors and their weights are

Table 2. Hail Predictors and Associated Weights Used to Determine the Likelihood of Hail

Predictor	Weight
1. Highest detectable component ≥ 8 km.	17
2. Maximum reflectivity value within a storm ≥ 55 dBz.	7
3. Centroid of lowest level component is to the north of centroid at any higher level.	15
4. Direction of tilt is between 45° and 180° to right of storm movement direction.	8
5. Area of component at any mid-level elevation greater than that at lowest elevation.	0
6. Maximum reflectivity of component at mid-levls (5 to 12 km) ≥ 50 dBz.	20
7. Overhang at mid-levels extends to at least 4km beyond lower level components.	15
8. Highest elevation components exist above a mid-level overhang.	18

listed in Table 2. A positive identification of a predictor occurs when sufficient information exists and this information satisfies the predictor being tested. For example, when the maximum reflectivity in a storm is 55 dBz or more, then Predictor 2 in Table 2 becomes positively identified and its associated weight, 7, is included in a positive sum. On the other hand, if the maximum reflectivity in a storm is less than 55 dBz, then no identification occurs; and its associated weight, 7, is not included in any sum.

A probable identification of a predictor occurs when there is insufficient information available to satisfy the predictor being tested. For example, if a storm's top is unknown because the radar could not scan above a fixed elevation angle, then the weight, 17, associated with Predictor 1 would be included in a probable sum.

After each predictor has been tested, the positive and probable sums are calculated. The positive sum is simply the sum of the weights associated with those predictors that have been positively identified. Likewise, the probable sum is the sum of the weights associated with those predictors whose identifications are probable.

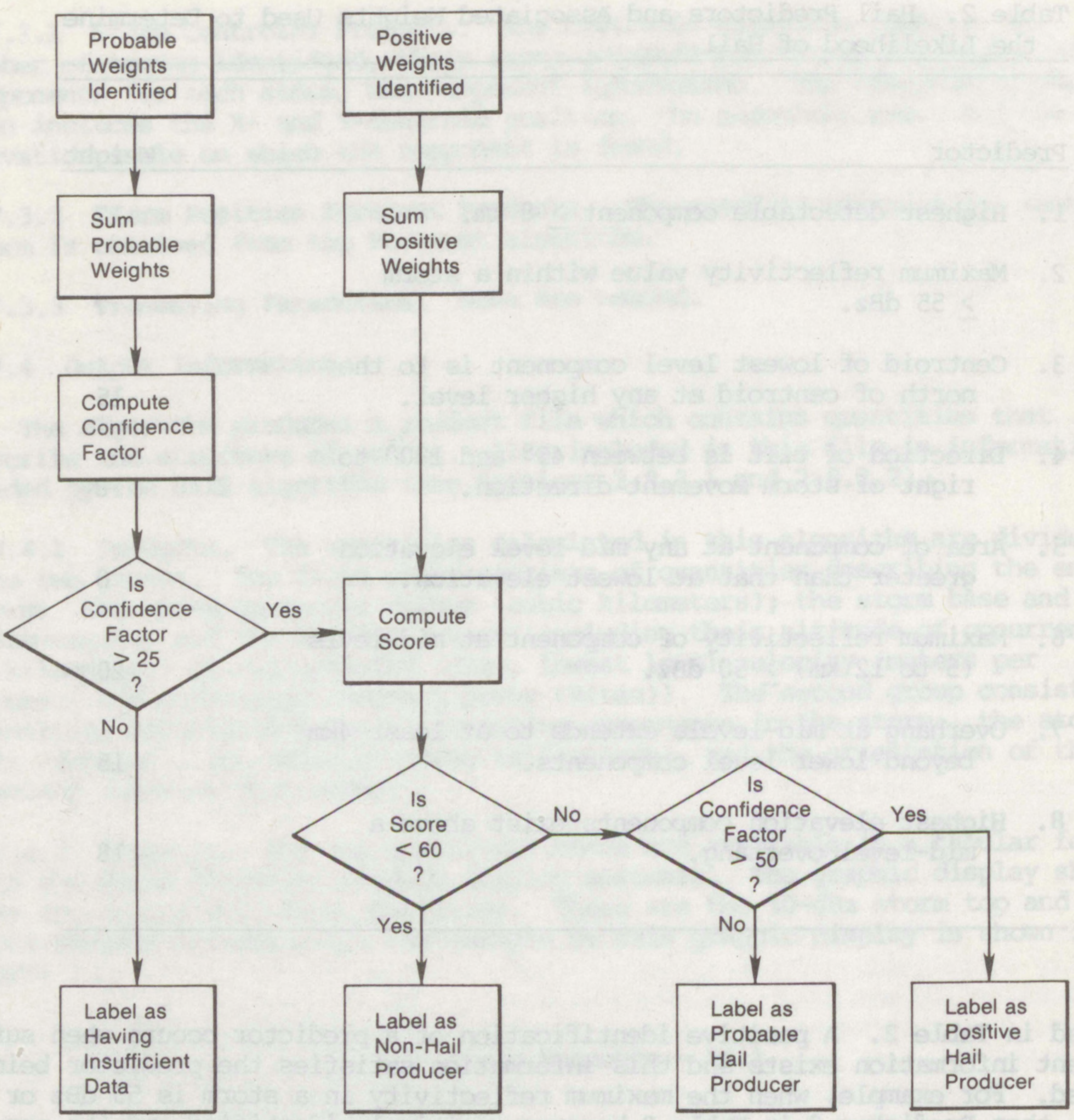


Figure 11. Flow diagram of the decision process used in the Hail algorithm for deciding whether a storm is producing hail.

As shown in Figure 11, the probable sum is used to calculate a confidence factor (confidence factor = 100 - probable sum). The confidence factor indicates the amount of weight identified with insufficient information. If the confidence factor for a given storm is small, then the algorithm will not make a decision concerning the storm's likelihood to produce hail. In this case the storm is identified as not having sufficient information. If sufficient information does exist for a given storm, then the positive sum is used together with the confidence factor to decide whether a given storm is positively hail-producing, probably hail-producing, or not hail-producing.

2.8.2 Processing Environment

The Hail algorithm is the last of the Storm Sequence algorithms to execute for a volume scan. All of the information needed in the algorithm is obtained from the Storm Structure and Storm Position Forecast product files. This information is passed to a module that executes the Hail algorithm software. After a storm is analyzed, it is given a label indicating its likelihood to produce hail. This label is written to a product file that may be accessed for display.

2.8.3 Input Requirements

The Hail algorithm software requires both Storm Position Forecast and Storm Structure algorithm products and processing parameters as inputs.

2.8.3.1 Storm Position Forecast Products. The Hail algorithm software uses information from the Storm Position Forecast product file. This information includes the speed, direction of movement, and the forecast positions for each storm.

2.8.3.2 Storm Structure Products. The Hail algorithm software obtains information from the Storm Structure product file. This information includes the number of storms, storm centroid positions, storm maximum reflectivity, storm top, the number of components in each storm, the component centroid positions, component area, and component maximum reflectivity.

2.8.3.3 Processing Parameters. The weights associated with the predictors, as well as several threshold values, are the parameters needed in the algorithm software. The entire list of processing parameters is found in the Hail section (HAL) of the Parameter File listed in Appendix J.

2.8.4 Output Information

Every storm processed by the algorithm obtains one of four possible labels indicating the storm's likelihood to produce hail. The four possibilities are 1) insufficient data were available, 2) no hail-producer, 3) probable hail-producer, and 4) positive hail-producer. For the purpose of displaying these labels, the storm rank and centroid positions are included in the output file.

2.8.4.1 Products. The algorithm product information is available in tabular form with the Hail product display software. The table is divided into the four categories described above. Within each category, the storms that have obtained the corresponding label are identified by their rank numbers.

2.8.4.2 Displays. An alphanumeric identifier representing the likelihood of hail for a storm can be displayed on a graphics monitor. The character "A" is used to indicate positive hail production. The characters "A-" and "A=" are used to indicate probable hail production and no hail, respectively. The character "-" indicates insufficient data existed. An example of this graphic display is given in Figure 2.

2.9 Velocity-Azimuth Display (VAD)

The Velocity-Azimuth Display (VAD) is the only algorithm of the Non-Storm category coded and exercised by PROFS. It is also one of the two algorithms using Doppler velocity data, the other being Mesocyclone Detection. The purpose of VAD is to determine the atmospheric wind field structure surrounding the radar. Specifically, VAD is used to obtain vertical profiles of horizontal wind speed, horizontal wind direction, divergence, and vertical wind speed.

We calculate these profiles using full sweeps of radar data at one or more slant ranges and several elevation angles. The VAD algorithm as implemented here uses a technique first proposed by Lhermitte and Atlas (1961) and further developed by Browning and Wexler (1968), Rabin and Zrnic (1980), Rabin (1982), and others. We implemented NEXRAD version NX-DR-03-007/11. The current VAD algorithm description may be found in the NEXRAD Algorithm Report (1983c).

2.9.1 Functional Description

The algorithm performs a discrete Fourier transform analysis on the Doppler velocities to obtain a least squares fit composed of the zeroth and first harmonics. Functions of these harmonics produce the horizontal divergence and the horizontal wind speed and direction. We obtain the vertical profiles of these parameters by performing the VAD analysis at several different heights (radar elevation angles) at the same slant range. The vertical wind speed profile is obtained by integration of the horizontal divergence profile. We compute the horizontal divergence and vertical wind speeds with precipitation fall-velocity formulations for rain, snow, and clear air.

The algorithm first performs the harmonic analysis on the Doppler velocities, obtaining the zeroth and first harmonics. Next, the horizontal wind direction is calculated from the coefficients of the first harmonic. Using the wind direction, VAD then determines the quality of the Fourier fit by calculating the square root of the mean squared deviations of the Doppler velocities from the Fourier least squares fitted curve. If this RMS error is below a specified threshold (typically 7.5 meters per second) then VAD continues processing by calculating the remaining vertical wind profile values; otherwise an error flag is set.

2.9.2 Processing Environment

VAD requires both the Doppler velocity and reflectivity data. The current version assumes that the data have been conditioned, if necessary. This would include elimination of ground clutter, anomalous propagation, weak signals, etc. Each volume scan of data is processed independently; hence, no time continuity requirements exist. The results of the VAD algorithm are not presently used in any other algorithm coded by PROFS. Due to its nature, VAD would normally be executed with data from non-storm meteorological conditions.

2.9.3 Input Requirements

The VAD algorithm requires two types of input: Doppler data and processing parameters.

2.9.3.1 Doppler Data. Three channels of information from the radar are required. These channels provide data values for velocity, reflectivity, and normalized coherent power, R(tau), or alternatively, spectral width. The velocity and reflectivity data are used directly in the VAD calculation. The normalized coherent power data are used to help eliminate unreliable velocities by comparing the individual velocity values against the corresponding R(tau) values. All velocities whose corresponding R(tau) values are below a specified minimum (typically 0.12) are eliminated. Velocity values which lie within a region centered about zero velocity can also be eliminated. However, since no velocity elimination was desired in processing data for this project, the width of the region was set to zero. Reflectivity data are used both in the calculation of the precipitation fall-velocities and for color-coding the observed velocities for one of the output displays.

2.9.3.2 Processing Parameters. The present version of VAD allows for three types of thresholding parameters, as well as information specifying which slant ranges to process. The first threshold is the RMS error threshold, used for determining the quality of the Fourier fit. If the observed velocities are too scattered about the fit line, i.e., RMS error is greater than the specified threshold, then the data are of poor quality and are not used. The threshold value used is 7.5 meters per second. The second threshold is the R(tau) value. The quality of individual observed velocity values is indicated by their corresponding R(tau) values. Typically, if R(tau) falls below 0.12 then the corresponding velocity value is rejected. The last threshold defines a narrow velocity band about zero. We have found that ground clutter can be removed if we reject all velocity values which lie within this band. For normal execution of the algorithm during this project, this band had zero width. Finally, the number of slant ranges to process and their individual values are specified as the last of the processing parameters. Processing parameter values for the three thresholds and specific slant ranges used are given in the Velocity-Azimuth Display section (VAD) of the Parameter File listed in Appendix J.

2.9.4 Output Information

2.9.4.1 Products. VAD produces an output product file of information each time it processes a volume scan of data. For each slant range and elevation angle, the algorithm outputs four arrays. These arrays contain the value of the azimuth angle, the corresponding observed Doppler velocity and reflectivity, and the velocity determined by the Fourier fit to the observed velocities. Typically these arrays each contained approximately 360 values, one for each degree of azimuth of a full scan. In addition to the array data, the resultant RMS error and calculated values for the various profiles are also provided. These values are for the profiles of horizontal wind speed and wind direction and the three profiles of horizontal divergence and vertical wind speed for the precipitation fall-velocity formulations of rain, snow, and clear air.

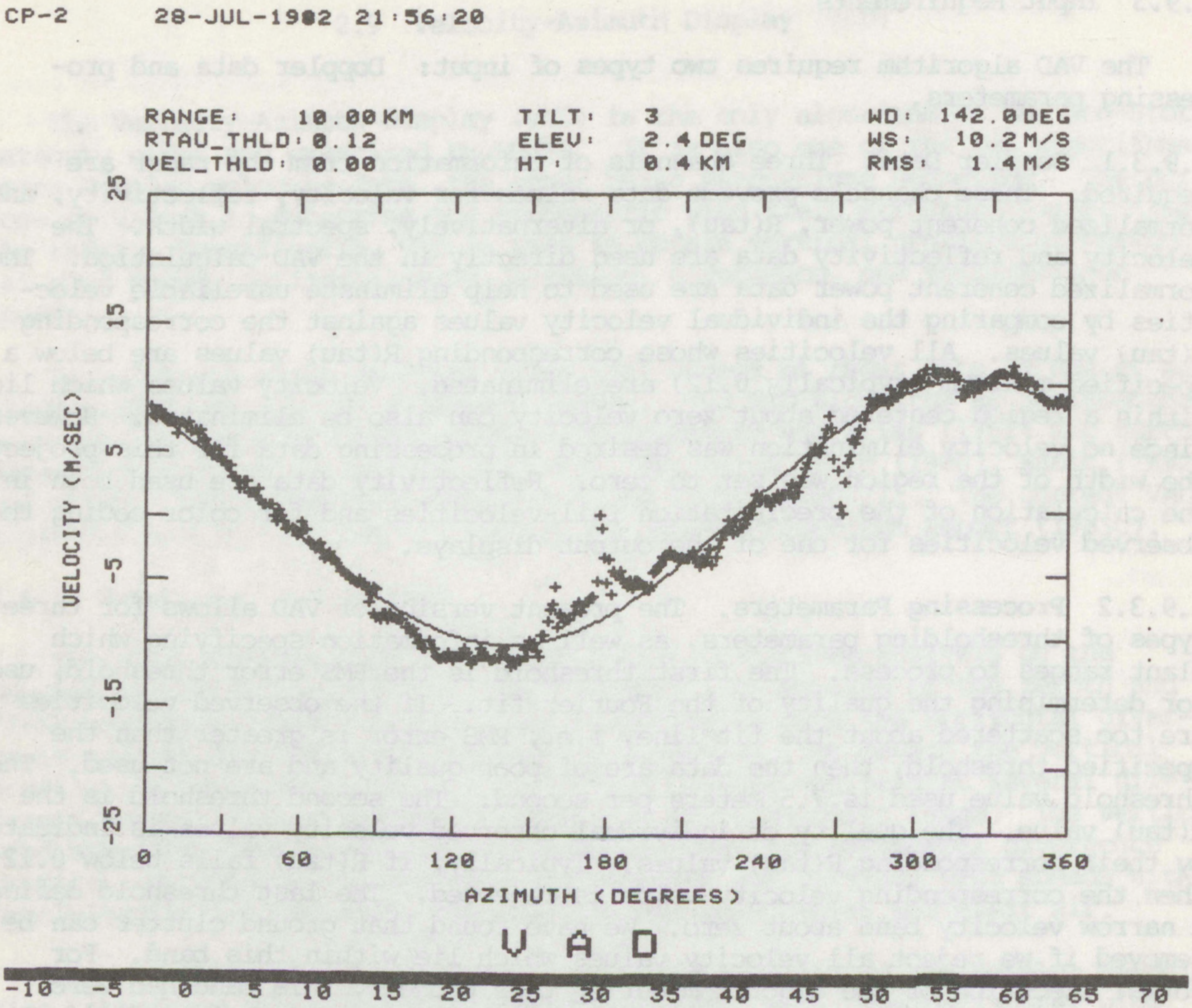


Figure 12. An example of the graphic presentation of a velocity versus azimuth analysis for a single sweep. Here, the scattered points indicate the observed Doppler velocities and the solid line represents the calculated fit to those data. The information above the plot characterizes the data and the results of the analysis. Data for 28-JUL-1982 21:56:20.

2.9.4.2 Displays. The output products of VAD are displayed in three forms; two are graphical, and the third is tabular. An example of the first graphic display (Fig. 12) shows the Fourier least squares fitted curve as the smooth sinusoidal curve and the observed velocities scattered about this curve. On a color display, the individual observed velocity values are color-coded according to their corresponding reflectivity values. Reflectivity values normally range from -10 to 70 dBz as shown in the bar at the bottom of the figure, which would appear in color on a graphic display. The curve shows velocities on a scale of -25.0 to 25.0 meters per second versus azimuth angles

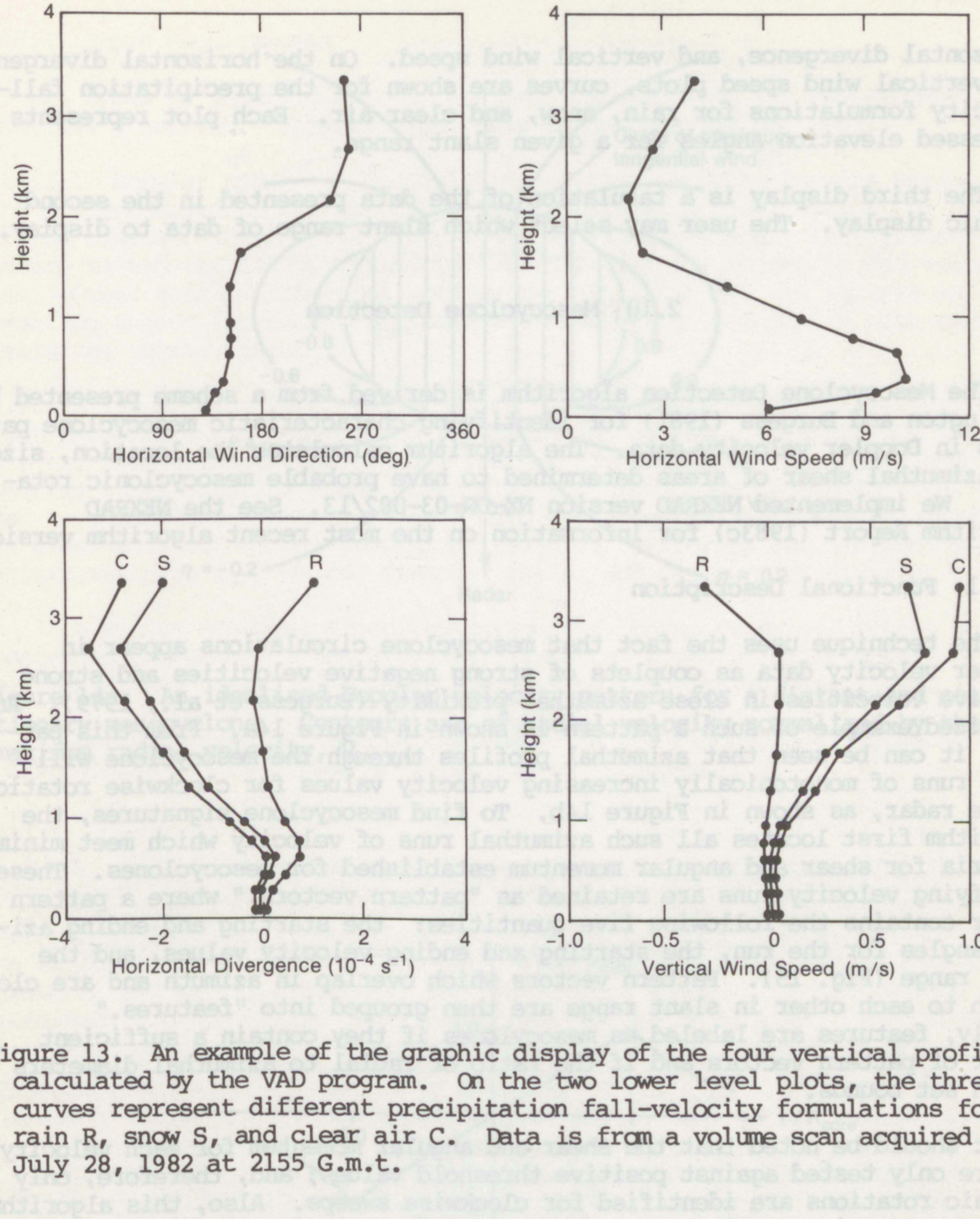


Figure 13. An example of the graphic display of the four vertical profiles calculated by the VAD program. On the two lower level plots, the three curves represent different precipitation fall-velocity formulations for rain R, snow S, and clear air C. Data is from a volume scan acquired on July 28, 1982 at 2155 G.m.t.

from 0.0° to 360.0°. Additional information is also given regarding the data's date and time; the specific slant range, elevation angle, and height above ground of the data; the corresponding RMS error of the fit; and the calculated wind speed and direction. Threshold values for R(tau) and velocity are also given.

The second graphic display, as shown in Figure 13, consists of a set of four height profiles, one each for the horizontal wind direction, wind speed, horizontal divergence, and vertical wind speed. On the horizontal divergence and vertical wind speed plots, curves are shown for the precipitation fall-velocity formulations for rain, snow, and clear air. Each plot represents all processed elevation angles for a given slant range.

The third display is a tabulation of the data presented in the second graphic display. The user may select which slant range of data to display.

2.10 Mesocyclone Detection

The Mesocyclone Detection algorithm is derived from a scheme presented by Hennington and Burgess (1981) for identifying characteristic mesocyclone patterns in Doppler velocity data. The algorithm calculates the location, size, and azimuthal shear of areas determined to have probable mesocyclonic rotation. We implemented NEXRAD version NX-DR-03-002/13. See the NEXRAD Algorithm Report (1983c) for information on the most recent algorithm version.

2.10.1 Functional Description

The technique uses the fact that mesocyclone circulations appear in Doppler velocity data as couplets of strong negative velocities and strong positive velocities in close azimuthal proximity (Burgess et al., 1979). An idealized example of such a pattern is shown in Figure 14a. From this pattern, it can be seen that azimuthal profiles through the mesocyclone will yield runs of monotonically increasing velocity values for clockwise rotation of the radar, as shown in Figure 14b. To find mesocyclone signatures, the algorithm first locates all such azimuthal runs of velocity which meet minimum criteria for shear and angular momentum established for mesocyclones. These qualifying velocity runs are retained as "pattern vectors," where a pattern vector contains the following five quantities: the starting and ending azimuth angles for the run, the starting and ending velocity values, and the slant range (Fig. 15). Pattern vectors which overlap in azimuth and are close enough to each other in slant range are then grouped into "features." Finally, features are labeled as mesocylones if they contain a sufficient number of pattern vectors and if the ratio of radial to azimuthal diameters is within set bounds.

It should be noted that the shear and angular momentum for each velocity run are only tested against positive threshold values; and, therefore, only cyclonic rotations are identified for clockwise sweeps. Also, this algorithm version has no interaction among the sweeps of a volume scan. Thus, strictly speaking, probable mesocyclone locations, rather than positive mesocyclone identifications, are determined.

2.10.2 Processing Environment

The algorithm description specifies that data processed by this algorithm must be free of folded velocities and ground clutter. Also, the algorithm assumes that the data are not noisy. Because no preprocessing of data was

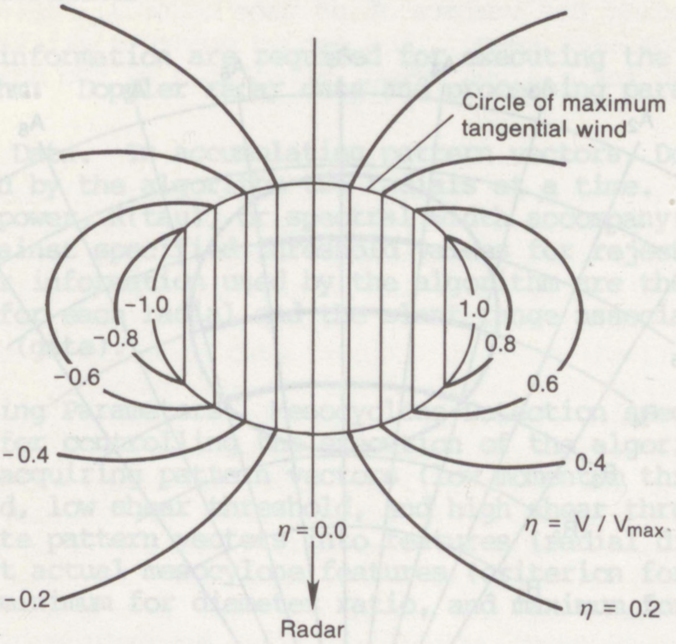


Figure 14a. An idealized Doppler velocity pattern for a distant and stationary mesocyclone. Contours are of radial velocity normalized by the maximum radial velocity, V_{max}.

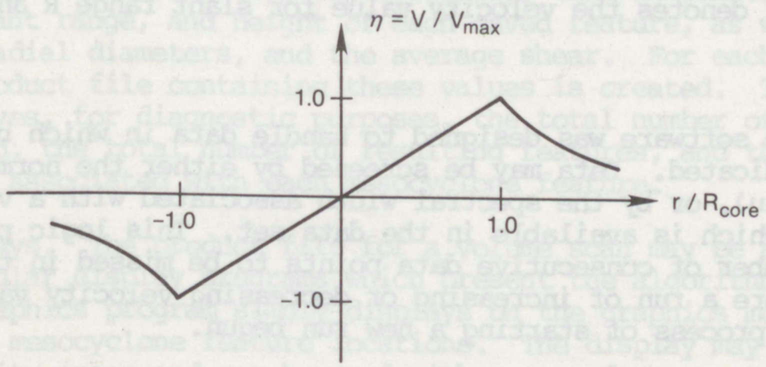


Figure 14b. Normalized Doppler velocity versus azimuthal distance, r, through the center of the idealized mesocyclone shown in Figure 14a. R_{core} is the distance from the mesocyclone's center to the point of maximum radial velocity.

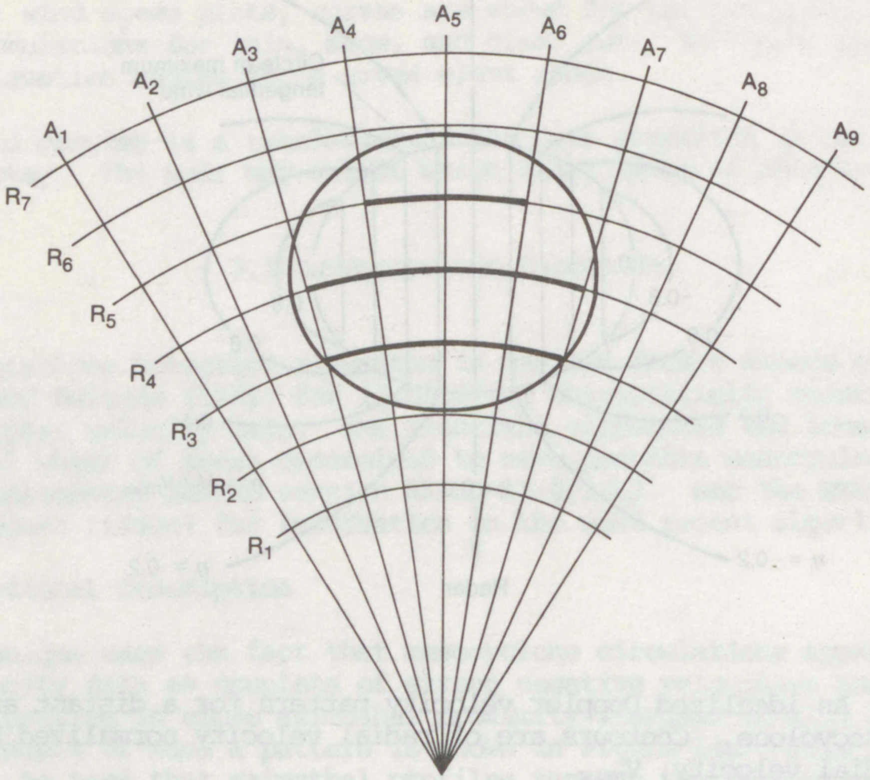


Figure 15. A mesocyclone on the radar slant range and azimuth grid. Bold lines indicate velocity runs which qualify as pattern vectors. The pattern vector at range R_4 contains the five values: R_4, A_3, A_7, $V_{(4,3)}$, $V_{(4,7)}$, where $V_{(R,A)}$ denotes the velocity value for slant range R and azimuth A.

performed, the software was designed to handle data in which bad velocity values are indicated. Data may be screened by either the normalized coherence estimate, R(tau), or by the spectral width associated with a velocity value, depending on which is available in the data set. This logic permits a selectable number of consecutive data points to be missed in the azimuthal direction before a run of increasing or decreasing velocity values is terminated and the process of starting a new run begun.

Another assumption of the algorithm description is that the radials of a sweep are processed in a clockwise manner. The case of a counter-clockwise sweep is handled by the software which simply processes the radials in reverse order. Of course, this feature of the software might not be practical for a real-time implementation. On occasions when the radar changes direction in the middle of a sweep, the software cannot successfully process the data for that sweep.

2.10.3 Input Requirements

Two types of information are required for executing the Mesocyclone Detection algorithm: Doppler radar data and processing parameters.

2.10.3.1 Doppler Data. In accumulating pattern vectors, Doppler velocity data are processed by the algorithm two radials at a time. Values of normalized coherent power, R(tau), or spectral width accompany the velocity data and are tested against specified threshold values for rejecting bad velocities. Other radar information used by the algorithm are the azimuth and elevation angles for each radial and the slant range associated with each resolution volume (gate).

2.10.3.2 Processing Parameters. Mesocyclone Detection specifies a set of eight parameters for controlling the execution of the algorithm. Of these, four are used in acquiring pattern vectors (low momentum threshold, high momentum threshold, low shear threshold, and high shear threshold); one is used to consolidate pattern vectors into features (radial distance); and three are used to select actual mesocylone features (criterion for minimum number of pattern vectors, minimum for diameter ratio, and maximum for diameter ratio).

In addition to these algorithm parameters, three other quantities are treated as parameters: the R(tau) and spectral width velocity qualifiers, and the maximum number of consecutive missing velocity values allowed.

Values for these parameters are stored in the Mesocyclone Dectection Section (MES) of the Parameter File, listed in Appendix J.

2.10.4 Output Information

2.10.4.1 Products. In processing Doppler data, the Mesocyclone Detection algorithm generates information on the location, size, and intensity of the mesocyclone feature it identifies. Specifically, this information includes the azimuth, slant range, and height of each saved feature, as well as the azimuthal and radial diameters, and the average shear. For each volume scan processed, a product file containing these values is created. The product file also receives, for diagnostic purposes, the total number of pattern vectors accumulated, the total number of resulting features, and the number of pattern vectors associated with each mesocyclone feature.

2.10.4.2 Displays. The product file for a volume scan may be accessed by graphic and tabular display programs which present the algorithm results to a user. The graphics program simply displays on the graphics monitor an "M" centered on the mesocyclone feature locations. The display may be overlaid on a Doppler velocity image and may be used with roam and zoom options. A special color table for these graphics is available, producing flashing white M's over a red and green image.

The tabular display program provides a formatted presentation of the processing results for a selected volume scan. Two separate types of information are available. The first type presents a general summary of results for the volume scan, including the numbers of pattern vectors, features, and mesocyclone features for each sweep. The second presents all the mesocyclone feature information for a given sweep, selectable by the user.

3. IDENTIFICATION OF A TEST DATA SET

3.1 A Representative Radar

A stringent test of the algorithm software involves real Doppler data from a radar with characteristics similar to those specified by NEXRAD. The data set must include a variety of meteorological phenomena with solid corresponding verification data. Because it met these criteria, the NCAR CP-2 radar was used as a primary data source for Doppler data. An additional consideration was the established data transfer between the CP-2 radar and PROFS computer facility. Although the current 56-kbaud link allows transmission of data only at greatly reduced resolution, a microwave link now under construction will allow transmission of all Doppler radar moments into PROFS' facility. During the summer of 1982, the CP-2 radar was operated, from its location east of Boulder, by PROFS and the Joint Airport Weather Study (JAWS) program (McCarthy et al., 1982). Data were recorded on magnetic tape during a variety of meteorological conditions which included the formation, growth, and movement of many storms over both the plains and the mountains. The data also included storms which produced hail, high winds, heavy rains, mesocyclones, and tornadic circulations. Verification data included sightings of tornadoes and measurement of hail size.

Shown in Table 3 are the characteristics of the CP-2 radar capability, the specific characteristics of CP-2 during the 1982 operation (labeled CP-2 PROFS Data), and NEXRAD specifications. Comparison of the columns titled CP-2 Capability and NEXRAD Specifications reveals that the CP-2 can emulate a NEXRAD radar since all the NEXRAD specifications are in the range of CP-2 capabilities. Comparison of the columns titled CP-2 PROFS Data and NEXRAD Specifications, however, indicates that the parameters set during the acquisition of the CP-2 PROFS data do not precisely agree with the NEXRAD Doppler specifications. The key point in this comparison is whether the CP-2 PROFS data reasonably represent data from a NEXRAD Doppler: would the algorithm software executed on the data yield significantly different results from those generated from a NEXRAD radar data set taken under the same meteorological conditions?

Although the radar characteristics during the time the CP-2 PROFS data were taken do not precisely duplicate NEXRAD Doppler specifications, most are the same or quite close. The pulse repetition frequency (PRF) and the angular velocity both indicate that the CP-2 was acquiring data at a rate slightly slower than a NEXRAD radar would have. This simply means that it would take a few tens of seconds longer to complete a volume scan. It is difficult to believe that this would significantly alter the products.

The CP-2 PROFS data were taken at nine elevation angles, one of the two modes specified by NEXRAD. This mode has the potential for generating slightly different products for given meteorological conditions. For example, determinations of storm top could differ slightly. In very special cases, this could result in prediction of hail under one set of elevation scans and

Table 3. NCAR CP-2 Doppler Capability and the NEXRAD Specifications

Characteristics	CP-2 Capability	CP-2 PROFS Data	NEXRAD Specifications
Measured fields	Z, V, SIG	Z, V, SIG	Z, V, SIG
Wavelength (cm)	10	10	10
PRF (Hz)	400-1600	960	1000
Beamwidth (deg)	.98	.98	1.0
Angular velocity (deg/s)	0-20	15	18
Pulse width (microseconds)	.5-2.0	1.0	1.0
Unambiguous range - velocity and spectral width (km)	93-375	156	230
Range for reflectivity coverage (km)	Selectable	312	460
Unambiguous velocity (m/s)	10-40	24	25
Number of range gates	256-1024	512	1024
Number of hits (sample size)	32-4096	64	32
Range gate spacing (m)	60-990	300	≤250
Number of elevation angles	Programmable	9	9, 14
Resolution (bits)	8	8	8
Time for sweep (s)	≥18	24	20
Time for volume (min)	Programmable	3.6	4.7
Volume scan interval (min)	Programmable	5	5

predictions of no hail under a different set of angles. However, only in very special conditions would elevation scan differences result in significantly different products. A large storm would surely be detected and tracked under either set of elevation angles. A mesocyclone of any appreciable depth would certainly be detected under either condition. The VAD algorithm would accurately depict conditions at any particular elevation angle.

Another difference between the CP-2 PROFS data and the NEXRAD specifications is the range gate spacing, 300 meters for the CP-2 and $\leq$ 250 meters for NEXRAD. Although this could have an effect under a very limited set of conditions, it is difficult to argue that it affected the detection or analysis of most storms. A reflective mass very close to the detectable size might have a segment that is not quite long enough and, therefore, would not be identified as a storm component for that volume sweep. However, this is a very marginal case and, if the storm grew, it would surely be detected on subsequent volume sweeps. This resolution difference could affect other algorithm computations in a similar manner, but it is doubtful that this would make any significant difference in most cases.

3.2 NCAR CP-2 Radar Characteristics

The CP-2 radar is a S-band, or 10-cm wavelength, Doppler radar. It was designed as a weather research tool and was built and is operated by the Atmospheric Technology Division of the National Center for Atmospheric Research (NCAR). It is located 23 kilometers east of Boulder. The 156-km range coverage extends north into Wyoming, east almost to Nebraska, south past Colorado Springs, and west past Vail and nearly to Aspen (Fig. 16). Slightly more than half of this area is plains with the remainder mountains. Both the Cheyenne and Limon National Weather Service radars are included in this range circle.

CP-2 radar is a completely coherent system constructed around an FPS-18 surveillance radar. To accomplish this, NCAR upgraded an FPS-18 extensively to bring the essential components of the transmitter and receiver up to present standards of power and sensitivity. The CP-2 antenna is a 9-m parabolic type housed in a 15-m diameter radome. It has a peak power output of 1 megawatt. Its range of characteristics is given in Table 3; many are interrelated. Given in the column titled CP-2 PROFS Data are characteristics of the CP-2 during the time PROFS data were taken.

Figure 17 is a block diagram of the radar measurement system. It consists of three parts: the CP-2 Doppler radar, the RP6 signal processor, and the Nova 4/X computer.

The CP-2 radar produces the following signals: a logarithmic video from the non-coherent receiver, in-phase and quadrature signals from the coherent receiver, and a trigger pulse coincident with the transmitted pulse. A more detailed discussion is given by Gray (1982).

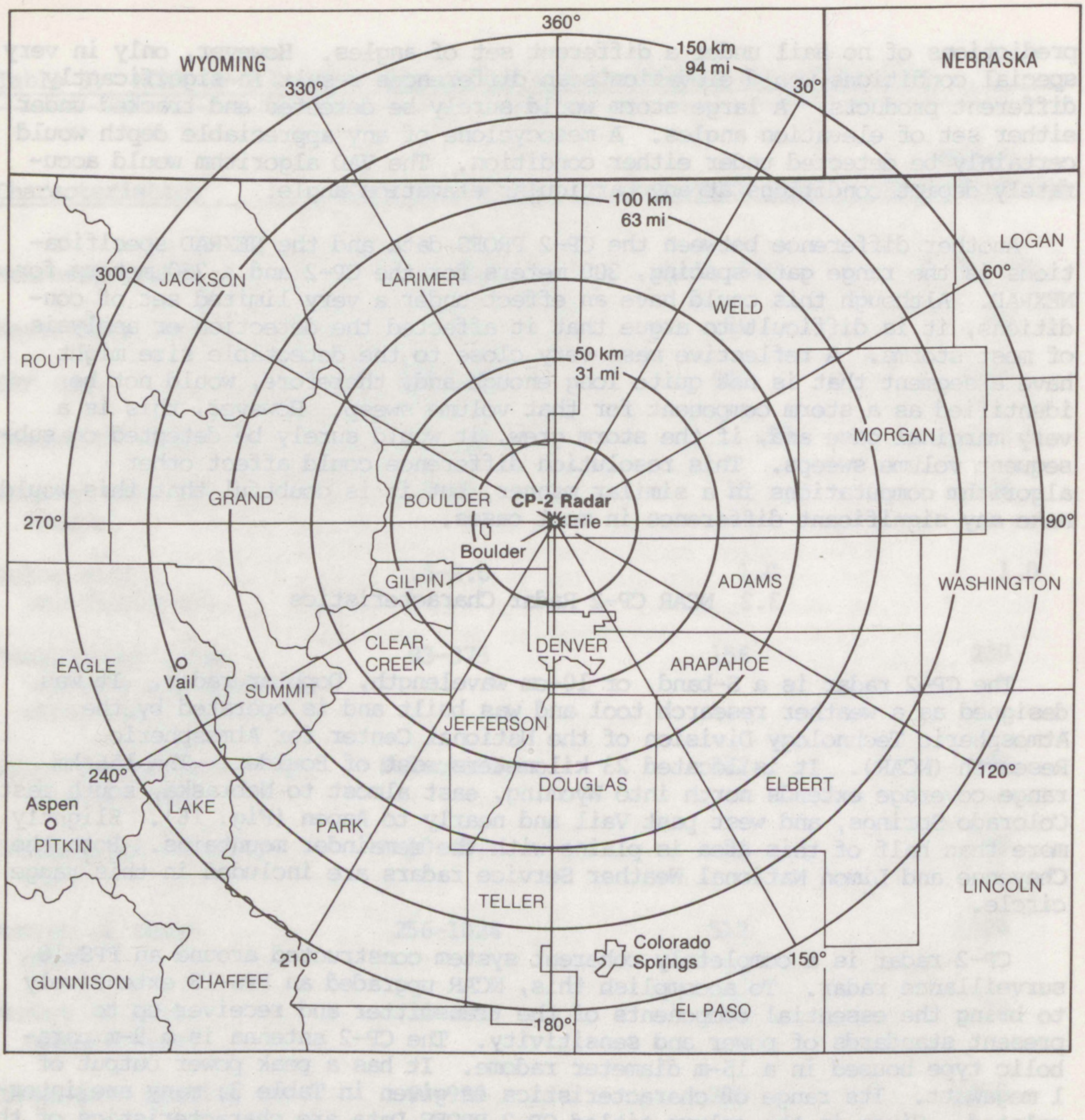


Figure 16. Coverage of the NCAR CP-2 Doppler radar at the 150-km range. The radar is located 23 kilometers east of Boulder, and the coverage extends over both plains and mountains. Range marks are every 25 kilometers.

The RP6 signal processor was constructed by NCAR specifically for the CP-2 radar. It has two components: a pulse-pair Doppler processor and a signal power averager. The pulse-pair processor provides estimates of the mean Doppler velocity, the linear signal power, and the average signal correlation coefficient. These last two, the linear signal power and the average signal

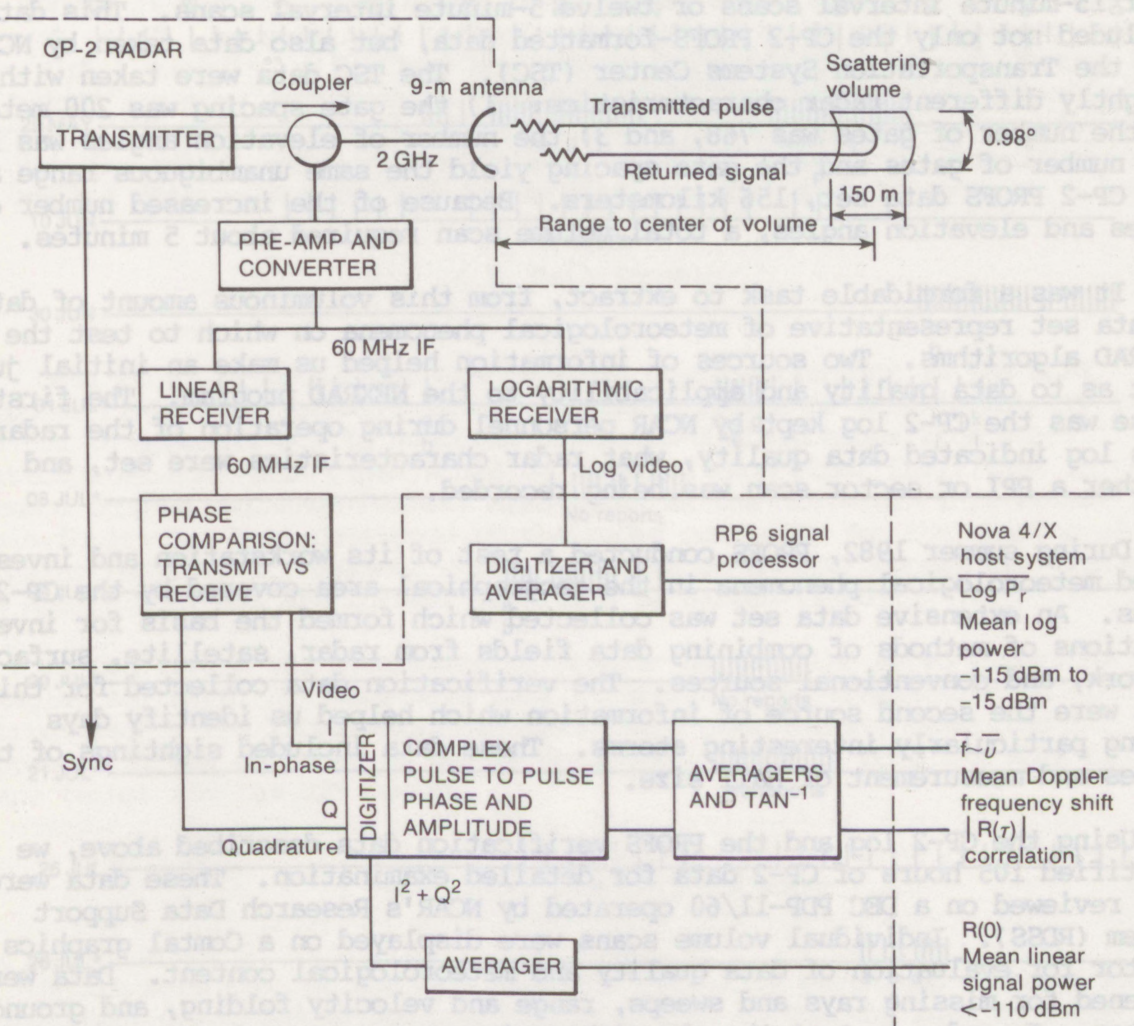


Figure 17. A block diagram of the NCAR CP-2 Doppler radar, the RP6 signal processor, and the Nova 4/X computer.

correlation coefficient, are used to estimate the spectral variance. The Nova computer formats the data, adding sufficient housekeeping data to make each processed ray a self-standing data set. The computer writes this data to tape, as well as handling any real-time data transmissions.

3.3 The 1982 CP-2 PROFS Data Selection

During the 1982 summer measurement period, more than 700 hours of CP-2 data were recorded on about 2000 tapes. An hour of data consists of either four 15-minute interval scans or twelve 5-minute interval scans. This data included not only the CP-2 PROFS-formatted data, but also data taken by NCAR for the Transportation Systems Center (TSC). The TSC data were taken with slightly different radar characteristics: 1) the gate spacing was 200 meters, 2) the number of gates was 768, and 3) the number of elevation angles was 11. The number of gates and the gate spacing yield the same unambiguous range as the CP-2 PROFS data set, 156 kilometers. Because of the increased number of gates and elevation angles, a total volume scan required about 5 minutes.

It was a formidable task to extract, from this voluminous amount of data, a data set representative of meteorological phenomena on which to test the NEXRAD algorithms. Two sources of information helped us make an initial judgment as to data quality and applicability to the NEXRAD program. The first of these was the CP-2 log kept by NCAR personnel during operation of the radar. This log indicated data quality, what radar characteristics were set, and whether a PPI or sector scan was being recorded.

During summer 1982, PROFS conducted a test of its workstation and investigated meteorological phenomena in the geographical area covered by the CP-2 scans. An extensive data set was collected which formed the basis for investigations of methods of combining data fields from radar, satellite, surface network, and conventional sources. The verification data collected for this test were the second source of information which helped us identify days having particularly interesting storms. These data included sightings of tornadoes and measurement of hail size.

Using the CP-2 log and the PROFS verification data described above, we identified 105 hours of CP-2 data for detailed examination. These data were then reviewed on a DEC PDP-11/60 operated by NCAR's Research Data Support System (RDSS). Individual volume scans were displayed on a Comtal graphics monitor for evaluation of data quality and meteorological content. Data were screened for missing rays and sweeps, range and velocity folding, and ground clutter. In order to test the algorithms, it was necessary to have two or more volume scans contiguous in time. Finally, significant meteorological information, including mesocyclone signatures and hail-producing storms, was logged during this data examination.

The information acquired from this detailed examination was kept in the data screen log. The log contained comments indicating areas of both clean and contaminated data, the meteorological content of the volume scan, which algorithm should be exercised by the data, and a photograph of a representative sweep in the volume scan. This log was used later to indicate which data should be transferred to the PROFS/NEXRAD data base, described in Section 4.

Fifty-two hours of data were selected for transfer to the PROFS system. Indicated in Figure 18 are sequences of contiguous volume scans selected for 14 days of 1982. Of the 52 hours selected, 37 hours were identified by the

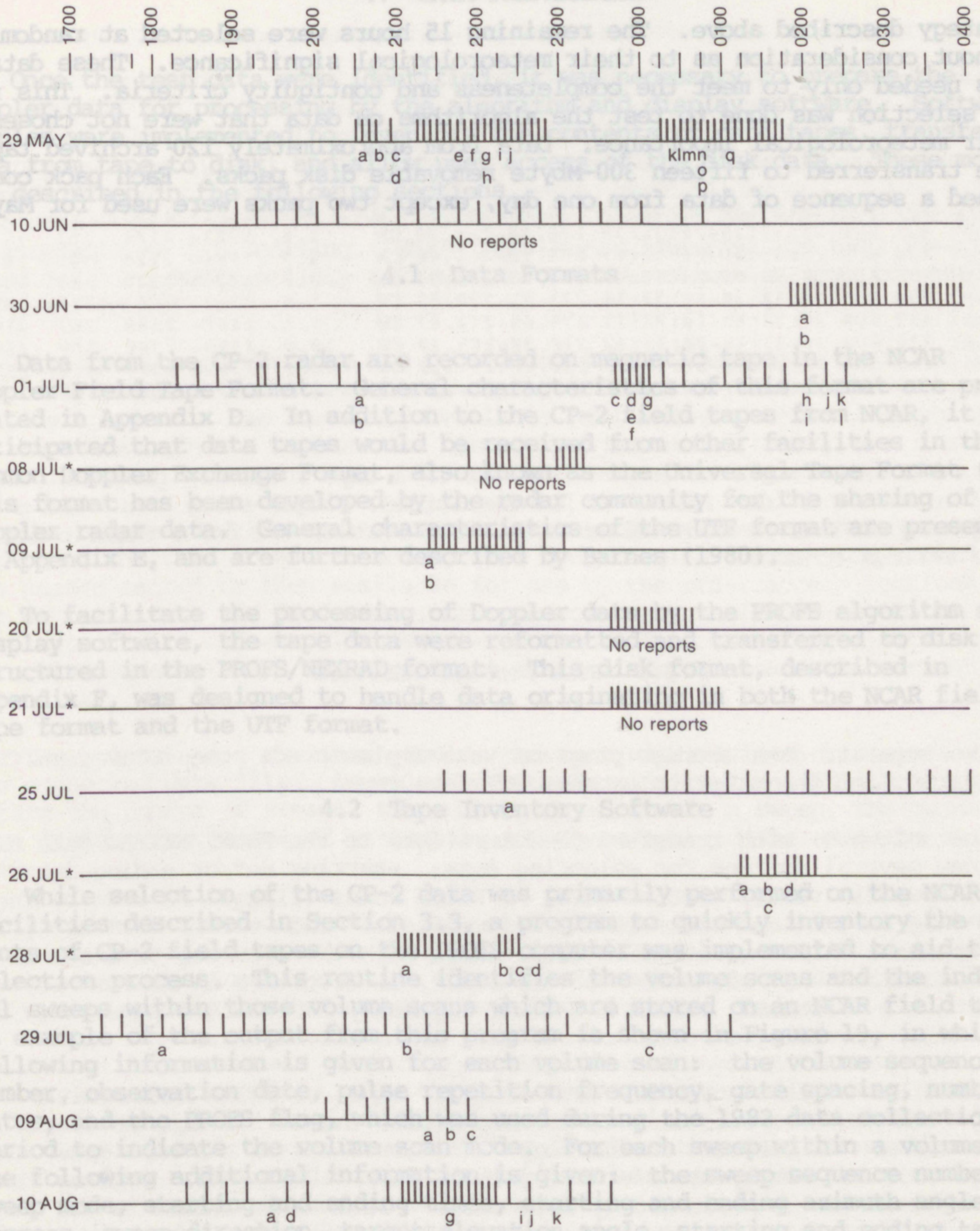


Figure 18. Spikes on the time lines represent PROFS CP-2 volume scans selected for processing. An asterisk (*) next to the date indicates a TSC volume scan. Letters under each line refer to a verification datum which can be found in Appendix C. Lack of verification data is indicated by "No Reports."

strategy described above. The remaining 15 hours were selected at random without consideration as to their meteorological significance. These data sets needed only to meet the completeness and contiguity criteria. This random selection was done to test the algorithms on data that were not chosen for their meteorological importance. Data from approximately 120 archived tapes were transferred to fifteen 300-Mbyte removable disk packs. Each pack contained a sequence of data from one day, except two packs were used for May 29.

4. DATA PREPARATION

Once the test data were identified, it was necessary to prepare the Doppler data for processing by the algorithm and display software. Software modules were implemented to inventory the contents of data tapes, transfer the data from tape to disk, and allow user access of the disk data. These modules are described in the following sections.

4.1 Data Formats

Data from the CP-2 radar are recorded on magnetic tape in the NCAR Doppler Field Tape Format. General characteristics of this format are presented in Appendix D. In addition to the CP-2 field tapes from NCAR, it was anticipated that data tapes would be received from other facilities in the Common Doppler Exchange Format, also known as the Universal Tape Format (UTF). This format has been developed by the radar community for the sharing of Doppler radar data. General characteristics of the UTF format are presented in Appendix E, and are further described by Barnes (1980).

To facilitate the processing of Doppler data by the PROFS algorithm and display software, the tape data were reformatted and transferred to disk files structured in the PROFS/NEXRAD format. This disk format, described in Appendix F, was designed to handle data originating in both the NCAR field tape format and the UTF format.

4.2 Tape Inventory Software

While selection of the CP-2 data was primarily performed on the NCAR RDSS facilities described in Section 3.3, a program to quickly inventory the contents of CP-2 field tapes on the PROFS computer was implemented to aid the selection process. This routine identifies the volume scans and the individual sweeps within those volume scans which are stored on an NCAR field tape. An example of the output from this program is shown in Figure 19, in which the following information is given for each volume scan: the volume sequence number, observation date, pulse repetition frequency, gate spacing, number of gates, and the PROFS flag, which was used during the 1982 data collection period to indicate the volume scan mode. For each sweep within a volume scan, the following additional information is given: the sweep sequence number, sweep mode, starting and ending times, starting and ending azimuth angles in degrees, sweep direction, target elevation angle, starting and ending tape record numbers, number of radials, and the total number of disk records which would be created for the volume scan disk file.

```
Vol:  35  29-MAY-1982  PRF: 480.0  Gate spacing: 600  #Gates: 512  Pflag: 2

Swp: 431 SUR 20:30:12-20:30:35  99.32- 89.25 CW    0.7  1801- 1859  176  177

Vol:  36  29-MAY-1982  PRF: 960.0  Gate spacing: 300  #Gates: 512  Pflag: 2

Swp: 432 SUR 20:30:50-20:31:15 270.59-269.27 CW    0.7  1861- 1983  367   368
Swp: 433 SUR 20:31:15-20:31:39 270.26-268.20 CW    1.5  1983- 2104  363   731
Swp: 434 SUR 20:31:39-20:32:03 269.19-269.08 CW    2.5  2104- 2225  365  1096
Swp: 435 SUR 20:32:03-20:32:28 270.07-272.94 CW    3.5  2226- 2348  368  1464
Swp: 436 SUR 20:32:28-20:32:53 273.91-278.72 CW    5.5  2348- 2467  358  1822
Swp: 437 SUR 20:32:53-20:33:17 279.69-277.67 CW    7.5  2468- 2588  363  2185
Swp: 438 SUR 20:33:18-20:33:42 290.50-285.42 CW   10.0  2589- 2708  360  2545
Swp: 439 SUR 20:33:42-20:34:06 298.28-293.12 CW   13.0  2709- 2828  360  2905
Swp: 440 SUR 20:34:07-20:34:31 305.95-300.92 CW   17.0  2829- 2948  360  3265
```

Figure 19. An example of output from the CP-2 tape inventory program. Here, two volumes are identified: a low-resolution, single-sweep scan, and a high-resolution, 9-sweep volume scan. See Section 4.2 for a description of the various columns.

4.3 Data Storage Software

Two separate data storage programs were implemented: one to process CP-2 data tapes, and the other to process UTF data tapes.

The software which transfers CP-2 tape data to the PROFS/NEXRAD data disk provides several options for selecting data: starting and/or ending data times, specific volume scan numbers, or PROFS scans (flagged as such by the PROFS flag in the tape housekeeping information). Each volume scan is written to a separate direct-access disk file, named with the starting time of the volume scan and the code number of the radar, as described in Section 6.4. As the data are processed, volume and sweep inventory information is written to a log file, providing a record of the data set's attributes.

A similar routine transfers data from UTF tapes to PROFS/NEXRAD disk files, with the user selecting to process either the entire tape or specific volume scans. Rather than perform any transformations on the tape data, the storage routines simply transfer the data to disk, bit-by-bit. Scaling to calibrated values is performed by data access routines described in the next section. Thus, CP-2 files contain raw 1-byte digital count values for each data element; files with UTF data contain 2-byte scaled integer words for each data element. One conversion that does occur for CP-2 data is a change from mountain daylight time (m.d.t.) to Greenwich mean time (G.m.t.). For UTF data, latitude and longitude values are converted from a degrees/minutes/seconds form to a scaled 2-byte integer form to conform with the method by which CP-2 values are stored. It should be noted that azimuth angle values are maintained in the radar coordinate system, in which zero degrees indicates north and angle values increase in the clockwise direction.

4.4 Data Access Software

A set of library subroutines and functions serve as an interface between the PROFS/NEXRAD data base and the various applications programs. These routines allow concurrent access to a data file by different software modules and protect the files from inadvertent modification. Also, this system provides a common access method for all users, permits changes to the data base structure to be transparent to the users, and allows for such software enhancements as data editing. Where possible, these routines have incorporated existing PROFS utility software for such functions as time conversions and error handling.

The data access software may be separated into four categories: file connect and disconnect routines, housekeeping information, data retrieval, and utility routines. The currently available data access routines, which reside in the NEXDAL (NEXRAD Data Access Library) object code library, are listed in Appendix K.

The connect routine opens a specified file for shared, read-only access, and reads the file's header record. The housekeeping information stored in the header record is then available for use by the other access routines. If the data are from the CP-2 radar, the connect routine also reads the CP-2 calibration file and establishes look-up tables for converting digital counts to scaled data values. The disconnect routine simply closes the currently open file.

The housekeeping routines provide the user with various characteristics of the connected data file. Examples of housekeeping routines are those that return the number of sweeps, the number of radials in a sweep, the number of data fields, the number of gates, the gate spacing, and the range to the first gate.

The data retrieval routines return calibrated data to the user. One routine returns data for all gates of all fields within a selected radial of a selected sweep. Another routine allows the user to specify a range of sample volumes or gates within the radial for which data are desired.

The utility routines perform a variety of functions which assist in the processing of a data file. For example, routines are available to find the index of the closest sample volume for a given slant range, find the index of the closest radial to a given azimuth angle, find the slant range for a given sample volume index, and obtain the sample volume and radial indexes for a given sector. Another routine allows a user to set a desired threshold value of R(tau) or spectral width which is used to reject velocity estimates during data retrieval.

Table 4. Processing Timing Summary

Action	Average Time
Tape Operations	
Store NCAR volume scan	3 min
Store entire NCAR tape (approx. 5 volume scans)	15 min
Inventory NCAR tape	6 min
Disk Operations	
Access and calibrate sweep (360 rays x 512 gates x 4 fields)	30 s
Access and calibrate 1 slant range for 1 elevation (360 rays x 1 gate x 4 fields)	5 s

4.5 Timing Summary

Timing statistics for various data processing operations are presented in Table 4. A typical CP-2 data tape contains about 20 minutes of data; transferring the data to disk requires about 75 percent of real time. The tape inventory procedure is considerably faster since no disk activity is required. The process of retrieving calibrated data for a sweep from the disk using the current access software actually takes about 25 percent longer than the nominal real-time sweep duration of 24 seconds (Table 3).

5. COMPUTER HARDWARE CONFIGURATION

All software generated for NEXRAD was coded, tested, and executed on a Digital Equipment Corporation (DEC) VAX-11/750 (DEC, 1980). This is a multi-user and multi-tasking computer. It has a 32-bit architecture and a speed of about .6 MIPS. The VAX also has a virtual memory operating system which allows a user to address much more memory than is available in core, more than 1 billion bytes. A large number of peripherals can be accommodated by the VAX. Figure 20 shows the configuration used during the NEXRAD project. The computer dedicated to NEXRAD is designated PROFS4. It is connected by a communications network to another PROFS computer.

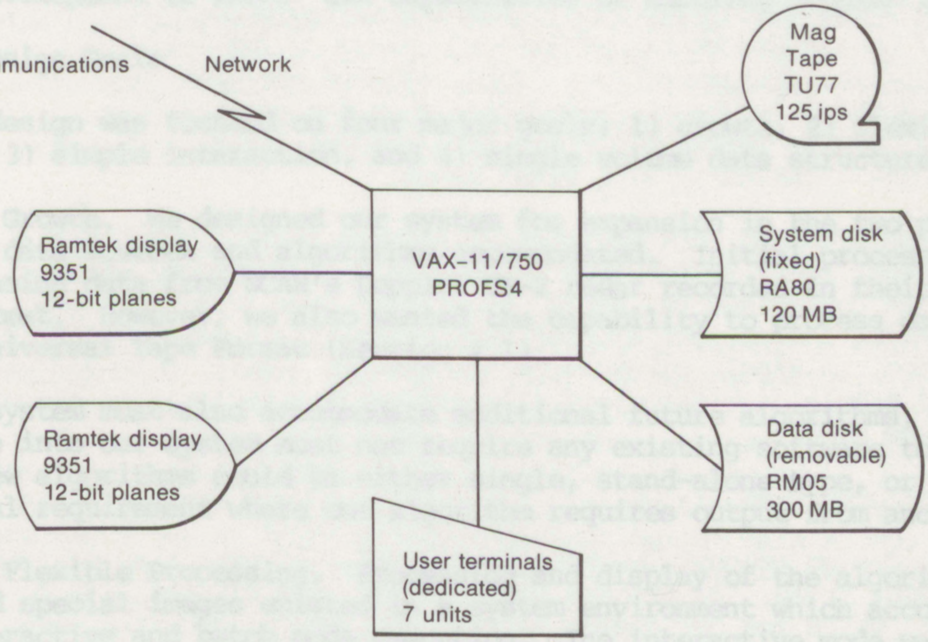


Figure 20. The hardware configuration of the Digital Equipment Corporation (DEC) VAX-11/750 computer used by PROFS to perform the NEXRAD software tasks.

PROFS4 has two disk drives, a 120-Mbyte system disk and a 300-Mbyte removable disk. This larger removable disk is mainly used for data storage. The 300-Mbyte removable disk typically holds about 28 full volume scans. Products generated from this data are also placed on the disks. The 125 IPS tape drive is used to transfer Doppler radar data recorded on magnetic tape onto one of the disk drives.

Seven terminals as well as two Ramtek 9351 display systems (Ramtek, 1979; Ramtek, 1978) are also connected to PROFS4. The Ramtek display system comprises a host processor, a display processor and refresh memory, and a display monitor. The Ramtek uses a raster-scan refresh technique and has a resolution of 512 x 512 picture elements. The refresh memory for the display contains storage of 12 bits per picture element. Each bit plane can be selectively enabled. Thus the Ramtek accommodates the storage of several displays which can be rapidly accessed to allow for such things as simulated loops.

6. SYSTEM SOFTWARE

6.1 Design Process

Our contractual agreement with NEXRAD required that we code eight specified algorithms, exercise them on a set of Doppler data, and produce a simple tabular display of the resulting algorithm products. However, we decided to make these requirements the core of a more versatile Doppler radar processing and display system. The scope of the software effort required a system design to assure coherent, maintainable, and technically correct software. Our aim was to design and implement a system which would not only satisfy NEXRAD's requirements, but would also make a significant contribution to the development of PROFS' own capabilities of handling Doppler radar data.

6.1.1 Design Goals

Our design was founded on four major goals: 1) growth, 2) flexible processing, 3) simple interaction, and 4) single volume data structure.

6.1.1.1 Growth. We designed our system for expansion in the two primary areas of data sources and algorithms accommodated. Initial processing would be done using data from NCAR's Doppler CP-2 radar recorded in their standard field format. However, we also wanted the capability to process data recorded in the Universal Tape Format (Section 4.1).

The system must also accommodate additional future algorithms, and their inclusion into our system must not require any existing software to be modified. New algorithms could be either single, stand-alone type, or have a sequential requirement where one algorithm requires output from another one.

6.1.1.2 Flexible Processing. Processing and display of the algorithm products and special images existed in a system environment which accommodated both interactive and batch mode execution. The interactive mode was normally used for software development and processing of a single algorithm. The user selected which algorithm program to execute and which data to use. Special versions of the support files containing data sectors, algorithm processing parameters, etc., could be easily generated and associated with the executing algorithm. Any diagnostic error messages were displayed directly to the user's terminal.

The batch mode was typically used for algorithm product production for a set of algorithms over a sequence of volume scans. All input required by the programs was provided by a run-time executive command procedure rather than by interactive user input. In this mode, production runs of many hours could be performed unattended. Production versions of the support files were given special names depending on the type of production. Any run-time diagnostic error messages were stored in a log file for later inspection.

6.1.1.3 Simple Interaction. Both user and programmer interactions were minimized and kept very simple. Primary program input consisted only of specifying which volume scan of which radar was to be processed. This information was transferred to the program through a logical name definition mechanism. All other input to the program was via disk files. Names of the physical disk files were either determined internally in the programs or assigned through use of logical names. This approach allowed a great deal of run-time flexibility without any software code modification. All output information, with the exception of run-time error messages, was via disk files. Likewise, the names of the output files were determined internally in the programs.

To keep programmer interaction at a minimum, processing of input/output was done by a set of tightly controlled I/O software modules. The number of different file structures was minimized. A set of file handling utilities was also provided for programmer use. This software included file connect/disconnect modules, software needed to access the various support files, specific read/write software for the data/product files, and software needed for the proper translation of logical names.

6.1.1.4 Single Volume Data Structure. A major design decision was to handle each individual radar volume scan of data as an independent entity. This decision then determined the basis of our data base file structure, influenced the file naming conventions adopted, and inherently characterized the entire system design. A single volume scan of data for a given date and radar was stored in a single disk file, and all software was designed to execute on a single volume scan.

6.1.2 Techniques Used

For the design process, we used four tools: 1) data flow diagrams, 2) requirement trees, 3) process descriptions, and 4) data item descriptions. As a general guide for our design methodology, we incorporated techniques developed by DeMarco (1978). Examples of the four different design documents may be found in Appendix G.

PROFS had previously developed software to aid in the system design process of various projects undertaken by PROFS. The examples in Appendix G were generated on our VAX computer system using this software. The software not only provided the necessary graphical output displays, but also did internal consistency checks among the four main design areas.

6.1.2.1 Data Flow Diagrams. For the design, we created 10 separate data flow diagrams. These diagrams showed how various kinds of data (Doppler data, algorithm products, processing parameters, etc.) flowed through the individual processes of our system. Data flow diagrams were generated for three levels of refinement. Each flow diagram has a unique name and its number indicates the refinement level; 0 = top level, N.0. = first refinement, and N.m. = subsequent refinement levels. The hexagonal boxes indicate various software processes. The lines indicate the flow of data items to and from the boxes and the arrows indicate the direction of flow.

6.1.2.2 Requirement Trees. A requirement tree document existed for each of the 10 data flow diagrams. These documents were very helpful in showing the hierarchical definition of the system's functional requirements. Each of the processes shown as a hexagonal box in the data flows is pictured as a rectangular box in the requirement tree documents.

6.1.2.3 Process Descriptions. Major functional areas were defined as system processes. Our design specified 33 separate processes. The data flows between these processes were graphically given by the data flow diagrams and the hierarchy of their functionality by the requirement trees. Their corresponding process descriptions simply itemized, in a structured text form, the major functions of each process.

6.1.2.4 Data Item Descriptions. For each major data structure or data item shown in the data flow diagrams, a description existed. The design specified 39 of these items. These descriptions, like those for the processes, were done in a structured text form, but in more detail. We made liberal use of relational operators and appropriate indenting to visually emphasize the top-down partitioning of each data item structure.

6.2 Software Environment

Since all software of the NEXRAD project was to reside on a dedicated VAX-11/750 computer, we were free to establish a custom software environment. The computer was configured with the standard VAX/VMS operating system software. Included in this system software were the necessary text editors, VAX-FORTRAN, and network communications software for access to other computers housed within the PROFS computer facility. In addition to the system software, there was a set of PROFS standard meteorological data access and processing libraries. An extensive software library containing display modules was also included.

6.2.1 Directory Structure

After the computer system software was configured, two major user accounts were created. The NEXRAD account contained all software and associated documentation. The second account, called NEXDAT, was used for the Doppler data, resulting algorithm output products, and special display images. These two accounts were further subdivided into various directories. Primary directories in the NEXRAD account were established for the major categories of design, documentation, program source code, subprogram source code, object libraries, executable program images, and data acquisition/access code.

In the NEXDAT account, three primary directories were created. The first was for the Doppler data, the second was for the output products generated by the algorithms, and the third was for any specially prepared display images.

6.2.2 General Guidelines

In the early stages of the project we established a set of guidelines for use in design and coding of the algorithms. These guidelines were reviewed and approved by both project sponsor and PROFS management. Adherence to the guidelines helped achieve one of the primary goals of the project: to produce well-documented code which faithfully represented the supplied algorithm descriptions. These guidelines can be grouped into the six general areas given below.

6.2.2.1 Exclusive Use of FORTRAN. All code was written in the FORTRAN language as available under the VAX operating system. This particular version of FORTRAN was an implementation of the ANSI X3.9 FORTRAN-77. It also incorporated a substantial set of VAX-oriented extensions including additional data typing, file handling capabilities, and constructs used in structured programming.

6.2.2.2 Emphasis on Modularity and Readability. All members of the programming team were encouraged to write code that was, above all, modular and readable. Tricky or clever code, at the expense of clarity, was discouraged. Any programmer with basic skills must be able to easily grasp both the flow and details of the code. Modularity was of particular importance both for understanding the code and for ease of error tracing. Segmenting the code into small, individual modules resulted in considerable time saving during the testing and debugging phases of the software development.

6.2.2.3 Minimum Optimization. Little effort was directed to writing code that was highly optimized. With the exception of code written for accessing the raw data, emphasis on execution speed was minimum. The amount of memory required (resident and/or virtual) was of little concern. Mass storage requirements were considered only when limited by our available resources.

6.2.2.4 Minimum Real-Time Processing Considerations. Although these algorithms would eventually have to execute in a real-time operational environment, there was no major effort to write code for real time. Each algorithm was designed and coded to operate as a totally independent software module. This fact precluded real-time processing optimization, since many of these algorithms could be processed in a semi-parallel fashion.

6.2.2.5 Minimum Algorithm Interdependence. The interfaces and communication between the algorithms were kept to a minimum. Therefore, transfer of information among algorithms would be limited, controlled, and well-defined. The mechanism for these transfers was disk files. All input and output for an individual algorithm was provided by reading from or writing to a disk file. This scheme clearly limited execution speeds since disk I/O operations are notoriously slow compared with other methods of information transfer.

6.2.2.6 Minimum Emphasis on Transportability. All code written was specific to our particular hardware configuration, a VAX-11/750 computer equipped with Ramtek 9351 display hardware. If deemed appropriate, full advantage was taken of the hardware and associated software features available.

6.2.3 Coding Standards

In parallel with the general design and coding guidelines, the project incorporated PROFS coding standards which have been used very successfully for the last several years. These standards emphasized good structured programming techniques and incorporated some local coding conventions. Three aspects of these standards were of particular note: 1) module and variable names, 2) error trace-back procedures, and 3) use of dynamic error diagnostic mode.

6.2.3.1 Naming Conventions. Module names of subprograms typically contained 11 characters, FST_PDT_WRT_XA. The first three characters are the algorithm's assigned 3-character code (SEG - Storm Segments, CEN - Storm Centroids, TRK - Storm Tracking, FST - Storm Position Forecast, STR - Storm Structure, HAL - Hail, VAD - Velocity-Azimuth Display, and MES - Mesocyclone Detection). The second three characters indicate the subject of the module's code. The third three characters indicate the process applied to the module's subject. The last two characters are sentinel characters. The first is the PROFS character assigned to the NEXRAD project, X. The second character indicates the functionality of the module; A for data Access, D for Display, G for General, M for general Meteorology, and Q for acQuisition/data management.

6.2.3.2 Error Trace-Back. PROFS has developed rigorous error trace-back procedures which were included in all of its project controlled software. These procedures were also incorporated in all of the NEXRAD software. After execution, each module returned a piece of information, typically called ISTATUS, which indicated whether the module executed as expected or had problems. Upon return of the module, ISTATUS was set to the value of one of a set of established error return codes. These errors could be fatal, non-fatal, or informative. Standard programming techniques then required checking of the value of ISTATUS before using any of the results of the called module. If at all possible, no module, called by a higher level module, was allowed to terminate execution unexpectedly, but rather was required to pass error information "back up the line." The ultimate result was a complete error track-back capability from the lowest level subprogram module back up through the highest level module.

6.2.3.3 Dynamic Error Diagnostic Mode. A companion capability to error trace-back was the dynamic error diagnostic mode. Software existed which allowed diagnostic output of the error trace-back information. The output could either be directed to the programmer's console for interactive debugging or to a disk file for later error analysis. This diagnostic mode could be invoked by methods external to the FORTRAN code so that software did not have to be recompiled or relinked for the diagnostic mode. Normally, during the testing and debugging phases of software, the diagnostic mode was invoked. Later, after thorough testing had been completed, the diagnostic mode was disabled.

6.2.4 Documentation Requirements

For each software module of code, two forms of documentation were written: internal and external.

6.2.4.1 Internal Documentation. The module's internal documentation consisted primarily of a project "box" near the beginning of each module, a "history" section, and a generous number of in-line comments. The box helped identify the module as being part of the particular PROFS project: "PROFS/NEXRAD '83 ALGORITHM EVALUATION TASK." The history section tracked all changes made to the code. Included in this section were the programmer's name, date, and a brief description of the action being taken. This section was extremely valuable for tracking changes to the code, who made them, and why. It was also an integral part of software configuration management control. In addition to the project box and history section, there were internal comment lines. Programmers were encouraged to liberally comment the flow of the code. Comment lines started with the character "!". An example of internal documentation is given in a partial listing of the module READPARMS_XA in Appendix H. For cases where the code directly implemented a particular line from the algorithm description, an internal comment was used to cross reference that line. All lines of the algorithm description had a numeric label (Appendix A).

6.2.4.2 External Documentation. The module's external documentation consisted of an individual write-up, in a standardized format, giving the module's functional description and detailed instructions for usage. This document also included the programmer's name, date of the software's current version, and instructions for linking to the library where the module's object code was stored. Each document contained sufficient detail that it was not necessary to inspect the code to properly use the module. An example of a complete document for module READPARMS_XA can be found in Appendix I. In addition to the individual module write-ups, various index listings were also maintained. These indexes could be organized alphabetically or according to subject matter. The index contained the module's name and a one-line functional description. Four indexes, listed in Appendix K, were maintained for the NEXRAD project: 1) data access routines stored in the library NEXDAL, 2) algorithm processing routines stored in the library NEXALGOR, 3) display routines stored in the library NEXDISP, and 4) programs and command procedures.

6.2.5 Conformance with NEXRAD Standards

The NEXRAD JSPO has established standards which their system contractors must follow for the final operational systems. We were not contractually committed to follow these standards. However, looking at the NEXRAD standards as they apply to computer software, we noted that we had followed most of their standards. These standards are discussed in detail in Section 3.3.8 in NEXRAD (1983a). Table 5 summarizes PROFS compliance with the NEXRAD standards. Three categories of our software are listed. The first, Data Restoration/Preparation, consisted of all the software required to prepare the Doppler data for processing. The second, Algorithm Processing, consisted of the software which actually implemented the algorithm descriptions. The third, Algorithm Evaluation, consisted of software written for display of both Doppler data and algorithm output products. Note that all software of the Algorithm Processing category was in total compliance where applicable.

Table 5. Compliance With NEXRAD Software Requirements

Requirement Description	Data Restor./Prep.	Algorithm Processing	Algorithm Evaluation
Computer program functional composition	P	X	P
Design techniques	P	X	P
Documentation constraints	P	X	P
Coding conventions and standards	X	X	X
Language standards	X	X	X
System software augmentation	X	X	X
Microprogramming	N/A	N/A	N/A
Microprocessor software	N/A	N/A	N/A
Program regeneration	N/A	X	P

P = partial compliance
X = total compliance

6.3 Algorithm Software

In order to implement the system design within the confines of our customized software environment, several characteristics of the algorithm software had to be established. First, upper bounds, or processing limitations, were set for various items such as the number of radars that the system could handle. Second, we decided how the software would be organized in terms of executive command procedures, programs, and subprograms. Third, we defined what type of support data files would be required. Finally, we specified exactly what mode of user interface to the system would be used. Each of these four system characteristics is discussed in the following sections.

6.3.1 Processing Limitations

When FORTRAN code is written, storage must be allocated for various arrays. Hence, it was necessary to assign program upper bounds for these arrays. These bounds were defined by the processing limitations agreed upon

Table 6. NEXRAD Processing Limitations

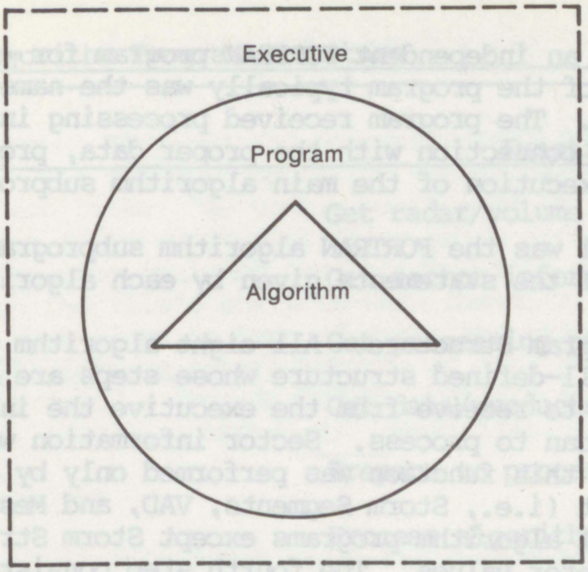
Item	Maximum Number
Algorithms	25
Radars	10
Sweeps/volume	20
Radials/sweep	720
Gates/radial	1024
Storms/volume	50
Positions/storm	50
Slant ranges/VAD	20

by PROFS and NEXRAD management and compatible with our software environment. The limitations addressed the number of algorithms and individual radars that the system can accommodate. Maximum data resolution, both spatial and temporal, was also addressed. Table 6 itemizes our eight current processing limitations.

6.3.2 Software Design Implementation

To implement the software design, we used one or more independent FORTRAN programs for each of the four major processes defined by the top level of our design (see Appendix G, Data Flow Diagrams; 0 - Evaluate Algorithms). If appropriate, the program would be initiated by an executive command procedure which would commence execution of the program with the proper data files. Guidelines for the implementation structure of the first two processes, Select CP-2 Data and Ingest Tape Data, were loosely defined with details left to programmer discretion. The third process, Execute Algorithms, which was part of the realization of our contractual commitments to NEXRAD, was very tightly specified. Details of this specification appear in the following two sections. The fourth and last process, Perform Evaluation, followed the Execute Algorithms specifications.

6.3.2.1 Generalized Product Generation Structure. The product generation software used for the Execute Algorithms process is graphically depicted in Figure 21. This structure had three levels: executive, program, and algorithm subprogram. On the very top level was the executive. For execution



EXECUTIVE:

Determines algorithm to execute
Passes radar/time to program

PROGRAM:

Gets processing information
Gets input data/products
Connects files
Initiates algorithm subprogram
Cleans up and terminates

ALGORITHM:

Implements ONLY algorithm code

Figure 21. Graphic representation of the product generation software structure. An executive controls the selection and processing details of an algorithm FORTRAN program. The program, in turn, establishes run-time connection to all input/output files and acts as a driver for the algorithm subprogram which actually executes the coded statements of the algorithm description.

of the algorithms in a batch environment, this executive took the form of a command procedure file containing command strings in DCL (Digital Command Language). This procedure initiated the appropriate algorithm program and made necessary logical translations required to execute with the proper radar data volume scan and support files. In the interactive/development environment, the executive was normally one or two DCL commands.

The next level was an independent FORTRAN program for each of the eight algorithms. The name of the program typically was the name of the algorithm, shortened if necessary. The program received processing information from the executive, established connection with the proper data, product, and support files, and initiated execution of the main algorithm subprogram.

On the lowest level was the FORTRAN algorithm subprogram which was the coded implementation of the statements given by each algorithm description.

6.3.2.2 Algorithm Program Structure. All eight algorithm product generation programs followed a well-defined structure whose steps are given in Table 7. The first function was to receive from the executive the information specifying which volume scan to process. Sector information was obtained in the second step. However, this function was performed only by algorithms which used Doppler data input (i.e., Storm Segments, VAD, and Mesocyclone Detection). Third, all algorithm programs except Storm Structure read in their processing parameter values. The fourth step consisted of reading either the Doppler input radar data or another algorithm's output product file(s). The fifth step included various housekeeping functions needed for processing the algorithm. One function was to establish an appropriate output product file for storage of results generated by the algorithm. It was at this step that appropriate housekeeping information, including processing parameter values used, was written to the product file. The sixth program step was to invoke the main algorithm subprogram. This subprogram, which actually implemented the algorithm description statements, stored the algorithm's products in the output file previously established. After all statements were executed, the subprogram returned control to the main program, which then closed and disconnected all input/output data/product files and performed any other necessary housekeeping functions. The last step involved logging any run-time errors or diagnostic status messages to a file and terminating the program.

In conformance with software guidelines discussed in Section 6.2.2, we required the code implementing the algorithm description statements to exist within a single main subprogram. However, this main subprogram could invoke other subprograms. This subprogram contained the internal comments that tracked the statements of the algorithm description (see Section 6.2.4.1). Each main algorithm subprogram followed a naming convention requiring the name to be of the form xxx_PROCSS_XM where xxx is the algorithm's appropriate 3-letter abbreviation (SEG - Storm Segments, CEN - Storm Centroids, MES - Mesocyclone Detection, etc.).

6.3.3 Support Files

One design goal was a system that could expand to accommodate additional radars and algorithms. A second goal was simple interaction for both user and programmer. To achieve these design goals, we implemented four support files: Radar, Algorithm, Sector, and Parameter. These disk files contained all the information which was constant for a particular version of the system but which could be easily modified as conditions change. This technique allowed us to easily add radars or algorithms or modify the algorithm's processing environment. The files had sequential access and their contents were in ASCII text.

Table 7. Algorithm Program Structure

Step	Function
1.	Get radar/volume time to process.
2.	Get sector information.
3.	Get processing parameters.
4.	Get data/products to process.
5.	Prepare to process.
6.	Process algorithm subprogram.
7.	Clean up.
8.	Errors and termination.

The purpose and details of these four files are given in the following sections. Examples of each support file are given in Appendix J.

6.3.3.1 Radar. The Radar support file specified which radars could be processed by the system. It informed the system how many radars existed, their unique numeric codes, their individual code names, and geographic locations.

6.3.3.2 Algorithm. The Algorithm support file specified which algorithms could be executed. It gave the number of available algorithms, their unique one-character codes, and algorithm code names.

6.3.3.3 Sector. The Sector file allowed data sectors to be defined for each algorithm and for each specified radar. Geographic location of a given radar or an algorithm's particular sensitivity to certain data anomalies might require that certain sectors be excluded from the processed data. This sector was defined by beginning and ending azimuth values and beginning and ending slant range values.

6.3.3.4 Parameter. The Parameter support file provided for algorithm processing flexibility. All current algorithms, with the exception of Storm Structure, required a number of different processing parameters. These parameters were often empirically derived values or might be user-supplied values in an operational environment. Some might be dependent upon surrounding terrain, could have a seasonal or diurnal dependence, or even be dependent upon the resolution of the raw radar data processed. The numeric values used were jointly established by PROFS, NEXRAD management, and the source scientists. These values can be changed without modifying the algorithm's code.

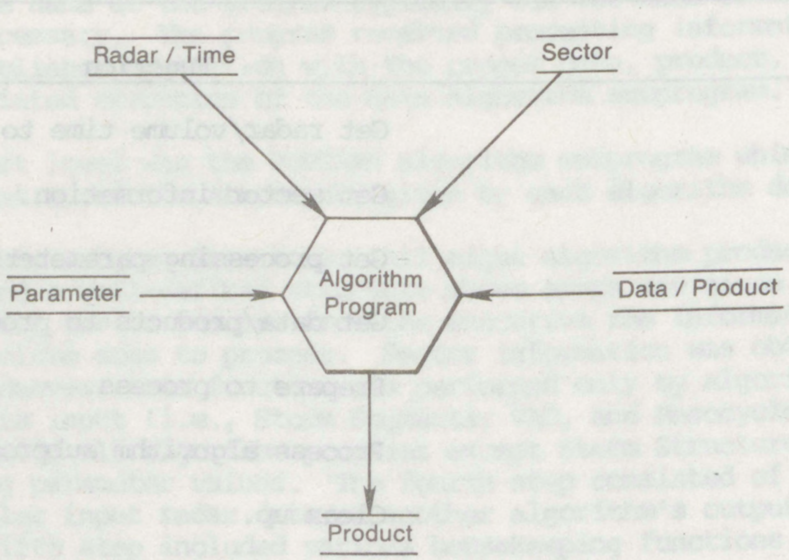


Figure 22. Graphic representation of an algorithm program construct showing four input structures and one output structure.

6.3.4 User Interface

The final system characteristic was the mode of user interface with the various algorithm programs. We specified a user interface with minimum input. The user first issued the command statement, INPUT. The system responded by asking the user two questions: 1) which radar? and 2) which volume scan? The user identified the radar by numeric code and the volume scan by a date/time group. The user needed only to specify the algorithm to be executed. This user interface format was consistent throughout the system both for programs which generated the algorithm products and for programs which displayed data and products.

Figure 22 schematically represents how major file structures related to the algorithm program. All programs which produced algorithm products adhered to this form. Input was provided by the user interface as indicated by the "Radar/Time" area, support files indicated by "Sector" and "Parameter" areas, and required data or products as shown in the "Data/Product" area. The program's only output was an algorithm product as illustrated by the "Product" area. All subprograms written to support algorithm product generation were collected into one software library called NEXALGOR. Appendix K lists the library modules along with corresponding programs and command procedures for algorithm product generation.

6.4 Data and Product Files

All input and output to algorithm processing and display programs was through disk files. The computer operating system permitted some flexibility in the naming of these files. We took advantage of this by establishing a method of naming data and product disk files which informed the user about the file's contents. The name included the date and time of the beginning of the volume scan on which the file's contents were based, it indicated which radar acquired the data, what class or type of data, and primary use of the data. In addition, all file formats adhered to one of four basic file structures described in Section 6.4.2. Imposing these controls on our files yielded software uniformity, easy use, and clear identification.

6.4.1 Naming Conventions

6.4.1.1 File Name. All file names had a basic form of a 9-character file name followed by a period and a 3-character file extension. The format of the file name was "YYJJJHHMM" where

YY = last two digits of year (82 = 1982)
JJJ = Julian day of the year (222 = August 10, 1982)
HH = G.m.t. hour (00-23)
MM = G.m.t. minute (00-59).

The date and time always referred to the beginning of the volume scan.

6.4.1.2 File Extension. The format of the data and product file extension was RCS where

R = radar code
C = class
S = subclass.

Current radar codes, R, were

1 = CP-2 NCAR, Boulder, CO
2 = CHILL, Chicago, IL
3 = NSSL, Norman, OK.

Classes or types of files, C, were

V = volume scan of raw Doppler data containing all data fields
P = algorithm product data.

Subclasses, S, were

n = sequence number used with Class V file
A = algorithm code where

C = Storm Centroids
E = Storm Segments

F = Storm Position Forecast
H = Hail
M = Mesocyclone Detection
S = Storm Structure
T = Storm Tracking
V = Velocity-Azimuth Display.

The format of the special images file extension was SDN where

S = Sweep number
D = Data source
N = Sequence number.

Sweep numbers, S, were in hexadecimal notation. A code of 0 indicated several different sweeps in an image.

Current data sources, D, were

A = Velocity-Azimuth Display products
C = Storm Centroids products
D = Doppler velocity data
E = Storm Segments products
F = Storm Position Forecast products
H = Hail products
M = Mesocyclone Detection products
R = Reflectivity data
S = Storm Structure products
T = Storm Tracking products
X = Combination of data and products.

The sequence number, N, varied from 0 to 9.

6.4.2 File Structure

6.4.2.1 Data. These files provided reflectivity, Doppler velocity, and normalized coherent power (R(tau)) values to algorithm processing and display software. The three programs directly using the data files were Storm Segments, Mesocyclone Detection, and Velocity-Azimuth Display. Data from a single volume scan was stored as an unformatted, direct (random) access file. Shared access in "READ" mode allowed several programs to read the file simultaneously. Each file had a header section giving file organizational information and a data section containing the actual numeric fields. See Section 4.1 for a general description of the data file structure and Appendix F for the details of the file format.

6.4.2.2 Product. These files contained output products generated by a specified algorithm process. Although a file was written by the software that generated the products, the file could be read by several other programs. One file was generated by each algorithm for each volume scan. The file name was the same as the data file name of the volume scan used to generate the products, and the file extension indicated the code of the algorithm producing the file. Except for the Storm Segments product file, the files were formatted sequential-access files. Because of data volume, Storm Segments produced an unformatted direct-access product file. All product files had shared access in read mode. Although each file had a format unique to the generating algorithm, the files all had a housekeeping section followed by a data product section. The housekeeping section contained not only file organizational information, but also the values of the processing parameters used to produce the output products along with ancillary information needed for display purposes. The data product section contained the values of the products.

6.4.2.3 Special Images. These files were generated to provide rapid display of Doppler data. The files were written for a specific display device. The file was in unformatted direct-access form with an optional header record, followed by 512-byte fixed-length records. Each record corresponded to one line on the display device. All data contained in the file were converted to the coordinate system used by the Ramtek display device.

6.5 Display Software

6.5.1 Overview

For the data selection and algorithm evaluation phases of the project, we wrote software to display both Doppler data in PPI format and the products of individual algorithms. Command procedures were used to process sequences of the display programs through user-interactive, menu-driven software. Image and graphic display software were written for Ramtek 9351 display devices, and tabular values were displayed on a standard alphanumeric terminal.

There were three categories of displays: an image display which showed Doppler data in a PPI format, a graphic which displayed the algorithm products using vector graphics, and a tabular display which summarized the results of an algorithm.

Image and graphic displays had several common features which allowed generation of special display products. A roam and zoom function allowed magnification of any selected area. The display screen could be partitioned for viewing multiple products with automatically scaled range and azimuth marks. Several color tables were available for enhancing features in the display and selected features could be flashed on and off.

6.5.2 Image Displays

Image displays showed radar data in a PPI format. Data source options included the display of an entire volume scan with all sweeps shown on one screen, or the display of a single sweep at a selected elevation angle. The three data fields available for display were reflectivity in dBz, Doppler velocity in meters per second, and R(tau), a measure of signal coherence. Reflectivity could be thresholded at any level, and a velocity qualifier could be used as a crude ground clutter suppression mechanism. An R(tau) qualifier could be used to minimize the effects of noisy velocity data.

6.5.3 Graphic Displays

Vector graphic displays illustrated algorithm output products. Data source options included Storm Sequence products, Mesocyclone Detection products, and VAD products.

Storm Sequence products were used to generate 1) plots of all storm centroids for a given scan, labeled with numbers showing each storm's rank, 2) plots of storm tracks, and 3) plotted storm structure information and forecast positions for all storms with a track history (Fig. 2).

The Mesocyclone Detection algorithm product was used to plot the locations of areas of cyclonic circulation for selected sweeps.

VAD products were used to plot vertical profiles of horizontal wind speed, horizontal wind direction, horizontal divergence, and vertical wind speed (Fig. 13). Velocity versus azimuth plots were also generated (Fig. 12).

Graphics products could be overlaid on each other or on an image, in any combination.

6.5.4 Tabular Displays

Numeric tables summarized the algorithm products on a standard alphanumeric display device. All displays had a header to indicate which algorithm products were being displayed. Each display included all output products specified by NEXRAD in addition to other information, including processing parameters used. Tabular displays were designed to be interactive. Upon execution, the software presented appropriate brief summaries of algorithm products. The software then permitted the user to display greater levels of detail.

6.6 Software Implementation

During the implementation phase, algorithm software was coded under very strict controls. These controls addressed changes in software configuration and design, NEXRAD approved changes to the algorithm descriptions with accompanying documentation, and a final software critique.

6.6.1 Configuration Control

After the first stages of code had been written and verified, the design and all accompanying software were placed under configuration control. Table 8 itemizes these data structures and process modules.

Changes in code required approval and proposed design changes were carefully analyzed before implementation. We also required all the software to be documented to PROFS specifications, which include the internal and external

Table 8. Software Tracking and Control

Major Data Structures (With Data Item Descriptions)	Number
Support	15
Products	17
Display	7
	39

Major Processes (With Process Descriptions)	Number
Data selection	9
Data storage	3
Algorithm execution	15
Algorithm evaluation	6
	33

documentation discussed in Section 6.2.4. We also documented all data structures through use of the Data Item Description (example given in Appendix G). Table 9 summarizes software under configuration control.

6.6.2 Conference Questions

A computer conference link was established to document communications between NEXRAD, PROFS, and the algorithms' source scientists. When suitable questions arose, these were transmitted to NEXRAD via the conference link. An example is given in Figure 23. To stay within our very tight schedule, we not only posed the question, but also suggested a solution which would be followed unless NEXRAD directed us differently. This conference link was extemely valuable for tracking questions and their associated answers. Since some questions led to modifications of the original algorithm descriptions, complete documentation was essential.

Table 9. Software Statistics

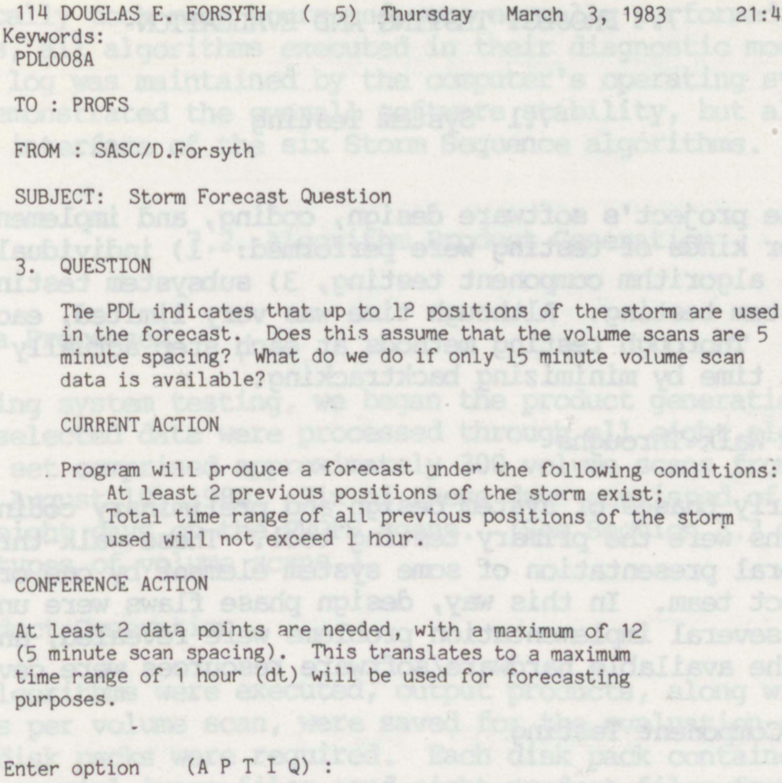
	Algorithm Description Lines	FORTRAN Lines	Comment Lines	Total Size (bytes)
Algorithm				
Segments	26	398	322	258,048
Centroids	47	519	336	152,576
Tracking	15	581	433	102,400
Forecast	11	390	192	62,976
Structure	77	367	271	54,784
Hail	79	399	325	56,320
VAD	23	269	112	92,160
Mesocyclone	41	415	398	385,024
New NEXRAD Code		17,000 (approx.)	6,000 (approx.)	
Existing PROFS Code		3,200 (approx.)		

After completing the software coding, we categorized all questions into various topics. Table 10 gives a summary of the number and types of questions.

6.6.3 Algorithm Critiques

After finishing the algorithm coding, the programming team prepared critiques of their respective algorithms. The purpose was to provide a uniform assessment of processes involved in translating individual algorithm description statements into technically correct, executable computer code.

Each programmer used a standarized critique form which addressed algorithm specification, implementation, and operation. Questions asked via the conference link were addressed on the algorithm level. The strengths and weaknesses of the algorithm description were summarized for later analysis. Appendix L contains a copy of the complete critique for the Storm Sequence's Hail algorithm.



```
114 DOUGLAS E. FORSYTH   ( 5) Thursday   March 3, 1983      21:42.27  IOTF
Keywords:
PDL008A

TO : PROFS

FROM : SASC/D.Forsyth

SUBJECT:  Storm Forecast Question



3.  QUESTION

    The PDL indicates that up to 12 positions of the storm are used
    in the forecast.  Does this assume that the volume scans are 5
    minute spacing?  What do we do if only 15 minute volume scan
    data is available?

    CURRENT ACTION

    Program will produce a forecast under the following conditions:
        At least 2 previous positions of the storm exist;
        Total time range of all previous positions of the storm
        used will not exceed 1 hour.

CONFERENCE ACTION

    At least 2 data points are needed, with a maximum of 12
    (5 minute scan spacing).  This translates to a maximum
    time range of 1 hour (dt) will be used for forecasting
    purposes.


Enter option    (A P T I Q) :
```

Figure 23. An example of a conference link communication between NEXRAD and PROFS concerning algorithms.

Table 10. Summary of Conference Questions

Topic	Number of Questions
Equations or units	16
Parameters	12
Radar characteristics	8
Algorithm clarification	4
Storm labeling	4
Coordinate system	3
Input/Output	2
Precision	2
TOTAL	51

7. PROJECT TESTING AND EVALUATION

7.1 System Testing

Throughout the project's software design, coding, and implementation phases, four major kinds of testing were performed: 1) individual structured walk-throughs, 2) algorithm component testing, 3) subsystem testing, and 4) integrated system testing. Although time was very limited, each phase was carefully tested. Thorough testing methods at each step actually shortened total development time by minimizing backtracking.

7.1.1 Structured Walk-Throughs

During the early phases of system design and preliminary coding, structured walk-throughs were the primary testing tool. These walk-throughs consisted of an oral presentation of some system element by one or more members of the project team. In this way, design phase flaws were uncovered and resolved. Also, several implementation problems were revealed, and solutions compatible with the available hardware/software resources were devised.

7.1.2 Algorithm Component Testing

Completed algorithm programs consisted of up to 25 directly called software modules. All I/O modules requiring direct access to the Doppler data files were the responsibility of one team member. Likewise, all I/O modules which read or wrote the algorithms' product files were the responsibility of another team member. Assignment of coding tasks to these two members resulted in a certain amount of control, uniformity, and consistency of module interface across the entire system. Each responsible member thoroughly tested these modules before the routines were released for use within the individual algorithm programs. In addition to thorough testing of all I/O modules, all programmers were responsible for testing individual modules of their programs.

7.1.3 Subsystem Testing

After all modules of an individual algorithm program were tested and verified, the programmer then performed a subsystem test. This test consisted of executing the particular algorithm on several sets of Doppler data, selected so that all features of the algorithm were exercised. In some cases, synthetic data sets were generated for thorough testing. Programs were individually executed on a set of volume scans, and the results of the output product files were carefully examined. Some programs incorporated intermediate diagnostics.

7.1.4 Integrated System Testing

After all eight algorithm programs had been individually tested, an integrated system test was performed. Several test data sets, each comprising approximately 15 sequential volume scans at either 5- or 15-minute frequency, were selected. A command procedure was created which would sequentially execute all the algorithms on all of the volume scans of each data set. These tests typically took many hours and were normally performed at night. During these tests, all algorithms executed in their diagnostic mode and a detailed processing log was maintained by the computer's operating system. The tests not only demonstrated the overall software stability, but also tested the sequential interface of the six Storm Sequence algorithms.

7.2 Algorithm Product Generation

7.2.1 Data Processed

Following system testing, we began the product generation phase. At this time, all selected data were processed through all eight algorithms. The total data set comprised approximately 300 volume scans from 14 days between May 29 and August 10, 1982. Six of these days consisted of the TSC volume scans and eight days of the PROFS scans. (See Section 3.3 for details of these two types of volume scans.)

7.2.2 Product Generation

When algorithms were executed, output products, along with four special image files per volume scan, were saved for the evaluation activity. Fifteen 300-Mbyte disk packs were required. Each disk pack contained the Doppler data, four special image files, and eight product files from each of the algorithms. Approximately 40 minutes of dedicated computer time were required to process one volume scan through all the algorithms and to produce the four special image files. (Detailed timing statistics are given in Section 7.3.)

7.2.3 Product Review

After a selected day's volume scans were processed, we carefully reviewed results for reasonableness of the algorithm output. When possible, results were compared against PROFS' corresponding verification data base of meteorological events. As data were reviewed, particularly interesting cases were noted which later formed the basis of a demonstration prepared for use during the evaluation activity.

7.3 Product Generation Statistics

During the product generation phase, we maintained computer processing logs which provide statistics of resources required to process the 300 volume scans of data through the algorithms. Besides providing overall timing figures, the statistics also gave gross time requirements for each algorithm. Although the software was written for readability and modularity rather than speed and efficiency, one could use these figures for future algorithm timing questions.

Table 11. Algorithm Processing Statistics

Algorithm	Time/Volume (min)
Segments	4
Centroids, Tracking, Forecast Structure, Hail	1
VAD (6 ranges)	6
Mesocyclone Detection	7-9
Total time for one volume	20 minutes

7.3.1 Algorithm Processing Statistics

Overall averages of the processing times are given in Table 11. Note that the three algorithms which access the Doppler data (Storm Segments, VAD, and Mesocyclone Detection) use most of the processing time. Times given are clock times of a dedicated VAX-11/750 whose configuration is shown in Figure 20. We processed 300 volume scans in all, at 20 minutes per scan, for a total processing time of 100 hours. There are also 1200 special image files used in the evaluation activity, which required 90 hours for processing. Total processing time for all algorithms and special images was 190 hours.

7.3.2 Algorithm Timing Statistics

Whereas Table 11 gives overall timing averages, Table 12 gives detailed statistics as a function of the type of volume scan. The numbers in the column titled "Direct I/O" are proportional to the number of disk accesses required by each algorithm. Pertinent volume scan information is included for comparison purposes.

7.4 On-Site Evaluation Activity

7.4.1 Purpose

One of the two deliverables to NEXRAD was an on-site evaluation of the results of coding and exercising the eight algorithms. The purpose was to acquaint various government agencies and members of private industry with the results of this effort and to allow them the opportunity to review the scientific validity and operational usefulness of the algorithms.

Table 12. Algorithm Timing Statistics for CP-2 TSC and CP-2 PROFS Data

Algorithm	CPU Time		Clock Time		Direct I/O	
	TSC	PROFS	TSC	PROFS	TSC	PROFS
Segments	9.5 min	4.8 min	10.3 min	5.0 min	2343	1097
Centroids	7.4 s	6.0 s	9.4 s	7.5 s	24	18
Tracking	1.3 s	1.2 s	3.0 s	3.0 s	7	7
Forecast	1.1 s	1.1 s	2.5 s	2.5 s	5	6
Structure	1.4 s	1.2 s	2.9 s	2.7 s	6	5
Hail	1.4 s	1.3 s	3.0 s	2.9 s	7	7
VAD	6.0 min	4.3 min	9.0 min	5.7 min	9139	5122
Mesocyclone	11.0 min	6.0 min	11.5 min	6.3 min	1509	843

CP-2 TSC volume information:	CP-2 PROFS volume information:
July 28, 1982 (Day 209)	August 10, 1982 (Day 222)
Sweeps/Volume = 11	Sweeps/Volume = 9
Radials/Sweep = 360	Radials/Sweep = 360
Gates/Radial = 768	Gates/Radial = 512
Mbytes/Volume = 12.2	Mbytes/Volume = 6.5

7.4.2 Scope

The evaluation phase comprised five independent sessions, each lasting 1½ days. Each session included both oral presentations and computer sessions. The oral presentations covered an overview and status of the project, a summary of the Doppler radar data used to exercise the algorithms, a description of each algorithm, and a synopsis of the software system used for coding and exercising the algorithms. The computer sessions involved a variety of displays of both the Doppler radar data and results of the eight algorithms. We presented scenarios showing the algorithm results with data from many meteorological conditions.

7.4.3 Participants

Participants for the on-site evaluation were members of the NEXRAD Joint System Program Office (JSPO) and Technical Advisory Committee (TAC); representatives from the NEXRAD Interim Operational Test Facility (IOTF) at Norman, Oklahoma; staff of the Systems and Applied Sciences Corporation (SASC); the source scientists for the eight algorithms; representatives from the two system contractors, Raytheon and Sperry Corporations; and representatives from the three principal NEXRAD users, the National Weather Service (NWS), Air Weather Service (AWS), and the Federal Aviation Administration (FAA).

8. CONCLUDING REMARKS

One goal of the NEXRAD JSPO was to evaluate the algorithm description methodology. These descriptions must adequately define and specify the functional procedures for processing Doppler radar data. We were able to transform the NEXRAD algorithm description statements into executable computer code, apply that code to Doppler data, and obtain meteorological results consistent with available verification data. This exercise demonstrated that the algorithm descriptions can be used to communicate software functional procedures. Although some descriptions lacked the completeness needed for actual coding, necessary enhancements were minor.

We processed approximately 300 volume scans of data, taken during a variety of meteorological conditions, through the eight algorithms and reviewed the output. This exercise showed that the algorithms produce useful results. However, Table 11, Algorithm Processing Statistics, shows that the current software configuration, as we implemented it, is not suitable for real-time processing. Nevertheless, these results are extremely useful since they provide NEXRAD with well-documented baseline performance characteristics.

During the first phase of this work, PROFS has developed a solid foundation for a system which processes and displays Doppler radar data. Our system was carefully designed to allow growth in two areas: data sources used and meteorological algorithms implemented. NEXRAD has received a set of well-documented and controlled software modules implementing eight algorithms. The experience gained by PROFS will be extremely useful to NEXRAD's continuing need to assess, test, and upgrade current and future algorithms.

9. ACKNOWLEDGMENTS

David E. Small, formerly of PROFS and now with the Wave Propagation Laboratory (ERL, NOAA), provided very admirable technical leadership for the NEXRAD/PROFS task team under the supervision of Ron Alberty, Chief, Advanced Data Systems. Raja Tallamraju, also a former member of the NEXRAD/PROFS task team, coded the VAD algorithm and assisted with the display software.

The PROFS/NEXRAD team is extremely grateful to the National Center for Atmospheric Research (NCAR) for support in various areas of the project. Specifically we thank Richard E. Carbone, Manager of the Field Observing Facility (FOF), for providing us with the CP-2 Doppler radar data tapes; Robert J. Serafin, Director of the Atmospheric Technology Division, who guided and supported the FOF's participation; John McCarthy, Director of the NCAR/JAWS project, who provided support and cooperation during the 1982 summer data collection period; and James W. Wilson, scientist with FOF, for his expert technical assistance.

We would also like to thank Grant Gray, a principal member of the team that designed the CP-2 radar pulse-pair processor, for his help in providing details of the radar hardware.

Significant support for this project was provided by the NEXRAD Joint System Program Office, according to the terms and conditions of Contract Number I83-RZ01, dated November 19, 1982.

10. REFERENCES

Alberty, R. L., 1982: NEXRAD Doppler radar and the role of data integration in PROFS. Proc. NEXRAD Doppler Radar Symposium/Workshop, CIMMS, Norman, OK, 216-220.

Barnes, S. L., 1980: Report on a meeting to establish a common Doppler radar data exchange format. Bull. Am. Meteorol. Soc., 61, 1401-1404.

Beran, D. W., and A. E. MacDonald, 1981: Design of a short-range forecasting system. Proc. IAMAP Symposium, Hamburg, 25-28 August, 347-350.

Bjerkaas, C. L., and D. E. Forsyth, 1980: An automated real-time storm analysis and storm tracking program (WEATRK). Air Force Geophysics Laboratory Tech. Report 80-0316, Hanscom AFB, MA.

Boak, T. I. S., III, A. J. Jagodnik, Jr., R. B. Marshall, D. Riceman, M. J. Young, 1977: R and D equipment report: tracking and significance estimator. Air Force Geophysics Laboratory Tech. Report 77-0259, Hanscom AFB, MA.

Browning, K. A., and R. Wexler, 1968: A determination of kinematic properties of a wind field using Doppler radar. J. Appl. Meteorol., 7, 105-113.

Burgess, D., R. J. Donaldson, T. Sieland, and J. Hinkelman, Jr., 1979: Final report on the joint Doppler operational project (JDOP), 1976-1978. Part I, NOAA Tech. Memo. NSSL-86, Norman, OK.

Carbone, R. E., R. I. Serafin, C. Frush, V. N. Bringi, and T. Seliga, 1982: CP-2 radar developments. Unpublished preprint, NCAR, Boulder, CO.

DEC, 1980: VAX technical summary. Digital Equipment Corporation, Maynard, MA.

DeMarco, T., 1978: Structured analysis and system specification. Yourdon Inc., New York, NY.

Forsyth, D. E., C. L. Bjerkaas, and P. J. Petrocchi, 1981: Modular radar analysis software system (MRASS). Preprints Twentieth Conference on Radar Meteorology, Am. Meteorol. Soc., Boston, MA, 696-699.

Gray, G., 1982: CP-2 Doppler radar data characteristics. Internal document, PROFS, NOAA, Boulder, CO.

Hennington, L. D., and D. W. Burgess, 1981: Automatic recognition of mesocyclones from single Doppler radar data. Preprints Twentieth Conference on Radar Meteorology, Am. Meteorol. Soc., Boston, MA, 704-706.

Lemon, L. R., 1980: Severe thunderstorm radar identification techniques and warning criteria. NOAA Tech. Memo. NWS NSSFC-3, Kansas City, MO.

Lemon, L. R., 1978: On the use of storm structure for hail identification. Preprints Eighteenth Conference on Radar Meteorology, Am. Meteorol. Soc., Boston, MA, 203-206.

Lhermitte, R. M., and D. Atlas, 1961: Precipitation motion by pulse Doppler. Proc. Ninth Weather Radar Conference, Am. Meteorol. Soc., Boston, MA, 218-223.

McCarthy, J., J. W. Wilson, and T. T. Fujita, 1982: The joint airport weather studies project. Bull. Am. Meteorol. Soc., 63, 15-22.

NCAR, 1982: Field observing facility radar description document, in Appendix E, JAWS Field Operations Manual, NCAR, Boulder, CO.

NCAR, 1981: CP-2 tape format, internal document, NCAR, Boulder, CO.

NEXRAD, 1983a: Technical requirements, NEXRAD Joint System Program Office, Washington, DC.

NEXRAD, 1983b: Algorithm documentation standards handbook. NEXRAD Joint System Program Office, Washington, DC.

NEXRAD, 1983c: Algorithm report. NEXRAD Joint System Program Office, Washington, DC.

NEXRAD, 1983d: Research and development plan. NEXRAD Joint System Program Office, Washington, DC.

Petrocchi, P. J., 1982: Automatic detection of hail by radar. Air Force Geophysics Laboratory Tech. Report 82-0277, Hanscom AFB, MA.

PROFS, 1980: PROFS proposed policy, mission, goals, and organizational structure. NOAA Tech. Memo. ERL PROFS-2, Boulder, CO.

Rabin, R., 1982: Vertical velocity calculation from VAD. Memo for the record, NSSL, Norman, OK.

Rabin, R., and D. Zrnic, 1980: Subsynoptic-scale vertical wind revealed by dual Doppler-radar and VAD analysis. J. Atmos. Sci., 37, 644-654.

Ramtek, 1979: RM-9050 Series graphic display system: system description manual. Pub. No. 503744, Revision C, Ramtek Corp., Santa Clara, CA.

Ramtek, 1978: 9050 Series graphic display system: programming manual. Part No. 503746, Ramtek Corp., Sunnyvale, CA.

Reynolds, D. W., 1983: Prototype workstation for mesoscale forecasting. Bull. Am. Meteorol. Soc., 64, 264-273.

Roberts, W. F., R. L. Alberty, and D. E. Small, 1983: Automated single-Doppler data processing: a systematic approach. Preprints Twenty-First Conference on Radar Meteorology, Am. Meteorol. Soc., Boston, MA, 727-732.

APPENDIX A. ALGORITHM DESCRIPTION EXAMPLE - STORM CENTROIDS

NX-DR-03-005/19
12/27/82

1.0 PROLOGUE

1.1 FUNCTIONAL DESCRIPTION

The purpose of this algorithm is to determine the centroids of storms. The continuous significant reflectivity ($\geq$ 30 dBZ) values collected along the radials (identified as Segments) of a volume scan are used to determine the location of centroids or centers of mass of storms. The determination of the centroid is carried out in two steps. In step one, regions of significant reflectivity at known elevation angles are determined by finding contiguous segments as a function of azimuths. These regions are defined to be Storm Components. The azimuthal spacing between radials is expected to be approximately equal to the radar beamwidth so that no gaps exist between segments. A 2D center of mass is computed for each identified Storm Component in a rectangular coordinate system with the radar at the origin; where the X-axis denotes East-West directions and the Y-axis denotes North-South directions. In step two, storm components at different elevation angles are correlated vertically along the Z-axis denoting heights above the radar at corresponding elevations. Vertical correlation means finding those storm components at different elevation angles whose centers of mass are closest in proximity with respect to the X-Y plane. The vertically correlated storm components constitute a single storm and the effective center of mass of the storm is computed. This 3D center of mass is projected onto the X-Y plane denoting the center of mass or centroid of a storm.

1.2 SOURCE

This algorithm has been implemented at the Air Force Geophysics Laboratory (AFGL), Sudbury, Massachusetts. Originally developed by Raytheon Corporation

(AFGL-TR-77-0259) as a 2D system, D. Forsyth and C. Bjerkaas modified it to a 3D system at AFGL, Sudbury.

REFERENCES

Bjerkaas, C.L., and D.E. Forsyth, 1980: An Automated Real-Time Storm Analysis and Storm Tracking Program (WEATRK). AFGL-TR-80-0316, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts 01731.

Boak, III, T.I.S., A.J. Jagodnik, Jr., R.B. Marshall, D. Riceman, M.J. Young, 1977: R & D Equipment Report - Tracking and Significance Estimator. AFGL-TR-77-0259, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts 01731.

1.3 PROCESSING ENVIRONMENT

The STORM CENTROIDS algorithm is a precursor to the STORM TRACKING algorithm (NX-DR-03-004) and the data elements included in this description are only those relevant to the storm tracking. The algorithm requires outputs from the STORM SEGMENTS algorithm (NX-DR-05-007).

2.0 INPUTS

2.1 IDENTIFICATION

AZIMUTHS = Azimuth angles at which radial data are collected.

DISTANCE = Maximum horizontal distance between centroids at different elevation angles of the same storm.

MASS WEIGHTED LENGTH = The sum over a SEGMENT of the mass of a resolution volume times the radial distance to the center of the resolution volume.

MASS WEIGHTED LENGTH SQUARED = The sum over a SEGMENT of the mass of a resolution volume times the squared radial distance to the center of the resolution.

ELEVATIONS = Elevation angles at which radial data are collected.

OVERLAP = Minimum number of resolution volumes that must overlap in range for adjacent azimuths to signify the same storm component. The AFGL implementation used a value of five resolution volumes for OVERLAP.

RESOLUTION = Minimum number of resolution volumes to define a segment. AFGL used a value of 14, derived as a function of 300m length of each resolution volume, i.e., the minimum length of a segment is 4200m.

SEGMENTS = Sets of numbered continuous resolution volumes along a radial identified by reflectivity factor above a specified threshold. AFGL used a value of 30 dBZ to define these SEGMENTS.

2.2 ACQUISITION

AZIMUTHS and ELEVATIONS are acquired as measured data from the radar's instantaneous position during radial data collection.

DISTANCE, RESOLUTION, and OVERLAP are supplied as parameters whose values are based on empirical and theoretical studies of storms.

MASS WEIGHTED LENGTH and MASS WEIGHTED LENGTH SQUARED are acquired from the STORM SEGMENTS algorithm (NX-DR-05-008). REFLECTIVITY FACTOR is acquired directly from the Doppler radar hardware.

SEGMENTS are the outputs produced by the Tracking and Significance Estimator (Boak, III *et al*., 1977).

3.0 PROCEDURE

3.1 ALGORITHM

```
BEGIN ALGORITHM (STORM CENTROIDS)
1.0 DO FOR ALL (ELEVATIONS)
    1.1 DO FOR ALL (AZIMUTHS)
        1.1.1 DO FOR ALL (SEGMENTS)
              1.1.1.1 Look for overlapping resolution volumes among a pair of contiguous SEGMENTS indicated by identical resolution volume numbers.
              1.1.1.2 IF (Number of overlapping resolution volumes between SEGMENTS is greater than or equal to OVERLAP).
                      THEN (Identify SEGMENTS to be part of the same STORM COMPONENT).
                      END IF
              END DO
        END DO
    1.2 Identify all possible STORM COMPONENTS at one ELEVATION.
    1.3 DO FOR ALL (STORM COMPONENTs)
        1.3.1 Count the number of AZIMUTHS in current STORM COMPONENT
        1.3.2 COMPUTE (ADJACENCY)
        1.3.3 IF (number of AZIMUTHS is equal to or greater than ADJACENCY)
              THEN COMPUTE (MASS)
                   COMPUTE (X-POSITION)
                   COMPUTE (Y-POSITION)
              END IF
        END DO
END DO
```

```
2.0  DO FOR ALL (ELEVATIONS)
     2.1  DO FOR ALL (STORM COMPONENTs)
          2.1.1  COMPUTE (X-DIFFERENCE)
          2.1.2  COMPUTE (Y-DIFFERENCE)
          2.1.3  IF (X-DIFFERENCE AND Y-DIFFERENCE are less
                     than DISTANCE)
                 THEN (Identify as UNORDERED STORM)
                 ELSE COMPUTE (X2-DIFFERENCE)
                      COMPUTE (Y2-DIFFERENCE)
                      IF (X2-DIFFERENCE AND Y2-DIFFERENCE
                          are less than DISTANCE)
                          THEN (Identify as UNORDERED
                                STORM)
                      END IF
                 END IF
          END DO
     END DO
3.0  DO FOR ALL (UNORDERED STORMs)
     3.1  COMPUTE (VMASS)
     3.2  COMPUTE (X-CENTROID)
     3.3  COMPUTE (Y-CENTROID)
     END DO
4.0  DO FOR ALL (UNORDERED STORMs)
     4.1  Select the next storm with greatest VMASS
     4.2  WRITE (STORMS)
     4.3  WRITE (X-CENTROID)
     4.4  WRITE (Y-CENTROID)
     END DO
END ALGORITHM (STORM CENTROIDS)
```

3.2 COMPUTATION

3.2.1 NOTATION

A = ADJACENCY, minimum number of adjacent azimuths to be defined a storm component.

RHO = RESOLUTION, in resolution volumes.

r = Resolution volume number ($1 \leq r \leq 1024$, for AFGL implementation).

M = MASS, mass weighted area of a storm component in kton/m. M_i indicates one such quantity.

θ_e, θ_s = Azimuth numbers between one and the maximum azimuth where s indexes the first (starting) segment in a STORM COMPONENT and e indexes the last (ending) segment in a STORM COMPONENT.

MWL = MASS WEIGHTED LENGTH of a SEGMENT.

dr = Length of a resolution volume in meters.

$d\theta$ = Width of a segment (beam width) or azimuthal increment in degrees where it is assumed that the azimuthal step is approximately equal to the beamwidth.

$\bar{X}$ = X-POSITION of the center of mass of a storm component in kilometers. $\bar{X}_i$ indicates $\bar{X}$ at the i^{th} elevation.

MWS = MASS WEIGHTED LENGTH SQUARED of a SEGMENT.

Θ = Azimuthal position of the STORM COMPONENT center of mass in degrees.

ϕ = One of the ELEVATIONS in degrees.

$\bar{Y}$ = Y-POSITION of the center of mass of a storm component in kilometers. $\bar{Y}_i$ indicates $\bar{Y}$ at the i^{th} elevation.

XD = X-DIFFERENCE in kilometers.

$|Q|$ = Absolute value of a quantity Q.

X_{ni} = X-POSITION of the center of mass of n^{th} storm component at i^{th} elevation in kilometers.

n,j = index of the n^{th} or j^{th} storm component. n and j vary from zero to the maximum number of storm components per i^{th} elevation.

YD = Y-DIFFERENCE in kilometers.

Y_{ni} = Y-POSITION of the center of mass of n^{th} storm component at i^{th} elevation in kilometers.

X2 = X2-DIFFERENCE in kilometers.

Y2 = Y2-DIFFERENCE in kilometers.

VOL = VMASS, Mass weighted volume of a storm in ktons.

dH = Height difference between centers of mass of storm components at two successive elevations in kilometers.

dH_i indicates one such height difference.

X_c = X-CENTROID in kilometers.

Y_c = Y-CENTROID in kilometers.

Note: This algorithm was implemented using 32 bit precision at AFGL.

3.2.2 SYMBOLIC FORMULAS

COMPUTE (ADJACENCY)

A = Maximum of $[(RHO/r)/d\theta]$ or 2.1

COMPUTE (MASS)

$$M = \sum_{i=\theta_s}^{\theta_e} (MWL)_i (dr)(d\theta)$$

COMPUTE (X-POSITION)

$$\bar{x} = 1/M \sum_{k=\Theta_s}^{\Theta_e} (\mathrm{MWS})_k \ \sin\Theta \ dr \ d\Theta \ \cos\phi$$

COMPUTE (Y-POSITION)

$$\bar{X} = 1/M \sum_{k=\Theta_s}^{\Theta_e} (\mathrm{MWS})_k \ \cos\Theta \ dr \ d\Theta \ \cos\phi$$

COMPUTE (X-DIFFERENCE)

$$XD = | X_{ni} - X_{j(i-1)} |$$

COMPUTE (Y-DIFFERENCE)

$$YD = | Y_{ni} - Y_{j(i-1)} |$$

COMPUTE (X2-DIFFERENCE)

$$X2 = | X_{ni} - X_{j(i-2)} |$$

COMPUTE (Y2-DIFFERENCE)

$$Y2 = | Y_{ni} - Y_{j(i-2)} |$$

COMPUTE (VMASS)

$$VOL = \sum_{i} M_i \, (dH_i) \quad \text{where}$$

$$dH_i = (\bar{X}_i^2 + \bar{Y}_i^2)^{0.5} \; (\sin d\phi) \, / \, \cos \phi_i$$

$$\text{where } d\phi = (\phi_{i+1} - \phi_{i-1}) \, / \, 2$$

COMPUTE (X-CENTROID)

$$X_c = 1/VOL \sum_{i} (\bar{X}_i)(M_i)(dH_i)$$

COMPUTE (Y-CENTROID)

$$Y_c = 1/VOL \sum_{i} (\bar{Y}_i)(M_i)(dH_i)$$

4.0 OUTPUTS

4.1 IDENTIFICATION

X-CENTROID and Y-CENTROID are locations of mass weighted volumetric centers of mass of identified storms.

STORMS is a set of identifiers for storms detected, ranked in the order of their mass weighted volumes.

4.2 DISTRIBUTION

The outputs from this algorithm are extremely useful, therefore they can be expected to be used for many applications. As an example, the location of the storms could be used by precipitation application algorithms to reduce the amount of data required to be processed. At AFGL, an implementation uses outputs from this algorithm to track and forecast locations of storms. The referencing algorithms are described in STORM TRACKING (NX-DR-03-004) and STORM FORECAST (NX-DR-03-008) algorithm descriptions.

5.0 INFERENCES

5.1 LIMITATIONS

The algorithm is designed for isolating storms defined by region of high reflectivity. This algorithm has not been tested extensively in cases of wide spread regions of light precipitation. Its performance under these conditions are unknown.

5.2 FUTURE DEVELOPMENTS

There is no knowledge of future plans to modify the STORM CENTROIDS algorithm.

APPENDIX B. TABLE OF SELECTED VOLUME SCAN TIMES

The calendar date and Julian day head each column of times selected for a particular day. Volume times are in G.m.t. (G.m.t. = m.d.t. + 6 hours).

29 May 1982 (Day #149)		10 June 1982 (Day #161)	30 June 1982 (Day #181)	
2030	2240	1900	0150	0350
2035	2245	1915	0155	0355
2040	2250	1930	0200	
2045	0015	1945	0205	
2100	0020	2000	0210	
2115	0025	2015	0215	
2120	0030	2100	0220	
2125	0035	2115	0225	
2130	0040	2130	0230	
2135	0045	2145	0235	
2140	0050	2200	0240	
2145	0055	2215	0245	
2150	0100	2230	0250	
2155	0105	2245	0255	
2200	0110	2300	0300	
2205	0115	2315	0310	
2210	0120	2330	0315	
2215	0125	2345	0325	
2220	0130	0000	0330	
2225	0135	0030	0335	
2230	0140	0045	0340	
2235	0145	0130	0345	

01 July 1982 (Day #182)		08 July 1982 (Day #189)	09 July 1982 (Day #190)
1815	0030	2150	2120
1830	0100	2205	2125
1845	0115	2210	2130
1915	0130	2215	2135
1920	0145	2220	2140
1930	0200	2230	2150
1945	0215	2235	2155
2015	0230	2245	2200
2030		2255	2205
2340		2300	2210
2345		2305	2215
2350		2310	2220
2355		2315	2225
0000			2230
0005			2245
0015			2300

20 July 1982 (Day #201)

2335
2340
2345
2350
2355
0000
0005
0010
0015
0020
0025
0030
0035

21 July 1982 (Day #202)

2335
2340
2345
2350
2355
0000
0005
0010
0015
0020
0025
0030
0035
0040
0045
0050
0055

25 July 1982 (Day #206)

2130
2145
2200
2215
2230
2245
2300
2315
0015
0030
0045
0100
0115
0130
0145
0200
0215
0230
0245
0300
0315
0330
0345
0400

26 July 1982 (Day #207)

0110
0115
0125
0130
0135
0145
0150
0155
0200
0205

28 July 1982 (Day #209)

2055
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2210
2215
2220
2225

29 July 1982 (Day #210)

1715
1730
1745
1800
1815
1830
1845
1900
1915
1930
1945
2015
2030
2045
2100
2115
2130
2145
2200
2215
2230
2245
2300
2330
2345
0000
0015
0030
0045
0100

09 August 1982 (Day #221)	10 August 1982 (Day #222)
2000	1815
2015	1830
2030	1845
2045	1900
2100	1930
2115	1945
2130	2045
2145	2055
	2100
	2105
	2110
	2115
	2120
	2125
	2130
	2135
	2140
	2145
	2150
	2155
	2200
	2205
	2215
	2230
	2300
	2315
	2350

APPENDIX C. VERIFICATION DATA CORRESPONDING TO SELECTED VOLUME SCAN TIMES

29 May 1982

a.	2036Z	1/2" or more size hail on ground, 1" deep near Prospect Valley.
b.	2050Z	1/2"-3/4" size hail on ground, 2" deep on 10-mile stretch of Highway 52 by Prospect Valley, some crop damage.
c.	2050Z	One killed and two injured by lightning in Washington Park in Denver.
d.	2100Z	Up to 1" rain in south Lincoln County with hail.
e.	2145Z	3/4" slushy hail 7 miles SW of Fort Morgan.
f.	2200Z	40-50 mi/h winds judging from the trees as a storm passed east of Prospect Valley.
g.	2215Z	Marble-size hail 10 miles south of Fort Morgan.
h.	2215Z	Crop damage 5-12 miles south of Fort Morgan from up to 3" of rain.
i.	2228Z	Small hail and lightning in south-central Arapahoe County.
j.	2230Z	Lightning struck a tree 2 miles south of Elizabeth, no damage, 1/2" rain.
k.	0026Z	Much hail on road south of Akron.
l.	0030Z	Pea-size hail in Longmont.
m.	0034Z	1/2" of hail north of Longmont.
n.	0044Z	Funnel went over house 1/2 mile east of Highway 71 and 2 miles north of Limon, no touch down.
o.	0044Z	Walnut-size hail covering ground 1/2 mile east of Highway 71 and 2 miles north of Longmont.
p.	0045Z	Walnut-size hail 2 miles north of Limon.
q.	0113Z	Marble-size hail 4 miles east of Windsor.

10 June 1982 No reports.

30 June 1982

a.	0200Z	1/2" size hail 3 miles NE of Golden.
b.	0200Z	Small hail 4 miles NE of Ault.

01 July 1982

a.	2038Z	Confirmed report of funnel cloud 4 miles north of Castle Rock.
b.	2039Z	Four reports of funnel clouds by County Road and I-25.
c.	2341Z	1/2" hail at 20th and Sheridan.
d.	2345Z	1-1/2" size hail at 14th and Carr.
e.	2345Z	1" hail in Wheatridge.
f.	2347Z	1/2"+ size hail at 26th and Jay in Edgewater.
g.	2359Z	Pea-size hail at 3rd and Milwaukee.
h.	0200Z	Marble-size hail in the Arvada area, damaged trees.
i.	0200Z	1/2"-1" hail in Golden.
j.	0210Z	1/2"-3/4" hail at 60th and Ward.
k.	0214Z	1" hail in Arvada at 64th and Ward.
l.	0215Z	Funnel cloud at 120th and Colorado Boulevard.

08 July 1982 No Reports.

09 July 1982

a. 2121Z Funnel clouds and wall cloud in Briggsdale area.
b. 2122Z Funnel clouds east of Colorado State University, Fort Collins.

20 July 1982 No Reports.

21 July 1982 No Reports.

25 July 1982

a. 2222Z Funnel cloud north of Bennett.

26 July 1982

a. 0112Z Street flooding in Greeley.
b. 0130Z Pea-size hail SSW of Carr.
c. 0132Z Tornado 15-20 miles north of Eaton.
d. 0140Z Funnel cloud 2 miles east of Ault.

28 July 1982

a. 2055Z Funnel dipped down near the Arapahoe County Airport briefly.
b. 2217Z 1.46" of rain in two hours, 10 miles west of Castle Rock.
c. 2229Z Possible funnel cloud SW of Denver on Wadsworth.
d. 2237Z Funnel cloud due east of Morrison.

29 July 1982 Used primarily for VAD algorithm.

a. 1800Z Denver Supplemental Rawinsonde Sounding.
b. 2100Z Denver Supplemental Rawinsonde Sounding.
c. 0000Z Denver Rawinsonde Sounding.

09 August 1982

a. 2115Z Marble-size hail at the corner of Morgan and Weld Counties.
b. 2130Z Unconfirmed funnel cloud near Wiggins.
c. 2145Z A tornado or funnel cloud reported by public 15-20 miles NE of Prospect Valley.

10 August 1982

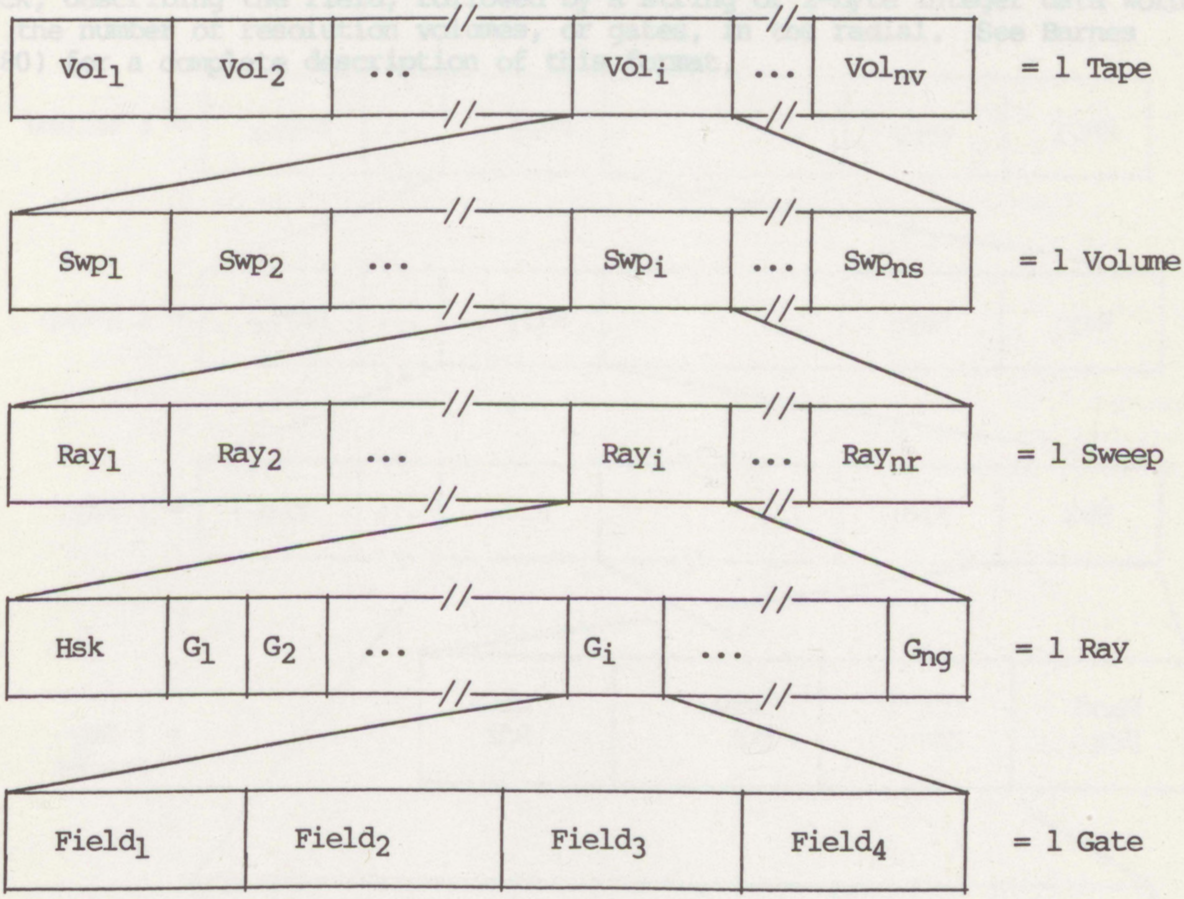
a. 1913Z Tornado 10 miles SE of Castle Rock on ground, moving north.
b. 1914Z Tornado north of Colorado Springs.
c. 1915Z Public report of funnel cloud in Elbert County.
d. 1923Z Funnel cloud south of the Arapahoe Airport.
e. 1930Z Four funnels in the Larkspur area.
f. 1930Z Funnel cloud reported over Parker.
g. 2132Z Marble-size hail covering the ground east of Castle Rock.
h. 2140Z Pea-size hail and heavy rain in Castle Rock.
i. 2230Z Small hail, over 1" of rain, and winds 16 miles west of Agate.
j. 2233Z Funnel cloud NW of Limon.
k. 2242Z Rotating wall cloud and a possible funnel 10 miles NW of Limon.

APPENDIX D. NCAR DOPPLER FIELD TAPE FORMAT

Data from the NCAR CP-2 radar are stored on tape in the NCAR Doppler Field Tape Format (Fig. 24). Each tape contains contiguous sets of data records for a number of volume scans, where each volume scan is identified by a volume sequence number. The volumes are subdivided into sweeps, identified by sweep sequence numbers. Each sweep then contains data for a number of radials (rays). Next each radial is represented by a housekeeping section followed by the data values for a number of resolution volumes (gates). The housekeeping section contains 256 2-byte integer words and provides such information as the volume and sweep sequence numbers, azimuth and elevation angles, and observation time, as well as various other radar characteristics. The details of the housekeeping section are listed in an internal document by NCAR (1981). Finally, for each gate, four 1-byte words contain the data for the four measured parameters: log channel received power, linear channel received power (R(0)), Doppler velocity, and normalized coherent power (R(tau)).

Features

- 1 file per tape.
- 256 housekeeping words per ray.
- 4 fields per gate.
- 8 bits per field for each gate.



Field 1 = Signal power counts
2 = R(tau)
3 = Velocity Counts
4 = R(0)

Figure 24. Block diagram of the NCAR Doppler field tape format.

APPENDIX E. COMMON DOPPLER EXCHANGE TAPE FORMAT

The Common Doppler Exchange Tape Format, also known as the Universal Tape Format or UTF, was developed by the Doppler radar community as a means for sharing data among installations. A schematic diagram of this format is given in Figure 25. Briefly, each volume scan, composed of a number of sweeps, is contained in a single tape file. Each sweep contains a number of radials, or rays, while a radial consists of a housekeeping section followed by data blocks for several parameter fields. The radial housekeeping section, which provides information about the radial and the radar, is further subdivided into a mandatory header block, an optional header block, a local header block, and a data header block. The blocks of field data each contain a field header block, describing the field, followed by a string of 2-byte integer data words for the number of resolution volumes, or gates, in the radial. See Barnes (1980) for a complete description of this format.

Features

- 1 file per volume scan.
- Approximately 110 housekeeping words per ray.
- n fields per gate.
- 16 bits per gate for each field.

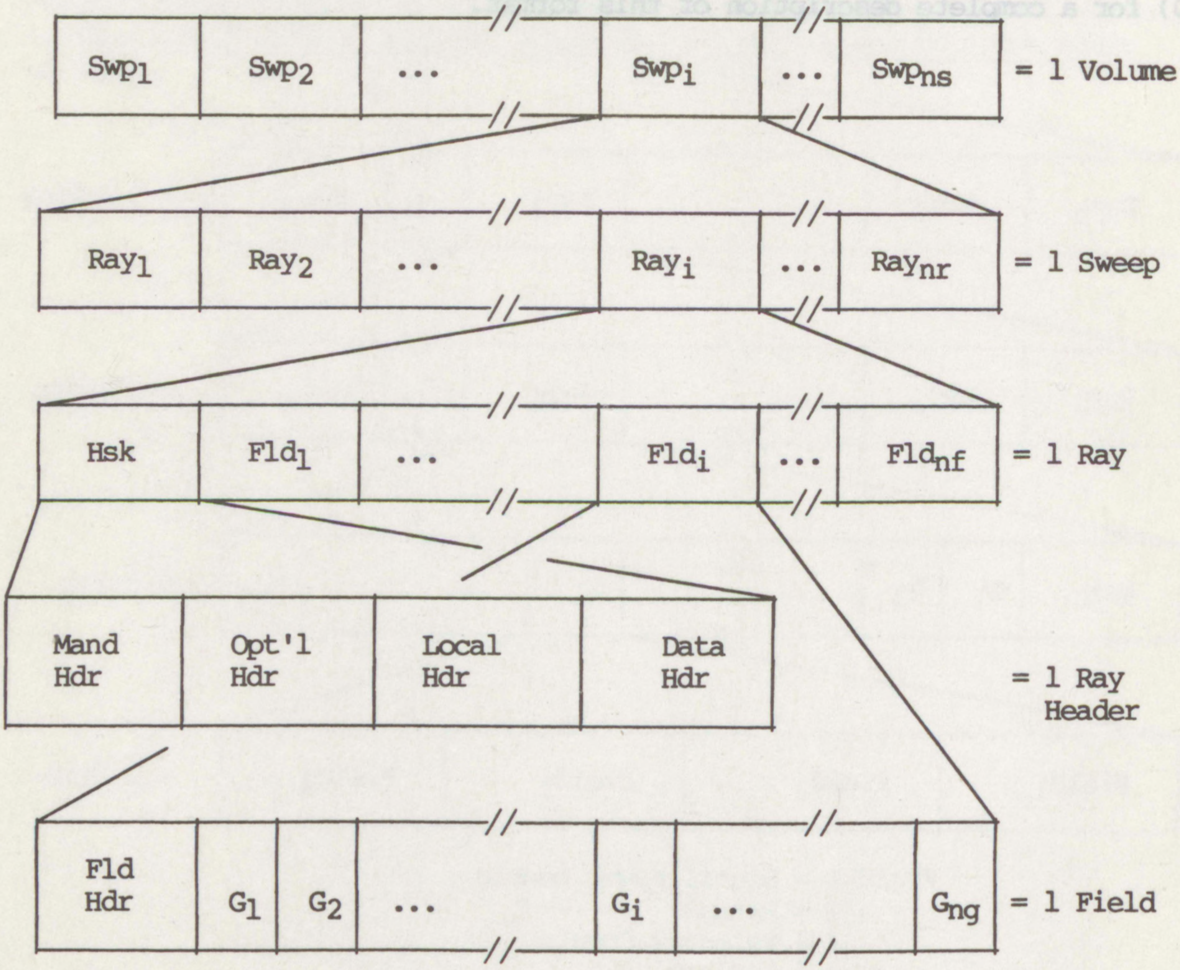


Figure 25. Block diagram of the Common Doppler Exchange Tape Format.

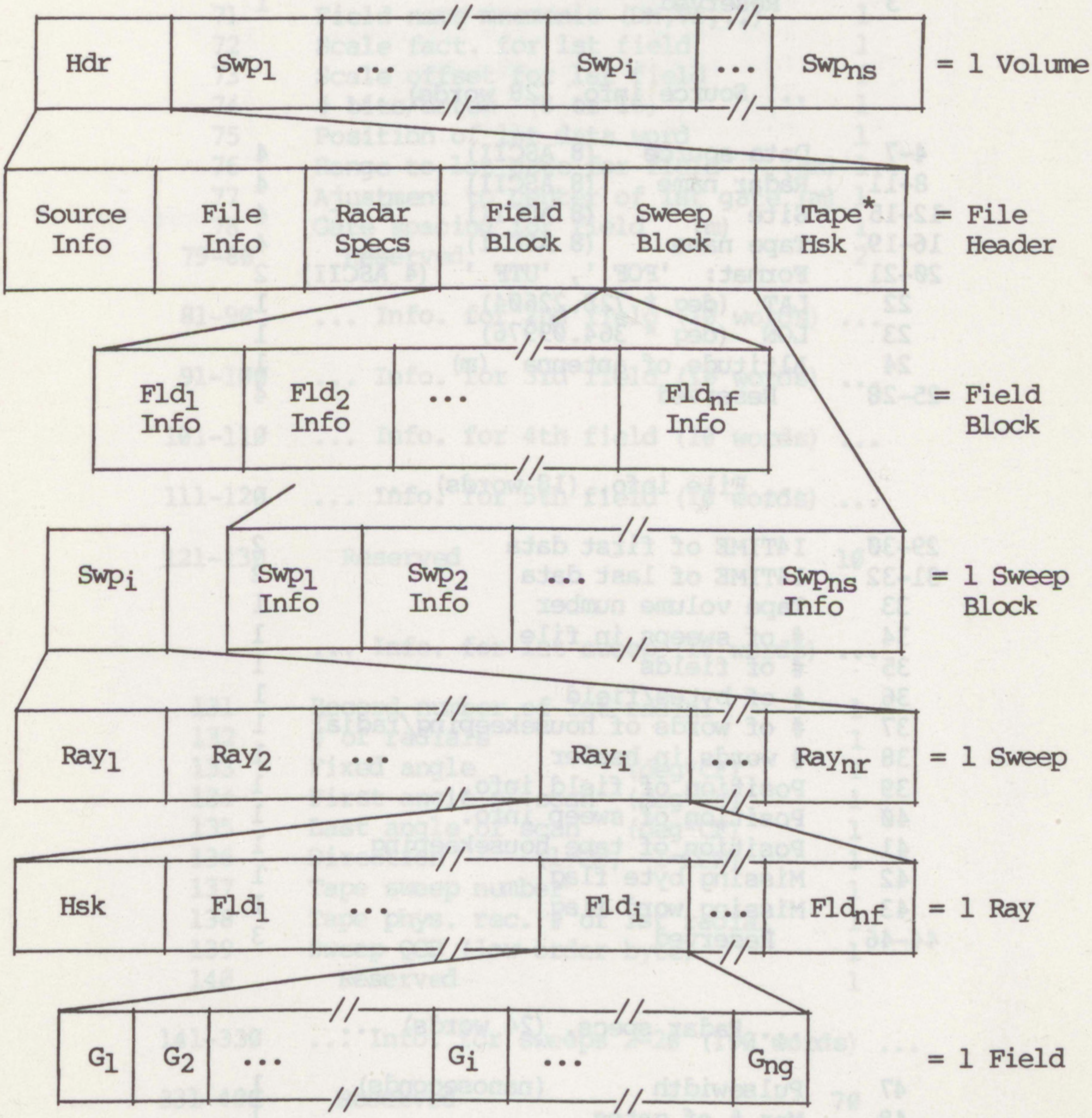
APPENDIX F. PROFS/NEXRAD DISK FILE FORMAT

As discussed in Section 4.1, Doppler radar data stored on tapes in the NCAR and UTF formats are transferred to disk files structured in the PROFS/NEXRAD format for processing by the NEXRAD algorithm software. This format, shown schematically in Figure 26, uses a direct-access file structure to allow selective processing of a volume scan, where a single volume scan is stored in a file. The first record of the file is a header record, detailed in Table 13. This header provides information about the radar and about the various sweeps and parameter fields contained in the file. The header is followed by the data for a number of sweeps, where a sweep is composed of a number of radials. Each radial contains a short housekeeping block, followed by data blocks for the various parameter fields. The housekeeping block contains such information about the radial as the azimuth and elevation angles, and the observation time; the specifics of this block are given in Table 14. Each data block contains values for all the resolution volumes observed for the particular field.

Values specified in the file's header record are assumed to be constants for the volume scan. In particular, the values of such items as the number of fields, the gate spacing for each field, the PRF, and the sweep mode are required to remain constant in a volume scan. As indicated in Table 13, the header is logically divided into blocks which describe the data source, the file characteristics, the radar characteristics, the parameter fields, and the sweeps. In addition, the header record contains a copy of the tape housekeeping section from the first radial of the volume scan to ensure that complete documentation of the radar and the data is available.

Features

- 1 direct-access file per volume scan.
- 1 file header per record.
- 13 housekeeping words per ray.



* Tape housekeeping for first ray of volume.

Figure 26. Block diagram of the PROFS/NEXRAD disk file format.

Table 13. PROFS/NEXRAD Disk File Header Record

2-byte Word	Contents	Number of Words
1	Logical record length	1
2	File format version	1
3	Reserved	1
	... Source info. (28 words) ...	
4-7	Data source (8 ASCII)	4
8-11	Radar name (8 ASCII)	4
12-15	Site (8 ASCII)	4
16-19	Tape name (8 ASCII)	4
20-21	Format: 'FOF ', 'UTF ' (4 ASCII)	2
22	LAT (deg * 728.22604)	1
23	LON (deg * 364.09976)	1
24	Altitude of antenna (m)	1
25-28	Reserved	4
	... File info. (18 words) ...	
29-30	I4TIME of first data	2
31-32	I4TIME of last data	2
33	Tape volume number	1
34	# of sweeps in file	1
35	# of fields	1
36	# of bytes/field	1
37	# of words of housekeeping/radial	1
38	# words in header	1
39	Position of field info.	1
40	Position of sweep info.	1
41	Position of tape housekeeping	1
42	Missing byte flag	1
43	Missing word flag	1
44-46	Reserved	3
	... Radar specs. (24 words) ...	
47	Pulsewidth (nanoseconds)	1
48	Max # of gates	1
49	Range to first gate (km)	1
50	Gate spacing (m)	1
51	PRF (Hz * 10)	1
52	Wavelength (cm * 100)	1
53	Azimuth correction	1
54	Sweep mode	1
55	Radar constant (dBm * 100)	1
56	Nyquist velocity (m/s * 100)	1

Table 13. Continued

57	Transmitted power (dBm * 10)	1
58-70	Reserved	13
	... Info. for 1st field (10 words) ...	
71	Field name mnemonic (DM,VE,..)	1
72	Scale fact. for 1st field	1
73	Scale offset for 1st field	1
74	# bits/datum (8 or 16)	1
75	Position of 1st data word	1
76	Range to 1st gate for field (km)	1
77	Ajustment to center of 1st gate (m)	1
78	Gate spacing for field (m)	1
79-80	Reserved	2
81-90	... Info. for 2nd field (10 words) ...	
91-100	... Info. for 3rd field (10 words) ...	
101-110	... Info. for 4th field (10 words) ...	
111-120	... Info. for 5th field (10 words) ...	
121-130	Reserved	10
	... Info. for 1st sweep (10 words) ...	
131	Record number of 1st radial	1
132	# of radials	1
133	Fixed angle (deg*CF)	1
134	First angle of scan (deg*CF)	1
135	Last angle of scan (deg*CF)	1
136	Direction (1=CW, -1=CCW)	1
137	Tape sweep number	1
138	Tape phys. rec. # of 1st radial	1
139	Sweep QCB (low order byte)	1
140	Reserved	1
141-330	... Info. for sweeps 2-20 (190 words) ...	
331-400	Reserved	70
401 │ 656 (FOF) or ~510 (UTF)	Copy of 1st radial's tape housekeeping info. (FOF=256 words; UTF~110)	

CF = 182.044 (= 2**16/360) for FOF data
= 64 for UTF data

Table 14. PROFS/NEXRAD Radial Housekeeping Block

2-byte Word	Contents	Number of Words
1	Azimuth (deg*CF)	1
2	Elevation (deg*CF)	1
3-4	PROFS' "I4TIME"	2
5	# of gates (< or = max gates)	1
6	Tape physical record number	1
7-8	Field QCB's (1 byte/field)	2
9	Edit code (2 ASCII)	1
10-13	Reserved	4
	Total	13 words

APPENDIX G. SYSTEM DESIGN DOCUMENT EXAMPLES

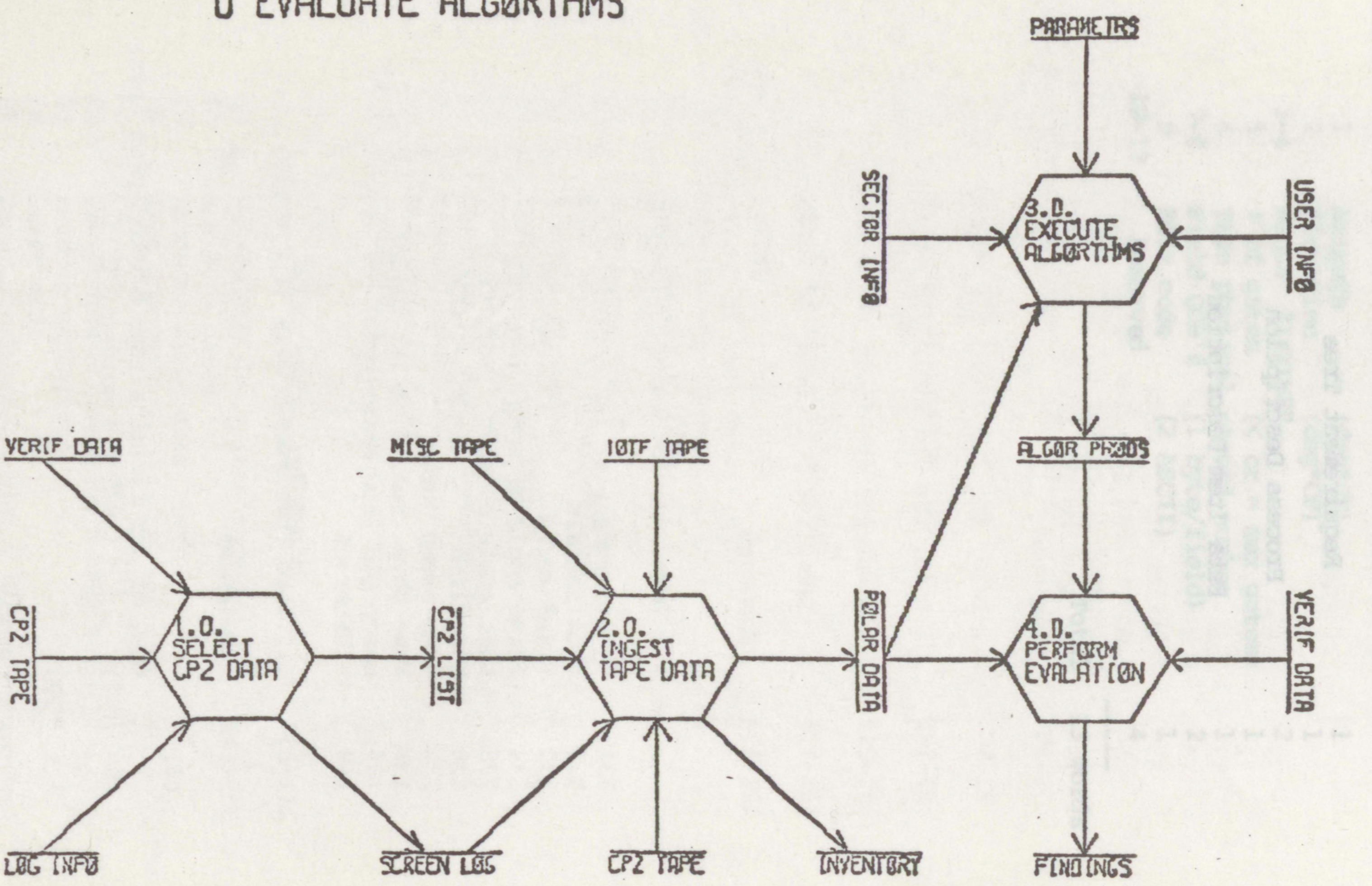
Data Flow Diagram

Requirement Tree

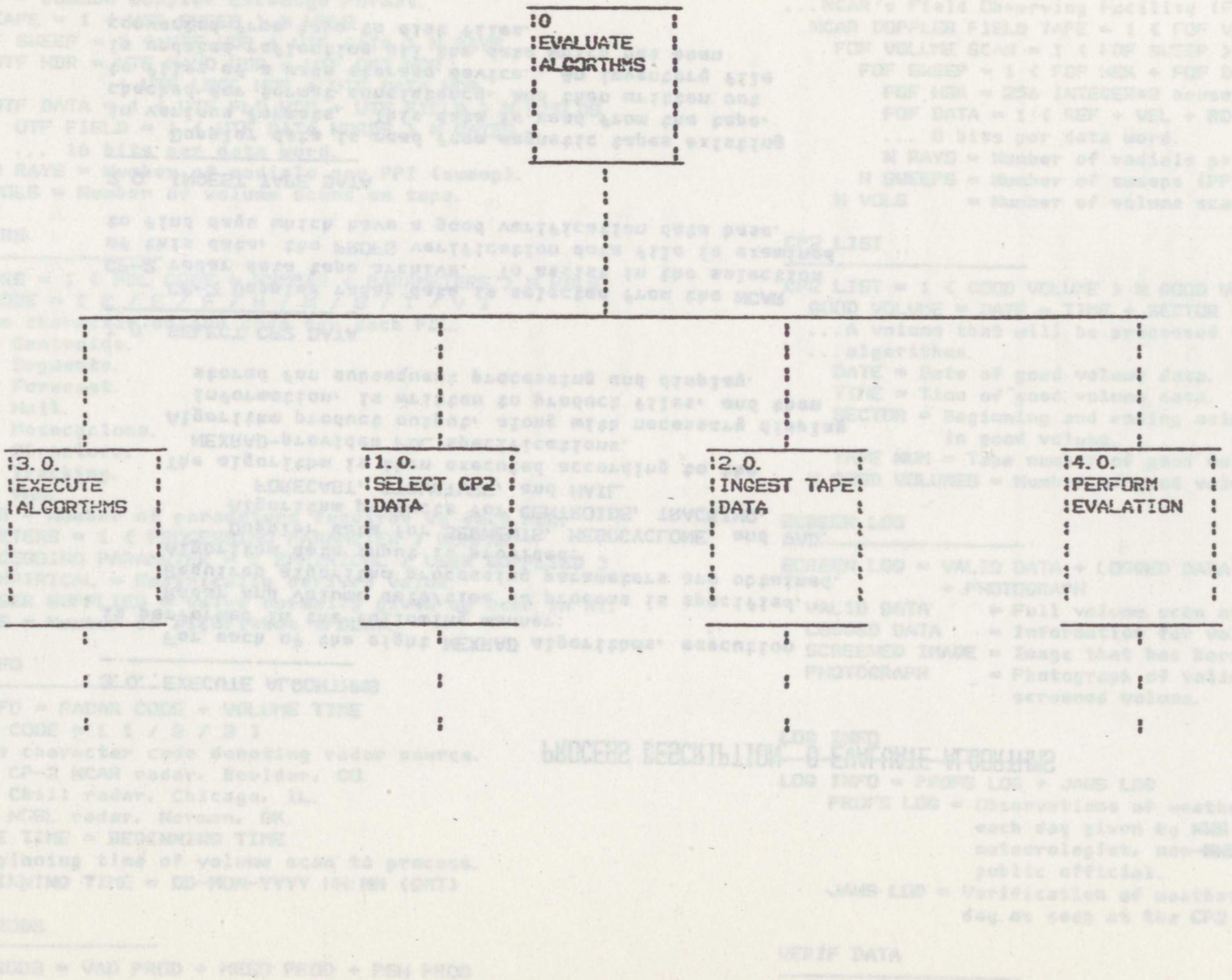
Process Description

Data Item Description

DATA FLOW DIAGRAM



REQUIREMENTS TREE



PROCESS DESCRIPTION: 0 EVALUATE ALGORTHMS

3. 0. , EXECUTE ALGORTHMS

For each of the eight NEXRAD algorithms, execution is performed in the following manner:

Radar and volume date/time to process is specified.
Required algorithm processing parameters are obtained.
Algorithm data input is provided:
  Doppler data for SEGMENTS, MESOCYCLONE, and VAD.
  Algorithm products for CENTROIDS, TRACKING, FORECAST, STRUCTURE, and HAIL.
The algorithm is then executed according to the NEXRAD-provided PDL specifications.
Algorithm product output, along with necessary display information, is written to product files, and then stored for subsequent processing and display.

1. 0. SELECT CP2 DATA

CP-2 Doppler radar data is selected from the NCAR CP-2 radar data tape archive. To assist in the selection of this data, the PROFS verification data file is examined to find days which have a good verification data base.

2. 0. INGEST TAPE DATA

Doppler data is read from magnetic tapes existing in various formats. This data is read from the tape, checked for format consistency, and then written out to files of a mass storage device. An inventory file is updated reflecting all the data which has been converted from tape to disk files.

4. 0. PERFORM EVALATION

Evaluation of the operational and functional meteorological performance of the NEXRAD algorithms consists of viewing the Doppler data and subsequent algorithm products resulting from the processing of the data. The user may select any volume scan contained in the extensive data base, display the Dopper data fields, and then select various types of display methods for any of the algorithm products which have been produced from the data.

DATA ITEM DESCRIPTION: (1 EVALUATE ALGORTHMS

MISC TAPE

```
MISC TAPE = Misc. NCAR field tape or misc. UTF tape.
```

IOTF TAPE

```
IOTF TAPE = UTF TAPE
...Doppler data tape supplied by IOTF.
...UTF = Universal Tape Format.
...CDEF = Common Doppler Exchange Format.
  UTF TAPE = 1 { UTF SWEEP } N VOLS
    UTF SWEEP = 1 { UTF HDR + UTF DATA } N RAYS
      UTF HDR = UTF MAND HDR + UTF OPT HDR
              + UTF LOCL HDR + UTF DATA HDR
      UTF DATA = 1 { UTF FLD HDR + UTF FIELD } N FIELDS
        UTF FIELD = 1 { UTF DATA WORDS } N GATES
        ... 16 bits per data word.
      N RAYS = Number of radials per PPI (sweep).
    N VOLS = Number of volume scans on tape.
```

PARAMETRS

```
PARAMETRS = 1 { PDL CODE + NUMPAR + PARAMETERS } N PDLS
  PDL CODE = [ C / E / F / H / M / S / T / V ]
  ...One character unique code for each PDL.
    C = Centroids.
    E = Segments.
    F = Forecast.
    H = Hail.
    M = Mesocyclone.
    S = Structure.
    T = Tracking.
    V = VAD.
  NUMPAR = Number of parameters required by each PDL.
  PARAMETERS = 1 { PROCESSING PARAMETER } NUMPAR
    PROCESSING PARAMETER = [ EMPIRICAL / USER SUPPLIED ]
      EMPIRICAL = Empirically derived value.
      USER SUPPLIED = Value normally given by user in RT.
  N PDLS = Number of algorithms (PDLs).
```

USER INFO

```
USER INFO = RADAR CODE + VOLUME TIME
  RADAR CODE = [ 1 / 2 / 3 ]
  ...One character code denoting radar source.
    1 = CP-2 NCAR radar, Boulder, CO.
    2 = Chill radar, Chicago, IL.
    3 = NSSL radar, Norman, OK.
  VOLUME TIME = BEGINNING TIME
  ...Beginning time of volume scan to process.
    BEGINNING TIME = DD-MON-YYYY HH:MM (GMT)
```

ALGOR PRODS

```
ALGOR PRODS = VAD PROD + MESO PROD + PSH PROD
...Products from an algorithm sequence.
  VAD PROD = Output files from VAD.
  MESO PROD = Output files from MESO.
  PSH PROD = Output from PSH sequence.
```

CP2 TAPE

```
CP2 TAPE = NCAR DOPPLER FIELD TAPE
...Data from CP-2 Doppler radar operated by
...NCAR's Field Observing Facility (FOF).
  NCAR DOPPLER FIELD TAPE = 1 { FOF VOLUME SCAN } N VOLS
    FOF VOLUME SCAN = 1 { FOF SWEEP } N SWEEPS
      FOF SWEEP = 1 { FOF HSK + FOF DATA } N RAYS
        FOF HSK = 256 INTEGER*2 housekeeping words.
        FOF DATA = 1 { REF + VEL + RO + RTAU } N GATES
        ... 8 bits per data word.
        N RAYS = Number of radials per sweep.
      N SWEEPS = Number of sweeps (PPIs) per volume scan
    N VOLS     = Number of volume scans per tape.
```

CP2 LIST

```
CP2 LIST = 1 { GOOD VOLUME } N GOOD VOLUMES
  GOOD VOLUME = DATE + TIME + SECTOR + TAPE NUM
  ...A volume that will be processed by the
  ...algorithms.
    DATE = Date of good volume data.
    TIME = Time of good volume data.
    SECTOR = Beginning and ending azimuths of good data
             in good volume.
    TAPE NUM = Tape number of good volume.
  N GOOD VOLUMES = Number of good volumes.
```

SCREEN LOG

```
SCREEN LOG = VALID DATA + LOGGED DATA + SCREENED IMAGE
             + PHOTOGRAPH
  VALID DATA       = Full volume scan at PRF = 960.
  LOGGED DATA      = Information for valid volume scan.
  SCREENED IMAGE = Image that has been validated and log
  PHOTOGRAPH       = Photograph of validated, logged and
                     screened volume.
```

LOG INFO

```
LOG INFO = PROFS LOG + JAWS LOG
    PROFS LOG = Observations of weather occurring for
                each day given by NWS spotter,
                meteorologist, non-NWS spotter, and
                public official.
     JAWS LOG = Verification of weather occurring for eac
                day as seen at the CP2 Radar console.
```

VERIF DATA

APPENDIX H. SOFTWARE MODULE CODE EXAMPLE - READPARMS_XA

```
          SUBROUTINE READPARMS_XA(ALGOR_CODE,
          1                  NUMPAR,PARVALS,
          1                  ISTATUS)

    !     Given a PDL algorithm code, ALGOR_CODE, the routine performs the following:
    !             Connects the NEXRAD PDL Algorithm Parameter file,
    !             Reads in the specific parameters for the given PDL,
    !             Returns a list of NUMPAR parameter values, PARVALS; and
    !             Disconnects the file.

          INCLUDE       'SYSDISK:[NEXRAD]NEXRADBOX.FOR/LIST'
  1 !
  1 !     *****************************************************
  1 !     *                                                   *
  1 !     *                     PROFS                         *
  1 !     *            PROFS/NEXRAD '83 PROJECT               *
  1 !     *                                                   *
  1 !     *            ALGORITHM EVALUATION TASK              *
  1 !     *                                                   *
  1 !     *               VAX-11/750 SOFTWARE                 *
  1 !     *                                                   *
  1 !     *****************************************************
  1 !
  1 !     History:
  1 !
  1 !          Name            Date              Description
  1 !     ---------------  -----------  --------------------------------
    !     Ackley, M. H.    14-JAN-1983  Original version.

          INCLUDE         'SYSDISK:[NEXRAD]ERRLIST.FOR/NOLIST'

          CHARACTER       ALGOR_CODE*1

          CHARACTER       DEFNAME*63,PARFILNAM*63,TRASH*1,UPCODE*1

          CHARACTER*1     CODES(25)

          INTEGER*4       NUMPAR,ISTATUS

          INTEGER*4       I,INDEX,LENFIL,LUN,NUMALGTOT,NUMSKIP

          INTEGER*4       NUMALGPAR(25)

          INTEGER*4       SYS$TRNLOG

          LOGICAL*4       LTEST_DIAG_GG

          REAL*4          PARVALS(*)

          DATA            DEFNAME        /'SYSDISK:[NEXRAD.DATA]PARAMETER.DAT'/

    !          *** Get information from system needed for opening parameter file. ***

   1      CONTINUE

    !             Get a system logical unit number that is free.
```

APPENDIX I. SOFTWARE MODULE DOCUMENTATION EXAMPLE - READPARMS_XA

READPARMS_XA reads the NEXRAD PDL (Program Description Language) algorithm parameter file and returns the number of processing parameters and their respective values given a specified PDL algorithm code.

USAGE

```
CALL READPARMS_XA (ALGOR_CODE, NUMPAR, PARVALS, ISTATUS)
```

Arguments:

ALGOR_CODE	Input	CHARACTER*1	PDL algorithm code. This is a unique one-character code assigned to each NEXRAD algorithm known to the system.
NUMPAR	Output	INTEGER*4	Number of processing parameters required by the algorithm associated with the given PDL algorithm code.
PARVALS	Output	REAL*4 Array	An array of length NUMPAR containing the processing parameter values.
ISTATUS	Output	INTEGER*4	Return status is a standard output argument. It contains an integer value that is equal to the value of one of the global variables listed below in RETURN STATUS. For further information on return status codes and how to use them, refer to APPENDIX A, RETURN STATUS CODES.

USER NOTES

1. This routine uses PARNAME as the logical name of the parameter file. The user must equivalence this logical name to the desired actual physical file name by using the DCL DEFINE statement. If the physical file name has not been defined, the routine will access a default file, SYSDISK:[NEXRAD.DATA]PARAMETER.DAT.

2. When called, READPARMS_XA connects the parameter file, reads it, and then disconnects it before returning.
3. If ISTATUS is anything other than SS$_NORMAL upon return, the output variables may be undefined and the parameter file may be left open.

RETURN STATUS

SS$_NORMAL	Normal successful completion.
EDF_DBNOTCNCT	Error in connecting the parameter file.
EDF_FILACCERR	Two error conditions result in this return: 1.) error in reading an individual record of the file, or 2.) error in disconnecting the file.
EDF_IVARG	The subroutine argument, ALGOR_CODE did not correspond to any existing in the parameter file, hence is invalid.
EDF_IVDATA	Two error conditions result in this return: 1.) the number of algorithms specified by the parameter file is not in the range of 1 to 25, or 2.) the number of an algorithm's processing parameters is negative.
EDF_IVFILSTR	An unexpected end-of-file was encountered while reading records of the file.
EDF_NODATA	No processing parameters existed for the given PDL algorithm code.
EDF_SUBERR	An error occurred while using the system services call to obtain or release an available logical unit number.

LOADING INFORMATION

LIBRARY:	[NEXRAD.LIBRARY]NEXALGOR/LIB
SYSTEM:	VAX-11/750
COMPILER:	FORTRAN

AUTHOR: Margot Ackley
READPARMS_XA: Gets NEXRAD algorithm parameter information.

SOFTWARE: 14-JAN-1983
DOCUMENT: 03-FEB-1983

APPENDIX J. SOFTWARE SUPPORT FILE EXAMPLES

Radar File

Algorithm File

Sector File

Parameter File

RADAR FILE

```
NEXRAD/PROFS Radar Information File - RADAR.DAT - 27-FEB-1983
4           !  Number of Radars known to the system.
0,RT-FISHL  !  Radar code for Simulated Data.
1,CP-2           !  Radar code for CP-2 NCAR Radar, Boulder, CO
2,CHILL 1        !  Radar code for CHILL Radar, Chicago, IL.
3,NSSL           !  Radar code for NSSL Radar, Norman, OK.
```

ALGORITHM FILE

```
NEXRAD/PROFS Algorithm Information File - ALGOR.DAT - 27-FEB-1983
8           !  Number of Algorithms known to the system.
V,VAD            !  Velocity-Azimuth Display
C,CENTROIDS      !  Storm Centroids
T,TRACKING       !  Storm Tracking
F,FORECAST       !  Storm Forecast
S,STRUCTURE      !  Storm Structure
H,HAIL           !  Hail
M,MESCYCLON      !  Mesocyclone Detection
E,SEGMENTS       !  Storm Segments
```

SECTOR FILE

```
NEXRAD/PROFS Sector Information File - SECTOR.DAT - 07-MAR-1983
3        !  Number of PDLs needing sector information.
V,M,E           !  Algorithm Codes (VAD,MES,SEG)
2        !  Number of different radars.
1,3,            !  Radar Codes.
1        -      CP2 NCAR Doppler Radar         -       25-JAN-1983
VAD      -         Velocity Azimuth Display       -       07-MAR-1983
-1.0,           !  Beginning azimuth - Met. deg.
-1.0,           !  Ending    azimuth - Met. deg.
10.,            !  Beginning slant range - km.
150.,           !  Ending    slant range - km. (maximum available)
MES      -         Mesocyclone Detection          -       27-JAN-1983
-1.0,           !  Beginning azimuth - Met. deg. (minimum available)
-1.0,           !  Ending    azimuth - Met. deg. (maximum available)
-1.0,           !  Beginning slant range - km. (minimum available)
-1.0,           !  Ending    slant range - km. (maximum available)
SEG      -         Storm Segment                  -       27-JAN-1983
0.0,            !  Beginning azimuth - Met. deg.
180.,           !  Ending    azimuth - Met. deg.
25.0,           !  Beginning slant range - km.
-1.,            !  Ending    slant range - km. (maximum available)
3        -      NSSL Doppler Radar             -       08-APR-1983
VAD      -         Velocity Azimuth Display       -       07-MAR-1983
-1.0,           !  Beginning azimuth - Met. deg.
-1.0,           !  Ending    azimuth - Met. deg.
10.,            !  Beginning slant range - km.
150.,           !  Ending    slant range - km. (maximum available)
MES      -         Mesocyclone Detection          -       27-JAN-1983
-1.0,           !  Beginning azimuth - Met. deg. (minimum available)
-1.0,           !  Ending    azimuth - Met. deg. (maximum available)
-1.0,           !  Beginning slant range - km. (minimum available)
-1.0,           !  Ending    slant range - km. (maximum available)
SEG      -         Storm Segment                  -       27-JAN-1983
0.0,            !  Beginning azimuth - Met. deg.
180.,           !  Ending    azimuth - Met. deg.
25.0,           !  Beginning slant range - km.
-1.,            !  Ending    slant range - km. (maximum available)
```

PARAMETER FILE

```
NEXRAD/PROFS PDL Algorithm Parameter File - PARAMETER.DAT - 22-APR-1983
8                   !  Number of PDLs.
V,C,T,F,S,H,M,E     !  Algorithm Codes (VAD,CEN,TRK,FST,STR,HAL,MES,SEG)
24,2,3,6,0,21,11,2  !  Number of parameters for each algorithm.
VAD         -       Velocity Azimuth Display       -        07-APR-1983
      7.5, !  Threshold
      0.12,         !  Rtau Thres
      0.0,       !  Velocity Threshold
       6,        !  # of slant ranges
     10.00,         !  1st slant range
     15.00,         !  2nd slant range
     20.00,      !  3rd slant range
     25.00,         !  4th slant range
     30.00,         !  5th slant range
     40.00,         !  6th
     45.00,      !  7
     50.00,      !  8
      0.00,      !  9
      0.00,      !  10
      0.00,      !  11
      0.00,      !  12
      0.00,      !  13
      0.00,      !  14
      0.00,      !  15
      0.00,      !  16
      0.00,      !  17
      0.00,      !  18
      0.00,      !  19
      0.00,      !  20
CEN         -       Storm Centroids                -        24-MAR-1983
      4.0,       !  Maximum horizontal distance between centroids (km.)
      1.5,       !  Minimum overlap range for adjacent azimuths (km.)
TRK         -       Storm Tracking                 -        07-MAR-1983
      2.0, !  Correlation-Speed (km/min.)
     16.0,       !  Maximum scan-to-scan time allowed (min.)
    180.0, !  Maximum time period for which storms are tracked (min.)
FST         -       Storm Forecast                 -        04-MAR-1983
     60.0, !  Maximum time period of data used in a forecast (min.)
        4,       !  Number of Forecast Periods
     15.0, !  Forecast Period - Number 1 (minutes)
     30.0,       !  Forecast Period - Number 2 (minutes)
     45.0,       !  Forecast Period - Number 3 (minutes)
     60.0,       !  Forecast Period - Number 4 (minutes)
STR         -       Storm Structure                -        15-DEC-1982
```

PARAMETER FILE (con't)

```
HAL       -      Hail                                -          17-MAR-1983
     17., !Weight # 1
      7., !       # 2
     15., !       # 3
      8., !       # 4
      0., !       # 5
     20., !       # 6
     15., !       # 7
     18., !       # 8
      8., !  Storm top threshold.
     55., !  Maximum storm refectivity threshold.
     50., !  Maximum component reflectivity threshold.
     45., !  Delta angle threshold low.
    180., !  Delta angle threshold high.
      5., !  Height threshold low.
     12., !  Height threshold high.
      4., !  Overhang threshold.
      8., !  Number of predictors (weights).
     -1., !  Y-displacement threshold.
     25., !  Confidence factor threshold low.
     50., !  Confidence factor threshold high.
     60., !  Score threshold.
MES       -      Mesocyclone Detection               -          22-APR-1983
  50000., !  Low momentum threshold      (m**2/sec)
 150000., !  High momentum threshold     (m**2/sec)
   0.002, !  Low shear threshold         (l/sec)
   0.004,     !  High shear threshold         (l/sec)
   1000., !  Max radial distance         (m)
     10.,     !  Criterion - min # of pattern vectors in feature.
     0.5,     !  Minimum diameter ratio for feature.
    2.0, !  Maximum diameter ratio for feature.
   0.35, !  R(tau) threshold for CP-2 data.
   9.28, !  Spectral Width threshold for NSSL data.
    1.0  !  Max. # of consecutive "bad" velocity values.
SEG       -      Storm Segment                       -          15-DEC-1982
  4200.,      !  Minimum length of a segment (meters)
   30.0,      !  Min. reflectivity of a resolution volume in a segment (dBz)
```

APPENDIX K. PROFS/NEXRAD SOFTWARE INDEX LISTINGS

Data Access Routines - Library NEXDAL

Algorithm Routines - Library NEXALGOR

Display Routines - Library NEXDISP

Algorithm Programs/Procedures

PROFS/NEXRAD Data Access Routines

Library NEXDAL

May 11, 1983

Connect/Disconnect

CNCT_DATAFIL_XA:	Connects a PROFS/NEXRAD radar data file.
DSCT_DATAFIL_XA:	Disconnects a PROFS/NEXRAD radar data file.

Data Retrieval

LD_RAY_XA:	Returns calibrated data for a selected ray.
LD_RAY_G_XA:	Returns calibrated data for a set of gates.

Housekeeping

AZM_OF_FIRST_XA:	Returns azimuth angle of first ray in sweep.
AZM_OF_LAST_XA:	Returns azimuth angle of last ray in sweep.
FIELD_NAME_XA:	Returns the ASCII identifier for a field.
FIXED_ANGLE_XA:	Returns target elevation angle for a sweep.
GATE_SPACE_XA:	Returns gate spacing in data file.
GATE_SP_FLD_XA:	Returns gate spacing for a given field.
GET_RAY_HSK_XA:	Returns housekeeping for a selected ray.
I_DIRECTION_XA:	Returns the rotation direction of a sweep.
I_SWEEP_NUM_XA:	Returns the sweep number as on tape.
I_TAPE_REC_1_XA:	Returns the tape record # for 1st ray in a sweep.
I_VOLUME_NUM_XA:	Returns the Volume Number as on tape.
NGATES_MAX_XA:	Returns the maximum number of gates per ray.
NRAYS_IN_SWP_XA:	Returns the number of rays in a sweep.
NUM_FIELDS_XA:	Returns the number of data fields per gate.
NUM_SWEEPS_XA:	Returns the number of sweeps in a connected file.
PRF_XA:	Returns the Pulse Repetition Frequency.
RADAR_NAME_XA:	Returns the ASCII radar name.
RANGE_0_FLD_XA:	Returns the range to the first gate for a field.
RANGE_0_XA:	Returns the range to the first gate (km).
SWEEP_MODE_XA:	Returns ASCII Sweep Mode.
WAVELENGTH_XA:	Returns the radar wavelength (cm).

Utilities

ANG_TO_REAL_XA:	Converts scaled FOF (NCAR) angle values to REAL*4.
CLOSEST_GATE_XA:	Returns index and range of gate closest to desired range.
CLOSEST_RAY_XA:	Returns index and angle of ray closest to desired angle.
DELTA_ANGLE_XA:	Computes the difference between two angles.
FOF_CAL_TBLS_XA:	Sets up look-up tables for calibrating FOF (NCAR) data.
GET_SL_RANGE_XA:	Returns slant ranges for a set of gates.
I_GATE_INDX_XA:	Finds index of gate closest to a desired slant range.
SECTR_INDXS_XA:	Returns ray and gate indexes for a desired sector.
SET_VTHRESH_XA:	Sets the velocity thresholding value of R(tau).
UTF_CAL_TBLS_XA:	Set up look-up tables for calibrating UTF data.

PROFS/NEXRAD Algorithm Routines

Library NEXALGOR

May 09, 1983

Connect/Disconnect

CNCT_DISPFIL_XA:	Connects an algorithm's display file.
CNCT_PRODFIL_XA:	Connects an algorithm's product file.
DSCT_DISPFIL_XA:	Disconnects an algorithm's display file.
DSCT_PRODFIL_XA:	Disconnects an algorithm's product file.

General

AVEARRAY_XG:	Calculates average value of real array.
DIFF_OF_TWO_XG:	Calculates two differences of two variables.
HEIGHT_AGL_XM:	Computes height above ground level.
MET_DIR_XM:	Calculates meteorological angle given (x,y).
MET_TO_XY_XM:	Converts meteorological to Cartesian coordinates.
SORT_DOWN_XG:	Sorts array indexes in decreasing order of array values.
TIM_DIF_MIN_XG:	Calculates I4 time difference in minutes.
XY_TO_MET_XM:	Converts Cartesian to meteorological coordinates.

Data Generation

CEN_CMP_BLD_XQ:	Builds synthetic CENTROID component data.
CEN_PHK_BLD_XQ:	Builds synthetic CENTROID housekeeping.
CEN_STM_BLD_XQ:	Builds synthetic CENTROID storm data.

Data Management

CK_RAD_CODE_XQ:	Checks validity of radar code.
GET_INPUT_XA:	Gets input information for algorithm programs.
READALGOR_XA:	Gets NEXRAD algorithm information.
READPARMS_XA:	Gets NEXRAD algorithm parameter information.
READRADAR_XA:	Gets NEXRAD radar information.
READSECTR_XA:	Gets NEXRAD sector information.

Hail

HAL_PDT_RED_XA:	Reads HAIL data from product file.
HAL_PDT_WRT_XA:	Writes HAIL data to product file.
HAL_PHK_RED_XA:	Reads HAIL housekeeping from product file.
HAL_PHK_RED_XA:	Writes HAIL housekeeping to product file.
HAL_PROCSS_XM:	Processes the NEXRAD algorithm HAIL.

Mesocyclone Detection

MES_FEA_TST_XM:	Tests for features passing certain criteria.
MES_FPV_DSP_XM:	Diagnostic display of pattern vertors by feature.
MES_GET_FEA_XM:	Consolidates pattern vectors into features.
MES_PV_DSP_XM:	Diagnostic display of unordered pattern vectors.
MES_PDT_RED_XA:	Reads MESOCYCLONE data from product file.
MES_PDT_WRT_XA:	Writes MESOCYCLONE data to product file.
MES_PHK_RED_XA:	Reads MESOCYCLONE housekeeping from product file.
MES_PHK_WRT_XA:	Writes MESOCYCLONE housekeeping to product file.
MES_PROCSS_XM:	Processes the NEXRAD algorithm MESOCYCLONE.

Storm Centroids

CEN_CK_OVLP_XM:	Checks overlap between segments in adjacent radials.
CEN_CMP_COR_XM:	Correlates storm components in the vertical.
CEN_LBL_CMP_XQ:	Labels storm components with appropriate storm label.
CEN_LBL_SEG_XM:	Assigns component numbers to storm segments.
CEN_PDT_RED_XA:	Reads CENTROIDS data from product file.
CEN_PDT_WRT_XA:	Writes CENTROIDS data to product file.
CEN_PHK_RED_XA:	Reads CENTROIDS housekeeping from product file.
CEN_PHK_WRT_XA:	Writes CENTROIDS housekeeping to product file.
CEN_PRC_CMP_XM:	Calculates characteristics of a storm component.
CEN_PROCSS_XM:	Processes the NEXRAD algorithm STORM CENTROIDS.
CEN_RAD_COR_XM:	Identifies storm components and correlates in azimuthal.
CEN_STM_CEN_XM:	Calculates storm's centroid.
CEN_STM_ORD_XQ:	Orders component information according to storm number.
CEN_STR_SEG_XM:	Read SEGMENTS data and stores for CENTROID processing.
CEN_SUMPAR_XM:	Summations for computing component characteristics.
CEN_VMASS_XM:	Calculates storm's mass-weighted volume.

Storm Forecast

FST_CALTIM_XG:	Calculates time of a forecasted storm.
FST_COEFS_XM:	Calculates forecast coefficients.
FST_DIRSPD_XM:	Calculates forecasted storm speed/direction.
FST_GET_VOL_XQ:	Selects volume scans for forecasting.
FST_PDT_RED_XA:	Reads FORECAST data from product file.
FST_PDT_WRT_XA:	Writes FORECAST data to product file.
FST_PHK_RED_XA:	Reads FORECAST housekeeping from product file.
FST_PHK_WRT_XA:	Writes FORECAST housekeeping to product file.
FST_POSIT_XM:	Calculates forecasted X and Y storm positions.
FST_PROCSS_XM:	Processes the NEXRAD algorithm STORM FORECAST.

Storm Segments

SEG_CMPRSS_XM:	Computes segment characteristics (ref, vel, etc.).
SEG_CNCTDB_XA:	Connects STORM SEGMENTS product file.
SEG_IDNTFY_XM:	Saves segment's beginning/ending identifiers.
SEG_MWL_MWS_XM:	Computes mass-weighted length and squared values.
SEG_PROCSS_XM:	Processes the NEXRAD algorithm STORM SEGMENTS.
SEG_REDHDR_XM:	Reads SEGMENTS housekeeping from product file.
SEG_WRTHDR_XM:	Writes SEGMENTS housekeeping to product file.
SEG_WRTSEG_XM:	Writes SEGMENTS data to product file.

Storm Structure

STR_PDT_RED_XA:	Reads STRUCTURE data from product file.
STR_PDT_WRT_XA:	Writes STRUCTURE data to product file.
STR_PHK_RED_XA:	Reads STRUCTURE housekeeping from product file.
STR_PHK_WRT_XA:	Writes STRUCTURE housekeeping to product file.
STR_PROCSS_XM:	Processes the NEXRAD algorithm STORM STRUCTURE.

Storm Tracking

TRK_ASN_LBL_XQ:	Assigns labels to newly formed storms.
TRK_COR_CHK_XM:	Determines condition of storm correlation.
TRK_COR_DST_XM:	Calculates storm correlation distance.
TRK_GET_LNM_XQ:	Gets name of last product file produced.
TRK_GET_STM_XQ:	Gets positional indexes of storm back in time.
TRK_PDT_RED_XA:	Reads TRACKING data from product file.
TRK_PDT_WRT_XA:	Writes TRACKING data to product file.
TRK_PHK_RED_XA:	Reads TRACKING housekeeping from product file.
TRK_PHK_WRT_XA:	Writes TRACKING housekeeping to product file.
TRK_PROCSS_XM:	Processes the NEXRAD algorithm STORM TRACKING.
TRK_PUT_CNM_XQ:	Puts product file name in logical name table.
TRK_TBL_INT_XQ:	Initializes information needed for correlation table.
TRK_TBL_UPD_XQ:	Updates information given in correlation table.
TRK_TIM_DIF_XQ:	Finds time difference in minutes between volume scans.

Velocity-Azimuth Display (VAD)

VAD_PDT_RED_XA:	Reads VAD data from product file.
VAD_PDT_WRT_XA:	Writes VAD data to product file.
VAD_PHK_RED_XA:	Reads VAD housekeeping from product file.
VAD_PHK_WRT_XA:	Writes VAD housekeeping to product file.
VAD_PROCSS_XM:	Processes the NEXRAD algorithm VAD.

PROFS/NEXRAD Display Routines

Library NEXDISP

May 11, 1983

General Utilities

ASK_FLT_QST_XD: Given text, displays floating question.
ASK_INT_QST_XD: Given text, displays integer question.
ASK_LOG_QST_XD: Given text, displays logical question.
PLOTIT_GD: General X-Y plot subprogram.
PUT_HEAD_IN_XD: Writes table heading for algorithm product display.
PUT_QUES_IN_XD: Inserts a question for a NEXRAD product display.

RAMTEK Utilities

AXIS: Draws axis for X-Y plot on Ramtek.
COMPUT_VIW_XD: Coordinate transformation for graphic roam and zoom.
DSP_AZI_LUT_XD: Computes look-up table for accessing data in polar form.
DSP_CMP_VIW_XD: Coordinate transformation for image roam and zoom.
DSP_COL_BAR_XD: Displays Ramtek look-up-table colors.
DSP_FST_MAP_XD: Polar to x-y conversion; fast, low resolution version.
DSP_GET_POL_XD: Accesses radar data in polar form.
DSP_LBL_MRK_XD: Labels radar range and azimuth marks.
DSP_RNG_MRK_XD: Draws radar range and azimuth marks.
DSP_SLW_MAP_XD: Polar to x-y conversion; slow, high resolution version.
RDISP_ARC_XD: Draws an arc on Ramtek.
RDISP_LINE_XD: Draws a line, given radar coordinates.
RDISP_SECTR_XD: Draws a box, given radar sector coordinates.
RM_COLOR_LB_XD: Labels color bar on Ramtek.
RM_PUT_FELD_XD: Displays field name, tilt number, and elev. angle.
RM_PUT_HEAD_XD: Displays radar name and date/time.
RMPRINT: Makes hardcopy of Ramtek display.

VT100 Utilities

CLR_ASK_INT_XD: Clears VT100 screen, displays integer question.
CLR_PUT_HED_XD: Clears VT100 screen, displays header information.
CLR_VT100_XD: Clears VT100 screen, positions cursor.

Hail

HAL_PRD_DSP_XD: Displays HAIL prediction information.
PLOT_STORM_XD: Plots storm locations and characteristics.

Mesocyclone Detection

MES_FEA_DSP_XD: Displays MESOCYCLONE feature information.
MES_VOL_DSP_XD: Displays MESOCYCLONE volume summary information.
PLOT_MESO_C_XD: Labels all mesocyclone locations with "M".

Storm Centroids

CEN_CMP_DSP_XD: Displays CENTROIDS storm component information.
CEN_STM_DSP_XD: Displays CENTROIDS storm information.
CEN_TSTOUT_XM: Diagnostic display of component's centroid.
PLOT_STORM_XD: Plots storm locations and characteristics.

Storm Forecast

CONECT_STM_XD: Marks and connects storm locations.
FST_FST_DSP_XD: Displays FORECAST storm forecast information.
FST_HIS_DSP_XD: Displays FORECAST storm forecast history.

Storm Segments

SEG_TSTOUT_XM: Diagnostic display of storm segments.

Storm Structure

PLOT_STORM_XD: Plots storm locations and characteristics.
STR_CMP_DSP_XD: Displays STRUCTURE storm component information.
STR_SCP_DSP_XD: Displays STRUCTURE successive component information.
STR_STM_DSP_XD: Displays STRUCTURE storm information.

Storm Tracking

CONECT_STM_XD: Marks and connects storm locations.
TRK_CTB_DSP_XD: Displays TRACKING storm tracking correlation table.
TRK_HIS_DSP_XD: Displays TRACKING storm tracking history.

Velocity-Azimuth Display (VAD)

VAD_COLOR_XD: Selects color code based on numeric range.
VAD_DRV_DSP_XD: Tabular display of VAD product information.
VAD_DRV_GRA_XD: Graphic display of VAD product information.
VAD_PRO_DSP_XD: Displays VAD wind and divergence profiles.
VAD_VEL_DSP_XD: Displays the velocity vs. azimuth information.

PROFS/NEXRAD Algorithm Programs/Procedures

May 11, 1983

Product/Image Display

DEMOCOPY: Copies products and images for demonstration disk packs.
LOOP: Permits looping of images over time.
MENU: Primary user interface for display of products and images.
RANGEMARK: Display's radar range and azimuth marks.
RDISP: Creates images from polar data.
RECALL: Allows recall and display of specially saved images.
SAVE: Allows special images to be saved for later recall.
SETCOLOR: Allows selection of display color tables.

Product/Image Production

IMAGGEN: Produces fast-load images for selected volume scans.
INPUT: Obtains algorithm input variables of radar and time.
PRODGEN: Produces algorithm product files for selected volume scans.

Hail

HAIL: Executes HAIL algorithm and produces product file.
HALDISPLA: Tabular display of HAIL algorithm product file.
HALGRAPHC: Graphic display of HAIL algorithm product file.

Mesocyclone Detection

MESDISPLA: Tabular display of MESOCYCLONE algorithm product file.
MESGRAPHC: Graphic display of MESOCYCLONE algorithm product file.
MESOCYCLN: Executes MESOCYCLONE algorithm and produces product file.

Storm Centroids

CENBUILD: Builds synthetic CENTROIDS product files.
CENDIAG: Gives diagnostic information for CENTROIDS algorithm.
CENDISPLA: Tabular display of CENTROIDS algorithm product file.
CENGRAPHC: Graphic display of CENTROIDS algorithm product file.
CENTROIDS: Executes CENTROIDS algorithm and produces product file.

Storm Forecast

FORECAST:	Executes FORECAST algorithm and produces product file.
FSTDISPLA:	Tabular display of FORECAST algorithm product file.
FSTGRAPHC:	Graphic display of FORECAST algorithm product file.

Storm Segments

SEGDISPLA:	Image display of SEGMENTS algorithm product file.
SEGMENTS:	Executes SEGMENTS algorithm and produces product file.

Storm Structure

STRDISPLA:	Tabular display of STRUCTURE algorithm product file.
STRGRAPHC:	Graphic display of STRICTURE algorithm product file.
STRUCTURE:	Executes STRUCTURE algorithm and produces product file.

Storm Tracking

TRACKING:	Executes TRACKING algorithm and produces product file.
TRKDISPLA:	Tabular display of TRACKING algorithm product file.
TRKGRAPHC:	Graphic display of TRACKING algorithm product file.

Velocity-Azimuth Display (VAD)

VAD:	Executes VAD algorithm and produces product file.
VADDISPLA:	Tabular display of VAD algorithm product file.
VADGRAPHC:	Graphic display of VAD algorithm product file.

APPENDIX L. ALGORITHM CRITIQUE EXAMPLE - HAIL

NEXRAD PDL/Algorithm Critique

Algorithm Name: Hail Version Number NX-DR-03-012/14

Implementor: J. Smart No. of Conference Questions: 4

A. SPECIFICATION

1.0 PROLOGUE:

Adequate? (Yes/No) Yes
If not, explain.

2.0 INPUTS

Adequate? (Yes/No) Yes; except, the definition of storm top is
If not, explain not consistent with STORM TOP calculation in STRUCTURE

3.0 PROCEDURE

3.1 Algorithm

Did your code follow the PDL exactly? (Yes/No) No

If not, attach revised PDL and explanation of change.

3.2 Computation

Was notation complete and accurate? (Yes/No) No (see below *)

Were equations correct? (Yes/No) No

If not, attach your equations.

Calculation of average direction has been changed.

Were units internally consistent? (Yes/No) Yes

*Were units externally consistent with other algorithms?
(Yes/No) No; units for speed are km/min while notation specified speed as km/sec.

A. SPECIFICATION (continued)

Did units conform to accepted meteorological standards?
(Yes/No) Yes

Attach any additional comments.

More care is necessary when dealing with units for storm direction and computations using this parameter.

4.0 OUTPUTS:

Adequate? (Yes/No) No

If not, explain.

Included with the labels for each storm are the storm identifier and its position.

5.0 INFERENCES:

Generally adequate? (Yes/No) No

If not, explain.

Further discussion of the storm top should be included since it plays such an important role in hail identification process.

All limitations addressed? (Yes/No) No

If not, attach additional limitations.

Centroids are adjusted by a distance based on the forecasted speed of the storm and the elapsed time between radar sweeps. When a forecasted speed is wrong (due to small number of previous storm positions or a poor track from which the forecast is made, to name two), the centroid adjustment is wrong leading to a possible false hail detection.

B. IMPLEMENTATION

1. Did PDL adequately specify the algorithm? (Yes/No) Yes

   If not, comment.

2. Is PDL adequately structured to implement as a real-time process?

   (Yes/No) Yes

   Comment.

   However, by including the storm component HEIGHT, RADIUS and RANGE as input from the STRUCTURE algorithm, one could eliminate component area, the elevation angles and section 3.8 from the HAIL algorithm.

3. Were "boundary conditions" properly handled? (Yes/No) Yes

   If not, explain.

C. OPERATION

1. List all problems you found when you ran the software (other than coding errors).

   None.

2. Describe the algorithm's sensitivity to "dirty" data?

   N/A. See limitations in A.5.

3. Describe the functional meteorological performance of the algorithm. Include, if appropriate, incidences of false detection (tracking), lack of detection (tracking), splitting of storms, etc.

   Many storms are labeled POSITIVE, more than one would imagine. However, verification data does not allow us to validate all the POSITIVE detections.

   Most hail verifications within CP-2 range where associated with POSITIVE hail detection, but not all. No PROBABLE or INSUFFICIENT DATA detections have been observed.

ACRONYMS AND ABBREVIATIONS

AEL	Algorithm Enunciation Language
AFGL	Air Force Geophysics Laboratory
ANSI	American National Standards Institute; organization which establishes standards used for computers and information processing
ASCII	American Standard Code for Information Interchange; a standard code used for the representation of text information
AWS	Air Weather Service
CEN	Storm Centroids, a NEXRAD algorithm
CHILL	Name of the Doppler radar owned and operated by the Illinois State Water Survey
CIMMS	Cooperative Institute for Mesoscale Meteorological Studies
CP-2	NCAR's 10-cm Doppler radar
DCL	Digital Command Language, a language used to communicate with the VAX operating system
DEC	Digital Equipment Corporation
ERL	Environmental Research Laboratories
FAA	Federal Aviation Administration
FOF	Field Observing Facility
FORTRAN	Formula Translation, high-level computer programming language used for all of the NEXRAD software
FST	Storm Position Forecast, a NEXRAD algorithm
G.m.t	Greenwich mean time (also GMT)
HAL	Hail, a NEXRAD algorithm
IAMAP	International Association of Meteorology and Atmospheric Physics
INPUT	A command issued by a user to the computer which initiates the NEXRAD software to request input information for the identification of a volume scan of radar data

I/O	Input and/or Output
IOTF	Interim Operational Test Facility of NEXRAD, in Norman, Oklahoma
IPS	Inches per second
ISTATUS	A code name used for a piece of information which provides error status information resulting from the execution of a module of NEXRAD computer software
JAWS	Joint Airport Weather Studies
JSPO	Joint System Program Office
m.d.t.	Mountain daylight time (also MDT)
MES	Mesocyclone Detection, a NEXRAD algorithm
MIPS	Million instructions per second
NCAR	National Center for Atmospheric Research
NEXALGOR	Symbolic name of a NEXRAD object code software library containing all modules used for processing for the NEXRAD algorithms
NEXDAL	Symbolic name of a NEXRAD object code software library containing all modules used for accessing the NEXRAD data files
NEXDAT	Name of the computer user account used for all of the NEXRAD raw radar data, algorithm products, and special display images
NEXDISP	Symbolic name of a NEXRAD object code software library containing all modules used for displaying the NEXRAD data, products, and images
NEXRAD	1) Next Generation Weather Radar; 2) name of the computer user account used for all of the NEXRAD software and associated documentation
NOAA	National Oceanic and Atmospheric Administration
NSSL	National Severe Storms Laboratory, Norman, Oklahoma
NWS	National Weather Service
PDL	Program Design Language
PPI	Plan Position Indicator, one sweep of radar at a given elevation angle
PRF	Pulse Repetition Frequency

PROFS	Program for Regional Observing and Forecasting Services
RCS	Symbolic representation of the file extension of NEXRAD data and product disk files where R = radar code, C = class, and S = subclass
RDSS	Research Data Support System of the Field Observing Facility at NCAR
READPARMS_XA	Symbolic name of NEXRAD software module which reads the algorithm processing parameter disk file
RMS	Root-mean-square
SASC	Systems and Applied Science Corporation, private contractor to NEXRAD
SDN	Symbolic representation of the file extension of NEXRAD special display image disk files where S = sweep number, D = data source, and N = sequence number
SEG	Storm Segments, a NEXRAD algorithm
STR	Storm Structure, a NEXRAD algorithm
TAC	Technical Advisory Committee of NEXRAD
TRK	Storm Tracking, a NEXRAD algorithm
TSC	Transportation System Center, program in Department of Transportation to study effects of weather on air traffic control
VAD	Velocity-Azimuth Display, a NEXRAD algorithm
VAS	VISSR Atmospheric Sounder
VAX	Virtual Address Extension, name of the DEC computers at PROFS
VAX-11/750	Computer used for NEXRAD project work at PROFS, made by DEC
VISSR	Visible and Infrared Spin Scan Radiometer
VMS	Virtual Memory System, the computer operating system used by the VAX
UTF	Universal Tape Format
YYJJJHHMM	Symbolic code for a date and time; YY = last two digits of year, JJJ = Julian day of the year, HH = G.m.t. hour, and MM = G.m.t minute