ETL-0569

AD-A238 937

Etak Navigator Modification

Final Report

Dr. Walter B. Zavoli
Gene E. Bloch

Etak, Incorporated
Menlo Park, California 94025

October 1990

Approved for public release; distribution is unlimited.

U.S. Army Corps of Engineers
Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060-5546

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE
October 1990

3. REPORT TYPE AND DATES COVERED
Technical report: Jun 89 - Aug 90

4. TITLE AND SUBTITLE
Etak Navigator Modification Final Report

5. FUNDING NUMBERS
DACA72-89-C-0007

6. AUTHOR(S)
Dr. Walter B. Zavoli
Gene E. Bloch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSE(S)
Etak, Inc.
1430 O'Brien Drive
Menlo Park, CA 94025

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Topographic Laboratories
Fort Belvoir, VA 22060-5546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ETL-0569

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
Etak modified its land vehicle navigation device, the Navigator, for test and evaluation by the U.S. Army Engineer Topographic Laboratories. The Navigator is a low-cost off-the-shelf commercial device that exhibits accurate navigation along with a highly useful electronic map display. The device uses a combination of dead reckoning and map matching. As part of this contract, Etak found that it could create the necessary maps from DMA 1:50,000 scale source material, to an accuracy of 50 meters, and that the Navigator could input and display vehicle positions and waypoints in UTM coordinates.

In almost 1400 km of drive testing in Fort Hood, Texas, the modified Navigator showed that as a dead-reckoning device it is accurate to 2% of distance traveled, while its map-matching algorithm gives the Navigator performance comparable to that of an absolute navigation device with an average error of 50 meters.

This navigation device demonstrates useful performance for certain classes of Army vehicles. Other vehicles may require more robust and hence more costly devices. It is suggested that digital map displays like that of the Navigator could be a useful standard presentation device for all Army vehicle navigation.

14. SUBJECT TERMS
Etak Navigator; dead reckoning; electronic map display; digital map display; vehicle navigation; off-road navigation; UTM conversion

15. NUMBER OF PAGES
239

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Preface

This report describes work performed under Contract DACA72-89-C-0007, "Modification of Etak Navigators," by Etak, Incorporated, Menlo Park, California 94025 for the U. S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, 22060-5546. The Contracting Officer's Technical Representative at ETL was Thomas M. Cox, Jr.

Contents

List of Tables ix

List of Figures xii

1. Summary 1

Document Overview 2

2. Introduction 3

Summary 3

Background 4

Phase I: Design 6

Use of UTM Coordinates 6

User Interface 7

Navigation 7

DMA Map Evaluation 8

Phase II: Implementation and Evaluation 9

3. Modifying the Navigator 11

Overview 11

Navigator Screen Modifications 11

Navigation Control 12

Coordinate Conversion Software 12

Detailed Description of User Interface 13

Menu Screen 13

Map Screen 13
Destination Options Screen 14
Map Information Screen 14
UTM Subfield Screen 14
UTM Easting and Zone Screens 14
Verify Screen 22
Using the Screens 22
Initializing and Updating via UTM Coordinates 22
Entering a Waypoint or Destination via UTM Coordinates 25
Outputting Position Coordinates and Heading to the Display 25
Displaying the Destination Distance and Heading to the Destination 25
Telling the Navigator when it is not Map Matching 25
Coordinate Conversion Routines 26
Constants of the Spheroids 26
Spheroids/Ellipsoids 27
Zone Choice and Validation 27
Zone Display Criteria 27
Entry Validation 28
Accuracy Tests 28
Timing Tests 29

4. Etak Maps 31
Introduction 31
Etak's Digital MapBase Structure 32
Etak's Map Production Capability 34
Capabilities of Topologically Structured Maps 35
Data Processing 35
Map Access Performance 36

Future Capabilities 37
5. Map Data Capture 39
Introduction 39
Features Captured 39
Map Data Capture Analysis 42
Checking the Results 43
Conclusions 44
6. Digital Map Testing 47
Introduction 47
Baseline Points 48
Baseline Data 51
Digital Map Tests 51
Source Material Validation 53
EtakMap Tests 65
Summary 65
7. Navigation Tests 71
Test Configuration 71
Methods of Navigational Accuracy Analysis 76
Methods of Data Presentation 77
Data Presentation 79
8. Discussion of Navigation Data Analysis 121
Comparisons between Dead Reckoning and Map Matching 121
A Close Look at Map-Matching Performance 124
9. Summary of Navigation Results and Discussion 137
Summary of Dead Reckoning Performance 137
Summary of Map-Matching Performance 139

10. Conclusions and Recommendations 143
Conclusions 143
Recommendations 144

References 145
Appendix 1. UTM Conversion Software 147
Introduction 147
What is Provided 147
Executables and the User Interface 148
How the Mode of Operation is Chosen 149
Examples of Dialog with the Programs 150
Appendix 2. Source Code Listings 153
Batch Files 153
Makefile 155
Header Files 156
Program Files 162
Appendix 3. Raw Digitization Data 195
Appendix 4. Raw Navigation Data 205

List of Tables

1. Coordinate Errors (mm) 29

2. Coordinate Conversion Times (msec) 30

3. N-Class Features Captured 41

4. EtakMap Data Capture 42

5. Terrain Features 43

6. Observation Scales 44

7. Road Priority Levels 44

8. Digitization Errors (meters) - Purmela 53

9. Digitization Errors (meters) - Gatesville 54

10. Digitization Errors (meters) - Fort Hood 55

11. Digitization Errors (meters) - Killeen 56

12. Digitization Errors (meters) - Test Roads 57

13. UTM Errors (meters) - Purmela 59

14. UTM Errors (meters) - Killeen 62

15. EtakMap Errors (meters) - Purmela 66

16. EtakMap Errors (meters) - Gatesville 67

17. EtakMap Errors (meters) - Fort Hood 68

18. EtakMap Errors (meters) - Killeen 69

19. EtakMap Errors (meters) - Test Roads 70

20. Counts of Runs 76

21. April WEMM Error Analysis (meters) 80
22. April EWMM Error Analysis (meters) 82
23. April SNMM Error Analysis (meters) 84
24. April NSMM Error Analysis (meters) 86
25. April WEDR Error Analysis (meters) 88
26. April EWDR Error Analysis (meters) 90
27. April SNDR Error Analysis (meters) 92
28. April NSDR Error Analysis (meters) 94
29. June WEMM Error Analysis (meters) 96
30. June EWMM Error Analysis (meters) 98
31. June SNMM Error Analysis (meters) 100
32. June NSMM Error Analysis (meters) 102
33. June WEDR Error Analysis (meters) 104
34. June EWDR Error Analysis (meters) 106
35. June SNDR Error Analysis (meters) 108
36. June NSDR Error Analysis (meters) 110
37. June WEDR Error Analysis (ETL) (meters) 112
38. June EWDR Error Analysis (ETL) (meters) 114
39. June SNDR Error Analysis (ETL) (meters) 116
40. June NSDR Error Analysis (ETL) (meters) 118
41. Dead Reckoning Errors (%) 139
42. Average Map-matching Errors (meters) 140
43. Raw Digitization Data - Purmela 196
44. Raw Digitization Data - Gatesville 196
45. Raw Digitization Data - Fort Hood 197
46. Raw Digitization Data - Killeen 197
47. Raw Digitization Data - Test Roads 198

48. Raw Map Data - Purmela 199

49. Raw Map Data - Killeen 200

50. Raw Navigator Data - Purmela 201

51. Raw Navigator Data - Gatesville 201

52. Raw Navigator Data - Fort Hood 202

53. Raw Navigator Data - Killeen 202

54. Raw Navigator Data - Test Roads 203

55. Raw Test Run Data - WEMM April 207

56. Raw Test Run Data - EWMM April 208

57. Raw Test Run Data - SNMM April 209

58. Raw Test Run Data - NSMM April 210

59. Raw Test Run Data - WEDR April 211

60. Raw Test Run Data - EWDR April 212

61. Raw Test Run Data - SNDR April 213

62. Raw Test Run Data - NSDR April 214

63. Raw Test Run Data - WEMM June 215

64. Raw Test Run Data - EWMM June 216

65. Raw Test Run Data - SNMM June 217

66. Raw Test Run Data - NSMM June 218

67. Raw Test Run Data - WEDR June 219

68. Raw Test Run Data - EWDR June 220

69. Raw Test Run Data - SNDR June 221

70. Raw Test Run Data - NSDR June 222

71. Raw Test Run Data - WEDR June (ETL) 223

72. Raw Test Run Data - EWDR June (ETL) 224

73. Raw Test Run Data - SNDR June (ETL) 225

74. Raw Test Run Data - NSDR June (ETL) 226

List of Figures

1a. Navigator Display 5
1b. Navigator Components 5

2. Menu Screen 15

3. Navigator Map Screen 16

4. Destination Options Screen 17

5. Compass Rose/Map Info Screen 18

6. Select UTM Subfield Screen 19

7. Select UTM Easting Screen 20

8. Select UTM Zone Screen 21

9. Verify Map-Match Change Screen 23

10. A Map in Topological Terms 33

11. Test Point Targets for Map Verification 49

12. Test Point Targets on Purmela Section 50

13. UTM Errors - Purmela 60

14. Digitization Errors - Purmela 61

15. UTM Errors - Killeen 63

16. Digitization Errors - Killeen 64

17. Navigator Installed in CUCV 72

18. Data Collection Checkpoints for Test Routes 73

19. Detailed Digital Map of Test Area 74

20. Eastern End of West-East Route WEMM 78

21. April WEMM Error Plot 81

22. April EWMM Error Plot 83

23. April SNMM Error Plot 85

24. April NSMM Error Plot 87

25. April WEDR Error Plot 89

26. April EWDR Error Plot 91

27. April SNDR Error Plot 93

28. April NSDR Error Plot 95

29. June WEMM Error Plot 97

30. June EWMM Error Plot 99

31. June SNMM Error Plot 101

32. June NSMM Error Plot 103

33. June WEDR Error Plot 105

34. June EWDR Error Plot 107

35. June SNDR Error Plot 109

36. June NSDR Error Plot 111

37. June WEDR Error Analysis (ETL) 113

38. June EWDR Error Analysis (ETL) 115

39. June SNDR Error Analysis (ETL) 117

40. June NSDR Error Analysis (ETL) 119

41. Map Plot for the June NSDR Set 122

42. Map Plot for the June SNDR Set 123

43. Map Plot for the June SNMM and SNDR Sets 125

44. Map Plot at 2-Mile Scale Showing Map Error 127

45-52. Map Plot at 1-Mile Scale for June EWMM Runs 128

53. Compendium of June Map-Matching Runs 136

54. Map Matching Position Error 141

Chapter 1

Summary

In accordance with Contract DACA72-89-C-0007, Etak has modified its land vehicle navigation device, the Etak Navigator®, to operate in an Army format. Specific modifications include:

1. Use of Universal Transverse Mercator (UTM) coordinates for all operator interfaces.
2. Selection of either of two types of navigation: dead reckoning or dead reckoning augmented with map matching.
3. Use of digital maps derived entirely from DMA 1:50,000 scale source maps.

A digital map of Fort Hood, Texas, and the surrounding area of approximately 800 square miles was prepared. An Etak Navigator was installed in an Army Commercial Utility Cargo Vehicle (CUCV) at Fort Hood and loaded with the map and the modified navigation software. Tests were performed on the accuracy of the resulting map and on the navigational performance of the Navigator®.

It was shown that features appropriate for land navigation can be extracted from the Defense Mapping Agency (DMA) source material and that positional accuracy can be maintained to 30 meters. This falls well within National Map Accuracy Standards.

In almost 1400 km of navigation field testing at Fort Hood, the Navigator demonstrated an average accuracy of 2% of distance travelled while operating

under dead reckoning alone and an accuracy of under 50 m when operating under dead reckoning augmented with map matching.

These results, combined with the added navigational benefit of a digital map display oriented to the vehicle heading, demonstrate the functionality and performance of this device and of on-board digital map applications.

Etak recommends that further work be done in enhancing digital map displays for Army purposes, standardizing navigation user interfaces to include a map display (even if different navigation sensors and methods are used for different classes of vehicles) and, finally, equipping an Army operating unit with Navigators linked to their command post via digital radio to test the operational advantages of these devices.

Document Overview

Chapter 2 provides an introduction to the project and its history and a brief summary of the changes to the Navigator. In chapter 3, the coordinate conversion software and the operation of the Navigator in UTM coordinates are described. Chapter 4 describes Etak's mapping capabilities in general terms, leading to the discussion in Chapter 5 of how the Fort Hood data were to be incorporated into an EtakMap. Chapter 6 details analysis of the digital mapping accuracy problems that were encountered, and the testing that helped in resolving those problems. The driving tests of the Navigator and the EtakMap at Fort Hood are discussed in Chapter 7. Chapter 8 discusses data analysis of the driving tests by giving a detailed interpretation of one set of runs. Chapter 9 gives summary accuracy figures for the test runs, and Chapter 10 concludes with Etak's recommendations.

The report is supplemented with several appendices. Appendix 1 describes the standalone version of the coordinate transformation software, while complete printed listings of this software are provided as Appendix 2. The raw data used for validating the digitization of the EtakMap are given in Appendix 3, and the data from the test drives are in Appendix 4.

Chapter 2

Introduction

Summary

This report constitutes the final deliverable to the U.S. Army Engineer Topographic Laboratories (ETL) as part of Contract DACA72-89-C-0007 ("the contract"). The contract was divided into two phases.

Phase I was a study and design phase, to determine and plan the actions required to implement changes to the Etak Navigator's software, and to design the symbols and digitizing methods required to create Navigator digital maps, called EtakMaps, from DMA maps. This phase called for two studies: one to design modifications to Etak software for the Navigator vehicle navigation system, and the other to investigate Etak's ability to digitize Defense Mapping Agency (DMA) 1:50,000 scale maps accurately, as well as to represent certain terrain features.

Phase II entailed the implementation of the requisite changes to the Navigator; the final digitization of the full DMA maps of the Fort Hood, Texas, area; the installation of Navigators in an Army vehicle; and testing of the system as installed.

Background

The stated objective of this contract effort is to evaluate the utility of Etak's navigation approach for the purpose of the Army's Position/Navigation requirements.

Accurate navigation is a requirement for proper functioning of all elements of the military. Electronic navigation aids have long been a part of navigation by sea and by air. In the past, the cost of such devices has been prohibitive in light of the large number of land vehicles, especially when considering performance issues and the fact that inexpensive paper maps already provide the pilot with a rich source of information to help him deduce his position and plan his route. Still, with the increased emphasis on mobility in evolving Army doctrine, navigational aids are being considered as an aid to more timely and more accurate in-vehicle navigation.

At the same time, navigation and digital map technologies have evolved to provide high-performance low-cost navigation aids with built-in electronic maps. These devices have the potential to aid the pilot in accurately and quickly determining where he is and where he wants to go, and in selecting a route to get there.

The Etak Navigator shown in Figures 1a and 1b was the first device to combine dead reckoning with map matching to achieve a highly accurate and low-cost commercial navigation product. The original Navigator has an electronic map display that shows the vehicle's current position on a road map at scales ranging from 1/8 mile to 40 miles. (The scale measures the distance from the vehicle indicator to the upper screen edge. This distance is about 2" on Etak's 4.5-inch display, corresponding to map scales from roughly 1:4,000 to 1:1,250,000 on screen). As the scale increases, more details of the map appear automatically. While the vehicle is moving, the electronic map is configured in a heading-up, vehicle-centered mode which continually displays to the driver a properly oriented map.

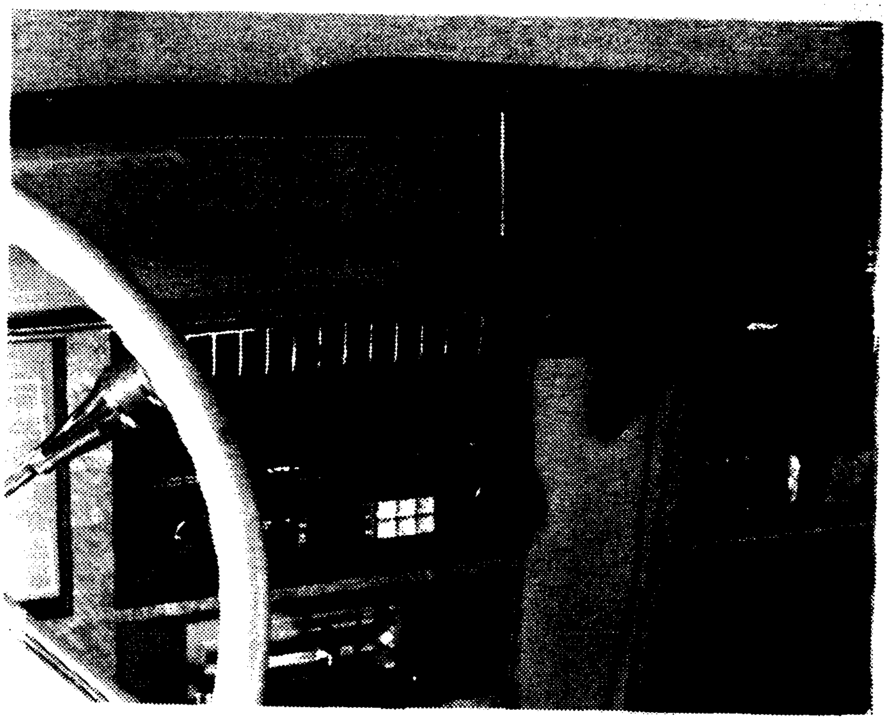

Figure 1a. Navigator Display
As Installed in a Typical Civilian Vehicle.

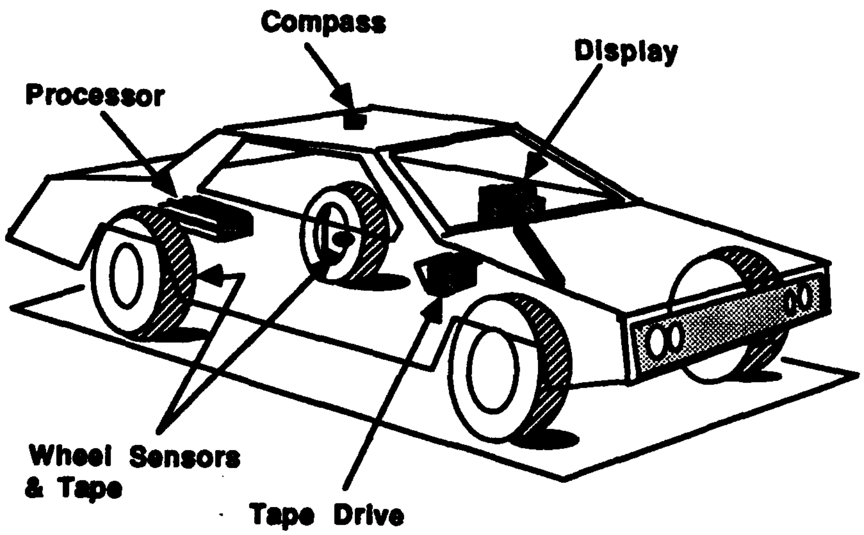

Figure 1b. Navigator Components
Typical Component Placement of the Navigator

Drivers may enter destinations, or waypoints, on the map. In the commercial version, these waypoints are entered as street addresses, either from buttons on the sides of the screen or through a communications link to a centralized dispatch operation, which may also be monitoring and coordinating, on its own electronic map, the positions, headings, status, and destinations of the vehicles in its fleet.

In 1986, ETL purchased an Etak Navigator to evaluate the performance of this commercial device in view of military navigation needs. At that time, no attempt was made to alter the user interface of the Navigator to make it more appropriate for Army operations. The device performed well enough that a further study of modifications was undertaken.

The work performed under this contract was a study of the design and modification of the Navigator and its digital maps for the purpose of operating with data and functionality consistent with Army operations. A digital map of Fort Hood was created and tests were performed to evaluate the digital map and Navigator performance. The study and modifications were in three general areas, as outlined below. This final report, which concludes the contract, discusses the Navigator modifications, gives the results of performance and feasibility studies, a description of the software and mapping design choices, and discusses coordinate conversions and map displays, the results of Etak's analysis of digitizing accuracy, and the results of navigation tests performed at Fort Hood.

Phase I: Design

Use of UTM Coordinates

The contract requires that the Navigator be able to accept and display coordinates in the Universal Transverse Mercator (UTM) coordinate system. This feature is central to its use in Army operations. The commercial Navigator uses latitude and longitude coordinates internally and for displays. Destination selection is done by entering street names and addresses. For the Army, UTM coordinates are required as the primary coordinate system for input and output.

The use of UTM required the addition of two conversion routines to the software, one to convert UTM coordinates to the internal (geographic) coordinates used by the Navigator, and the other to perform the inverse transform, from geographic coordinates and heading to UTM and grid heading.

User Interface

Several screens were modified or added to accommodate UTM input and output. Specifically, the operator needs to enter UTM coordinates directly, in order to adjust the Navigator's estimate of the vehicle's position. This required the addition of input screens and methods to add this capability to the current repositioning capabilities of the Navigator. A similar capability is needed for entering a destination location in UTM coordinates.

Output requirements include the ability to display the current position and heading of the vehicle in UTM coordinates. This required a redesign of the Navigator's Map Info screen, which in the commercial version displays position and heading in geographic coordinates.

The display of the distance and direction to the current destination must also be consistent with UTM. This again required modifying a capability already present in the Navigator, with a change of units to kilometers and a modification of the heading to incorporate the grid convergence angle.

Navigation

A final aspect of the user-interface modification involves the unique way in which the Navigator navigates, namely by a combination of dead reckoning and map matching. Dead reckoning uses a solid-state flux-gate compass with software compensation as well as high-resolution wheel-rotation counters. Over most road surfaces this provides accuracies typical of dead-reckoning devices: an error of a few percent of distance travelled.

For civilian applications, where virtually all travel is along streets and highways, the Navigator's accuracy is enhanced by proprietary map-matching technology. Periodically, the Navigator compares the recent history of heading and direction travelled to the configuration of roads in the vicinity, and updates the vehicle's position when it will improve the match between the vehicle's travel history and the map.

This map-matching navigational technique has a major effect on the accuracy of position estimation. It transforms the performance of the Navigator from a dead-reckoning device nearly to that of an absolute device, with errors on the order of 50 meters, independently of the distance travelled. It is this aspect of the Navigator's design which enables it to achieve accuracies comparable to that of equipment many times more expensive.

However, military vehicles are expected to travel off-road as part of their mission, sometimes while travelling parallel to a nearby road. In this situation, the map-matching algorithm might, on occasion, make reasonable but incorrect updates. For this reason, and to further test the Navigator, the contract required the capability to turn map-matching off and on. This was done by providing screens and corresponding internal logic to accept these commands.

DMA Map Evaluation

Etak's digital map technology offers unique functionality and performance for electronic maps. Typically, Etak creates its maps from a variety of photographic, paper, and machine-readable sources. For the purpose of evaluating this technology in Army operations, the contract requires Etak to produce its digital maps exclusively from 1:50,000 scale DMA maps. Four such maps, in addition to their color separates, have been provided for the Fort Hood, Texas, test area. The contract required Etak not only to digitize the maps for use in the Navigator and to evaluate their positional accuracy in UTM coordinates, but also to study the content of DMA maps to determine how their features can be adapted to Etak's technology.

Etak has approached this study with two separate objectives in mind. The first is in response to the more general question of which features might be captured and how they might be displayed. The second is in response to which specific features will be captured for display with the limited hardware capabilities of the Navigator.

Because Etak's business is in digital map publishing, it is continually improving its abilities to capture geographic data, to display it, and to use it electronically to perform geographically-oriented tasks. Today these capabilities far outstrip the limited functionality of the 4.5-inch vector-graphic monochromatic display, the slow microprocessor, and the slow and limited mass-storage device of the Navigator. Newer navigation hardware designs include color-graphic displays, faster microprocessors, and faster and larger mass-storage devices. For these reasons, Chapter 4 is devoted to describing Etak's map capability without reference to limitations embodied in the Navigator's hardware platform. Chapter 5 discusses the features that were captured from the DMA source material and how they appear on the Navigator's screen.

Phase II: Implementation and Evaluation

The modifications outlined above were implemented in the Navigator's software, and the maps of the Fort Hood, Texas area were digitized. Then a series of tests was performed to evaluate the accuracy of the resulting map. The tests and results are described in Chapter 6. The digital map of Fort Hood was processed into Etak's proprietary map format (EtakMap) to be used in conjunction with the modified software for the navigation tests.

After testing at Etak's factory, two Navigators were shipped to Fort Hood for installation in an Army CUCV. One unit was installed and calibrated and a series of tests was conducted by Etak and ETL representatives to determine navigation and map accuracies. ETL personnel were trained in the use of the units. Chapters 7 through 10 detail the findings of these tests.

Chapter 3

Modifying the Navigator

Overview

The objective of modifying the Navigator's user interface was to adapt the device for Army operations. This was accomplished in three ways:

1. By providing input and output screens and corresponding internal logic to accommodate UTM coordinates.
2. By providing a user interface to enable and disable the map-matching navigation function.
3. By installing algorithms to convert the user coordinate system (UTM) to and from the Navigator's internal coordinate system (latitude and longitude).

Details of the modifications are presented below.

Navigator Screen Modifications

The main Navigator map screen shows an electronic road map centered on the vehicle's position and oriented to the vehicle's heading. This screen was modified to show distance scales in metric units and directions with respect to grid north. The Map Information (Compass Rose) Screen was similarly modified.

Entering a waypoint is done through the Destination Options Screen, which has been modified to show, and to allow selection by, UTM coordinates. Additional screens are used for entering easting, northing, and zone information. Similar screens are used in conjunction with the Navigator's reposition sequence.

Navigation Control

The Navigator uses dead-reckoning navigation augmented with a map-matching algorithm. Map matching is very effective in minimizing the accumulation of dead-reckoning errors, but it operates under the assumption that the vehicle will probably be driving on the road network. This assumption is not always valid for Army operations, so a software switch, controlled by a button on the main map screen, was provided to allow the driver to enable or disable map-matching.

Coordinate Conversion Software

The major software programming task for Phase I of the contract was to convert UTM coordinates to geographic coordinates and vice versa. The design of the conversion software for the Navigator had to account for several characteristics of the task and of the Navigator's computer platform, including the required accuracy, the need for reasonable computational speed, and the lack of a floating point processor and of a fast floating-point library in the platform (the Navigator's computer system). This problem was resolved during the study phase by the creation of a test program called utm2geo. Utm2geo was developed from formulations given in Snyder, *Map Projections Used by the U.S. Geological Survey* (Reference 1) and *Universal Transverse Mercator Grid* (Reference 2).

In order to provide the initial proof of method and a reference for performance, the program was first written in double-precision floating-point arithmetic, using Microsoft C 5.10 on a PC compatible. When that version was debugged and its accuracy was verified, the program was converted to do all computations in long-integer arithmetic, using a custom integer math library written in assembly language. Although the integer program is slightly larger than the double-precision floating-point version, it is faster by a factor of about 5 (its running time depends on the coordinates, since some of the arithmetic and trigonometric functions bypass some calculations in certain ranges of their arguments).

Etak's testing demonstrated conversion accuracies on the order of centimeters and computational speeds on the order of 50 milliseconds. These performance results were much better than needed to insure that the Navigator's performance would not be limited by coordinate conversions.

This exercise both proved the mathematical validity of the approach and showed that the performance impact (on memory usage and on timing) would be moderate. It fully demonstrated the feasibility of making the necessary modifications to the Navigator software.

During Phase II, the conversion algorithms developed in utm2geo were incorporated into the Navigator software in order to provide the UTM capabilities called for in the contract. In addition, the source code for these conversion algorithms is provided as a contract deliverable. The methods used are discussed below, and Appendix 1 describes the stand-alone form of the program. The code is provided in machine-readable form on a 3.5" diskette, and a printed listing is provided as Appendix 2 of this report.

Detailed Description of User Interface

The eight figures presented in this section illustrate the new and modified Navigator screens which are used to enter and display locations and to choose the map-matching state, as described below. Detailed information on the screens is found in *Etak Navigator (UTM Version) Operating Instructions for EtakMap Series 1*, Reference 4.

Menu Screen

The Army version of the Menu Screen (see Figure 2) contains a new function: a previously unused button contains the symbol for the PAN function. In addition, the destination display will be in UTM coordinates if the current destination was entered from the UTM screen.

Map Screen

The Map Screen (see Figure 3) is the Navigator's key screen. There are three changes to this screen. First, all units are metric; that is, the scales and the distance to the destination are shown in kilometers rather than miles. Second, the direction arrows are shown oriented to grid north, rather than geographic north (the difference is never more than three degrees). Finally, one of the buttons now contains either "MM" or "MM in a barred circle". This button is used to turn map matching on and off.

Destination Options Screen

The Destination Options Screen (see Figure 4) contains a new button function to allow using UTM coordinates to enter a destination. In addition, the coordinates of the destination are displayed in UTM if the current destination was entered from the UTM screen. Although the Reposition Vehicle Screen is not shown separately in this report, it contains similar changes.

Map Information Screen

The modified version of the Map Information Screen (see Figure 5), like the Map Screen, shows all distances and scales in kilometers and headings in terms of grid north. The coordinates of the vehicle and of the destination are displayed in UTM if the current destination was so entered. The compass rose now contains the value of the vehicle's grid heading as a three-digit number of degrees. Note that the vehicle's heading is always represented as straight up on this screen.

UTM Subfield Screen

Figure 6 shows the Select UTM Subfield Screen. Buttons are used to choose UTM zone, easting, or northing for data entry.

UTM Easting and Zone Screens

The Select UTM easting and zone screens (see Figures 7 and 8) are the screens on which the values of easting and zone are entered. The screen for northing is identical, and therefore is not shown. These screens are conceptually the same as the original Enter Address Screen.

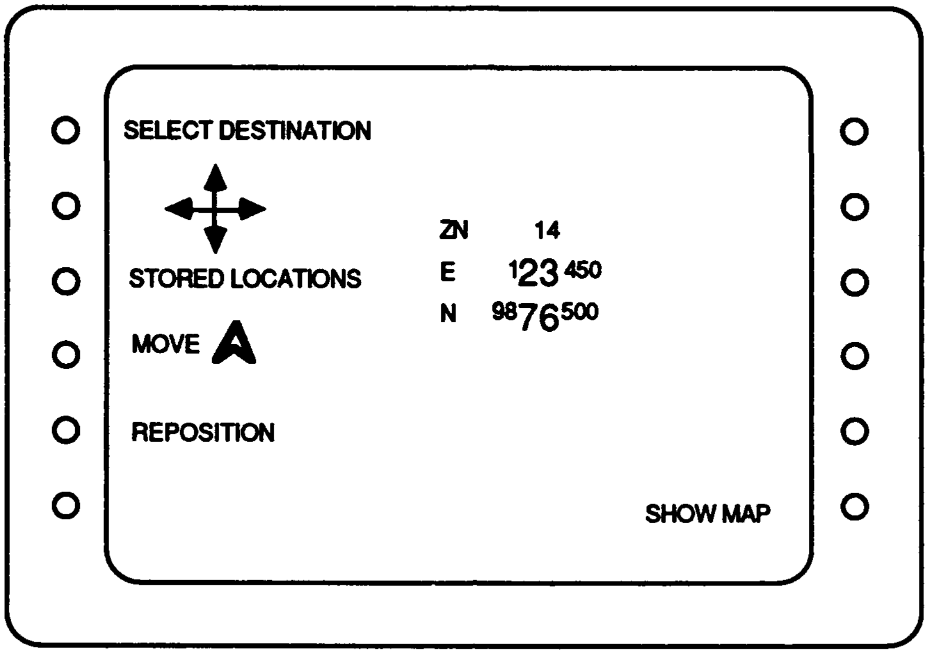

Figure 2. Menu Screen

Note: South Latitude is shown as a negative zone, as in "ZN-14".

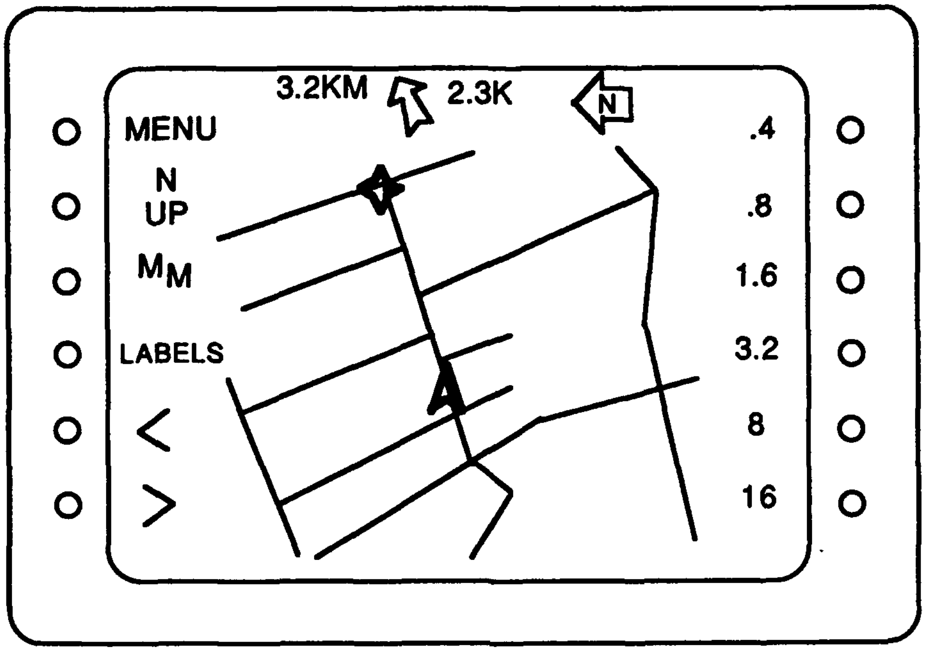

Figure 3. Navigator Map Screen

Notes:

1. Screen same as original, except units are metric.
2. Two successive pushes of .4KM button bring up Map info Screen.
3. When map-matching is ON, button 2 will select "MATCH OFF" (shown as MM in a barred circle). Pressing the button then brings up the Verify Screen to verify turning map-matching off. When map-matching is OFF, button 2 will select "MATCH ON" (shown as MM alone). Pressing the button then brings up the Verify Screen to verify turning map-matching on. This follows Etak's convention that the label describes the action that the corresponding button will cause.

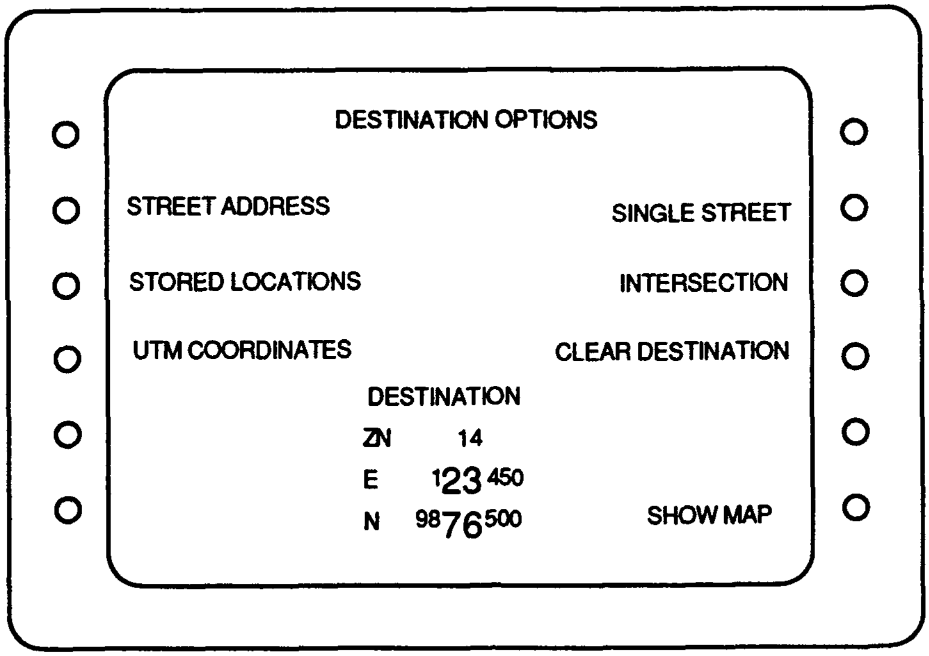

Figure 4. Destination Options Screen

Note: The format of the destination display reflects the format in which the destination was last entered (that is, UTM, street address, etc.).

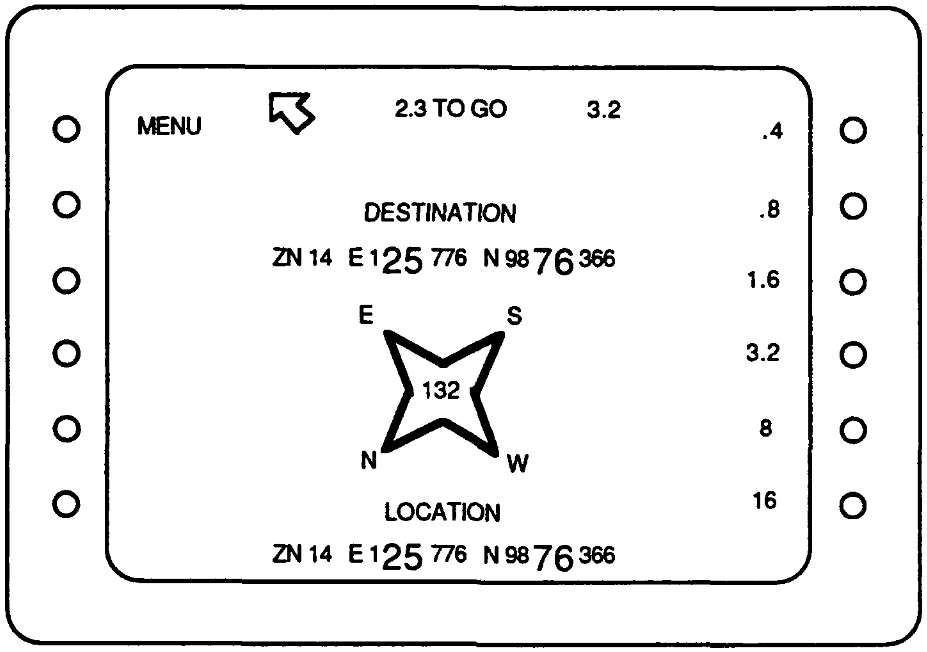

Figure 5. Compass Rose/Map Info Screen

Note: The format of the destination display reflects the format in which the destination was last entered (that is, UTM, street address, etc.).

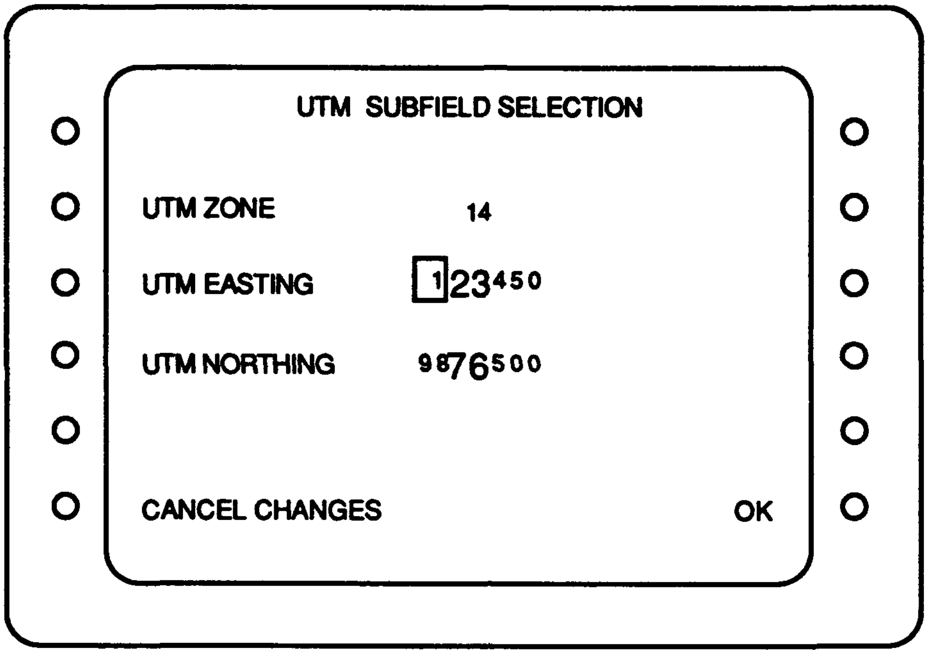

Figure 6. Select UTM Subfield Screen

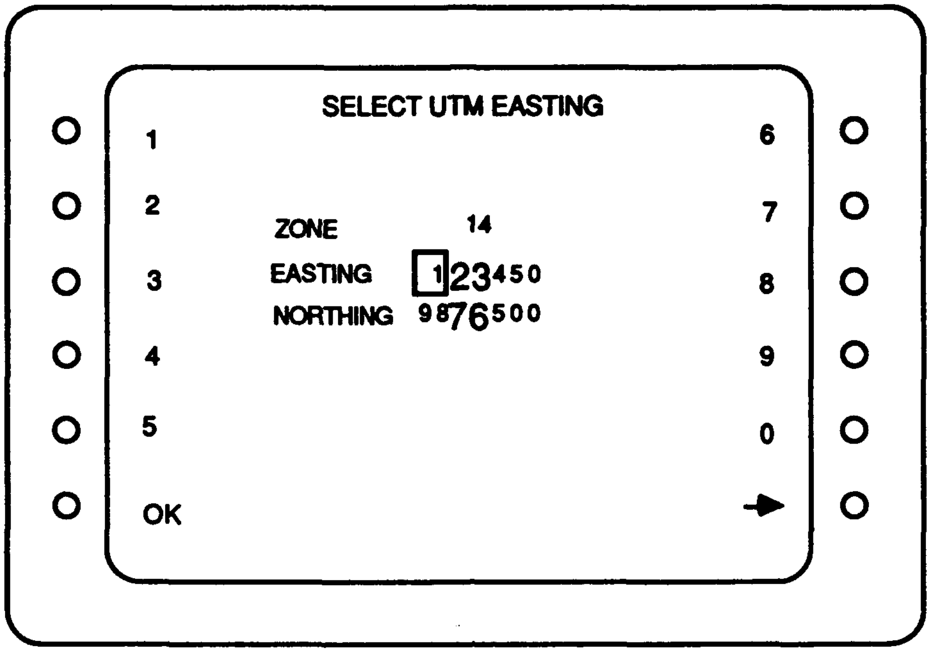

Figure 7. Select UTM Easting Screen

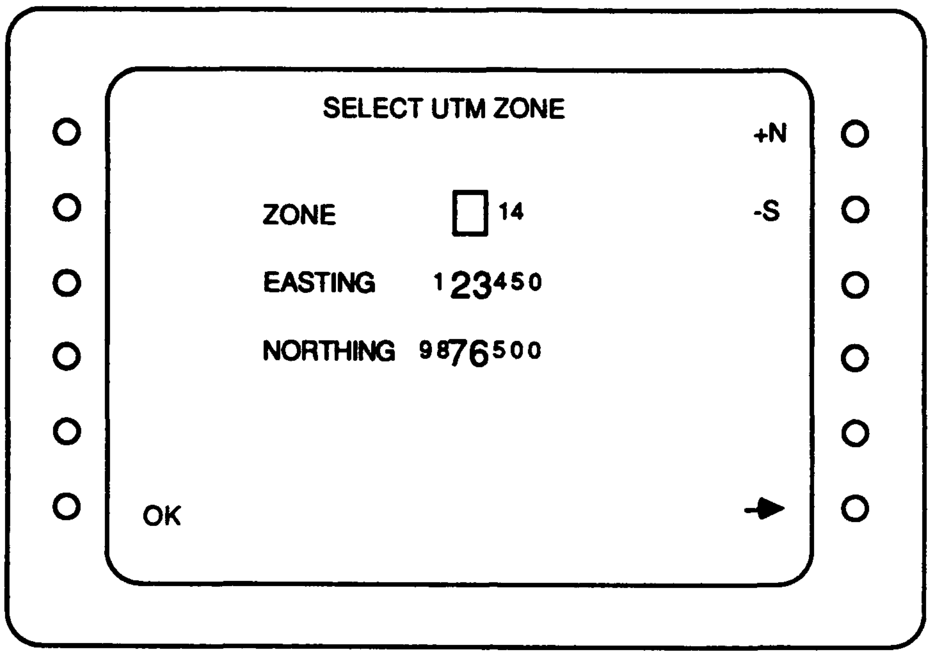

Figure 8. Select UTM Zone Screen

Verify Screen

The Verify Screen (see Figure 9) is used to forestall accidental changes in the map-matching status, since such changes can have an unwanted effect on the Navigator's vehicle location computations.

Using the Screens

In general, functions provided in the Navigator software are selected by pressing an appropriately labelled button on the current screen, which either generates an action or transfers the user to another screen where the desired action is selected and generated. This section describes the means by which the new functions are selected, in terms of both the buttons required and the screen sequences. Only the operations relevant to this contract are described here; pre-existing screens and sequences remain as detailed in the original Operating Instructions, except for the changes and new features described herein; Reference 4 details the operation of the UTM Navigator.

Initializing and Updating via UTM Coordinates

When the operator finds that the vehicle position differs from the Navigator's report, he can correct the Navigator's estimate by pressing the "REPOSITION" button on the main menu. To get to the main menu, press the "MENU" button from either the Map Screen or the Map Info Screen.

The Reposition Options Screen is identical to the Destination Options Screen (see Figure 4), except for its title, and has the same behavior, except that it adjusts the Navigator's estimate of the vehicle's position rather than setting a new destination. If the operator wishes to choose coordinates by some other means, he will follow the sequence given in the Operating Instructions. Otherwise, the operator would choose the "UTM Coordinates" Button from the Reposition Screen. This button causes the Navigator to display the Select UTM Subfield screen.

The Select UTM Subfield screen (see Figure 6) shows three field choices, zone, easting, and northing, and two actions, Cancel Changes and OK. The OK button

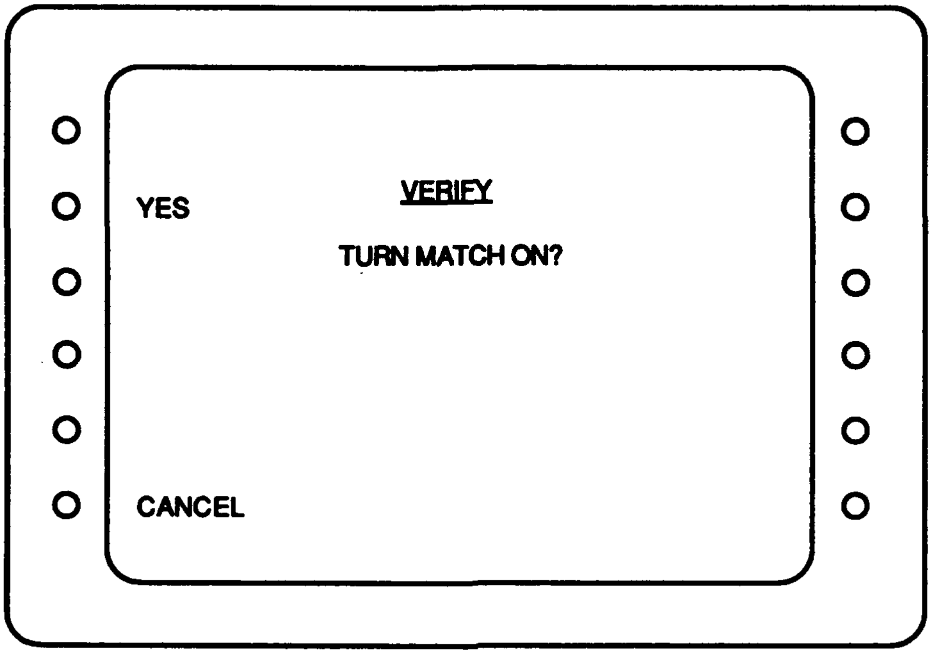

Figure 9. Verify Map-Match Change Screen

accepts the coordinates as they currently appear and returns to the Map Screen. The Cancel Changes button returns to the Map Screen without adopting any changes that the user might have entered. The other three buttons send the user to a data entry screen.

The data entry screens shown here are the Select UTM Easting Screen (see Figure 7) and the Select UTM Zone Screen (see Figure 8). The Northing Screen is nearly identical. The Navigator displays the Easting Screen in order to accept a new value of easting. This screen shows all ten digits, a right arrow, and an OK button. In the center are the current UTM coordinates, with a character highlighted by a box prompt. On the Easting Screen, this box is on one of the digits of the current easting, and cannot be moved to another field. To change the easting, the user pushes one of the digit buttons. The new digit replaces the old, and the cursor moves right to the next digit. If the cursor was on the last digit, it will move back to the first digit. If the user wants to keep the current digit unchanged, he can press the arrow button instead. This will move the cursor right one space. Again, the cursor will move from the last digit back to the first. If the user makes an error, he can press the arrow button until the cursor cycles back to the erroneous digit and then press the correct button.

When the value is satisfactory, the user presses the OK button and returns to the Select UTM Subfield Screen (see Figure 6). If another subfield is to be changed, the user will press the corresponding button and proceed as above to enter the new value. If the user wishes to abandon one of the Subfield screens, he can push the OK button, and then when the Reposition Screen appears, he can select Cancel Changes.

Please note that the UTM Zone Screen differs from the other two in one way. When the cursor cycles from the units digit, it goes to the space in front of the hundreds digit, and the screen displays "+N" and "-S" buttons instead of the ten digits (see Figure 8). This allows the user to enter a negative zone value to indicate latitude south of the Equator, and to change it back to north (with the plus).

Entering a Waypoint or Destination via UTM Coordinates

The procedure here is identical to that described above, except that the user pushes the Select Destination button on the Menu Screen, and the initial screen is called Destination Options. Of course, what is then selected is a destination instead of a vehicle position.

Outputting Position Coordinates and Heading to the Display

The Menu Screen (see Figure 2) and the Destination Options Screen (see Figure 4) show the destination. The Reposition Options Screen (not shown in this report, but virtually identical to Figure 4) shows the current position of the vehicle. The Map Info Screen (see Figure 5) shows both the current position and the current destination. The position of the vehicle is always shown in UTM on the Map Info Screen. However, all the remaining representations of a position depend on how it was last entered. Thus, when the Destination Options Screen is called up, it will show a UTM position if the last destination was entered as UTM. If it was last entered as a street address, however, the window will show it in that form, and the same with the other possible modes of entry. The destination position as shown on the Menu Screen and on the Map Info Screen behaves in the same manner. The Reposition Screen behaves the same way with regard to the last mode of repositioning.

Displaying the Distance and Heading to the Destination

The Navigator displays the distance and heading from the vehicle's current position to the destination on two screens. These are the Map Screen (see Figure 3) and the Map Info Screen (see Figure 5). The heading is displayed as the orientation of an arrow, and the distance as a decimal number. For showing UTM coordinates, the arrow reflects the grid heading, rather than the geographic heading, and the distance is expressed in kilometers.

Telling the Navigator when it is not Map Matching

The button for turning map matching on and off is found on the Map Screen. It is labelled with its action. If map matching is OFF, the button will show the letters MM, and pressing the button will bring up a Verify Screen (see Figure 9) which asks "Turn Match ON?". Pressing the YES button will then turn map matching ON. Pressing the CANCEL button will return to the Map Screen without changing the Map Matching status.

If map matching is ON, the button will show the letters MM with the familiar barred-circle logo for "NO". Pressing the button will bring up a Verify Screen (see Figure 9) which asks "Turn Match OFF?". Pressing the YES button will then turn map matching OFF. Pressing the CANCEL button will return to the Map Screen without changing the Map Matching status.

Coordinate Conversion Routines

The coordinate conversion is done by a logically straightforward process of computing first the power-series coefficients and then the power series itself, as presented on pp. 67-69 of Snyder (Reference 1), with some formulas for convergence, which Snyder does not cover, from UTM Grid (Reference 2). The difficult part is in using integer arithmetic, since each value in each computation must be individually normalized in order to meet the conflicting demands of retaining precision while avoiding arithmetic overflow.

Constants of the Spheroids

Each spheroid (currently only two are tabulated) is defined by only three values. In this implementation they are the major axis, the reciprocal of the flatness, and the central meridian scale factor. In addition, the data structure contains a selection number and an identifier (a text string naming the spheroid). The program uses only the Clarke 1866 ellipsoid.

In addition to the defining constants, there are many coefficients in the equations that depend on the spheroid but not otherwise on the coordinates being transformed. Examples include the eccentricity and several values derived from it, such as the coefficients in Snyder's Equations 3-21 (Reference 1, p. 68) and 8-19 (p. 69). These coefficients are computed at startup time; they will be recomputed later if the spheroid is ever changed (see the next section).

Spheroids/Ellipsoids

One problem not addressed in the contract or proposal is the fact that various parts of the world are best approximated by different spheroids, reflecting the fact that the earth is not a simple ellipsoid of revolution, but has higher harmonics. At the moment only one spheroid is needed because all of the current activity is confined to North America, which is best represented by the Clark 1866 ellipsoid.

Because the programmatic determination of the preferred spheroid is a lengthy computation requiring a considerable amount of stored data, and since under any conditions a vehicle will only rarely cross a spheroid boundary, Etak recommends using a menu selection to choose the active spheroid when it becomes time to implement a world-wide version of the program. All that is needed is a display of the available spheroids, listing the selection number and the descriptive identifying label of each. Selection can be accomplished by scrolling the cursor to a given line or by entering the selection number by the usual technique.

The cold-start default spheroid will be the Clarke 1866 spheroid, and the warm-start default will be the last one chosen.

Zone Choice and Validation

A user can enter vehicle or destination UTM coordinates that lie outside of the specified grid zone. For example, at a northing of 5653891, the smallest easting within the zone is 289504, yet the user might enter 281246. Additionally, if the current position is near the zone edge, the vehicle can easily cross into the next zone. The ambiguity is undesirable. The series approximations to the coordinate conversions lose accuracy rapidly outside the zone boundary, and the scale factor becomes excessive.

Zone Display Criteria

Normally, vehicle coordinates and destination coordinates will each be displayed in the zone in which the corresponding point lies. That is, the zone will be the one whose central meridian lies closest to the point in question.

When a pilot enters coordinates that are within four degrees of the zone center, the program will accept them, but subsequent display will show the point in its proper zone (and thus within three degrees of that zone's center). When coordinates outside the four-degree range are entered, the program will return to the coordinate entry screen with an error message.

When the vehicle enters a new zone, the coordinates will continue to be referred to the old zone until the vehicle is at least ten meters into the new zone. This is to prevent coordinates changing frequently in the event that the vehicle is skirting the zone edge.

Entry Validation

Since the UTM coordinates are immediately converted to geological coordinates on entry, validation consists of verifying that the longitude is within four degrees of the zone center and that the latitude is within the range 80°S to 84°N.

Accuracy Tests

For purposes of evaluating accuracy, the Etak transformation program was run on latitudes of 0°, 10°,... , 80°, as well as at latitudes of 44°59'59", 45°, and 45°00'01" (since the integer trig functions switch modes at 45°). At each latitude, longitudes of 0°, 1°, 2°, and 3° from the central meridian were selected. The actual angular coordinates were adjusted slightly, using a conversion program sent to us by the Army, so that all the UTM coordinates had exact integer values. All of the resulting easting and northing values were entered into the UTM-to-geo program, and the resulting geographic coordinates were compared to original coordinates from the Army program, and converted into distances. The UTM coordinates were treated similarly, except that, since the values are in whole meters, small errors are invisible outside of the debugging environment.

In the case of conversion to angular coordinates, the results were variable, with a tendency for the larger errors to be associated with latitudes nearer to 45 degrees and longitudes nearer to the zone edges. The errors in easting averaged 6 millimeters (mm) with a standard deviation of 11 mm. Twenty-one of the 48 values were zero and the maximum error was 46 mm. The errors in northing were larger, averaging 48 mm with a standard deviation of 34 mm. There were only ten zero errors in the sample, and the largest error was 94 mm. When the error was expressed as the Euclidean distance $(x^2 + y^2)^{1/2}$, there were nine zeros in 48 samples, and the maximum error was 96 mm. The mean total error was 50 mm with a standard deviation of 33 mm. See Table 1 for the breakdown by latitude.

Table 1. Coordinate Errors (mm)

Latitude	Average	Std Deviation
0 - 30	28	30
40 - 50	77	19
60 - 80	35	22

One integer northing value was off by a meter; its fractional part as shown in the Army program was 0.565 meters. Presumably, it was rounded in the wrong direction because, in effect, there was an error amounting to at least 65 mm in the wrong direction. This seems to be not serious.

In terms of consistency as shown by the difference between a starting value and the result of applying a transformation followed by its inverse, the Etak integer transformation and the Army program seem about on a par. Since the Etak program computes only integer values, this requires using the Army program to help show fractional meters, so that part of the test has to be viewed as qualitative. Of course, this accuracy is far more than is needed to support the tens-of-meters accuracy of the digital map and the Navigator.

Timing Tests

The conversion software was instrumented (in software) for measuring the execution time of the two conversions (UTM to geographic and vice versa), and the program was run on a PC-XT, which has the same CPU (Intel 8088®) and clock rate (4.77 MHz) as the Navigator's computer. The timings fell into four distinct groups, with little or no variation (beyond the granularity of the timing) within each group. These groups are: zero latitude and longitude, zero latitude and non-zero longitude, non-zero latitude and zero longitude, and both non-zero. The variation occurs mainly because some computations are bypassed when one or more operands are zero.

Table 2. Coordinate Conversion Times (msec)

	UTM to geo	geo to UTM
Both == 0	20	16
Lat == 0	25	20
Lon == 0	57	38
Both != 0	62	44

Note: Zero latitude means relative to the equator, i.e., when latitude is 0 degrees or northing is zero or ten million.
Zero longitude means relative to the central meridian of a zone; i.e., when longitude is an odd multiple of three degrees or easting is 500000.

Chapter 4

Etak Maps

Introduction

This chapter discusses the underlying structure of Etak's digital maps and its implications for possible future U.S. Army systems. The discussion concludes by defining the actual maps and features that were captured for the display on the Navigator screen for this contract.

There are many ways to represent map data digitally. Etak's internal map structure relies heavily on the mathematics of topology. The advantages of representing maps as such a mathematical database include the ability to:

- use it for intelligent data processing (for example, map matching, navigation, route planning)
- store data compactly and access data rapidly
- automate data integrity checking

These advantages are discussed in greater detail after an introduction to Etak's topologically structured map database and Etak's map capture process. Specific details of Etak's EtakMap format, access software, and map production process are proprietary. The material presented below is limited to a non-proprietary discussion of these items.

Etak's Digital MapBase Structure

Topology is the mathematics of connectivity. Its use in conjunction with cartography is more fully treated in Corbett, *Topological Principles in Cartography* (Reference 6) and White, *Technical Requirements for a Geographic System* (Reference 7).

For the surface of the earth, the world may be divided into points, lines, and areas. These are referred to as 0-, 1-, and 2-cells respectively. Two 0-cells can be connected by one or more 1-cells. Conversely, 1-cells are bounded by two 0-cells, one on each end. Similarly, a 2-cell is bounded by a set of 1-cells. This structure can be extended to higher dimensionality, but this is generally not needed for surface mapping.

These 0-, 1-, and 2-cell relations (the term "k-cell" is used when the dimension is arbitrary) define how map elements are connected. The connectivity, or topology, forms the necessary database structure, but does not in itself capture the geography (position) and features of the map. Each 0-cell has attached to it a set of coordinates.

Etak uses latitude and longitude for its internal coordinate representation. If a 1-cell (line) that connects two 0-cells cannot be accurately represented as a straight line, then a set of "shape points" is used to further refine the 1-cell. Shape points are useful for defining the geography (positional accuracy) but have no relation to the topology of the map. Figure 10 shows an example of a map depicted from 0-, 1- and 2-cells and shape points.

Each k-cell in the database can represent different data types. For example, a 1-cell may be a road, contour line, power line, shoreline, geopolitical boundary, and so forth. Different attributes can be associated with each k-cell of a different type. For example, a 1-cell representing a road may have attributes of name, priority, one/two way, and so forth. A 1-cell representing a power line may have attributes of height above ground, voltage, and so forth.

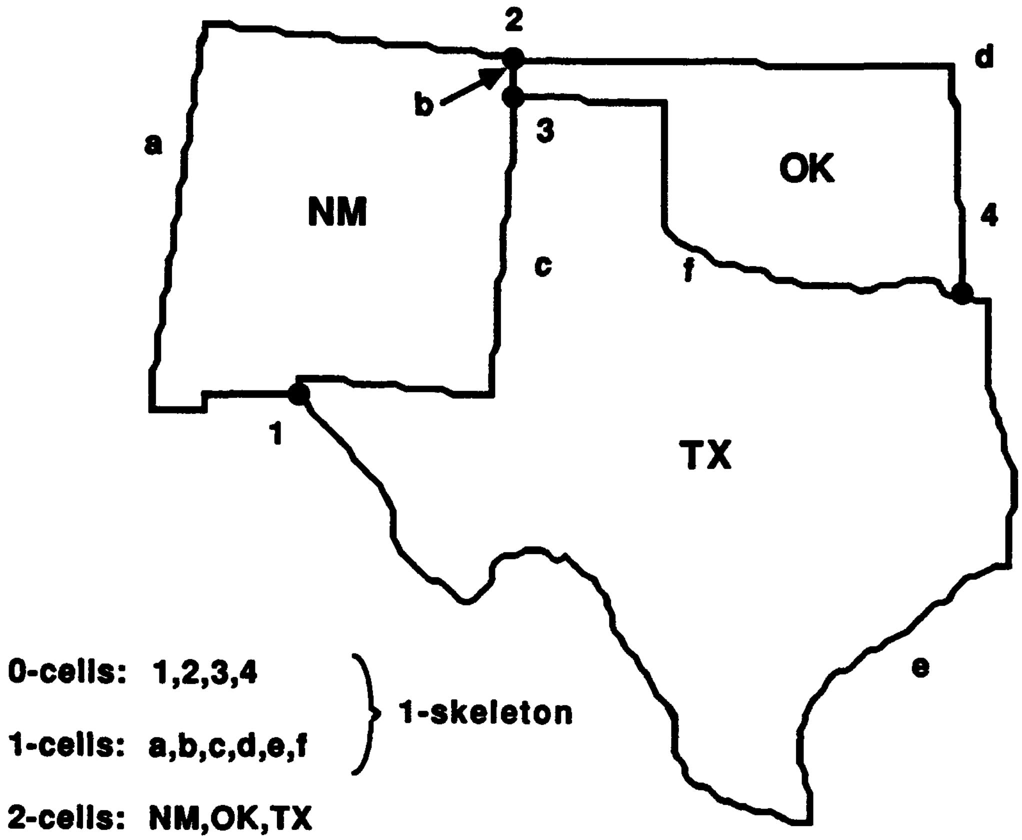

Figure 10. A Map in Topological Terms

Not all map attributes are necessarily tied to the topological structure of the map. A building may be a point of interest that lies within a 2-cell but is not connected to the k-cell structure. To add flexibility to maps, Etak has introduced the notion of landmarks (these are not compatible with the Navigator map formats). Landmarks enable Etak to capture arbitrary map information and display them in ways that are appropriate at different scales.

In summary, Etak stores its data topologically as a set of 0-, 1-, and 2-cells. Each k-cell is stored with an identification and the necessary pointers to capture its connectivity relations. Additional pointers allow us to attach attribute data to any cell or landmark. This structure enables Etak to store map data compactly and to retrieve it very efficiently. The benefits of such a map storage format are discussed later in this chapter.

Etak's Map Production Capability

Etak creates its map database using a proprietary semi-automatic data-capture process. While the specific details for the capturing process are not discussed, a general approach to Etak's data capture is described here.

Etak can capture map data from a wide variety of sources including paper maps, aerial photography, and machine readable maps such as the Census Bureau's DIME and TIGER files. In addition to these traditional cartographic resources, Etak can capture data from many other sources. For example, the U.S. Postal Service's ZIP + 4 files are used to update ZIP code information and help validate street names and addresses. Etak defines the process of merging data from different sources as *conflation.*

Under this contract, Etak is directed to only use the DMA 1:50,000 scale map sources provided. Because this test map is in the United States near Fort Hood, Texas, Etak could conflate many additional sources of data to provide a more current and complete EtakMap. For example, DIME and TIGER files could serve to add street names and address ranges to the road network. Aerial and satellite photographs could be used to add features that are newer than the DMA source maps. However, this is considered inappropriate because such additional data sources may not, in general, be available to the U.S. Army in other parts of the world.

In general, Etak accurately extracts features from the source material and places these features into our computer for processing into EtakMap format. To do this, Etak defines an extraction procedure which details the type of data to be extracted from the source. This is generally defined in terms of k-cells, shape points, attributes, and landmarks. For example, the extraction procedure would specify the capture of all road features.

Shape points were added to achieve Etak's nominal relative accuracy of 30 meters. Attributes of road features include road name when available and road priority. Etak defines a set of road priorities which range from dirt roads to superhighways. The specific data captured for Phase II of this contract are detailed later in this chapter. Once the procedure was established during Phase I a combination of digital and human intelligence was used to process the source material efficiently, format the data, and verify its topological consistency.

This procedure is flexible and can be used to capture much more information than can be effectively displayed on the Navigator screen. That screen is a monochromatic vector display with limited resolution. In contrast, high resolution raster scan color displays can be utilized to display maps that approach the quality of printed maps. This, coupled with the processing capability of digital maps, opens new and exciting possibilities to navigation and, more generally, Geographic Information Systems (GIS).

Capabilities of Topologically Structured Map Data Bases

Data Processing

Because each feature in the map is made up of a database element the computer has direct knowledge of all features. This structure differs from graphical representations of maps and enables sophisticated computer processing (not just image processing) to enhance map presentations. Examples of map processing done on the 8088-based Navigator include:

- heading up, moving map display
- scaling that selects which features to display and at what level of generalization to display them
- dynamic positioning of labels which remain principally upright even as the map is rotated
- labels that remain fixed in size as the map is scaled
- map matching which compares the heading and position of a sequence of dead reckoned points to the heading and position of the road network to improve navigation accuracy
- geocoding which quickly searches the database for locations associated with attribute data such as the street name and number of an address.

Map Access Performance

The above capabilities would be useless if they required the processing power of a mainframe or, conversely, took tens of minutes to accomplish on a microprocessor. Etak's proprietary topological structuring of map data enables a tenfold compaction of data and a hundred-fold improvement in access speed over traditional map formats. It is this performance enhancement that permits inexpensive, high-performance digital map capabilities.

To perform the processing functions outlined in the previous section, the map data must be structured in such a way that all pertinent data can be quickly read into cache memory from the mass storage device. The topological concepts described above form the building blocks for a map data base format that allows efficient searching through vast amounts of data to find the connected elements. The power of this technique was proven in the Navigator. Map data processing and displays described above were accomplished using an 8088 microprocessor and tape cassette mass storage device with average seek times of ten seconds.

Future Capabilities

Ruggedized compact disks, high resolution color monitors, and high-powered microprocessors exist today and are finding applications into next-generation Navigator-like products. With hardware limitations close to being removed and improvements being made to the functionality of Etak's map databases, it is useful to consider not only what maps in the near future might look like, but also what they would be capable of doing.

With the use of color and raster scan area fill, map presentations could be roughly equivalent to those of paper maps. Such presentations would eliminate problems encountered in the Navigator. For example, streams and lakes could appear blue-filled, and clearly differentiated from other lines which might represent a road. Landmarks and icons could be used to capture symbol data such as towers and bridges. These symbols would appear as icons and would be scale invariant. All of the features on current DMA maps could then be captured and presented unambiguously. Each class of icon could be prioritized to appear only in maps above a specified scale.

Contour lines and grids could be selected individually and presented with spacings depending on user preference. In general, the ability to select which data elements to present could be much more flexible, both under specific user commands and with more intelligent automatic feature selection. For example, elevation data would permit perspective terrain display from arbitrary viewpoints. Topological map access coupled with known hidden line and surface algorithms, particularly those operating in object space rather than display space, are likely to be very fast and, thus, possible to run in-vehicle on inexpensive hardware such as a version of the Navigator with a raster color display. For example, the Binary Search Partition Tree (BSP-tree) organizes objects in order for visible surface plotting. See Fuchs et al., *Near Real-Time Shaded Display of Rigid Objects* (Reference 8). Providing those objects in order may be more efficient and straightforward using topological retrieval.

The choices of map data processing will also improve. Drivers may wish to ask the map to select a best on-road or possibly off-road route. Dynamic data, such as the position of other military forces, may be displayed. Furthermore, integration with both on-board and remote systems becomes possible.

In summary, Etak has the capability to capture more date from the DMA maps than it is capable of unambiguously presenting on the Navigator. Newer formats are being developed to take advantage of more advanced hardware. The capabilities of next generation systems will have significant enhancements for U.S. Army applications.

Chapter 5

Map Data Capture

Introduction

During Phase I of this contract, Etak digitized part of the Fort Hood 1:50,000 DMA Map (Sheet 6446 III). A 5.8 x 8.25 mile area was chosen that contains all of the terrain features selected for capture. Included in the test area are the Fort Hood Main Post and a portion of Killeen, Texas.

For this project standard EtakMap data capture was augmented with special terrain feature capture. These features appear in digital map form on the Etak Navigator screen. Data captured from the Fort Hood DMA map is annotated for features and stored as attributes of 0-, 1-, or 2-cells, as defined earlier in Chapter 4.

The resulting map was used, in consultation with ETL personnel, to select the features to be captured for the Fort Hood tests, and to select the representations of these features.

Features Captured

The inclusion of the Fort Hood Main Post and a portion of Killeen in the test area provides a network of highways, streets, roads, and dirt roads, and the Fort Hood Military Reservation Boundary.

The features for capture beyond Etak's standard capture of the road network are shown in Table 3. All these features are captured in a class of non-navigable 1-cells, called "N-class" cells. These N-class features appear on the map but are not seen by the map matching algorithm. Thus, while a power-line may look like a road on the Navigator's monochrome screen, the map matching algorithm will not update the vehicle position estimate to a nearby powerline.

Features such as powerlines, telephone lines, railroads, streams, and creeks are naturally linear features and are topologically appropriate as 1-cell elements. Lakes and reservoirs would best be captured as 2-cells in a system which allowed for color area fill. Similarly, water towers, water holes, cemeteries, and buildings would best be captured as landmarks. In this way, an easily recognized icon could be placed on the map, unique for each class. These icons would be scale invariant so the direct use of landmarks would improve the appearance and utility of the map screen.

Etak resolved the Navigator's landmark limitations by using N-class 1-cells to capture all the terrain features. All 1-cells are scale proportioned so a symbol used to display a water tower changes size with scale. All terrain features have been given labels to distinguish them from roads. Some labels will probably not appear on the display because the corresponding 1-cells are too short for a label.

Etak analyzed the data to determine which scales would show the N-class features. Because the display is on a vector screen, the time required to display data depends on the amount of data such as the number of vectors and their total length. At smaller scales, the number of vectors increases, but the total length of lines may show little change. Smaller scales require a longer time to draw a whole screen. This then slows down the screen refresh rate, but at approximately 45 frames per second flickering becomes visible, and it gets worse rapidly at slower rates. Thus, there is a conflict between suppressing detail to obtain high repetition rates, and flickering at low repetition rates. At the half-mile scale, flicker can be kept tolerable in reasonably uncrowded areas, but this is not generally the case at the one-mile scale. As a result, Etak configured the N-class features to disappear at the one-mile scale. (These scales became the one- and two-kilometer scales when the Navigator was modified.)

All of the selected terrain features are included in the test area with two additions. Power transmission lines and telephone and electric service lines extend through the test area and appear as prominent features on the DMA map.

There are no water tower terrain features on the Fort Hood DMA map. However, some water tanks with obstruction symbols appear on the map. The obstruction symbol indicates a tower-like structure. All of the water tank features were captured.

Table 3 shows, for each N-class feature, the number captured and how each was labeled:

Table 3. N-Class Features Captured

Feature	Label	Count
Lake/Reservoir	LK/RES	9
Airfield	AF	1
Water Towers (Tanks)	WT	7
Streams/Creeks	STR/CRK	7
Cemeteries	CEM	2
Railroads	RR	1
Water Holes	WH	10
Power Transmission Line	PL	4
Telephone & Electric Service Line	TL	3

Map Data Capture Analysis

Data from the DMA map were captured using Etak's proprietary mapping process. The results were files containing 0-cells, 1-cells, 2-cells, names, and other associated attributes. Table 4 shows Etak's normal classification of linear features, while table 5 lists the features captured from the Fort Hood DMA map and indicates their representation.

Table 4. EtakMap Data Capture

Feature	Representation
Hard Surface Heavy Duty, 4 or More Lanes, Limited Access	1-cell
Hard Surface Heavy Duty, 2 Lanes	1-cell
Hard Surface Light Duty, 4 or More Lanes	1-cell
Hard Surface, Light Duty	1-cell
Improved, Light Duty	1-cell
Trails, Dirt Roads	1-cell
Low Speed Ramps	1-cell
High Speed Ramps	1-cell
Political Boundaries	1-cell

Table 5. Terrain Features

All features are N-class (non-navigable).

Feature	Representation
Lake/Reservoir	1-cell
Airfield	1-cell
Water Tower	1-cell (triangle)
Streams/Creeks	1-cell
Cemeteries	1-cell
Railroad	1-cell
Water Holes	1-cell (triangle)
Power Transmission Line	1-cell
Telephone & Electric Service Line	1-cell

Checking the Results

In order to validate the priorities assigned, four regions of the Fort Hood Test Area were observed on the Etak Navigator screen at the five different scales as tabulated in Table 6. During the investigative phase, scales in miles were used; for reference, Table 6 also shows the corresponding scales in kilometers (KM) as used on the modified Navigator. The observed visibility of features is tabulated in Table 7. The regions checked were the Machine Gun Range (Grid Square 1747), an interchange on U.S. 190 (Grid Square 1743), a section of the Main Post (Grid Square 1845), and the north end of the Robert Gray Airfield (Grid Square 1038).

Table 6. Observation Scales

Scale (miles)	Scale (KM)	Scale (fractional)
2	4	1:64,000
1	2	1:32,000
1/2	1	1:16,000
1/4	0.5	1:8,000
1/8	0.25	1:4,000

Table 7. Road Priority Levels

Level	Feature	Scale (miles) 2	1	1/2	1/4	1/8
1	Hard Surface Heavy Duty 4 or More Lanes, Limited Access	X	X	X	X	X
2	Hard Surface Heavy Duty, 2 Lanes	X	X	X	X	X
3	Hard Surface Light Duty, 4 or more Lanes		X	X	X	X
4	Hard Surface, Light Duty			X	X	X
5	Improved, Light Duty			X	X	X
6	Trails, Dirt Roads			X	X	X
7	Low Speed Ramps		X	X	X	X
8	High Speed Ramps		X	X	X	X
9	Political Boundaries					
10	N Features (terrain)			X	X	X

Conclusions

At the end of Phase I of the contract, the results of the digitization of the Fort Hood Test Section were discussed with members of the ETL staff, and demonstrated on both an Etak Navigator and an Etak Geocoder Workstation. At that meeting, final agreement was reached on which landmarks to capture, on the road and landmark priorities to use, and on the representations to be used for the landmarks. Approval was granted to Etak to digitize all four sheets of the DMA map of the Fort Hood, Texas area, to implement the design changes to the Navigator software, and to create an Etak tape cartridge incorporating the Fort Hood map and the modified software.

Chapter 6

Digital Map Testing

Introduction

One objective under this contract was to determine the positional accuracy of the digital map as derived from the DMA source material, in accordance with the ideas in Chapter 5. After having made the complete map of the Fort Hood area and just before the scheduled testing in Fort Hood, Etak's staff noted that a mapping error had shifted the map positions of much of the data on the EtakMap cassette that had been prepared for delivery. Since there was not sufficient time for creating a new map, the decision was made to proceed with the tests as planned. The major effects of these mapping errors were to prevent accurate calibration of the installed Navigator, and to cause map-matching errors during the test runs. On return to Etak, steps were taken to correct the map, including redigitizing the four DMA map sheets.

Etak performed extensive testing of the redigitized maps before the digitized maps were made into an Etak tape cassette for ETL, followed by further tests on the cassette. This chapter discusses the error analysis of the new map. In this discussion, the term "digital map" refers to a presentation on a computer screen of the results of digitizing the source material, and implies appropriate software to select points of the map and to read out their longitude and latitude (the necessary conversions to UTM were done separately with the Etak conversion software that was provided for this contract; see Appendix 1). The term "EtakMap" refers to the map on the Etak data cassette as used in a Navigator. The positional data from the Fort Hood EtakMap were obtained directly from the modified Navigator by using the Navigator's positioning arrows to place the car cursor on each test intersection and then reading the UTM coordinates from the Navigator's Information (Compass Rose) Screen.

Baseline Points

In order to validate the accuracy of the digitization process for the DMA 1:50,000 scale maps, a set of points was chosen for testing. On each of the four DMA map sheets making up the Fort Hood data base, sixteen points were chosen, as shown in Figures 11 and 12. On each chart, these sixteen points are at or near the intersections of four vertical strips and four horizontal strips. In each direction, two strips were chosen near each edge of the chart and two others roughly a third of the way in, so as to scatter the points fairly evenly over the chart. Road intersections were then chosen near each of the sixteen intersections. Each point was required to be completely boxed in by UTM grid lines, so that its UTM coordinates could be interpolated with the aid of a precision glass rule. Road intersections were used because their positions are readily identified on all the maps: the DMA paper maps, the corresponding separates, Etak's digital map, and the EtakMap cassette.

Figure 11 shows a representation of the four map sections. The vertical and horizontal lines represent the target positions for points to measure. They are tilted because they are parallel to the UTM grid lines. In several instances, there are jogs at the junctures between sheets where the continuation of certain lines would intersect the wrong edges of the adjacent sheets.

Figure 12 shows part of the coordinate grid for Purmela in an enlarged view. The vertical and horizontal strips are shown as pairs of UTM grid lines. The chosen road intersections are pictured as symbols representing a few meters of road on and near the intersections, and all other details are suppressed for clarity. The

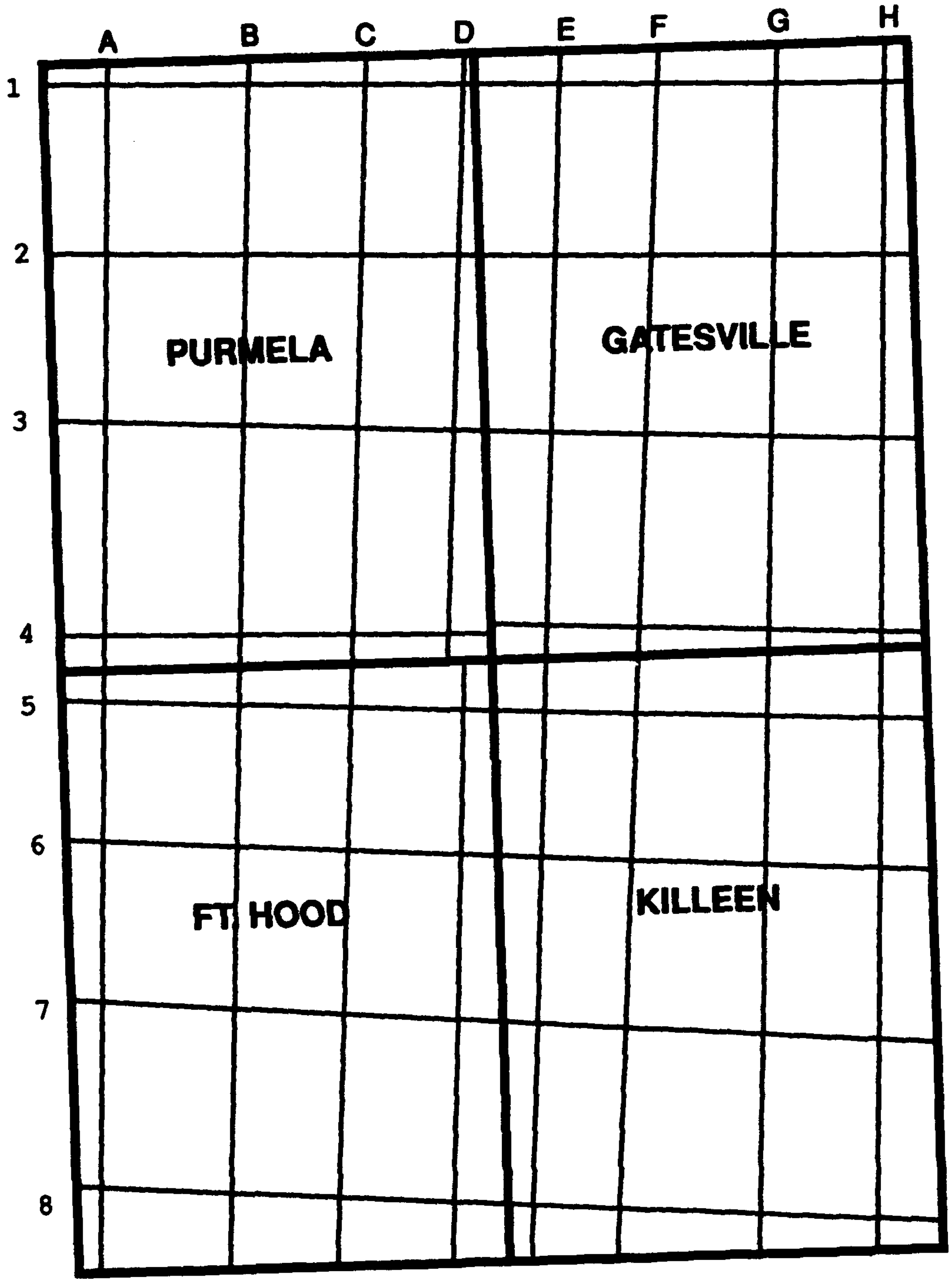

Figure 11. Test Point Targets for Map Verification

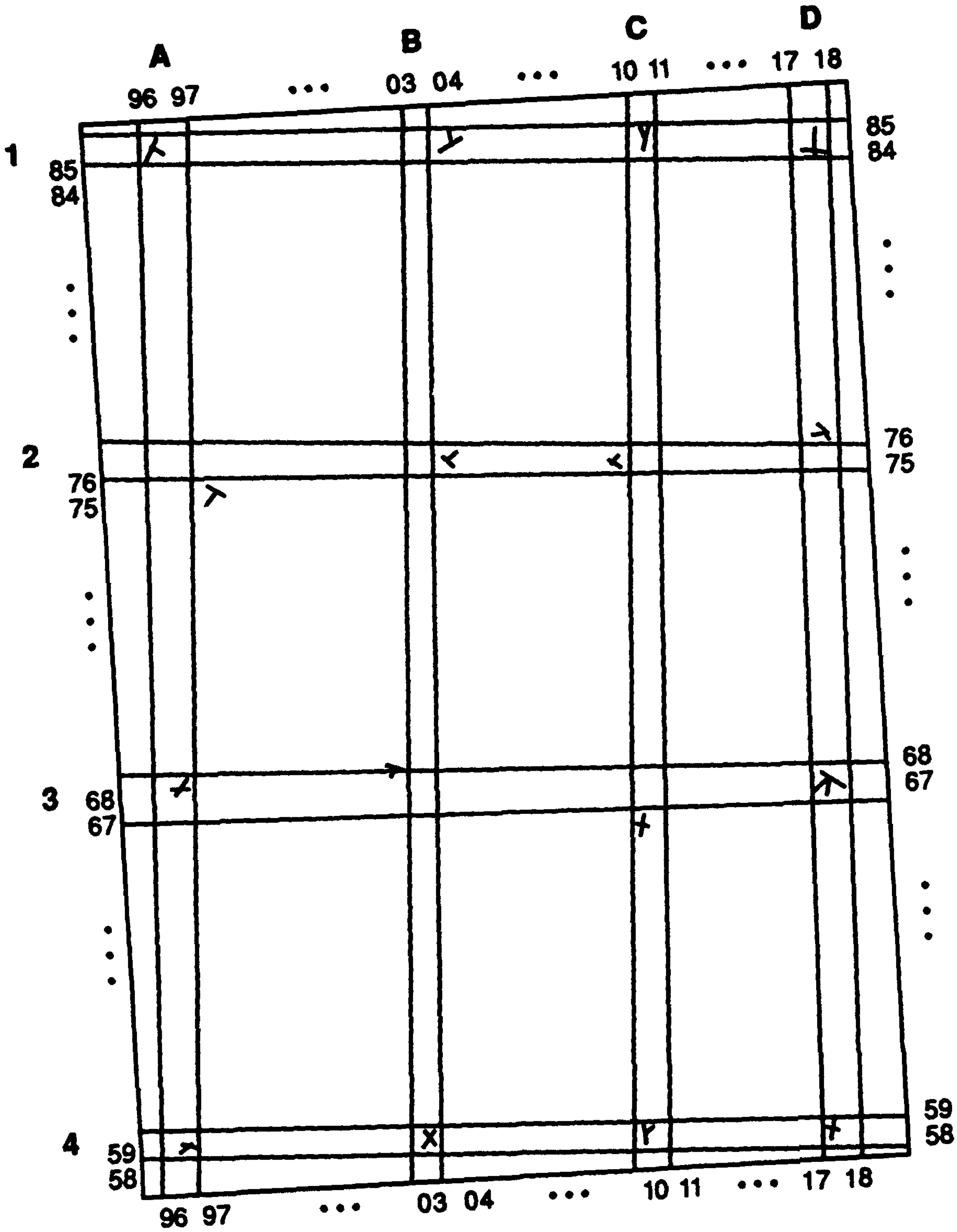

Figure 12. Test Point Targets on Purmela Section

intersections are not always in the target squares, because there was not always an intersection available, especially in the more rural areas.

In addition to the 64 points described above, the 22 points used as navigation checkpoints during the April 1990 tests were also measured; these points are all on the Killeen map sheet.

Baseline Data

First, in order to provide the baseline of "map truth," the test point UTM coordinates were read - using the precision glass rule - from the Mylar separates provided by ETL. The glass rule is calibrated in both English and metric units, so each point was measured twice (on different measurement runs) and the resulting interpolated UTM values were compared. Any points in which the resulting coordinates differed by more than a few meters were redone. Since the rulings are not directly commensurate (they are marked in intervals of 0.1 mm and 0.005 in), it was felt that the operator would make different kinds of errors on the two measurements, and that errors would therefore be unlikely to escape notice. This is also why the runs with the two scales were separated in time.

For the 86 points, the eastings derived from the metric measurements all agreed with the corresponding English measurements within 7.52 meters. Eighty-two values agreed within five meters, and 75 within three meters. The northings agreed within 8.09 meters; 84 values were within five meters and 75 within three. The UTM coordinates adopted as map truth for each point were the average of the two measured eastings and the average of the two measured northings.

Digital Map Tests

The baseline data were copied to spreadsheets for data comparison, one for each of the four map sheets used (Purmela, Gatesville, Fort Hood, and Killeen). The spreadsheets were set up to compute the differences in easting and northing between each digitized point and the corresponding map truth, as well as the magnitude of the error (square root of easting squared plus northing squared). In addition, summary data (mean and standard deviation) of the errors on the sheet were provided.

As the four map sheets for the Fort Hood area were digitized, the corresponding digital maps were tested immediately in order to provide early warning in the event of problems with the redigitization. The results of the tests on the digital map are shown in Tables 8 through 11, corresponding to the Purmela, Gatesville, Fort Hood, and Killeen map sheets. The 64 chosen intersections are labelled with a letter-number pair spreadsheet style: A through H stand for columns from west to east, and 1 through 8 for rows from north to south. Thus the four map sheets contain points numbered A1 (NW corner of Purmela) through H8 (SE corner of Killeen). See Figures 11 and 12 for examples of these labels. Each road intersection is labelled by the letter above it and the number to its left. For each point, the errors in easting and in northing are computed as digitized value minus map truth. The Euclidean distance is also shown. The maximum, minimum, mean, and standard deviation over the sixteen samples are shown for each sheet. Finally, the same data are shown in Table 12 for the 22 points on the two test roads, as tested in April.

Two of the sheets, Gatesville and Fort Hood, looked excellent, with average errors of only a few meters and most errors under ten meters. Killeen showed only small errors on the points at the top and bottom of the sheet, but errors of about 20 meters on the two middle rows. The Purmela sheet looked much worse, though, with errors around 30 meters everywhere. Investigation could uncover no human, procedural or computational explanation for these problems. However, the mapping staff felt that the maps themselves might be in error. It should be noted that these errors, although large, still fall within the desired accuracy and well within the National Map Accuracy Standard (Reference 5).

The raw data for these tests are included for reference in Appendix 3, Tables 43 through 47, pages 196 through 198.

Source Material Validation

In order to investigate the possibility of problems with the source material, another round of measurements was made on the problem maps of Killeen and Purmela. It was felt that the UTM grid lines themselves could be misplaced on

Table 8. Digitization Errors (meters) - Purmela

Error in:	Easting	Northing	Magnitude
A1	4	-16	16.5
B1	20	-21	29.0
C1	13	-28	30.9
D1	20	-25	32.0
A2	14	-24	27.8
B2	7	-34	34.7
C2	7	-29	29.8
D2	25	-31	39.8
A3	10	-33	34.5
B3	9	-38	39.1
C3	20	-31	36.9
D3	13	-30	32.7
A4	34	-10	35.4
B4	6	-27	27.7
C4	8	-32	33.0
D4	0	-28	28.0
Max	34	-10	40.0
Min	0	-38	16.0
Mean	13.1	-27.3	31.7
Std	8.5	6.8	5.4

Table 9. Digitization Errors (meters) - Gatesville

Error in:	Easting	Northing	Magnitude
E1	0	-3	3.0
F1	-9	-7	11.4
G1	-12	-8	14.4
H1	3	-3	4.2
E2	7	-12	13.9
F2	19	-9	21.0
G2	-5	-13	13.9
H2	4	-22	22.4
E3	3	-15	15.3
F3	5	-18	18.7
G3	-1	-11	11.0
H3	0	-23	23.0
E4	1	11	11.0
F4	3	3	4.2
G4	-1	-5	5.1
H4	9	7	11.4
Max	19	11	23.0
Min	-12	-23	3.0
Mean	1.6	-8.0	12.8
Std	6.9	9.3	6.2

Table 10. Digitization Errors (meters) - Fort Hood

Error in:	Easting	Northing	Magnitude
A5	-17	15	22.7
B5	-12	-10	15.6
C5	1	11	11.0
D5	-19	1	19.0
A6	-10	4	10.8
B6	-12	-20	23.3
C6	-4	2	4.5
D6	-16	-18	24.1
A7	-7	2	7.3
B7	-2	-1	2.2
C7	14	3	14.3
D7	7	7	9.9
A8	-33	1	33.0
B8	-16	-5	16.8
C8	1	-13	13.0
D8	-11	-13	17.0
Max	14	15	33.0
Min	-33	-20	2.2
Mean	-8.5	-2.1	15.3
Std	11.0	9.8	7.7

Table 11. Digitization Errors (meters) - Killeen

Error in:	Easting	Northing	Magnitude
E5	13	-16	20.6
F5	3	0	3.0
G5	3	-5	5.8
H5	2	0	2.0
E6	23	-1	23.0
F6	10	-17	19.7
G6	-3	-16	16.3
H6	MISSING	MISSING	MISSING
E7	3	-14	14.3
F7	7	-16	17.5
G7	-9	-34	35.2
H7	3	-22	22.2
E8	-6	0	6.0
F8	5	-8	9.4
G8	5	-18	18.7
H8	-5	1	5.1
Max	23	1	35.2
Min	-9	-34	2.0
Mean	3.6	-11.1	14.6
Std	7.7	9.9	9.0

Table 12. Digitization Errors (meters) - Test Roads

	Point	Easting	Northing	Magnitude
E/W Run	2	7	-16	17.5
	3	MISSING	MISSING	MISSING
	4	-5	-32	32.4
	5	4	-35	35.2
	6	20	-17	26.2
	7	5	-16	16.8
	8	26	-23	34.7
	9	5	-19	19.6
	10	-2	-25	25.1
	11	-1	-24	24.0
	12	3	-24	24.2
	13	3	-30	30.1
	14	-8	-16	17.9
	Max	26	-16	35.2
	Min	-8	-35	16.8
	Mean	4.8	-23.1	25.3
	Std	9.3	6.3	6.4
N/S Run	1	2	-16	16.1
	2	-1	-14	14.0
	2A	-2	-19	19.1
	3	1	-25	25.0
	4	0	-20	20.0
	5	15	-17	22.7
	6	6	-22	22.8
	6A	8	-19	20.6
	7A	2	-9	9.2
	8	8	-5	9.4
	Max	15	-5	25.0
	Min	-2	-25	9.2
	Mean	3.9	-16.6	17.9
	Std	5.0	5.7	5.3

the map sheets. To test this hypothesis, sixteen points were chosen exactly at intersections of UTM grid lines (using the same lines as before - see Figures 11 and 12), and their longitudes and latitudes were derived from the Mylar separates, using the glass rule to measure and interpolate from the borders of the maps. The validity of the approach was tested by also interpolating the coordinates of the tick marks along the borders, to see whether the procedure gave correct coordinates for the marks. It was discovered that the Killeen sheet UTM grids were quite accurate - only a few meters off, which is consistent with the observed small normal errors in measurements. However, the Purmela sheet showed a consistent bias in the placement of the UTM grid, roughly twenty meters, and in the same direction as the digitizing errors. Thus it was concluded that, after taking into account the source errors, the accuracy of digitization of the Purmela map was roughly ten meters.

In the process of capturing the geographic coordinates of the UTM lines on the Killeen map, it was noticed that the grid lines and other features on the northern part of the Mylar separates were double-exposed, and an error in choosing which line to read could easily create errors of several tens of meters. Also, in some cases the two exposures were of very different intensity, with the correct line being very faint, sometimes almost invisible. It is believed that this double-exposure problem had much to do with the mapping problems of April and explains the poorer results of the Killeen map.

The results of the test on the UTM grids are shown in Table 13 and Figures 13 and 14 for the Purmela map sheet. The sixteen chosen points are labelled by the same spreadsheet-like scheme as the nearby road intersections mentioned above. Table 13 shows the errors in the UTM grid for the Purmela map sheet using the method described above for Table 8, corresponding to points A1 thru D4. Figure 13 shows a scatter diagram for the data of Table 13; that is, each point is plotted at x = (easting error) and y = (northing error). Figure 14 is a similar scatter diagram for the errors in the road intersections of Table 8; these two diagrams give a graphic indication of where most of the coordinate error

Table 13. UTM Errors (meters) - Purmela

Error in:	Easting	Northing	Magnitude
A1	3	-8	8.5
B1	7	-14	15.7
C1	10	-19	21.5
D1	20	-17	26.2
A2	3	-5	5.8
B2	7	-12	13.9
C2	13	-18	22.2
D2	22	-18	28.4
A3	-1	-5	5.1
B3	5	-11	12.1
C3	7	-12	13.9
D3	13	-6	14.3
A4	-1	-20	20.0
B4	1	-25	25.0
C4	6	-28	28.6
D4	9	-24	25.6
Max	22	-5	28.6
Min	-1	-28	5.1
Mean	9.1	-15.0	19.4
Std	7.8	8.1	8.5

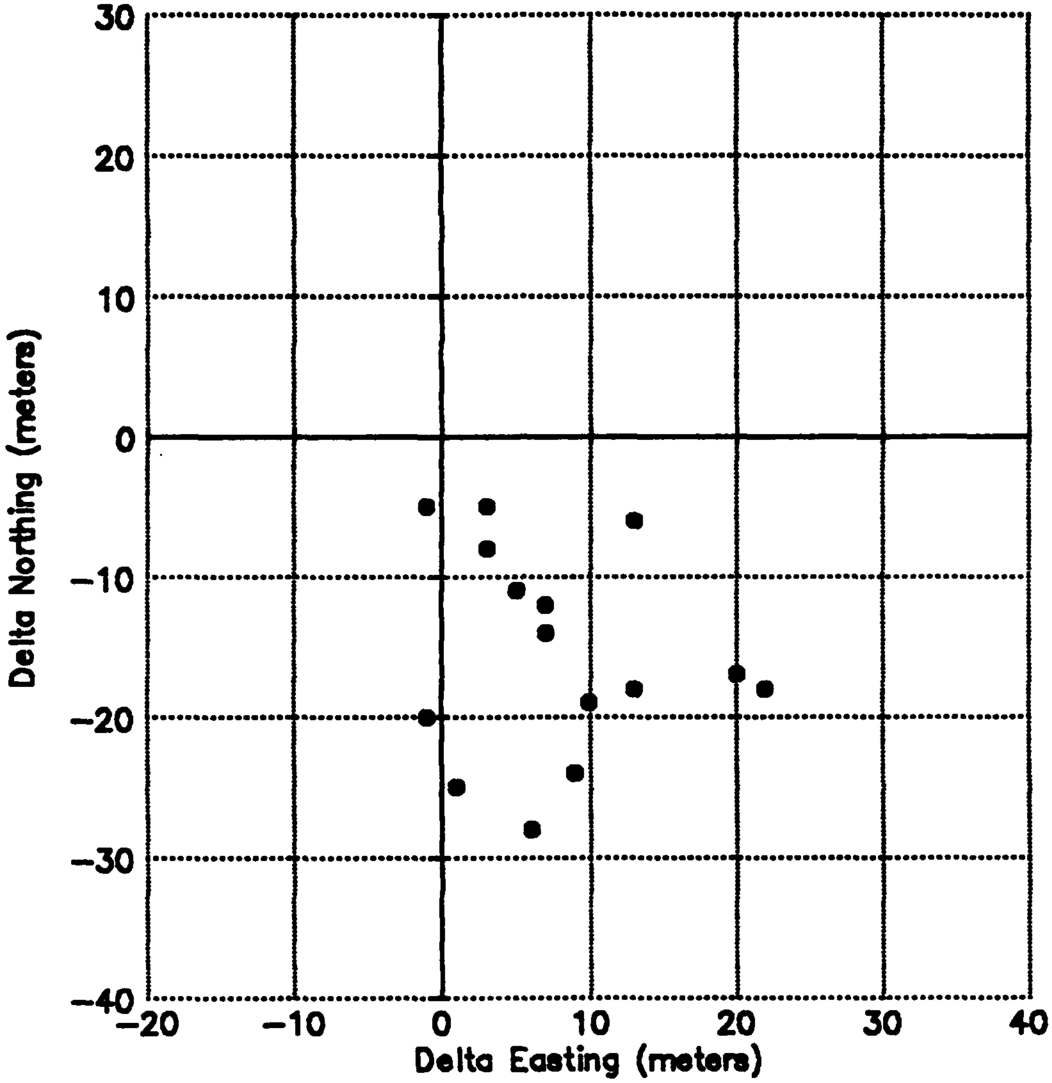

Figure 13. UTM Errors - Purmela

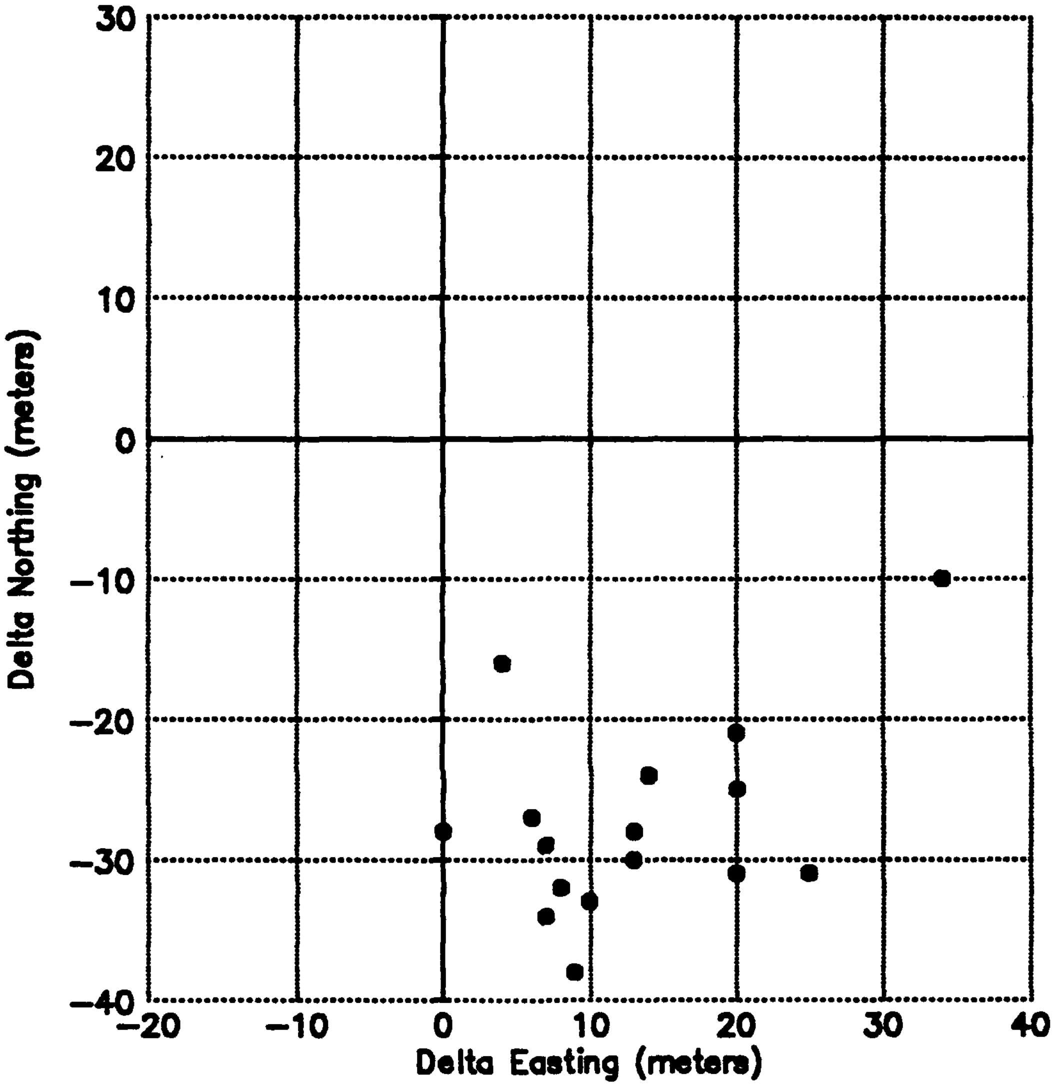

Figure 14. Digitization Errors - Purmela

Table 14. UTM Errors (meters) - Killeen

Error in:	Easting	Northing	Magnitude
E5	-4	5	6.4
F5	0	3	3.0
G5	-1	1	1.4
H5	-1	7	7.1
E6	6	7	9.2
F6	3	6	6.7
G6	-2	30	30.1
H6	-7	10	12.2
E7	1	-1	1.4
F7	-1	-3	3.2
G7	-6	-4	7.2
H7	-9	-2	9.2
E8	-5	-3	5.8
F8	-3	-1	3.2
G8	-2	-3	3.6
H8	-4	-1	4.1
Max	6	30	30.1
Min	-9	-4	1.4
Mean	-1.7	4.9	9.0
Std	4.6	10.9	8.8

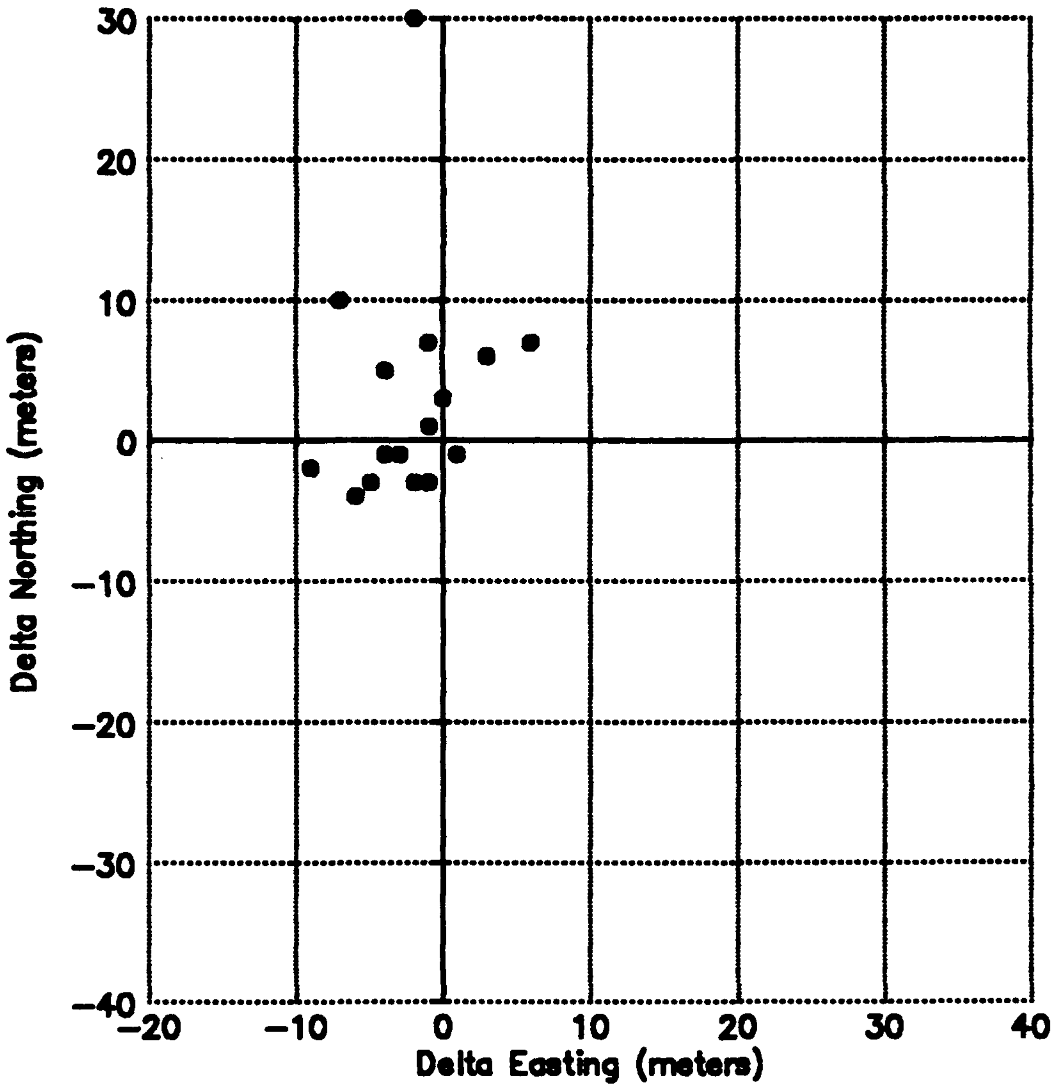

Figure 15. UTM Errors - Killeen

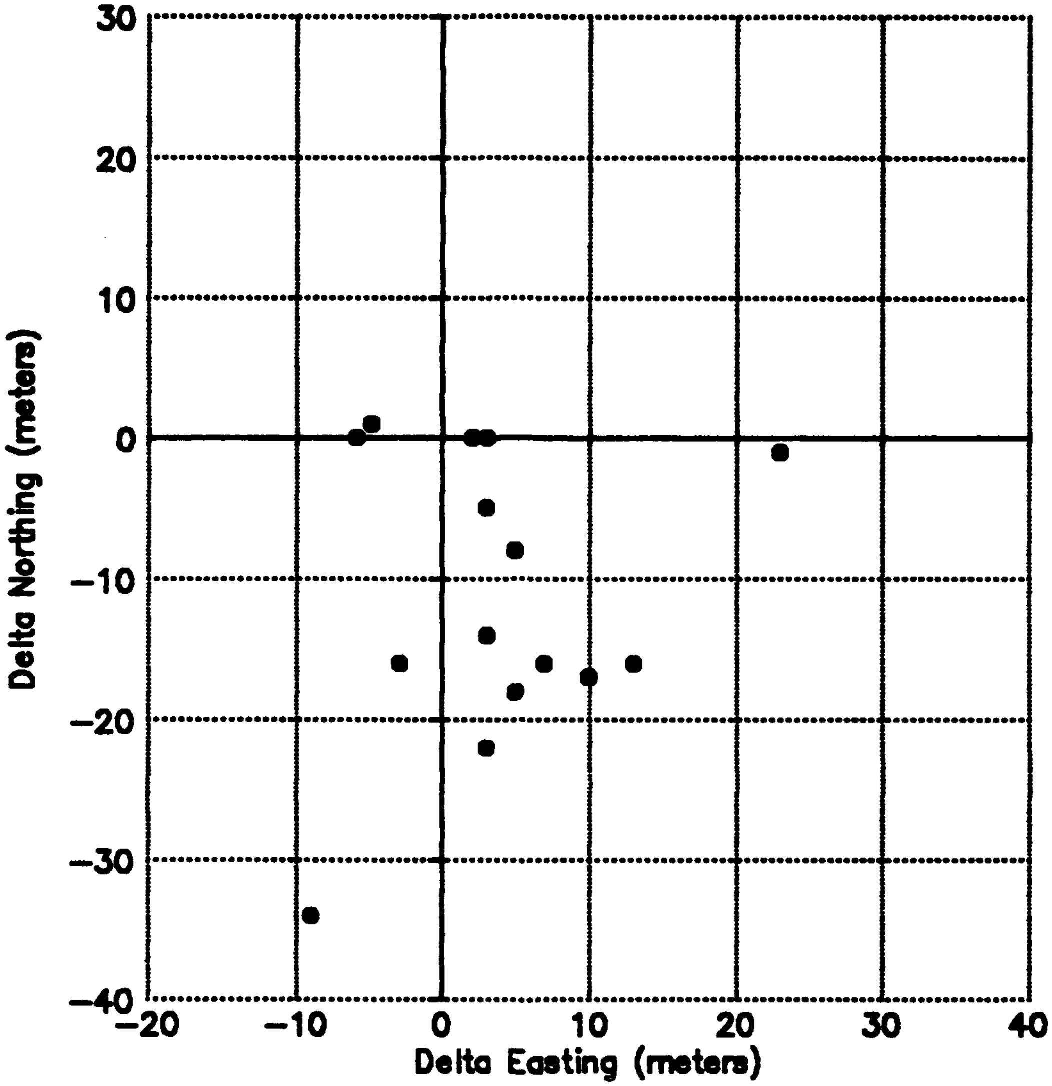

Figure 16. Digitization Errors - Killeen

comes from. Table 14 and Figures 15 and 16 give the same analysis for the Killeen map sheet, corresponding to points E5 through H8.

The corresponding sample data are given for reference in Appendix 3, Tables 48 and 49.

EtakMap Tests

As soon as the final EtakMap was available, another set of spreadsheets was made. These spreadsheets used the same map truth data, and were used to calculate the errors in the data as it appeared on the Navigator itself. Positional data for the test of Fort Hood EtakMap were obtained directly from the modified Navigator by using the Navigator's positioning arrows to place the car cursor on each desired intersection and then by reading the UTM coordinates from the Navigator's Information (Compass Rose) Screen. It was seen that the Navigator's EtakMap was accurate, so the cassette was accepted, and two copies of it were taken to Fort Hood for the June tests. The data from the Navigator screen are displayed in Tables 15 through 19, in the same manner as the data of Tables 8 through 12 are displayed. These tables can thus be compared directly to show that the EtakMap of the Fort Hood area corresponds well to the digital map created from the DMA maps, while giving an idea of the net effect of slight digital adjustments in creating the EtakMap and of the inevitable slight inconsistencies in reading the data from the Navigator.

The raw data for the Navigator EtakMap tests are tabulated for reference in Tables 50 through 54, pages 201 through 203 of Appendix 3.

Summary

In summary, extensive accuracy testing of the digitized maps measured an absolute error (averaging over the 86 points) of 18 m (under 15 m if Purmela is excluded). Inconsistencies observed on the Purmela map sheet were responsible for up to 18 m of the 26 m error found on that sheet.

Table 15. EtakMap Errors (meters) - Purmela

Error in:	Easting	Northing	Magnitude
A1	9	-11	14.2
B1	20	-19	27.6
C1	12	-24	26.8
D1	13	-24	27.3
A2	10	-24	26.0
B2	8	-34	34.9
C2	3	-31	31.1
D2	15	-35	38.1
A3	9	-34	35.2
B3	10	-36	37.4
C3	19	-31	36.4
D3	14	-24	27.8
A4	38	-11	39.6
B4	7	-23	24.0
C4	9	-29	30.4
D4	3	-23	23.2
Max	38	-11	40
Min	3	-36	14
Mean	12.4	-25.8	30.0
Std	8.1	7.5	6.5

Table 16. EtakMap Errors (meters) - Gatesville

Error in:	Easting	Northing	Magnitude
E1	-6	5	7.8
F1	-8	-2	8.2
G1	-7	-8	10.6
H1	4	-3	5.0
E2	14	-10	17.2
F2	19	-5	19.6
G2	-4	-10	10.8
H2	6	-22	22.8
E3	5	-13	13.9
F3	4	-17	17.5
G3	-1	-7	7.1
H3	1	-17	17.0
E4	-3	13	13.3
F4	-1	7	7.1
G4	3	-3	4.2
H4	8	5	9.4
Max	19	13	22.8
Min	-8	-22	4.2
Mean	2.1	-5.4	12.0
Std	7.2	9.3	5.4

Table 17. EtakMap Errors (meters) - Fort Hood

Error in:	Easting	Northing	Magnitude
A5	-17	22	27.8
B5	-10	-5	11.2
C5	2	8	8.2
D5	-17	4	17.5
A6	-4	5	6.4
B6	-7	-16	17.5
C6	-9	2	9.2
D6	-15	-13	19.8
A7	-4	1	4.1
B7	0	5	5.0
C7	15	6	16.2
D7	1	-1	1.4
A8	-31	4	31.3
B8	-16	-1	16.0
C8	1	-1	1.4
D8	-8	-7	10.6
Max	15	22	31.3
Min	-31	-16	1.4
Mean	-7.4	0.8	12.7
Std	10.3	8.5	8.5

Table 18. EtakMap Errors (meters) - Killeen

Error in:	Easting	Northing	Magnitude
E5	12	-12	17.0
F5	1	2	2.2
G5	3	2	3.6
H5	2	4	4.5
E6	21	0	21.0
F6	12	-12	17.0
G6	-9	-13	15.8
H6	MISSING	MISSING	MISSING
E7	0	-13	13.0
F7	5	-17	17.7
G7	3	-21	21.2
H7	5	-13	13.9
E8	-7	-1	7.1
F8	3	1	3.2
G8	4	-18	18.4
H8	-9	7	11.4
Max	21	7	21.2
Min	-9	-21	2.2
Mean	3.1	-6.9	12.5
Std	7.7	9.0	6.5

Table 19. EtakMap Errors (meters) - Test Roads

	Error in:	Easting	Northing	Magnitude
E/W Run	2	1	-16	16.0
	3	3	-23	23.2
	4	-6	-25	25.7
	5	0	-26	26.0
	6	23	-16	28.0
	7	3	-12	12.4
	8	26	-21	33.4
	9	4	-17	17.5
	10	1	-25	25.0
	11	1	-23	23.0
	12	-5	-30	30.4
	13	4	-28	28.3
	14	-6	-13	14.3
	Max	26	-12	33.4
	Min	-6	-30	12.4
	Mean	3.8	-21.2	23.3
	Std	9.5	5.6	6.2
N/S Run	1	1	-12	12.0
	2	0	-12	12.0
	2A	0	-14	14.0
	3	2	-18	18.1
	4	4	-15	15.5
	5	11	-15	18.6
	6	5	-19	19.6
	6A	7	-18	19.3
	7A	0	-5	5.0
	8	6	-1	6.1
	Max	11	-1	19.6
	Min	0	-19	5.0
	Mean	3.6	-12.9	14.0
	Std	3.5	5.5	5.0

Chapter 7

Navigation Tests

Test Configuration

The objective of the navigation tests was to determine the navigation performance at Fort Hood using the Etak Navigator and EtakMaps created entirely from DMA 1:50,000 scale maps.

After completion of the software modifications discussed in Chapter 3 and local testing, the modified Navigator software was combined with the Fort Hood database for final testing.

The navigation test phase involved installing a Navigator in a CUCV (as shown in Figure 17) at Fort Hood, Texas, and driving several predefined routes. Two routes were selected, one predominantly north-south and one predominantly east-west. Each route was traversed in both directions. A number of runs was conducted, about half with map matching enabled and the rest with map matching disabled. Position data were collected at identifiable check points along the way, consisting of points marked on the road surface near intersections. The map of Figure 18 shows the section of the Fort Hood digital map which includes the two test routes. The numbered circles indicate the data-collection check points. The path numbered 1 through 14 marks the East-West routes and the path numbered 1 through 8 marks the north-south routes. The data at the bottom indicate the map center in latitude and longitude and the scale (which is the distance corresponding to the height of the screen). Figure 19 is a

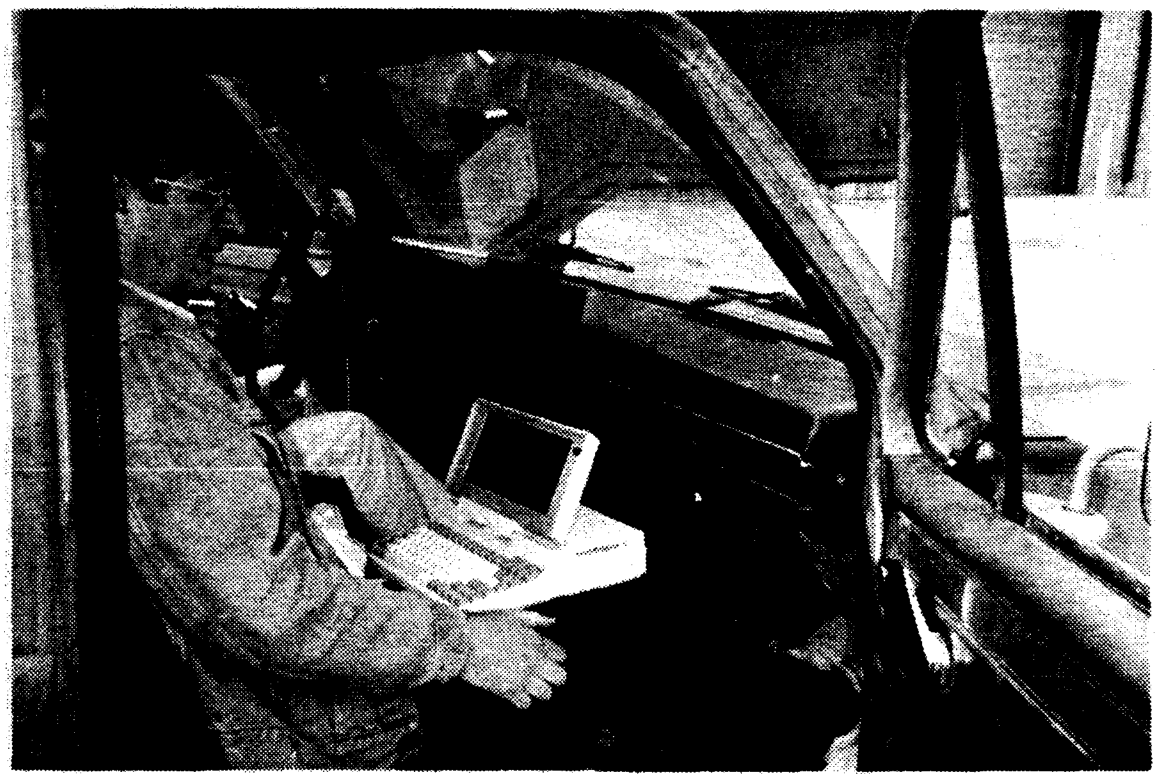

Figure 17. Navigator Installed in CUCV

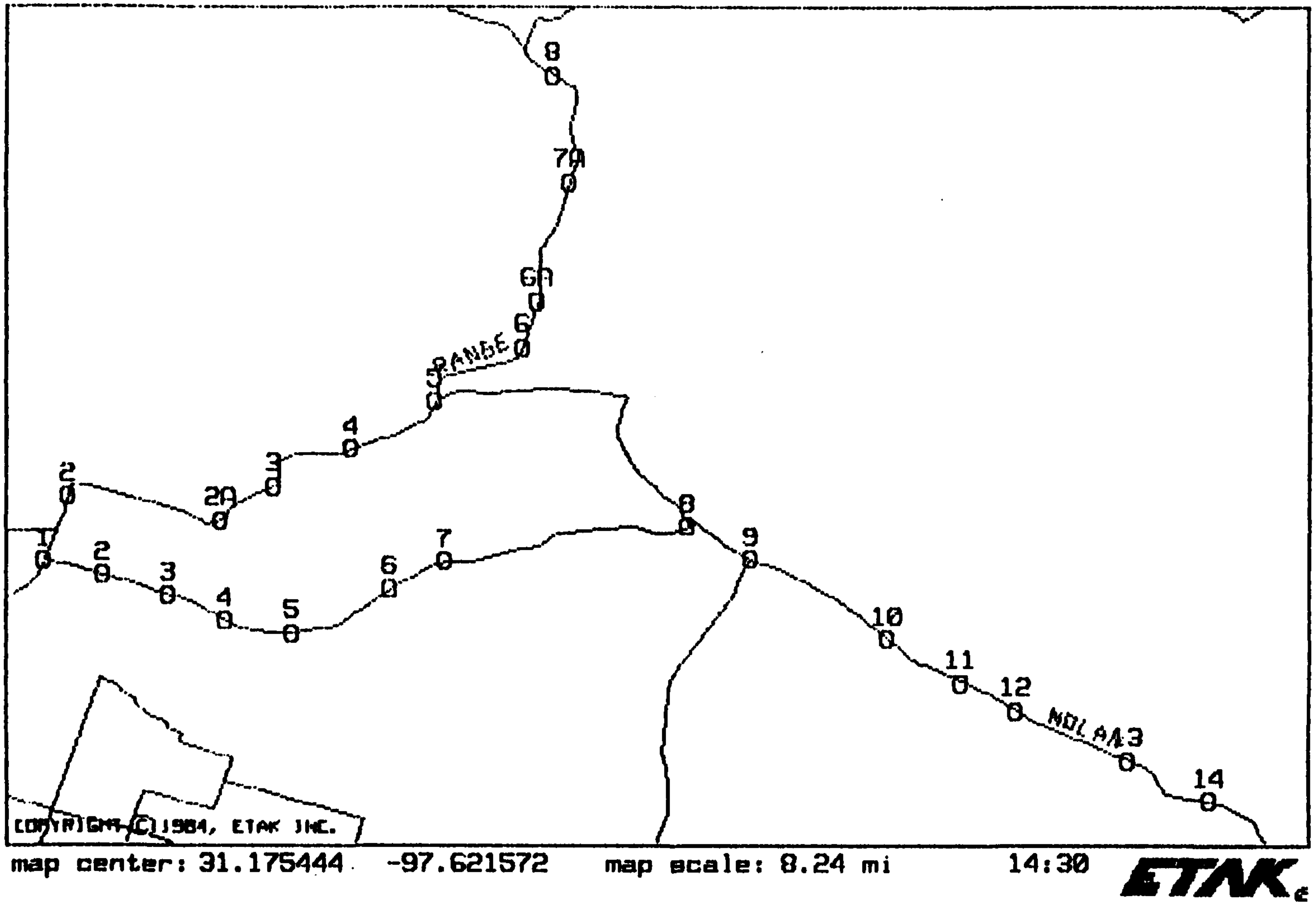

Figure 18. Data Collection Checkpoints for Test Routes

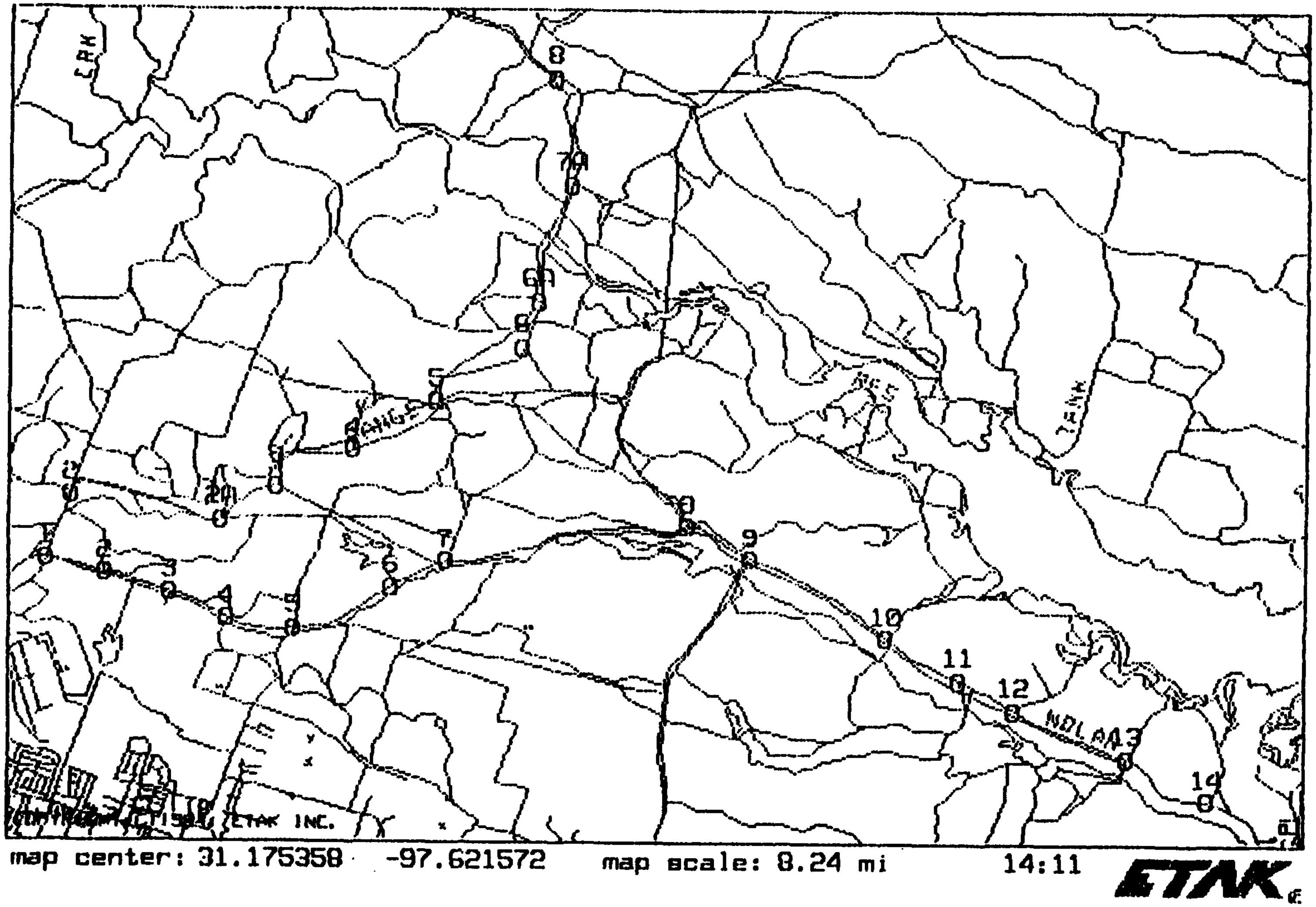

Figure 19. Detailed Digital Map of Test Area

plot of the same area showing all data in the database, including shorelines, residential roads, and tank trails.

Each run was named by an abbreviation defining the set and a run number. For example, the first west-to-east run in June with map matching turned on is given the name WEMM1. (Runs in April were numbered from 0.) Summary runs are named by the letters alone, for example, WEMM, which summarizes WEMM1 through WEMM6.

The selected routes consisted of paved roads. Cross-country tests involving unpaved tank trails were not performed, since after discussions with military authorities we concluded that because of road conditions and military operations it would not be safe to drive on the tank trails.

Data were captured using the digital communications feature of the Navigator. The RS-232C port on the Navigator was connected to a laptop computer which was programmed to poll the Navigator every 15 seconds to capture latitude, longitude, heading, speed, relative distance, UTM coordinates, and a map-matching indicator. At each checkpoint the vehicle was stopped and aligned with the mark on the road, and an additional - indexed - sample was collected at that point by pressing a function key on the laptop.

Two test trips were made to Fort Hood: the first in April, 1990, and the second in June, 1990. Just prior to the first test a mapping error was discovered in the Fort Hood EtakMap. The decision was made to proceed with the test. Subsequent analysis of the maps showed the magnitude of the map errors to be variable but with a maximum absolute displacement on the order of 70 meters. Even though the Navigator's performance is not strongly affected by absolute coordinate errors, it was decided to correct the maps and rerun the tests. Both data sets are presented.

In all, 88 runs totaling more than 1400 km were recorded. Thirty-four runs were recorded in April and 44 were recorded in June with Etak and ETL personnel participating. An additional 20 runs were recorded later in June by ETL personnel alone. Table 20 provides a count of the numbers of runs as a function of path and navigation.

Table 20. Counts of Runs

Run	Map Matching	Dead Reckoning
West-East	**WEMM**	**WEDR**
April	3	3
June	6	6
ETL	—	5
East-West	**EWMM**	**EWDR**
April	3	3
June	6	6
ETL	—	5
South-North	**SNMM**	**SNDR**
April	3	3
June	5	5
ETL	—	5
North-South	**NSMM**	**NSDR**
April	3	3
June	5	5
ETL	—	5

Methods of Navigational Accuracy Analysis

Because the Etak Navigator uses relative navigation techniques (dead reckoning and map matching), it was proposed that navigational accuracy would be evaluated in two parts: first in relative terms by comparing it to the digital map, and second by evaluating the absolute accuracy of the map separately. The map accuracy evaluation is presented in Chapter 6.

For the April tests the checkpoint data were compared to the coordinates of the map road intersections. Position errors were tabulated in meters. Just prior to the June (and ETL) tests, ETL personnel surveyed a revised set of checkpoints using differential GPS equipment. These points were used directly in the navigational accuracy analysis for the subsequent runs; for each run the difference between the GPS and Navigator coordinates at the starting point was subtracted from all coordinates. While this approach lumps navigation errors in with some mapping errors, the map errors had been quantified and it was felt that a direct comparison of Navigator to absolute coordinates was appropriate.

Methods of Data Presentation

The data are presented in two ways: in error tables and in graphs of error magnitude versus run length. Each presentation provides a different and useful way to view the data. An example of the presentation formats is described here. In the next section error tables and error plots are presented for each of the 20 sets of runs.

For example, the computed position errors for each of the June WEMM runs are shown in Table 29 on page 96. Errors for all six runs are shown along with summary statistics. For each of the checkpoints the measured distance from the start of the run and the computed navigation error at that checkpoint are shown. For the April runs, the navigation error is the distance in meters between the Navigator's position estimate and the corresponding EtakMap coordinates. For the June runs, the navigation error is the distance between the Navigator's position estimate and the surveyed position of the checkpoint, with the difference between navigation coordinates and GPS coordinates at the start of the run subtracted from all points.

The bottom sets of numbers in Table 29 show the average and standard deviation of the navigator's measured distances travelled, and the average and standard deviation of the navigator's position errors.

For purposes of data presentation the Fort Hood EtakMap database was loaded onto the Etak GeoCoder Workstation, a Geographic Information System (see the Geocoder Workstation User's Guide, Reference 3). This system has the capability to display a color map of the data base at an arbitrary center and scale. Data from the field test were plotted as landmarks and the resulting display was plotted on a monochrome printer.

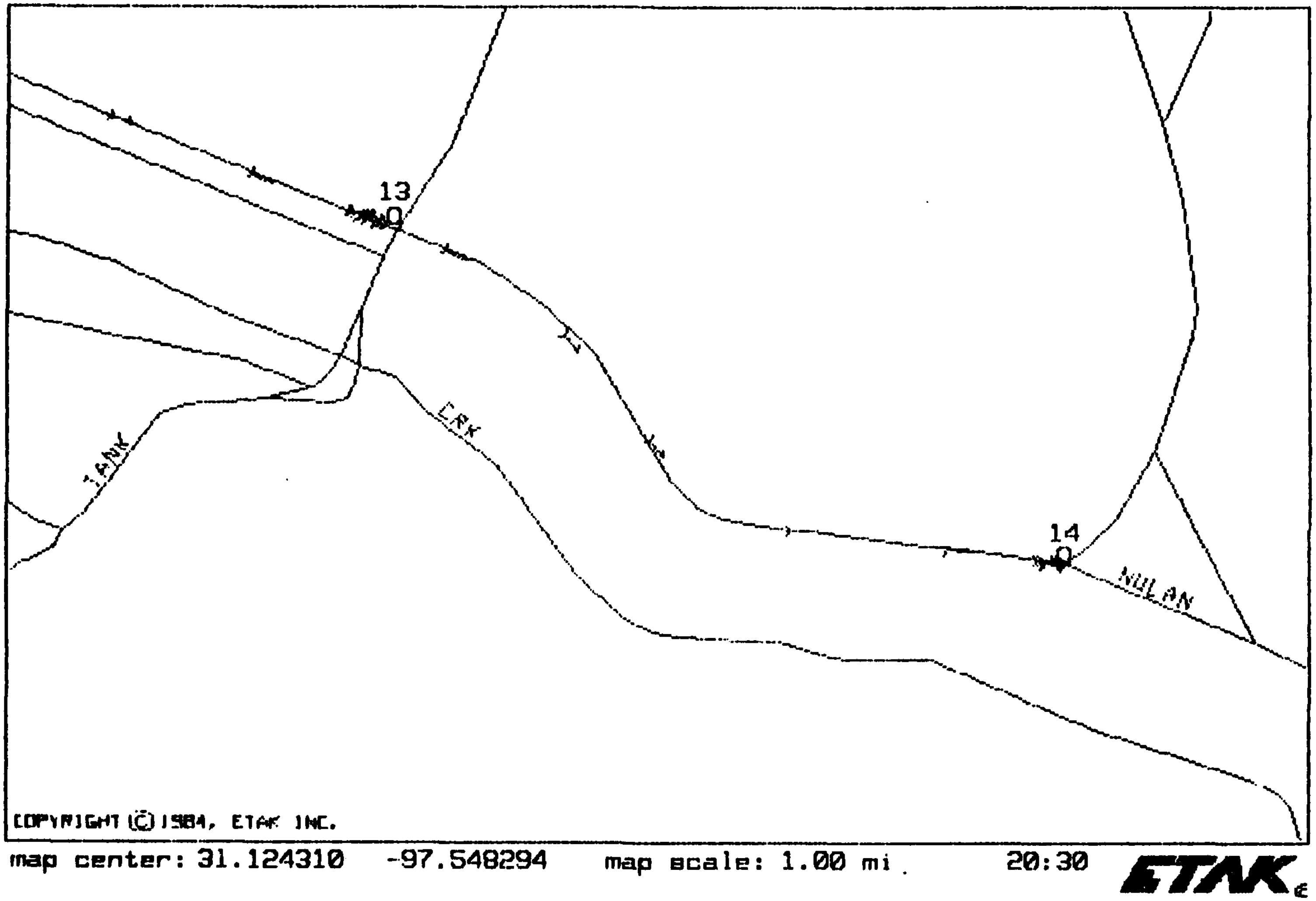

Figure 20. Eastern End of West-East Route WEMM

Figure 20 plots, at a 1-mile scale, the map at the east end of the east-west route, showing check points 13 and 14. Superimposed on the map are small arrows indicating the position and heading of the vehicle at various points. The shapes of the arrows indicate different types of data. Arrows with full tails represent data collected with map matching. Arrows with no tails represent data collected with dead reckoning. Arrows with half heads represent points automatically collected every 15 seconds (which are not correlated with checkpoints). Arrows with full heads represent data collected by function key at the check points.

In Figure 20 the checkpoint data for each of the six eastbound map-matching runs in June are shown. In addition, the 15-second data for one of the runs are shown. The 15-second data for a single run show details of the vehicle's path without cluttering the display with the checkpoint data clusters.

Data Presentation

The detailed results of the navigation tests are given by a table and a graph for each set of runs, starting at Table 21 and Figure 21 and ending at Table 40 and Figure 40. The data are presented in the order of the three data collection groups: April, June, and ETL. Within each group the runs are organized by navigation type: map matching, then dead reckoning. Finally, within each type the runs are organized by route in the order driven. In the tables, "Dist" is the distance travelled along the route from the beginning of the traverse, and "Error" is the distance between the UTM coordinates reported by the Navigator and the correct UTM coordinates of the checkpoint. Chapters 8 and 9, following the tables and figures at page 121, discuss and summarize the results.

The raw data as collected are included in Appendix 4 for reference.

Table 21. April WEMM Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
wemm0			wemm2		
2	968	9	2	961	10
3	1845	6	3	1836	13
4	2802	37	4	2791	38
5	4382	13	5	4368	8
6	5264	42	6	5247	39
7	8739	18	7	8719	29
8	9752	85	8	9729	80
9	12066	46	9	12040	51
10	13231	103	10	13204	100
11	14149	89	11	14121	89
12	15886	80	12	15856	81
13	17270	63	13	17238	55
wemm1					
2	962	5			
3	1838	10			
4	2793	37			
5	4372	14			
6	5252	44			
7	8724	21			
8	9736	84			
9	12047	48			
10	13211	105			
11	14128	94			
12	15864	87			
13	17246	68			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
2	964	2.84	2	7.93	1.98
3	1839	3.95	3	9.50	3.02
4	2795	4.66	4	37.06	0.56
5	4374	6.13	5	11.65	2.52
6	5254	7.05	6	41.81	2.39
7	8727	8.71	7	22.91	4.78
8	9739	9.51	8	83.08	1.86
9	12051	10.75	9	48.40	1.77
10	13215	11.43	10	102.63	2.30
11	14133	12.07	11	90.93	2.53
12	15869	12.85	12	82.47	2.96
13	17251	13.92	13	62.01	5.00

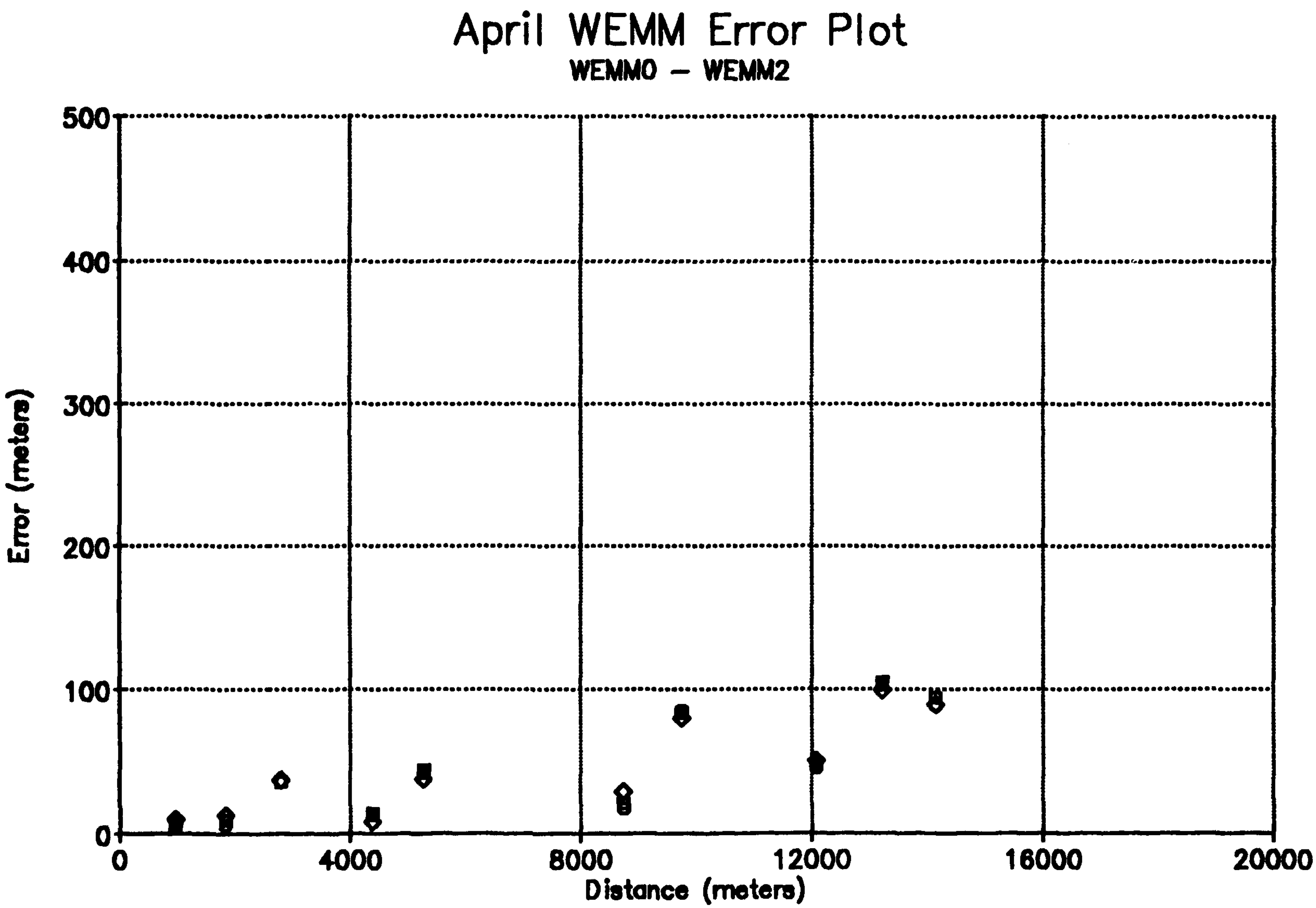

Figure 21. April WEMM Error Plot

Table 22. April EWMM Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
ewmm0			ewmm2		
13	1384	22	13	1383	14
12	3122	17	12	3119	20
11	4040	23	11	4036	19
10	5206	47	10	5200	55
9	7519	7	9	7511	18
8	8533	56	8	8523	64
7	12007	29	7	11994	33
6	12889	32	6	12875	33
5	14466	19	5	14452	38
4			4	15406	10
3	16297	25	3	16281	15
2	17258	28	2	17241	23
ewmm1					
13	1382	15			
12	3118	14			
11	4036	21			
10	5200	54			
9	7512	15			
8	8525	58			
7	11997	36			
6	12877	30			
5	14456	20			
4	15410	16			
3	16285	19			
2	17246	23			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
13	1383	0.89	13	16.98	14.20
12	3120	1.69	12	17.14	5.48
11	4037	2.20	11	21.07	16.48
10	5202	2.78	10	52.11	11.06
9	7514	3.73	9	13.51	10.12
8	8527	4.14	8	59.20	19.53
7	11999	5.62	7	32.54	14.67
6	12880	6.03	6	31.55	3.58
5	14458	5.69	5	25.57	7.65
4	10272	1.98	4	13.16	5.18
3	16287	6.84	3	19.79	4.54
2	17248	7.13	2	24.62	5.55

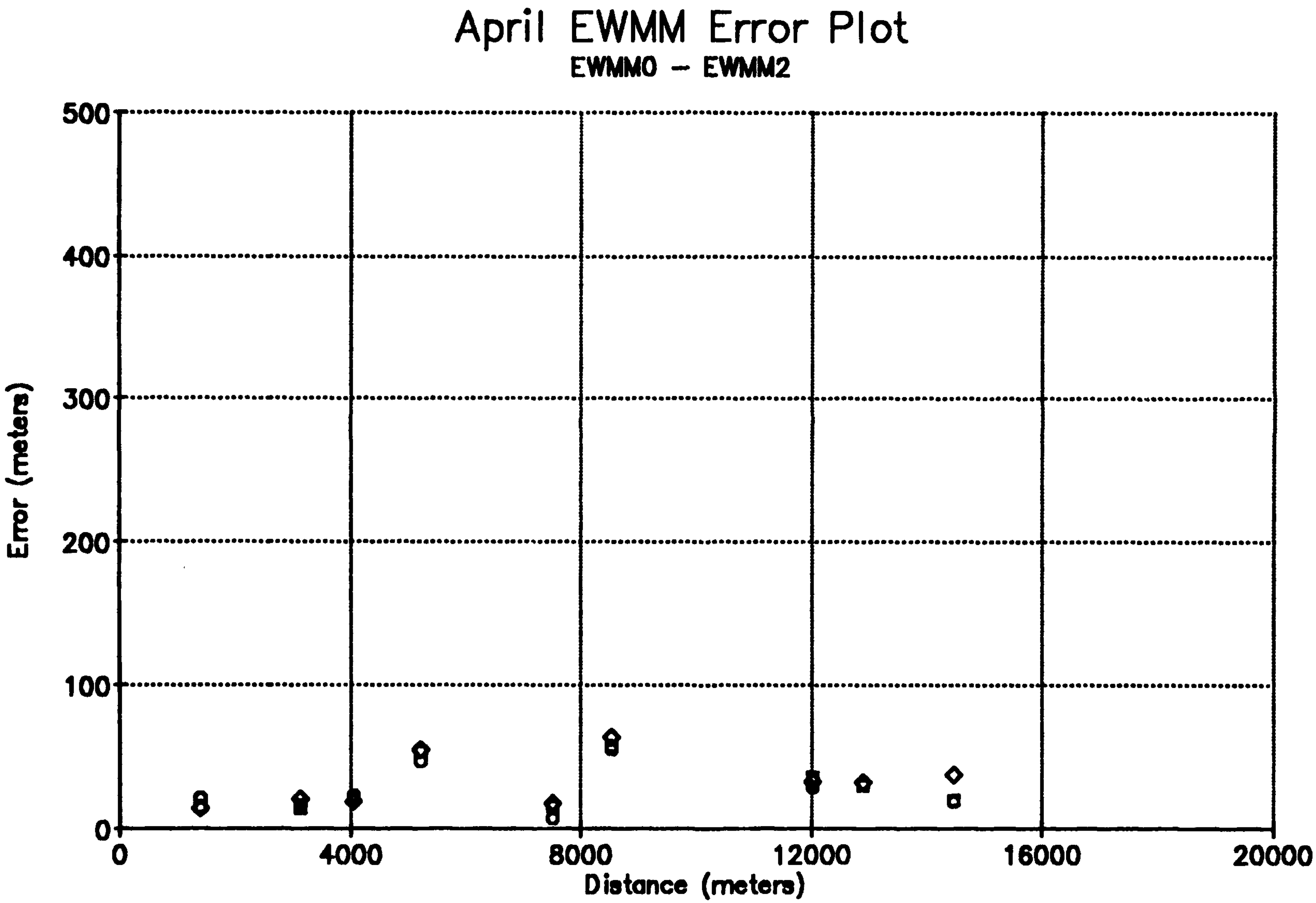

Figure 22. April EWMM Error Plot

Table 23. April SNMM Error Analysis (meters)

Point	Dist	Error
snmm0		
2	1101	14
2A	-	—
3	4374	19
4	5842	22
5	7305	52
6	9020	50
6A	9736	12
7A	11734	33
8	13555	76
snmm1		
2	1100	23
2A	3404	50
3	4375	21
4	5841	22
5	7303	19
6	9017	47
6A	9732	26
7A	11729	56
8	13550	71
snmm2		
2	1103	34
2A	3412	46
3	4384	20
4	5853	33
5	7320	17
6	9038	50
6A	9754	21
7A	11755	28
8	13579	63

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
2	1101	1.31	2	23.71	8.42
2A	3408	3.79	2A	48.19	1.80
3	4378	4.39	3	20.02	0.52
4	5845	5.63	4	25.60	5.24
5	7309	7.27	5	28.97	16.01
6	9025	9.06	6	49.17	1.24
6A	9741	9.52	6A	19.66	5.56
7A	11739	11.10	7A	39.28	12.31
8	13561	12.49	8	70.25	5.61

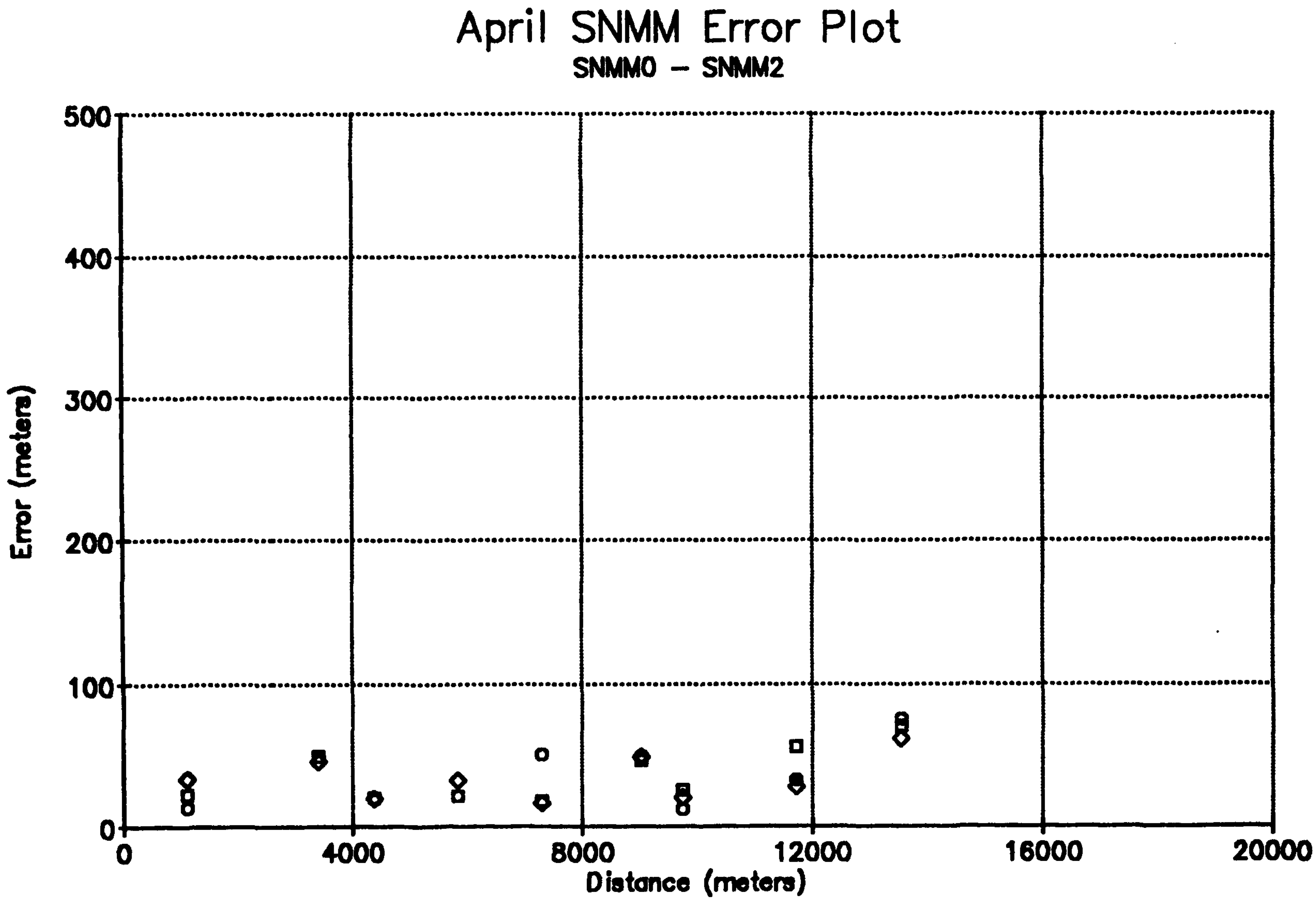

Figure 23. April SNMM Error Plot

Table 24. April NSMM Error Analysis (meters)

Point	Dist	Error
nsmm0		
7A	1818	25
6A	3814	49
6	4530	5
5	6244	67
4	7704	40
3	9169	22
2A	10137	101
2	12440	21
1	13542	18
nsmm1		
7A	1818	35
6A	3813	79
6	4529	34
5	6242	70
4	7702	41
3	9168	26
2A	10137	94
2	12442	18
1	13545	24
nsmm2		
7A	1820	32
6A	3819	73
6	4535	29
5	6251	67
4	7712	39
3	9176	19
2A	10145	91
2	12452	27
1	13555	24

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
7A	1818.43	0.88	7A	30.54	3.87
6A	3815.50	2.32	6A	66.94	13.20
6	4531.14	2.72	6	22.64	12.58
5	6245.62	3.67	5	68.29	1.29
4	7705.99	4.12	4	40.20	0.78
3	9170.87	3.69	3	22.11	2.88
2A	10140.00	3.71	2A	95.20	3.97
2	12445.08	5.24	2	21.82	3.79
1	13547.25	5.71	1	21.86	2.62

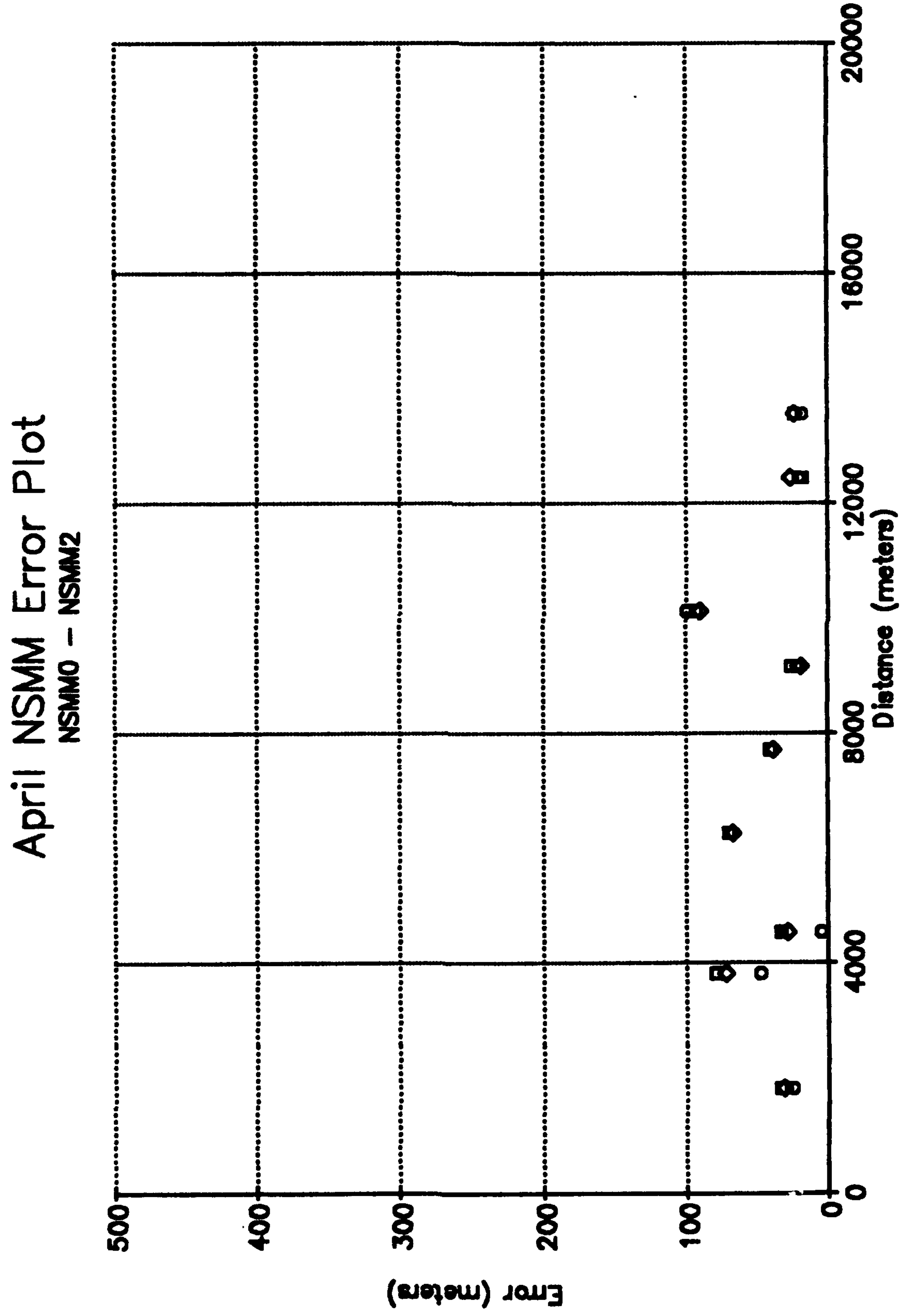

Figure 24. April NSMM Error Plot

Table 25. April WEDR Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
wedr0			wedr2		
2	963	29	2	962	42
3	1840	68	3	1838	57
4	2797	45	4	2793	17
5	4378	84	5	4372	34
6	5258	127	6	5252	90
7	8735	162	7	8724	79
8	9748	258	8	9735	175
9	12063	281	9	12047	144
10	13228	352	10	13212	209
11	14146	361	11	14129	197
12	15884	405	12	15864	191
13	17268	470	13	17246	213
wedr1					
2	963	19			
3	1838	46			
4	2794	14			
5	4375	44			
6	5255	106			
7	8729	140			
8	9741	230			
9	12055	232			
10	13219	301			
11	14137	302			
12	15874	331			
13	17256	389			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
2	963	0.50	2	29.91	9.17
3	1839	0.97	3	56.86	9.25
4	2795	1.64	4	25.30	13.95
5	4375	2.21	5	54.00	21.88
6	5255	2.66	6	107.74	15.38
7	8729	4.35	7	126.91	34.74
8	9741	5.03	8	220.73	34.63
9	12055	6.21	9	219.12	56.73
10	13220	6.84	10	287.44	59.13
11	14137	7.29	11	286.81	68.07
12	15874	7.94	12	309.21	88.55
13	17257	8.75	13	357.50	107.11

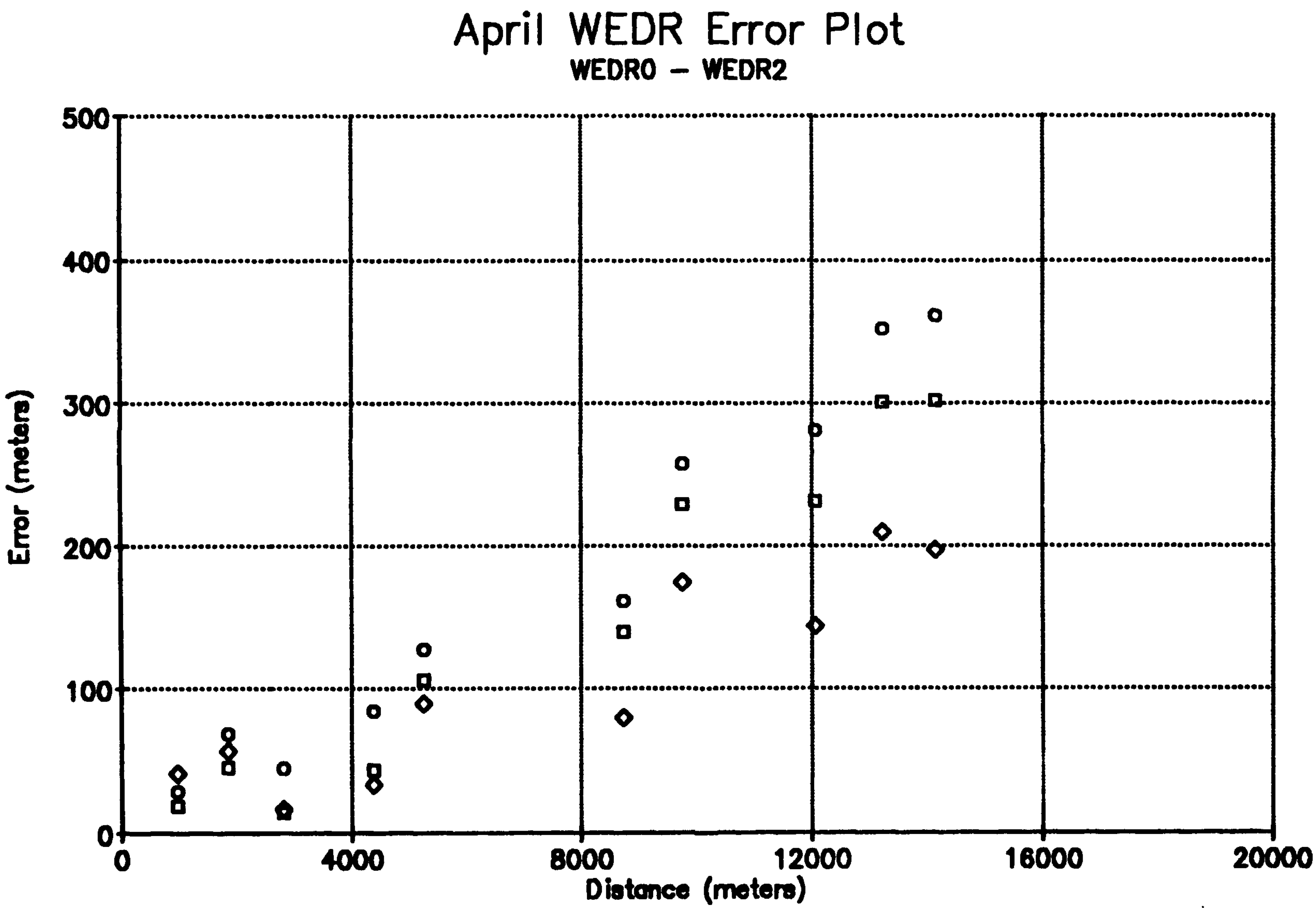

Figure 25. April WEDR Error Plot

Table 26. April EWDR Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
ewdr0			ewdr2		
13	1385	52	13	1382	1
12	3123	65	12	3117	38
11	4041	76	11	4034	53
10	5207	61	10	5197	126
9	7522	97	9	7508	131
8	8537	127	8	8521	157
7	12013	156	7	11993	114
6	12895	159	6	12874	137
5	14474	129	5	14452	178
4	15429	194	4	15406	134
3	16305	206	3	16281	145
2	17267	185	2	17242	172
ewdr1					
13	1383	43			
12	3120	42			
11	4037	43			
10	5202	46			
9	7515	33			
8	8529	75			
7	12003	72			
6	12884	95			
5	14463	104			
4	15418	135			
3	16293	142			
2	17255	134			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
13	1383	1.15	13	32.26	22.20
12	3120	2.30	12	48.33	11.85
11	4038	3.06	11	57.33	13.66
10	5202	4.01	10	77.79	34.73
9	7515	5.65	9	86.92	40.87
8	8529	6.25	8	119.54	33.62
7	12003	8.06	7	113.89	33.99
6	12884	8.49	6	130.37	26.42
5	14463	9.08	5	136.93	30.43
4	15418	9.52	4	154.31	28.13
3	16293	9.89	3	164.29	29.78
2	17255	10.45	2	163.96	21.60

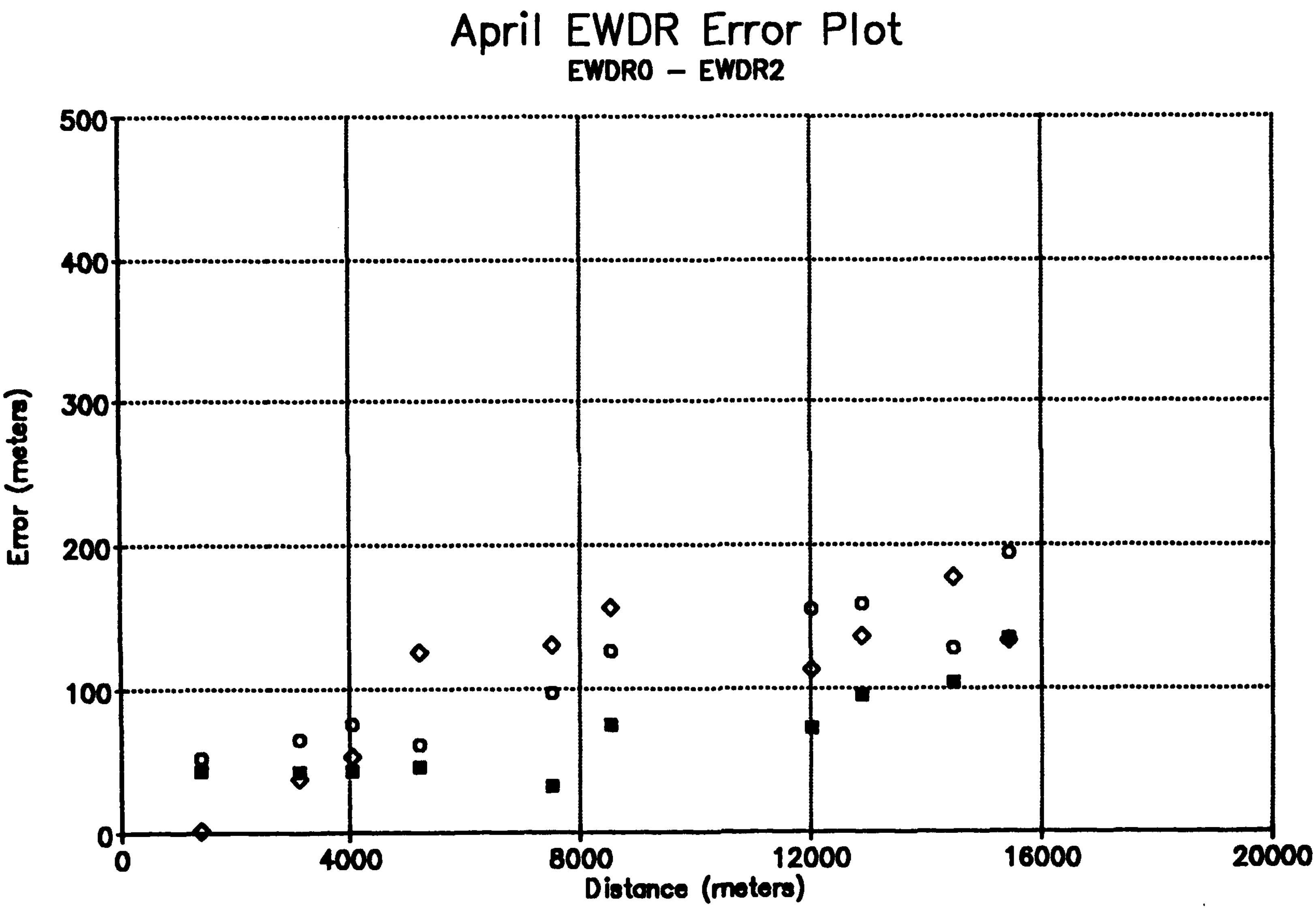

Figure 26. April EWDR Error Plot

Table 27. April SNDR Error Analysis (meters)

Point	Dist	Error
sndr0		
2	1100	84
2A	3405	154
3	4376	115
4	5843	109
5	7306	114
6	9021	146
6A	9737	173
7A	11735	167
8	13556	101
sndr1		
2	1101	128
2A	3412	256
3	4383	234
4	5854	405
5	7322	392
6	9042	394
6A	9760	418
7A	11764	444
8	13590	388
sndr2		
2	1104	44
2A	3415	86
3	4389	53
4	5861	35
5	7329	52
6	9049	24
6A	9767	50
7A	11771	29
8	13597	58

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
2	1102	1.55	2	85.38	34.10
2A	3414	1.68	2A	165.58	69.82
3	4383	5.37	3	134.21	75.24
4	5853	7.51	4	182.96	159.89
5	7319	9.32	5	185.92	147.91
6	9037	11.68	6	187.79	153.77
6A	9755	12.70	6A	213.38	153.06
7A	11757	15.42	7A	213.32	172.41
8	13581	17.88	8	182.19	146.40

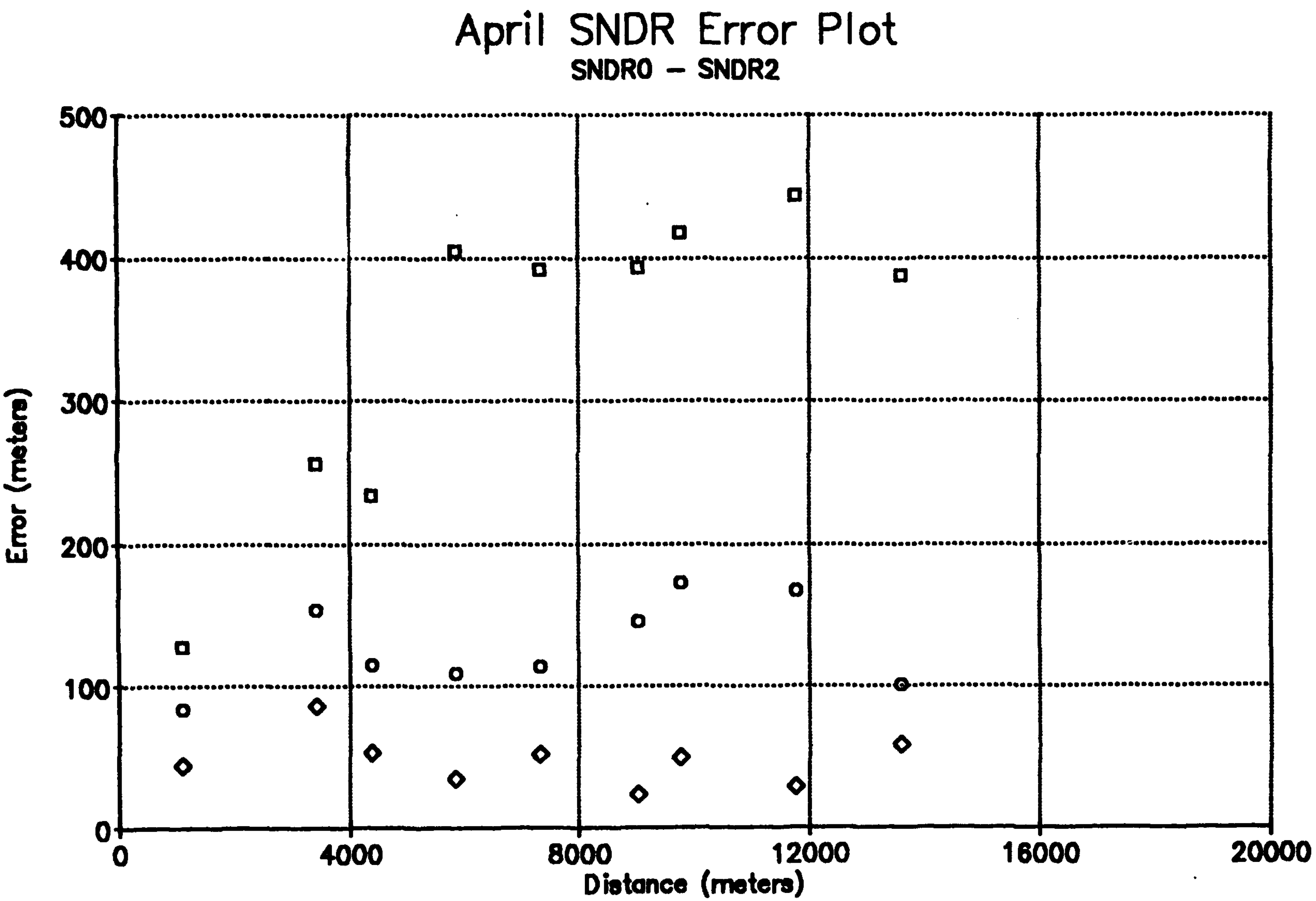

Figure 27. April SNDR Error Plot

Table 28. April NSDR Error Analysis (meters)

Point	Dist	Error
nsdr0		
7A	1819	130
6A	3815	163
6	4531	190
5	6245	158
4	7705	166
3	9171	89
2A	10140	108
2	12445	77
1	13547	37
nsdr1		
7A	1822	133
6A	3827	64
6	4544	109
5	6264	55
4	7729	68
3	9199	16
2A	10172	36
2	12485	22
1	13590	57
nsdr2		
7A	1824	130
6A	3827	150
6	4545	164
5	6265	135
4	7731	128
3	9201	69
2A	10174	98
2	12486	57
1	13591	31

Combined Statistics

Distance			Error		
Point	**Mean**	**STD**	**Point**	**Mean**	**STD**
7A	1821.49	2.17	7A	130.96	1.76
6A	3823.30	5.60	6A	125.56	44.15
6	4539.94	6.53	6	154.32	34.11
5	6258.08	9.45	5	115.91	44.16
4	7721.65	11.75	4	120.57	40.56
3	9190.42	14.01	3	58.09	30.86
2A	10162.03	15.75	2A	80.69	31.54
2	12472.04	18.77	2	51.93	22.75
1	13575.90	20.33	1	41.50	11.28

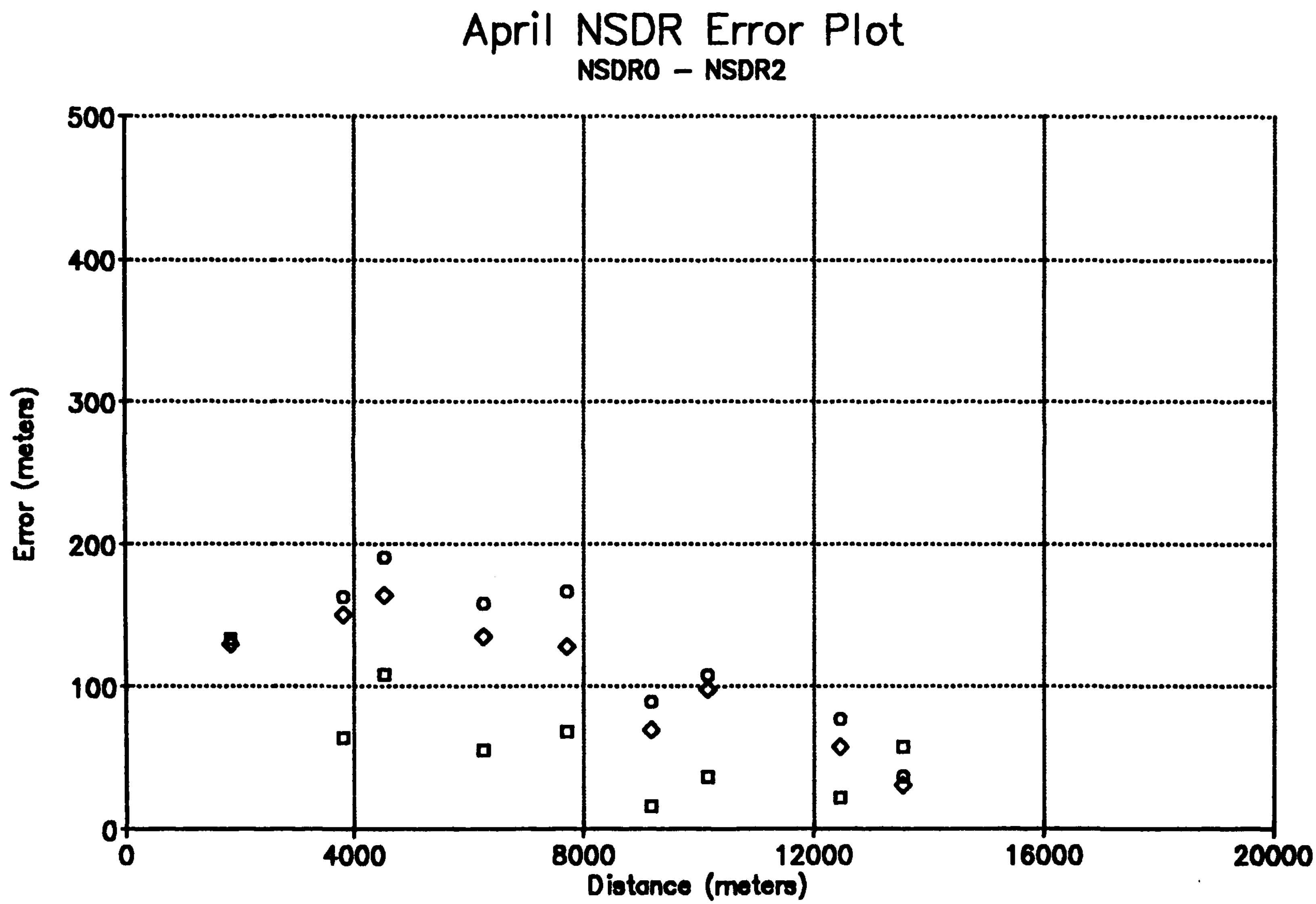

Figure 28. April NSDR Error Plot

Table 29. June WEMM Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
wemm1			wemm4		
3	1743	7	3	1746	6
5	3589	38	5	3593	35
6	5155	84	6	5161	83
7	6047	61	7	6056	57
8	9483	46	8	9498	31
9	10536	43	9	10552	34
10	12847	58	10	12867	48
12	14930	48	12	14954	33
13	16643	60	13	16670	40
14	18025	38	14	18054	27
wemm2			wemm5		
3	1743	8	3	1745	5
5	3587	40	5	3592	36
6	5153	93	6	5161	79
7	6045	64	7	6056	54
8	9481	22	8	9497	21
9	10533	32	9	10552	25
10	12844	48	10	12868	37
12	14927	35	12	14954	24
13	16641	45	13	16671	30
14	18023	32	14	18055	23
wemm3			wemm6		
3	1746	5	3	1743	9
5	3594	35	5	3588	42
6	5163	81	6	5154	83
7	6058	39	7	6048	61
8	9499	26	8	9486	45
9	10554	28	9	10539	36
10	12870	39	10	12852	48
12	14957	25	12	14937	30
13	16673	32	13	16651	40
14	18057	30	14	18034	34

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
3	1744	1.46	3	6.77	1.59
5	3590	2.83	5	37.75	2.61
6	5158	4.07	6	83.84	4.35
7	6052	4.97	7	55.81	8.39
8	9491	7.57	8	31.95	10.09
9	10544	8.58	9	32.97	5.68
10	12858	10.45	10	46.21	6.88
12	14943	12.14	12	32.46	8.01
13	16658	13.50	13	41.06	9.81
14	18041	14.41	14	30.91	4.76

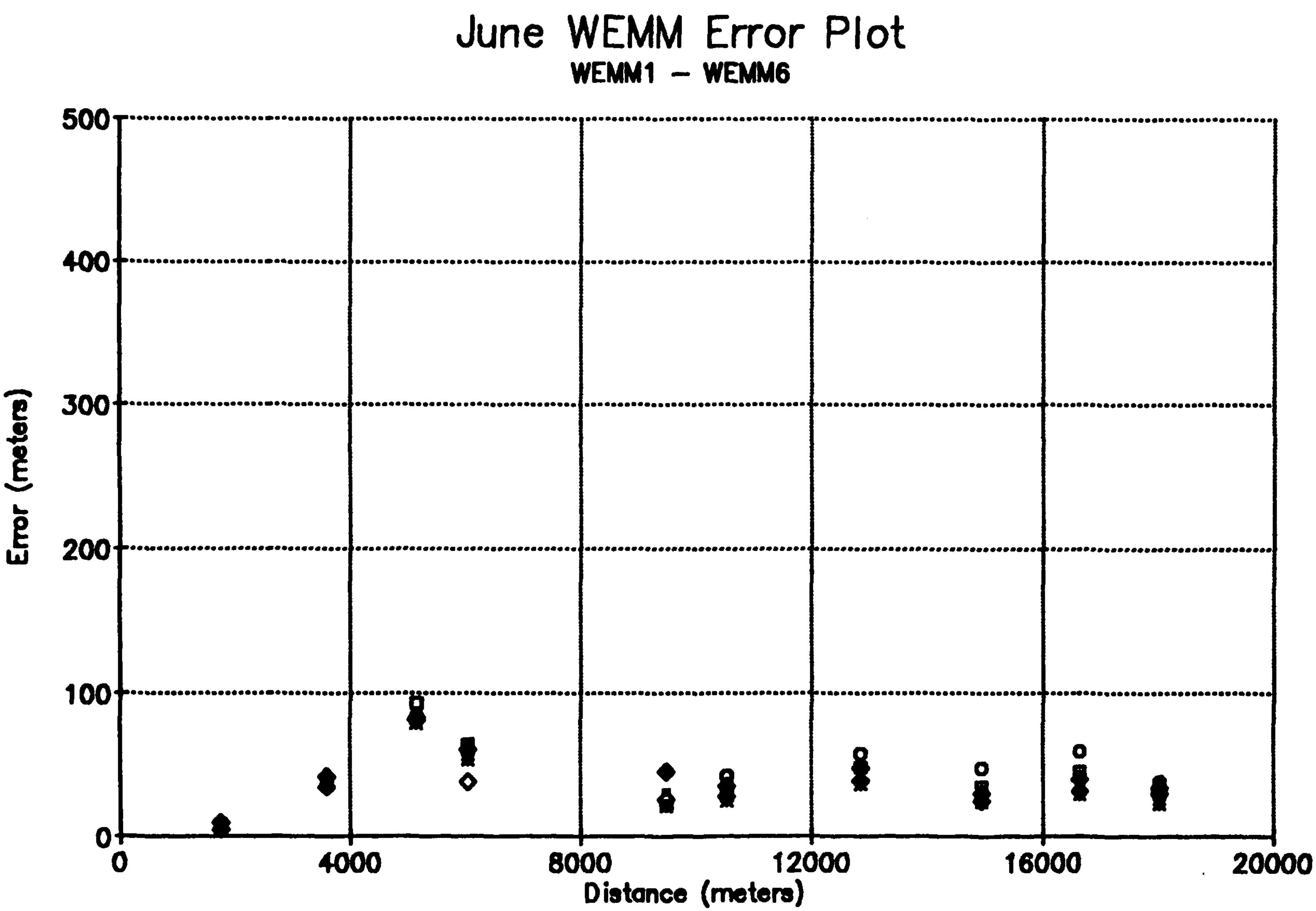

Figure 29. June WEMM Error Plot

Table 30. June EWMM Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
ewmm1			**ewmm4**		
13	1382	36	13	1384	58
12	3095	43	12	3099	61
10	5178	37	10	5184	78
9	7490	54	9	7498	90
8	8545	43	8	8554	66
7	11981	318	7	11994	68
6	12875	296	6	12889	57
5	14440	243	5	14457	41
3	16282	157	3	16301	33
A	18024	167	A	18045	40
ewmm2			**ewmm5**		
13	1382	56	13	1383	51
12	3095	62	12	3099	54
10	5178	65	10	5185	50
9	7490	82	9	7499	64
8	8545	127	8	8555	49
7	11981	70	7	11995	79
6	12875	72	6	12891	63
5	14440	52	5	14458	42
3	16283	43	3	16303	31
A	18025	51	A	18048	37
ewmm3			**ewmm6**		
13	1384	33	13	1382	34
12	3099	39	12	3096	42
10	5185	50	10	5179	28
9	7500	77	9	7491	41
8	8556	82	8	8546	45
7	11997	47	7	11983	84
6	12893	48	6	12877	52
5	14461	33	5	14443	46
3	16306	17	3	16285	41
A	18051	19	A	18029	46

Combined Statistics

Distance			Error		
Point	**Mean**	**STD**	**Point**	**Mean**	**STD**
Distance 13	1383	0.88	Error 13	44.62	10.60
12	3097	1.83	12	50.31	9.05
10	5182	3.14	10	51.36	16.53
9	7495	4.44	9	68.14	16.84
8	8550	4.95	8	68.68	29.38
7	11988	7.17	7	110.86	93.21
6	12884	7.86	6	97.95	88.98
5	14450	8.92	5	76.30	74.96
3	16293	10.20	3	53.56	47.17
A	18037	11.31	A	59.77	48.98

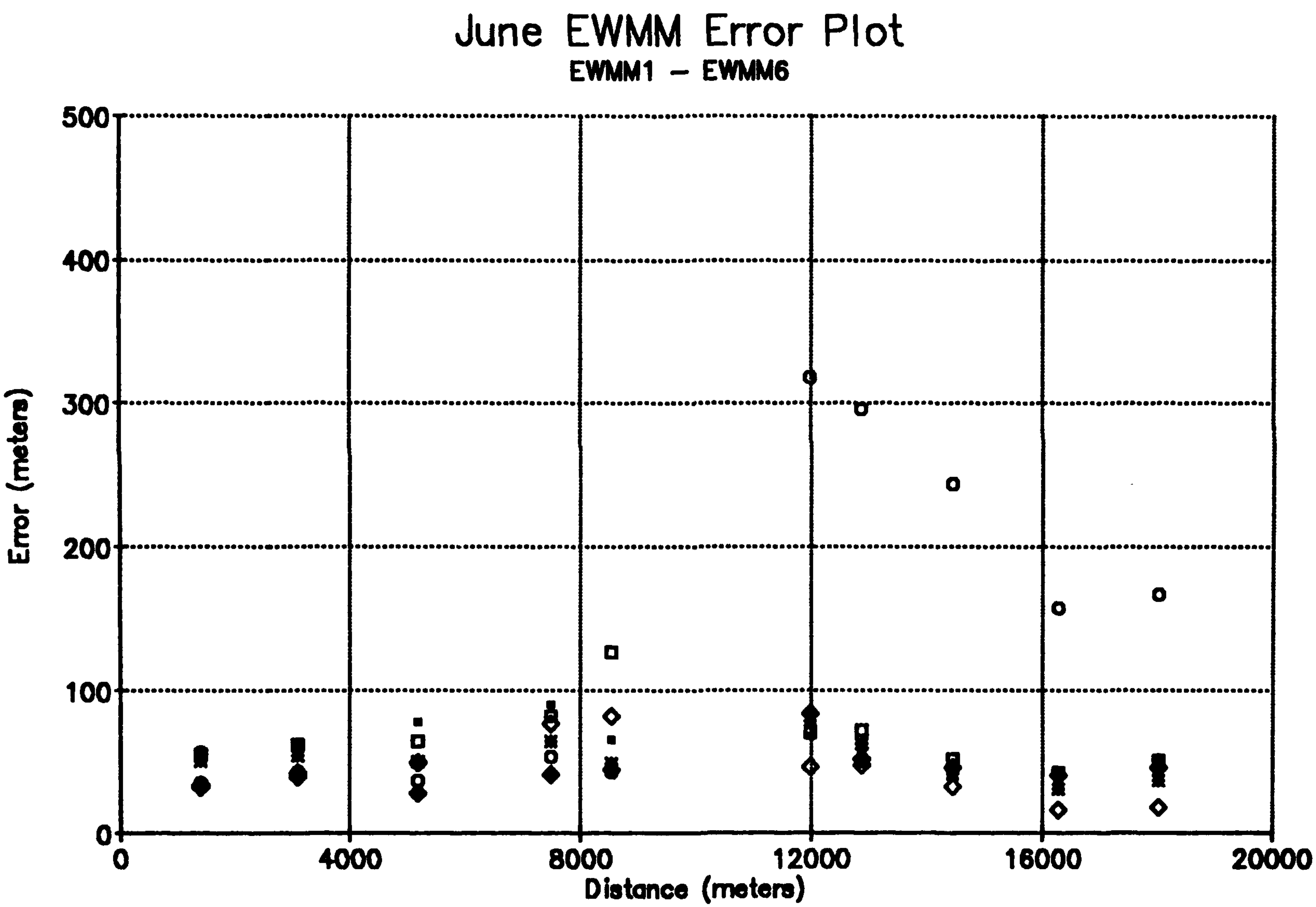

Figure 30. June EWMM Error Plot

Table 31. June SNMM Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
snmm1			snmm4		
2	1040	17	2	1040	17
2A	3327	22	2A	3327	28
3	4299	27	3	4298	38
4	5768	6	4	5766	29
5	7233	21	5	7230	61
6	8915	29	6	8910	58
6A			6A	9667	65
7A	11666	80	7A	11660	79
8	13489	57	8	13483	39
snmm2			snmm5		
2	1040	22	2	1040	21
2A	3329	26	2A	3326	30
3	4300	44	3	4298	61
4	5768	12	4	5765	30
5	7233	26	5	7229	56
6	8914	32	6	8910	60
6A	9672	41	6A	9666	70
7A	11666	39	7A	11660	63
8	13490	32	8	13483	40
snmm3					
2	1040	17			
2A	3327	24			
3	4299	31			
4	5766	33			
5	7230	51			
6	8910	59			
6A	9667	67			
7A	11661	86			
8	13483	33			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
2	1040	0.32	2	18.66	2.01
2A	3327	0.78	2A	25.92	2.97
3	4299	0.96	3	40.20	11.90
4	5767	1.25	4	21.89	10.91
5	7231	1.64	5	42.84	16.56
6	8912	2.36	6	47.63	14.12
6A	9668	2.43	6A	60.63	11.57
7A	11663	2.82	7A	69.60	17.17
8	13486	3.49	8	40.14	8.96

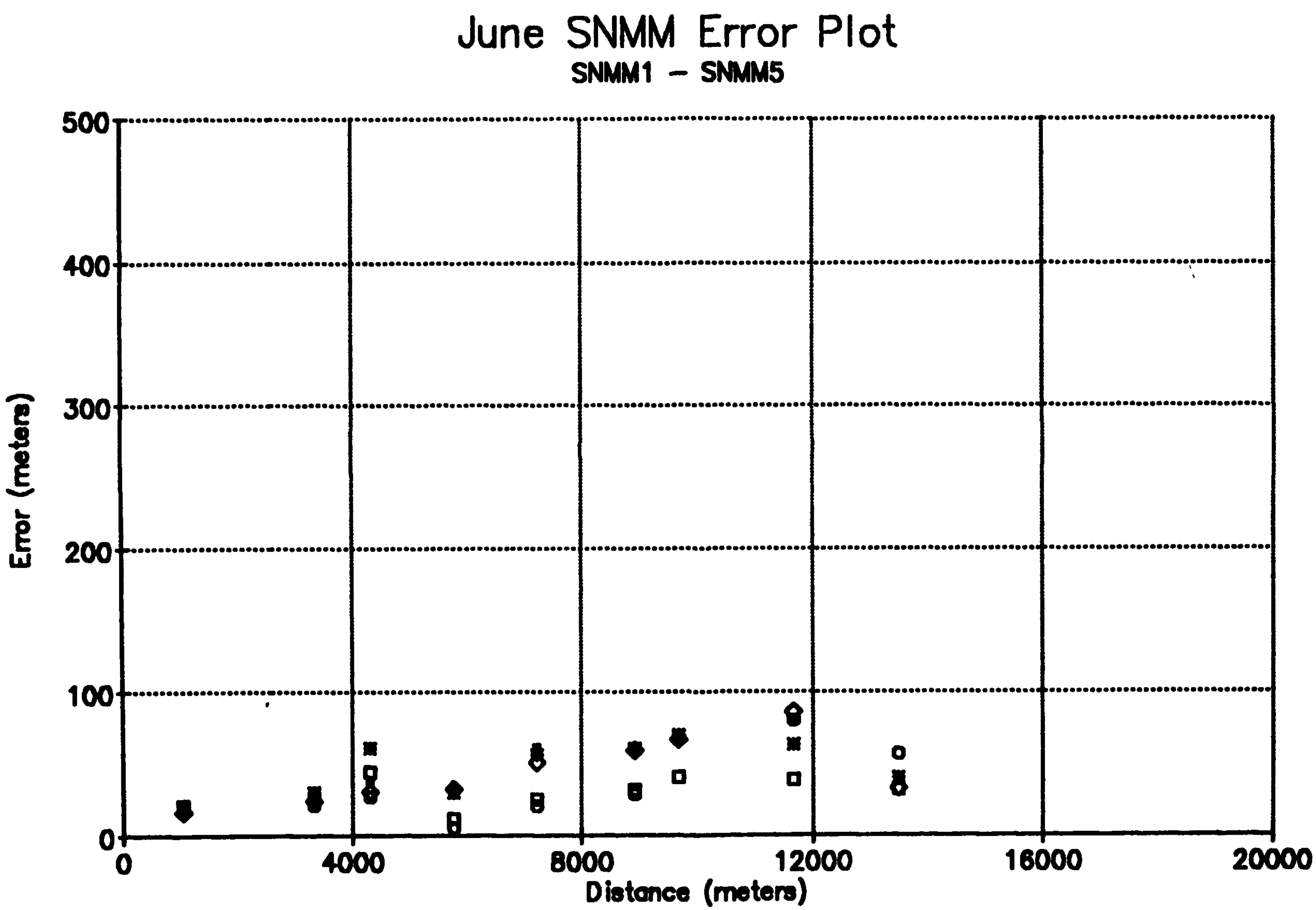

Figure 31. June SNMM Error Plot

Table 32. June NSMM Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
nsmm1			nsmm4		
7A	1822	20	7A	1819	28
6A	3815	79	6A	3813	87
6	4537	85	6	4569	88
5	6253	82	5	6249	63
4	7714	46	4	7709	48
3	9182	21	3	9176	26
2A	10153	28	2A	10144	42
2	12441	33	2	12431	34
C	13482	28	C	13471	29
nsmm2			nsmm5		
7A	1821	24	7A	1819	23
6A	3816	54	6A	3812	77
6	4572	53	6	4568	73
5	6252	67	5	6248	85
4	7714	43	4	7708	46
3	9182	27	3	9175	22
2A	10152	45	2A	10144	38
2	12440	30	2	12431	32
C	13481	27	C	13472	28
nsmm3					
7A	1819	19			
6A	3813	79			
6	4569	97			
5	6249	80			
4	7709	50			
3	9175	25			
2A	10145	39			
2	12432	26			
C	13473	26			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
7A	22.98	3.24	7A	1820.24	0.98
6A	75.07	11.18	6A	3813.76	1.44
6	79.02	15.00	6	4562.80	13.15
5	75.53	8.50	5	6250.05	2.00
4	46.60	2.19	4	7710.92	2.60
3	24.31	2.33	3	9177.91	3.37
2A	38.53	5.90	2A	10147.88	3.96
2	30.89	3.06	2	12434.82	4.62
C	27.58	0.98	C	13475.67	4.79

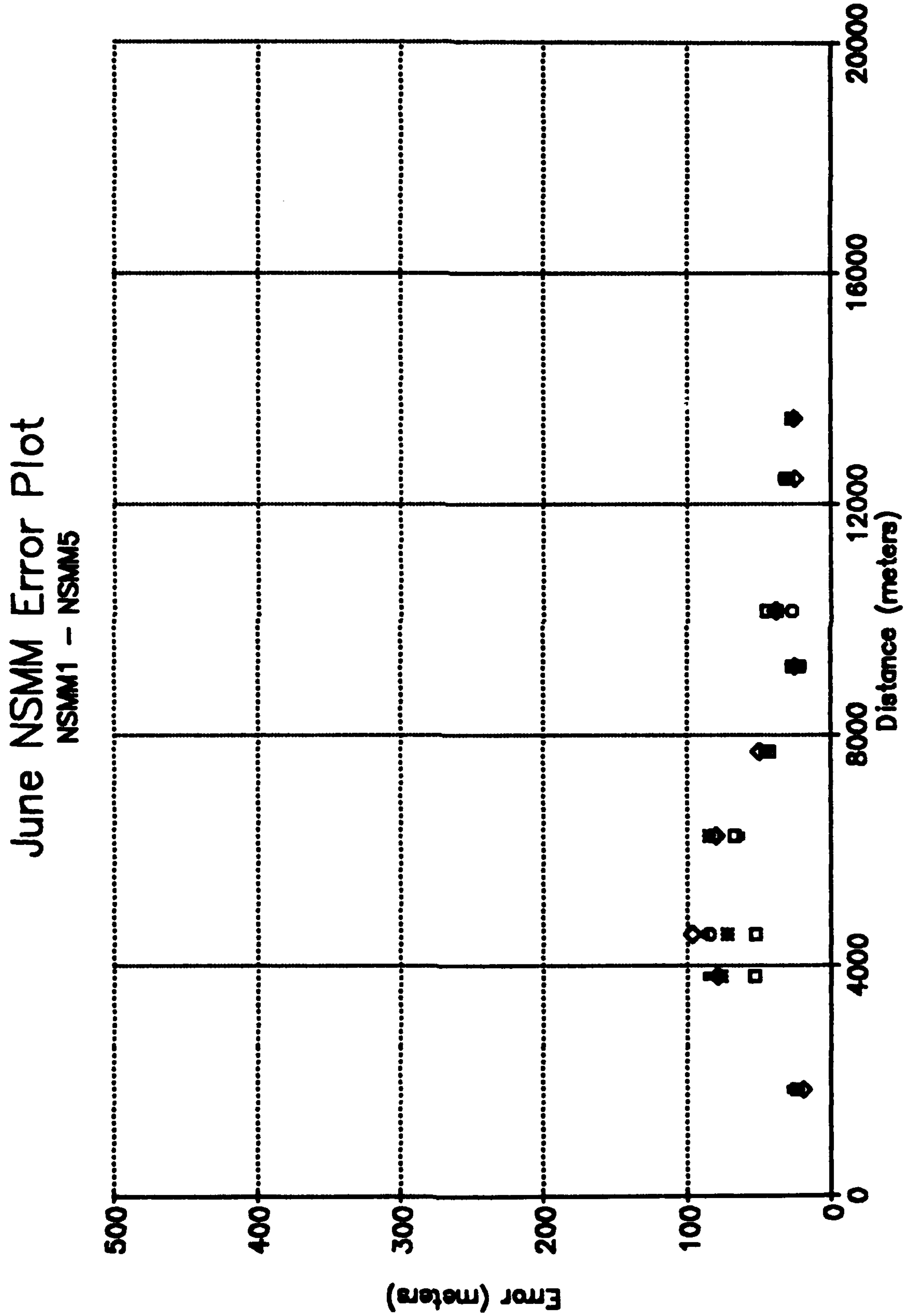

Figure 32. June NSMM Error Plot

Table 33. June WEDR Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
wedr1			wedr4		
3	1747	4	3	1742	62
5	3596	45	5	3584	153
6	5166	47	6	5150	145
7	6061	35	7	6042	139
8	9505	29	8	9476	148
9	10560	47	9	10527	143
10	12877	67	10	12837	156
12	14965	77	12	14919	176
13	16682	89	13	16632	186
14	18067	108	14	18012	195
wedr2			wedr5		
3	1742	11	3	1741	55
5	3585	72	5	3584	152
6	5149	69	6	5148	149
7	6042	106	7	6041	139
8	9474	129	8	9473	145
9	10526	142	9	10525	135
10	12836	156	10	12834	144
12	14918	166	12	14916	158
13	16630	176	13	16629	169
14	18011	187	14	18009	171
wedr3			wedr6		
3	1742	51	3	1744	25
5	3584	131	5	3590	78
6	5149	128	6	5158	69
7	6042	122	7	6051	80
8	9475	131	8	9490	93
9	10528	129	9	10543	105
10	12838	137	10	12857	121
12	14919	151	12	14942	130
13	16631	161	13	16657	143
14	18012	169	14	18040	159

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
3	1743	2.19	3	34.63	22.54
5	3587	4.65	5	105.27	41.94
6	5153	6.53	6	101.27	40.53
7	6047	7.50	7	103.37	36.68
8	9482	11.67	8	112.67	41.38
9	10535	12.79	9	116.88	33.52
10	12847	15.62	10	130.09	30.85
12	14930	17.93	12	143.17	32.86
13	16644	19.92	13	154.15	31.85
14	18025	21.37	14	164.86	28.01

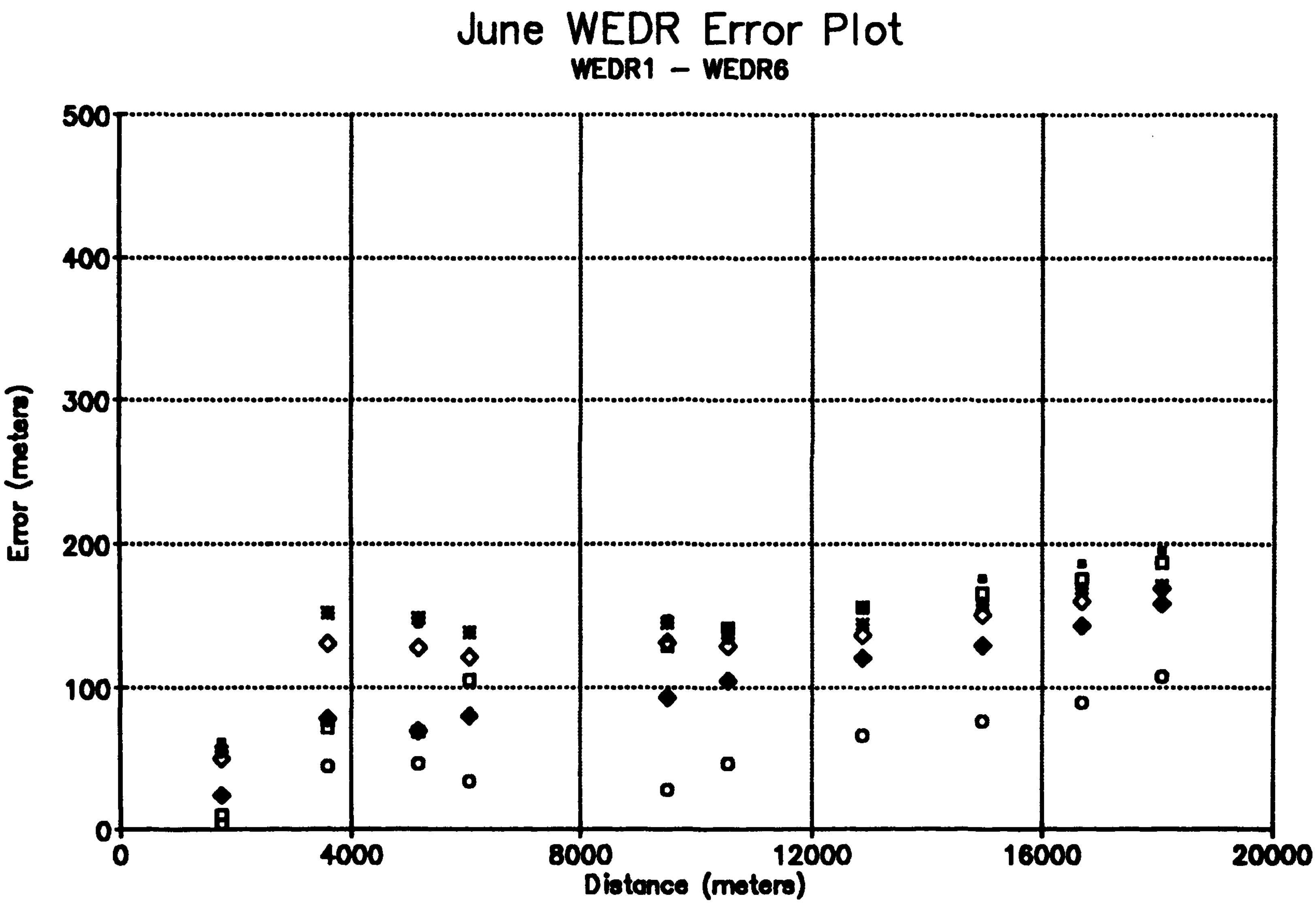

Figure 33. June WEDR Error Plot

Table 34. June EWDR Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
ewdr1			ewdr4		
13	1385	41	13	1380	15
12	3101	128	12	3092	58
10	5189	202	10	5173	97
9	7505	257	9	7482	117
8	8562	286	8	8536	123
7	12004	318	7	11968	171
6	12900	355	6	12861	236
5	14469	393	5	14424	298
3	16315	390	3	16264	287
A	18061	447	A	18005	303
ewdr2			ewdr5		
13	1381	9	13	1380	16
12	3093	31	12	3092	79
10	5173	48	10	5173	127
9	7483	48	9	7482	158
8	8537	49	8	8535	165
7	11969	110	7	11966	190
6	12862	188	6	12860	243
5	14426	274	5	14424	300
3	16266	256	3	16265	266
A	18007	263	A	18006	284
ewdr3			ewdr6		
13	1380	13	13	1382	16
12	3092	48	12	3097	76
10	5173	79	10	5181	125
9	7482	88	9	7493	159
8	8536	88	8	8549	160
7	11968	126	7	11985	215
6	12861	197	6	12880	277
5	14426	269	5	14446	339
3	16266	251	3	16289	312
A	18007	267	A	18033	333

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
13	1381	1.63	13	18.42	10.28
12	3094	3.52	12	70.03	30.49
10	5177	5.94	10	113.09	48.08
9	7488	8.68	9	137.87	65.78
8	8542	9.90	8	145.09	74.40
7	11977	14.07	7	188.41	68.17
6	12871	15.03	6	249.22	55.92
5	14436	16.70	5	312.12	42.86
3	16278	19.00	3	293.80	47.83
A	18020	21.07	A	316.07	62.89

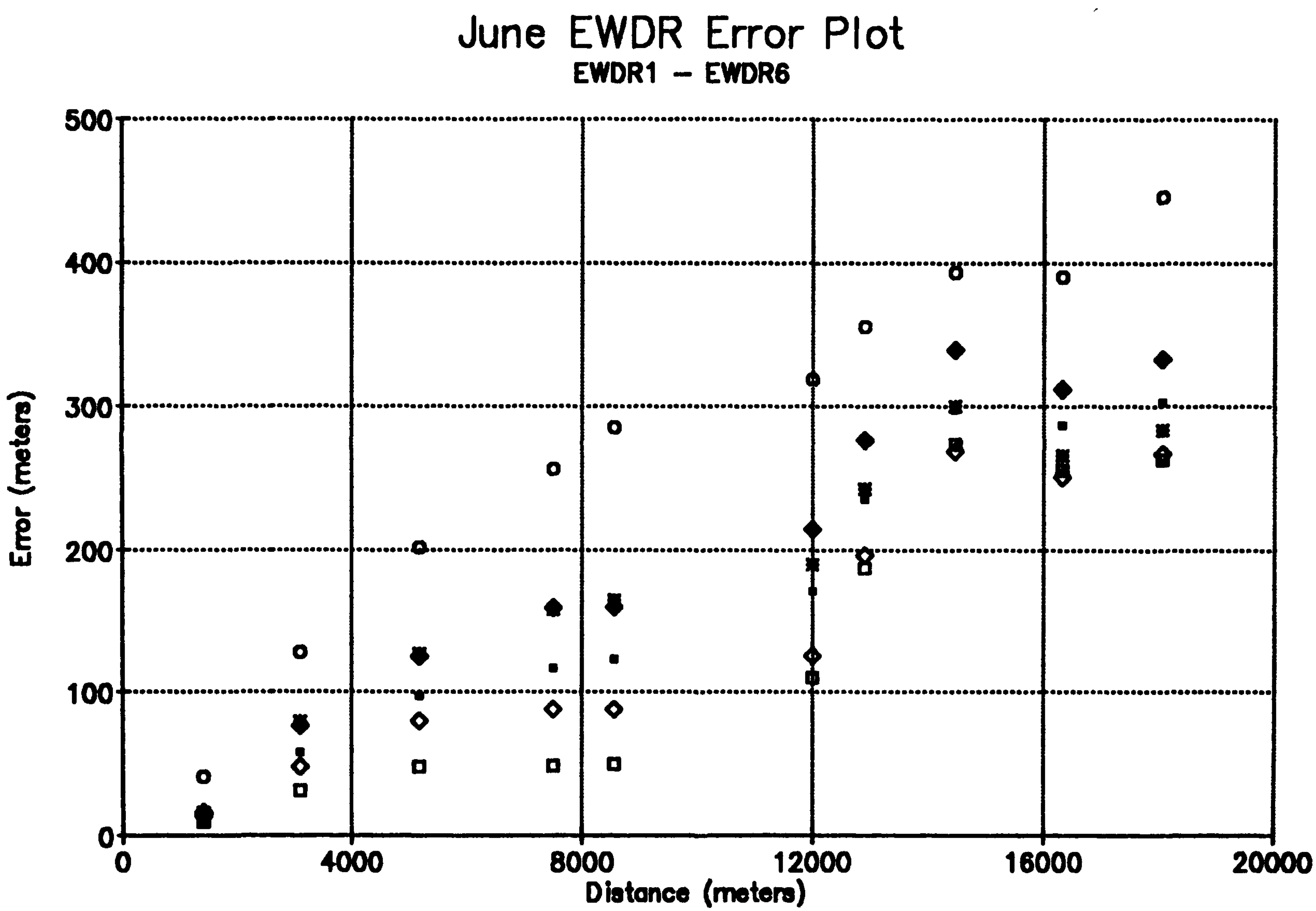

Figure 34. June EWDR Error Plot

Table 35. June SNDR Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
sndr1			sndr4		
2	1040	108	2	1041	119
2A	3326	147	2A	3327	156
3	4298	168	3	4299	179
4	5765	187	4	5766	207
5	7229	208	5	7229	248
6	8909	268	6	8908	317
6A	9666	315	6A	9665	364
7A	11659	379	7A	11657	419
8	13481	376	8	13479	410
sndr2			sndr5		
2	1039	106	2	1039	112
2A	3326	148	2A	3323	149
3	4297	170	3	4293	168
4	5764	193	4	5759	185
5	7227	236	5	7221	223
6	8907	299	6	8900	292
6A	9663	352	6A	9656	336
7A	11657	378	7A	11648	380
8	13478	375	8	13468	382
sndr3					
2	1039	117			
2A	3324	155			
3	4294	181			
4	5760	226			
5	7223	275			
6	8902	352			
6A	9658	407			
7A	11651	471			
8	13472	494			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
2	1039.60	0.76	2	112.32	4.89
2A	3325.30	1.71	2A	151.11	3.84
3	4296.10	2.31	3	173.31	5.78
4	5762.73	2.59	4	199.61	15.30
5	7225.70	3.09	5	238.18	22.76
6	8905.20	3.64	6	305.68	28.08
6A	9661.45	3.87	6A	354.78	30.63
7A	11654.36	4.25	7A	405.31	36.06
8	13475.56	4.66	8	407.24	45.00

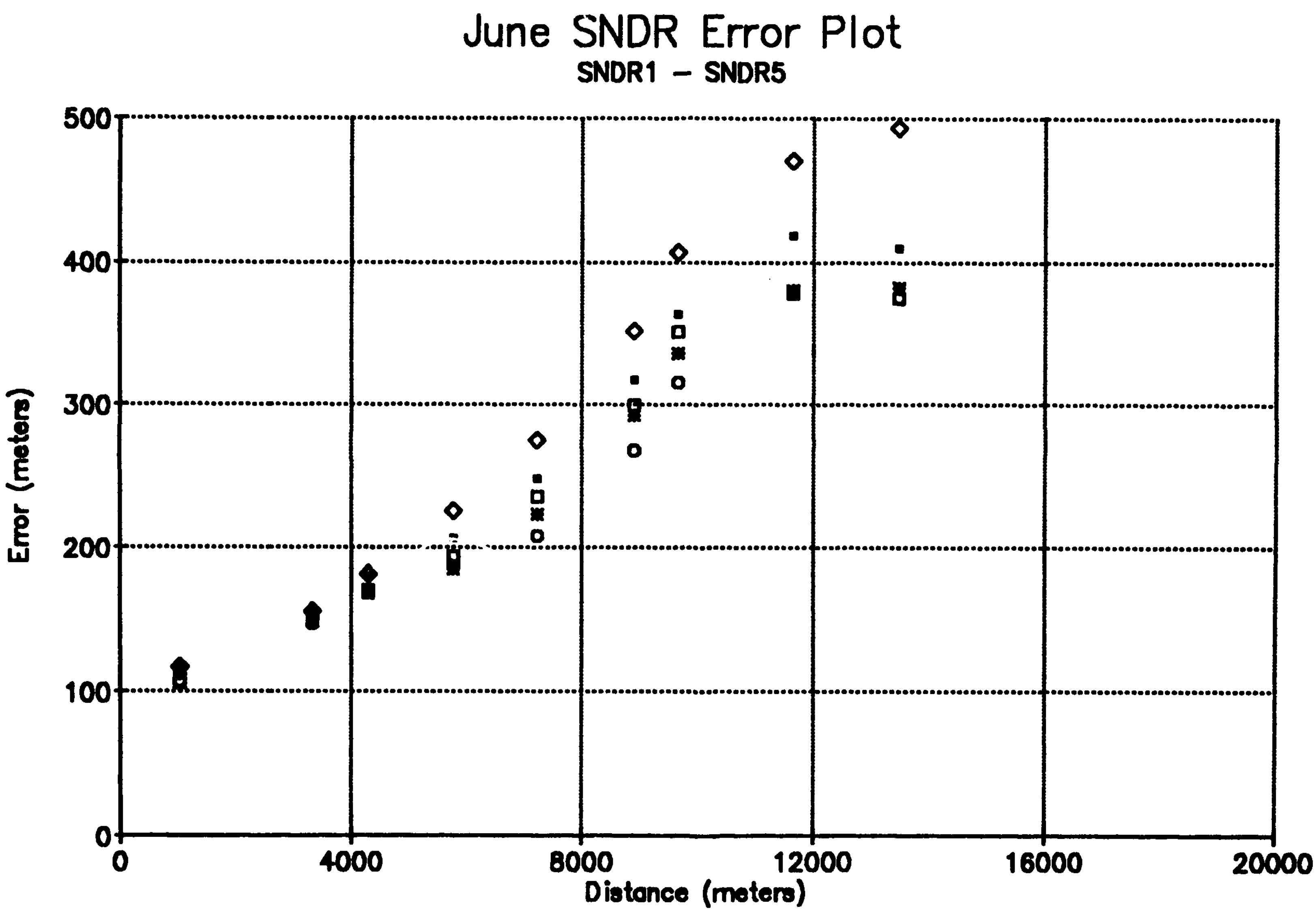

Figure 35. June SNDR Error Plot

Table 36. June NSDR Error Analysis (meters)

Point	Dist	Error	Point	Dist	Error
	nsdr1			nsdr4	
7A	1820	21	7A	1819	26
6A	3813	38	6A	3811	26
6	4569	65	6	4567	37
5	6247	128	5	6245	72
4	7708	201	4	7704	131
3	9174	295	3	9169	217
2A	10144	358	2A	10138	267
2	12430	382	2	12421	295
C	13470	448	C	13461	357
	nsdr2			nsdr5	
7A	1819	33	7A	1818	24
6A	3811	47	6A	3809	57
6	4567	41	6	4565	77
5	6245	67	5	6242	123
4	7706	142	4	7701	182
3	9172	233	3	9166	268
2A	10142	289	2A	10134	321
2	12426	316	2	12418	352
C	13466	381	C	13457	417
	nsdr3				
7A	1819	22			
6A	3810	92			
6	4566	122			
5	6245	174			
4	7704	247			
3	9169	333			
2A	10138	396			
2	12422	430			
C	13461	495			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
7A	1818.74	0.64	7A	25.09	4.12
6A	3810.72	1.25	6A	52.12	22.66
6	4566.55	1.34	6	68.36	30.42
5	6244.81	1.70	5	112.86	39.59
4	7704.69	2.25	4	180.60	41.85
3	9170.17	2.73	3	269.35	41.87
2A	10139.23	3.24	2A	326.21	46.69
2	12423.35	4.16	2	355.20	48.08
C	13463.00	4.41	C	419.53	48.64

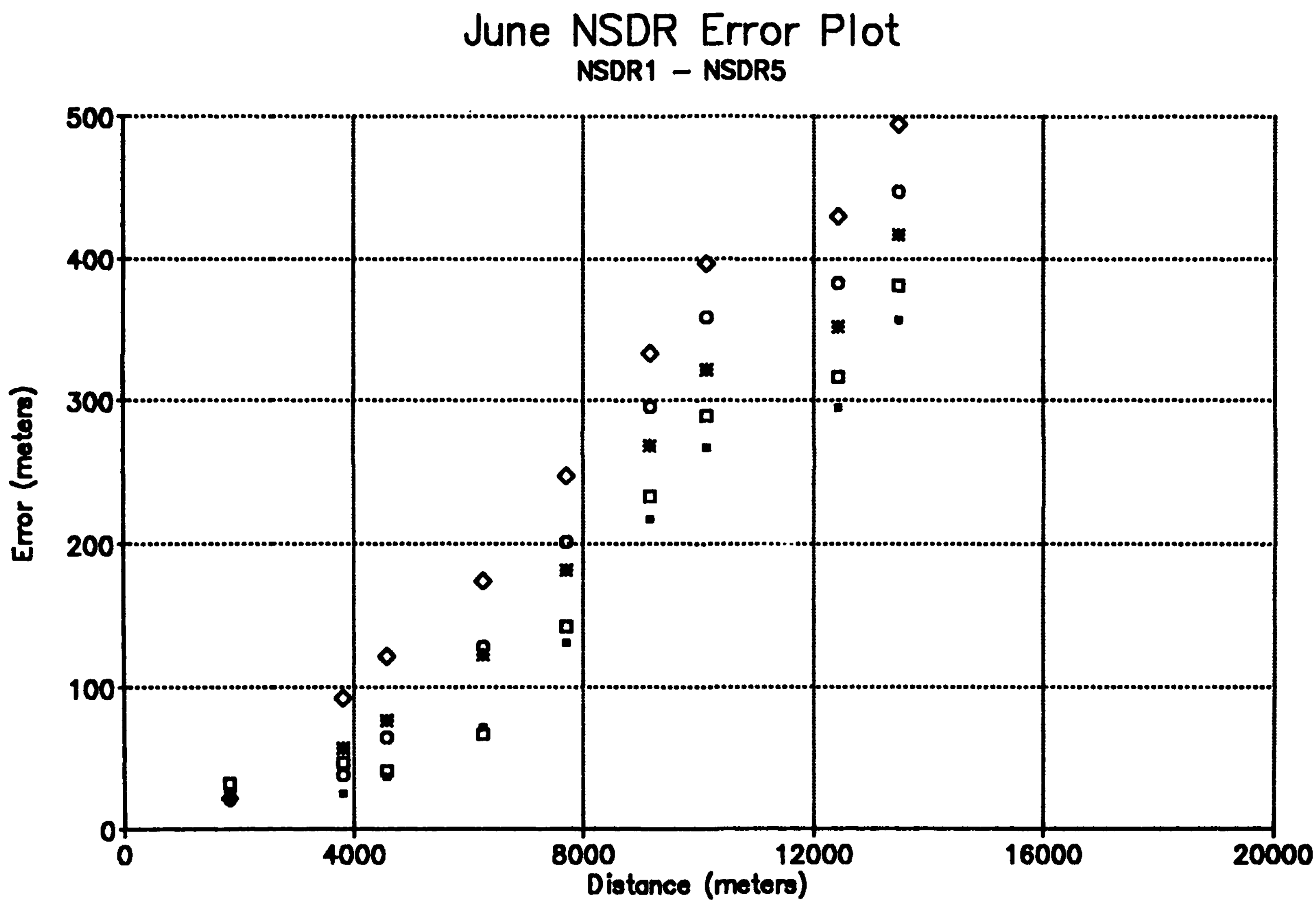

Figure 36. June NSDR Error Plot

Table 37. June WEDR Error Analysis (ETL) (meters)

Point	Dist	Error	Point	Dist	Error
wedr10			wedr13		
3	1744	5	3	1742	8
5	3589	34	5	3586	26
6	5156	80	6	5152	67
7	6050	170	7	6044	151
8	9488	271	8	9479	252
9	10542	307	9	10531	287
10	12854	337	10	12843	319
12	14939	359	12	14925	350
13	16653	391	13	16639	394
14	18036	419	14	18020	430
wedr11			wedr14		
3	1744	5	3	1742	25
5	3588	35	5	3585	82
6	5154	68	6	5150	74
7	6047	151	7	6043	86
8	9484	259	8	9477	97
9	10537	295	9	10529	107
10	12850	331	10	12839	122
12	14933	364	12	14922	137
13	16648	407	13	16635	155
14	18030	422	14	18016	172
wedr12					
3	1743	6			
5	3587	35			
6	5151	54			
7	6045	134			
8	9480	225			
9	10533	268			
10	12844	308			
12	14927	340			
13	16641	385			
14	18023	407			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
3	1743	0.71	3	9.84	7.50
5	3587	1.37	5	42.32	20.33
6	5153	2.18	6	68.66	8.74
7	6046	2.51	7	138.67	28.71
8	9481	4.04	8	220.89	63.90
9	10534	4.53	9	252.64	73.87
10	12846	5.35	10	283.50	81.52
12	14929	6.03	12	310.02	87.11
13	16643	6.67	13	346.26	96.15
14	18025	7.17	14	369.86	99.24

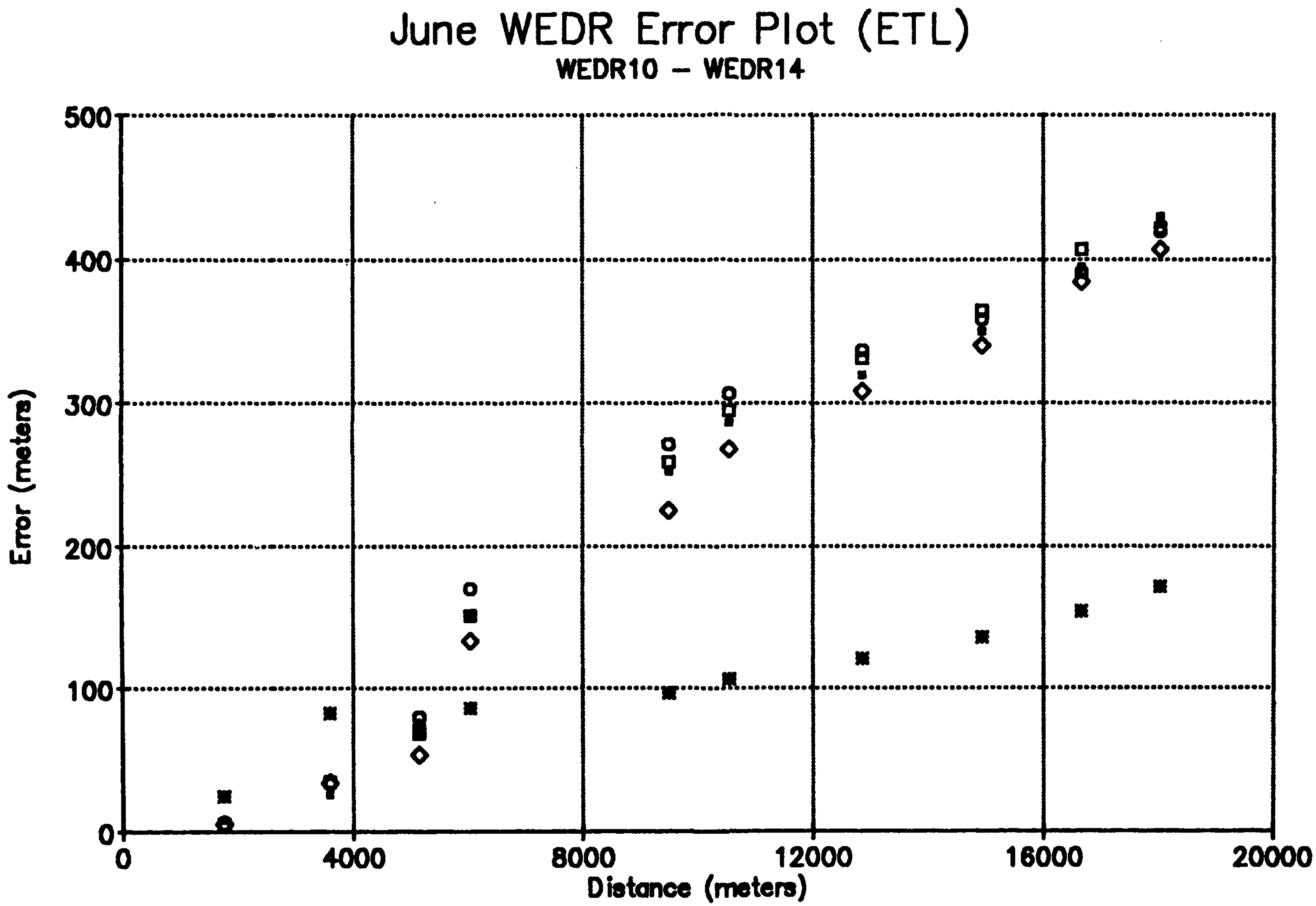

Figure 37. June WEDR Error Analysis (ETL)

Table 38. June EWDR Error Analysis (ETL) (meters)

Point	Dist	Error	Point	Dist	Error
ewdr10			ewdr13		
13	1383	27	13	1382	39
12	3098	89	12	3095	112
10	5182	145	10	5177	187
9	7496	188	9	7488	254
8	8551	202	8	8542	283
7	11989	189	7	11977	296
6	12884	213	6	12871	337
5	14451	234	5	14436	367
3	16294	212	3	16278	349
A	18037	235	A	18020	378
ewdr11			ewdr14		
13	1382	28	13	1381	24
12	3096	89	12	3093	102
10	5179	147	10	5175	176
9	7492	199	9	7486	227
8	8546	217	8	8539	256
7	11982	221	7	11972	306
6	12877	249	6	12866	346
5	14443	275	5	14431	392
3	16285	254	3	16272	391
A	18028	280	A	18013	430
ewdr12					
13	1382	35			
12	3095	113			
10	5178	189			
9	7490	234			
8	8544	247			
7	11980	265			
6	12874	301			
5	14439	328			
3	16281	315			
A	18023	344			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
13	1382	0.73	13	30.51	5.25
12	3095	1.47	12	100.90	10.48
10	5178	2.50	10	168.74	19.06
9	7490	3.47	9	220.48	23.95
8	8545	4.02	8	241.03	28.76
7	11980	5.62	7	255.10	44.53
6	12874	5.99	6	289.29	50.84
5	14440	6.73	5	319.53	58.08
3	16282	7.49	3	304.45	64.36
A	18024	8.12	A	333.58	69.31

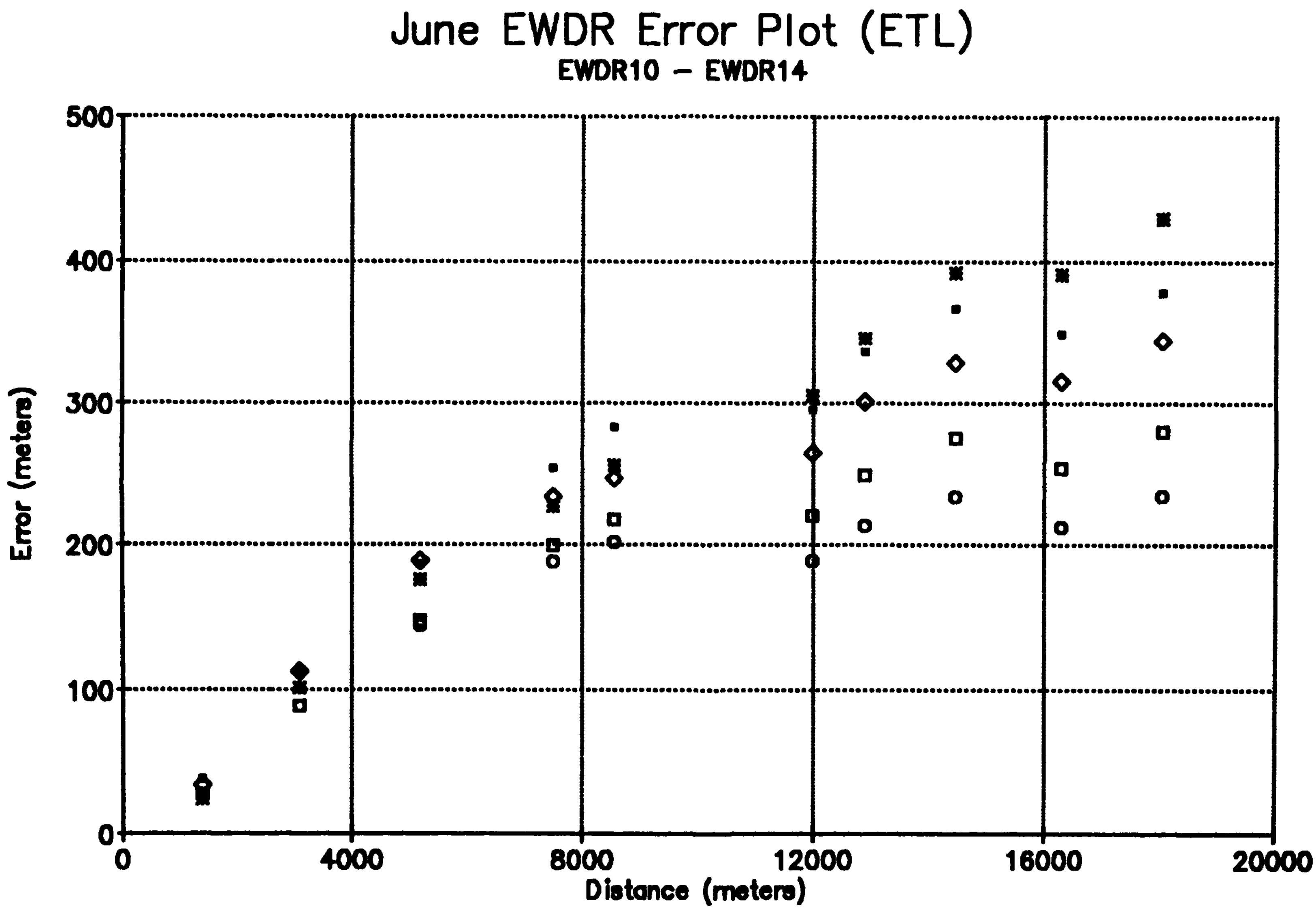

Figure 38. June EWDR Error Analysis (ETL)

Table 39. June SNDR Error Analysis (ETL) (meters)

Point	Dist	Error	Point	Dist	Error
sndr10			sndr13		
2	1038	87	2	1039	96
2A	3321	124	2A	3324	143
3	4291	140	3	4294	158
4	5758	136	4	5762	159
5	7220	152	5	7225	197
6	8898	189	6	8904	246
6A	9654	227	6A	9660	285
7A	11646	244	7A	11653	298
8	13466	220	8	13475	285
sndr11			sndr14		
2	1038	98	2	1038	103
2A	3322	140	2A	3323	148
3	4292	152	3	4293	158
4	5759	142	4	5761	148
5	7222	162	5	7224	175
6	8901	207	6	8903	223
6A	9657	234	6A	9659	262
7A	11649	228	7A	11651	285
8	13470	200	8	13473	270
sndr12					
2	1039	113			
2A	3325	158			
3	4296	172			
4	5765	174			
5	7229	203			
6	8908	263			
6A	9665	317			
7A	11658	381			
8	13480	390			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
2	1038.40	0.60	2	99.31	8.74
2A	3322.81	1.55	2A	142.70	11.26
3	4293.36	1.88	3	156.04	10.46
4	5760.94	2.45	4	151.73	13.38
5	7224.01	2.97	5	177.92	19.64
6	8902.84	3.37	6	225.68	26.50
6A	9658.83	3.60	6A	265.05	33.01
7A	11651.49	4.23	7A	287.07	53.67
8	13472.87	4.70	8	272.73	66.21

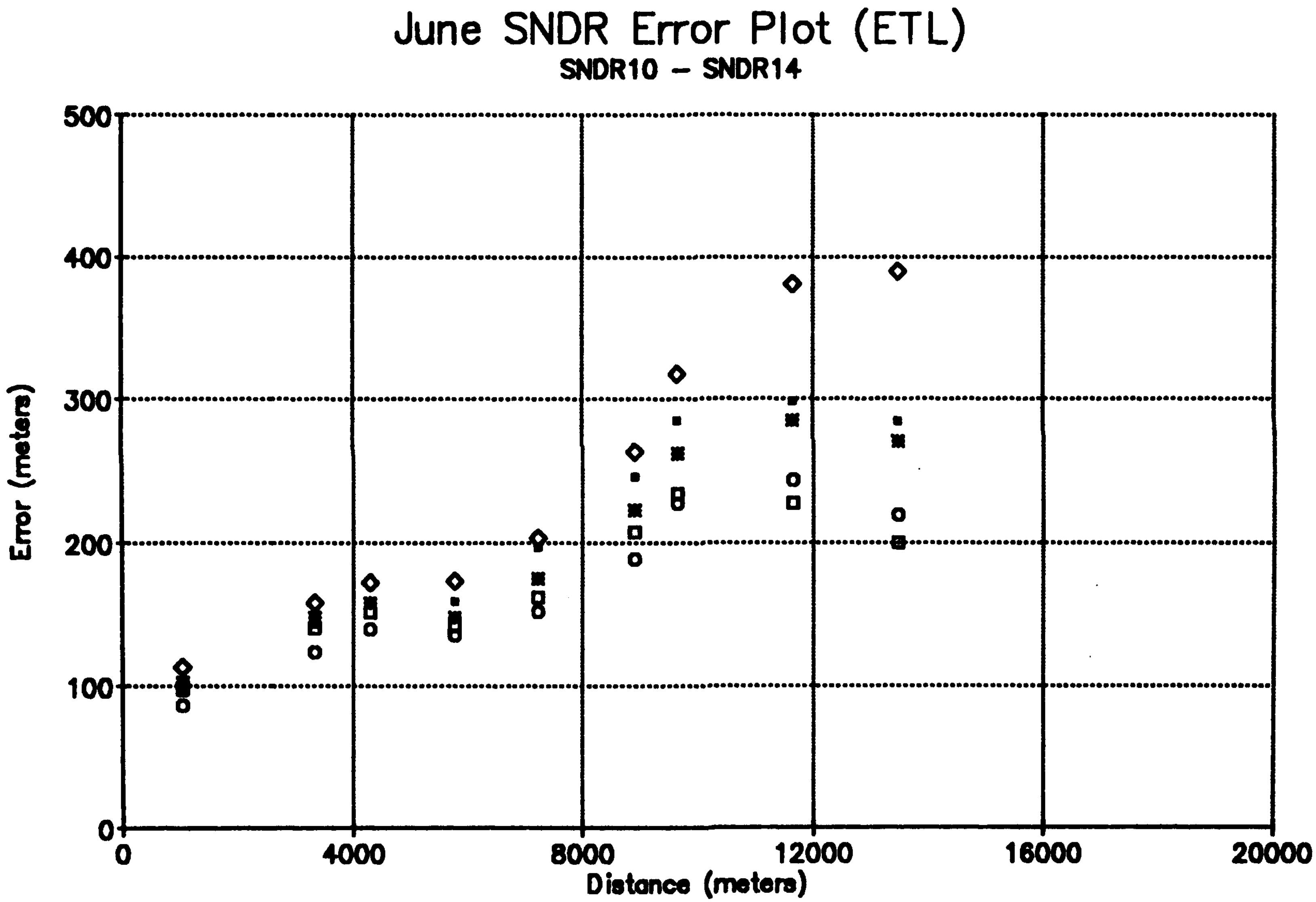

Figure 39. June SNDR Error Analysis (ETL)

Table 40. June NSDR Error Analysis (ETL) (meters)

Point	Dist	Error	Point	Dist	Error
nsdr10			**nsdr13**		
7A	1817	47	7A	1819	64
6A	3809	43	6A	3811	41
6	4564	41	6	4566	35
5	6242	47	5	6245	11
4	7701	93	4	7705	47
3	9163	179	3	9170	132
2A	10131	245	2A	10138	179
2	12416	279	2	12423	216
C	13455	320	C	13463	243
nsdr11			**nsdr14**		
7A	1817	32	7A	1818	63
6A	3809	53	6A	3811	60
6	4563	72	6	4566	53
5	6242	112	5	6244	19
4	7701	164	4	7704	20
3	9165	255	3	9167	99
2A	10132	322	2A	10135	150
2	12416	359	2	12419	181
C	13455	403	C	13458	208
nsdr12					
7A	1819	57			
6A	3812	51			
6	4569	46			
5	6248	4			
4	7708	55			
3	9172	140			
2A	10141	197			
2	12426	235			
C	13466	270			

Combined Statistics

Distance			Error		
Point	Mean	STD	Point	Mean	STD
7A	1818.27	0.77	7A	52.52	11.88
6A	3810.43	1.37	6A	49.63	7.01
6	4565.67	1.80	6	49.49	12.53
5	6244.24	2.09	5	38.58	39.59
4	7703.82	2.46	4	75.66	49.87
3	9167.24	3.04	3	161.30	53.50
2A	10135.50	3.50	2A	218.55	60.25
2	12420.11	4.09	2	253.99	61.34
C	13459.51	4.33	C	288.75	68.04

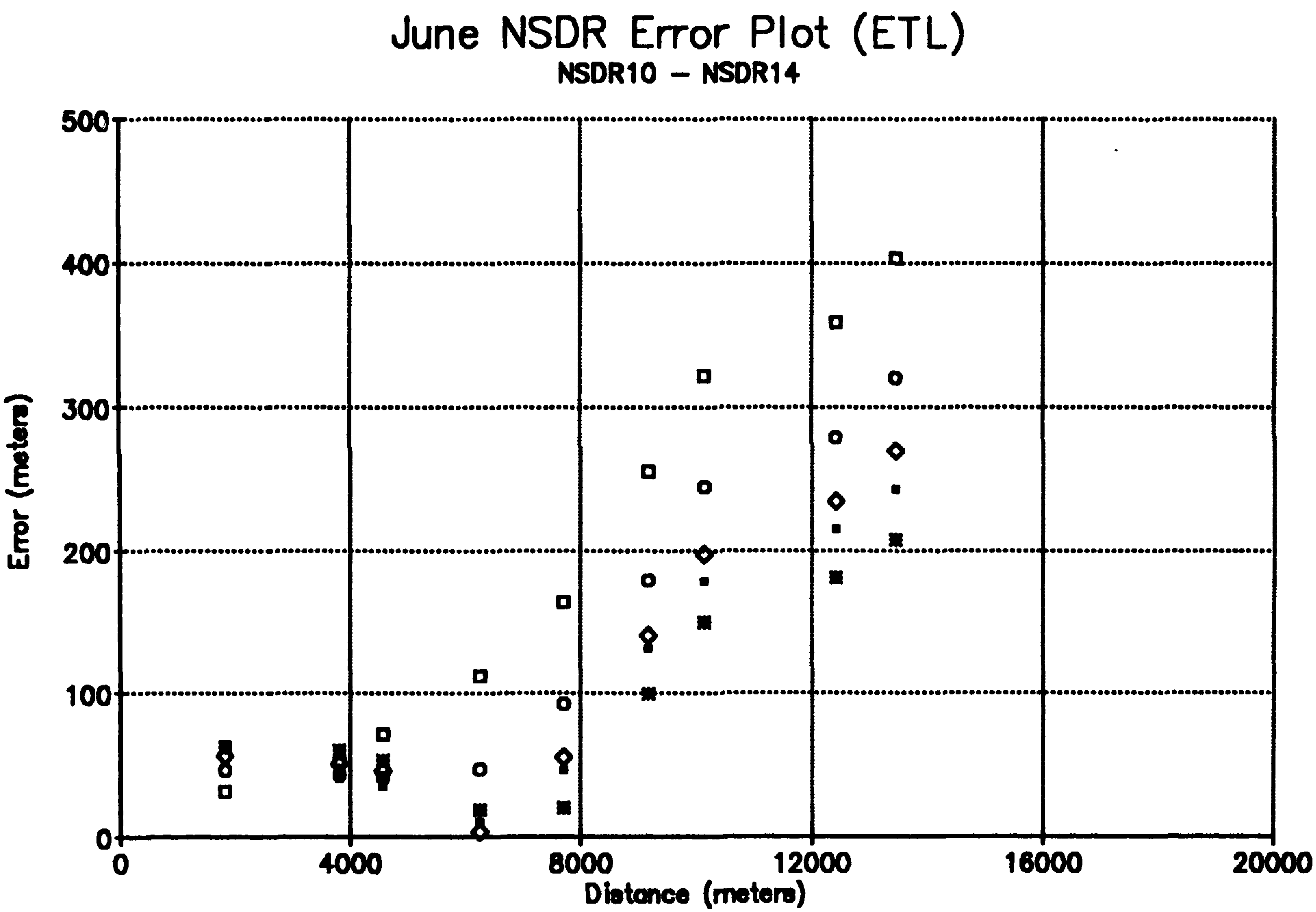

Figure 40. June NSDR Error Analysis (ETL)

Chapter 8

Discussion of Navigation Data Analysis

Comparisons between Dead Reckoning and Map Matching

The Etak Navigator was tested using its two methods of navigation: dead reckoning only and dead reckoning with map matching. Dead reckoning is a relative navigation technique; such techniques are usually characterized by a linear accumulation of errors. That is, such a navigation device may be described as being accurate within a fixed percentage of distance travelled. On the other hand, dead reckoning with map matching behaves somewhat like an absolute navigation device. Such navigation devices are typically characterized by an accuracy that is more or less constant regardless of distance travelled. Thus it can be seen that the method of computing the final accuracy statistics depends upon the method of navigation.

Before the navigation performance statistics were summarized, the data were reviewed to determine the performance characteristics of dead reckoning and map matching. Figure 41 is a map showing the checkpoint positions of the June NSDR test along with the 15-second positions for one of the runs. It can be seen that the error accumulates roughly linearly with distance travelled. Note also that the separate runs are tightly clustered, indicating that the bulk of the error is due to a bias in the sensor calibrations.

A similar plot is shown in Figure 42 for the June SNDR set. Notice a similar linear accumulation of error and a similar clustering of runs. Both the NS and SN

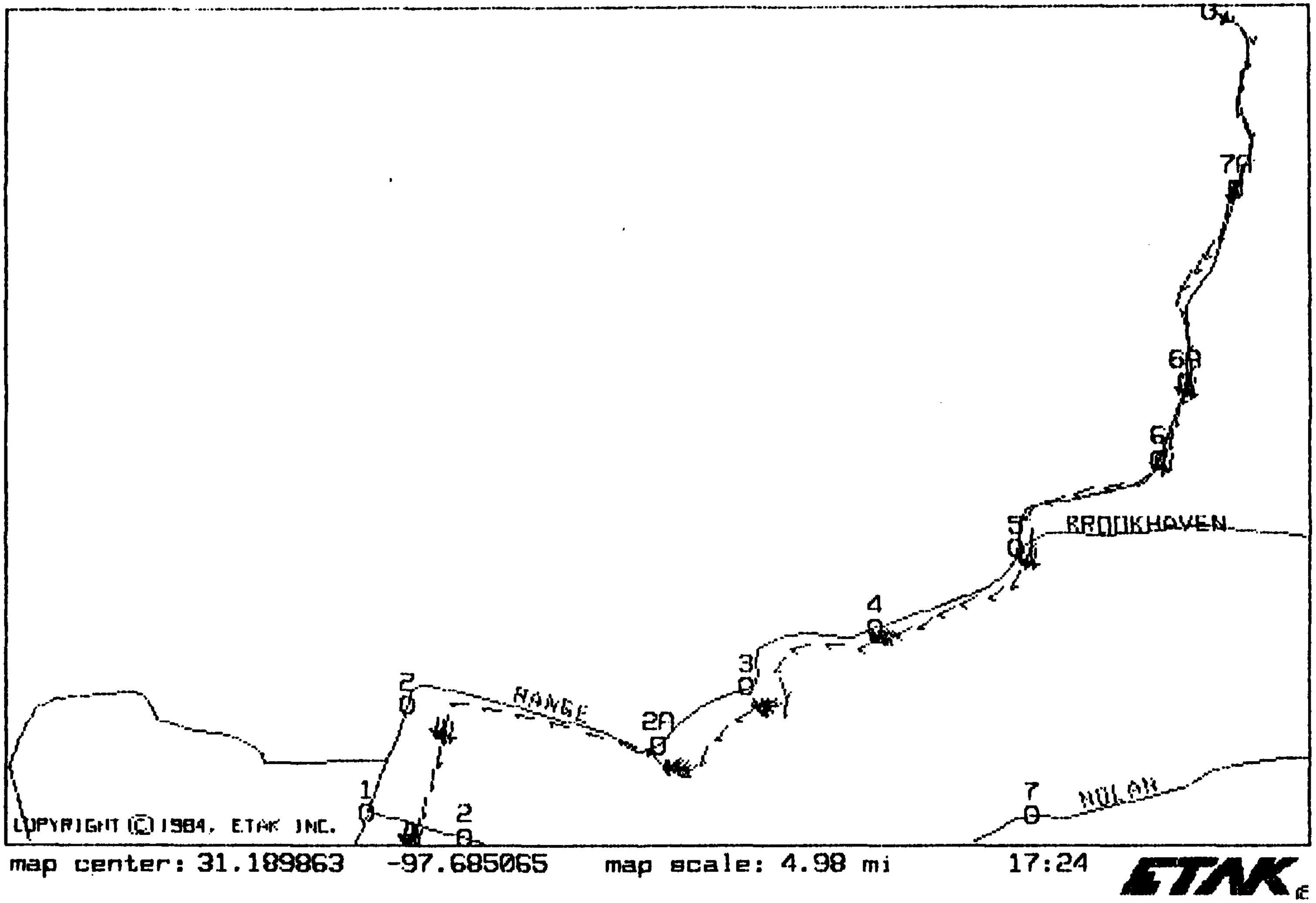

Figure 41. Map Plot for the June NSDR Set

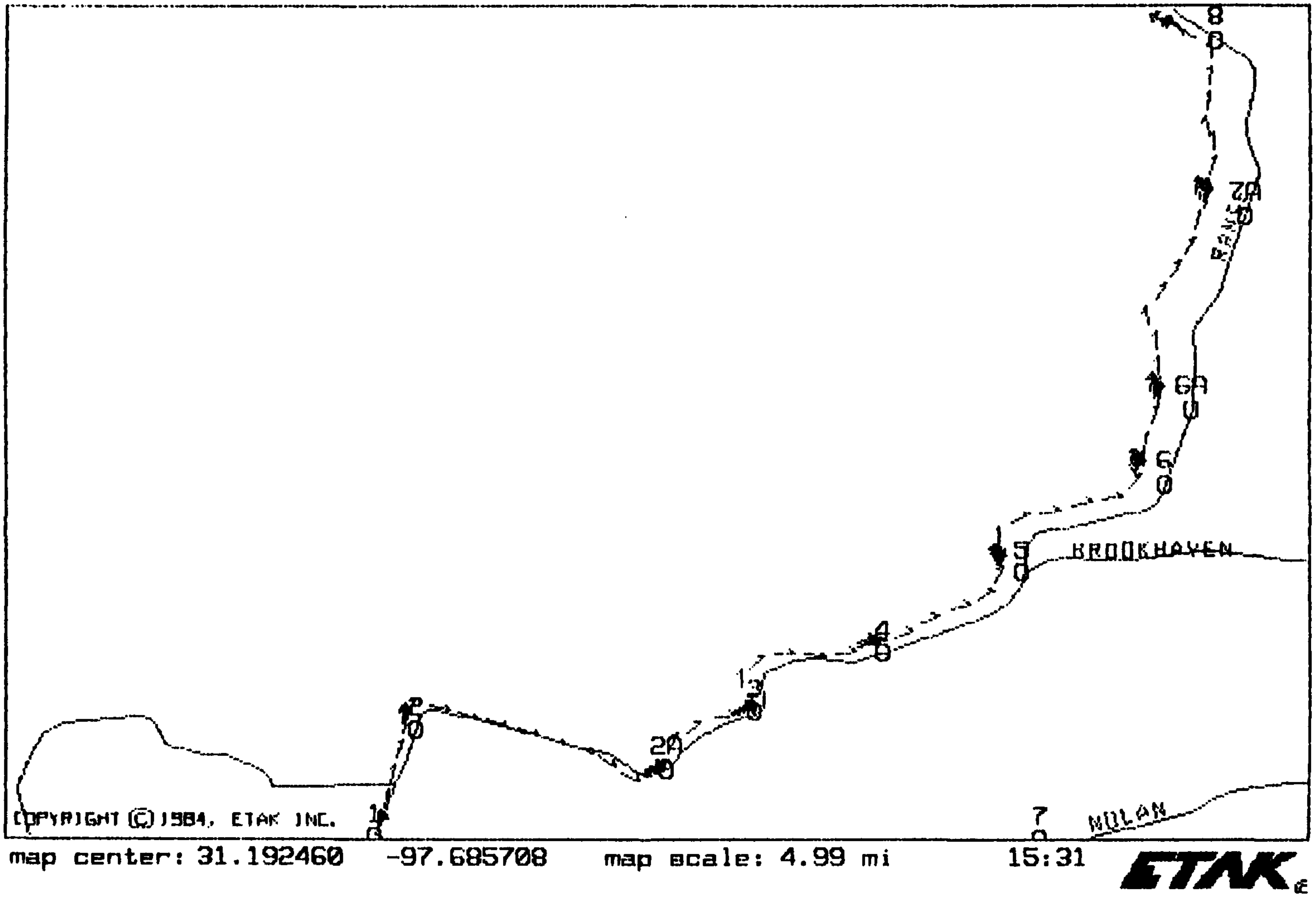

Figure 42. Map Plot for the June SNDR Set

runs show a consistent counter-clockwise bias indicating that the bulk of the error was probably due to a calibration error in the compass. In contrast, Figure 43 plots the June SNDR set as in Figure 42 but with one of the June map-matching runs added. Note how close the map-matched positions remain to the road. Map matching has the effect of eliminating the accumulated errors caused by calibration, computational inaccuracies, and other dead reckoning sensor errors.

A Close Look at Map-Matching Performance

Etak's map-matching algorithm is very flexible, designed to accommodate map omissions and errors, off-road driving, and sensor errors. One example is shown in Figure 44, which plots a part of the June SNMM set. The route near the center of the map crosses a bridge. Interpretation of the DMA paper map was difficult and the bridge was digitized in the wrong place. This problem was noted and thought to have been corrected between the April and June tests. However, the Navigator's position estimates appear to indicate that the bridge is still represented to the west of its position on the map. The map-matching algorithm was able to dead reckon for a short distance and then reacquire the road network.

A more informative look at the effects of map matching can be seen from studying the routes for the June EWMM set shown in Figures 45 through 52. Here all six runs are shown at 15-second intervals. All runs were initiated at checkpoint 14. At checkpoint 13 a tank trail starts, parallel to and approximately 30 meters south of the road. In Etak's civilian applications, dirt roads are not classified as navigable and the map matching algorithm would not consider them, whereas for the Army map all roads were classified as navigable.

The map matching algorithm behaved flawlessly through checkpoints 12 and 11 of Figure 46, but just at the eastern edge of Figure 47 one of the runs broke to the parallel tank trail. This might be explained by the combination of calibration bias plus some sensor noise. At the turn at checkpoint 8 in Figure 48, a second run diverged to the southern tank trail and two of the six runs diverged to the very

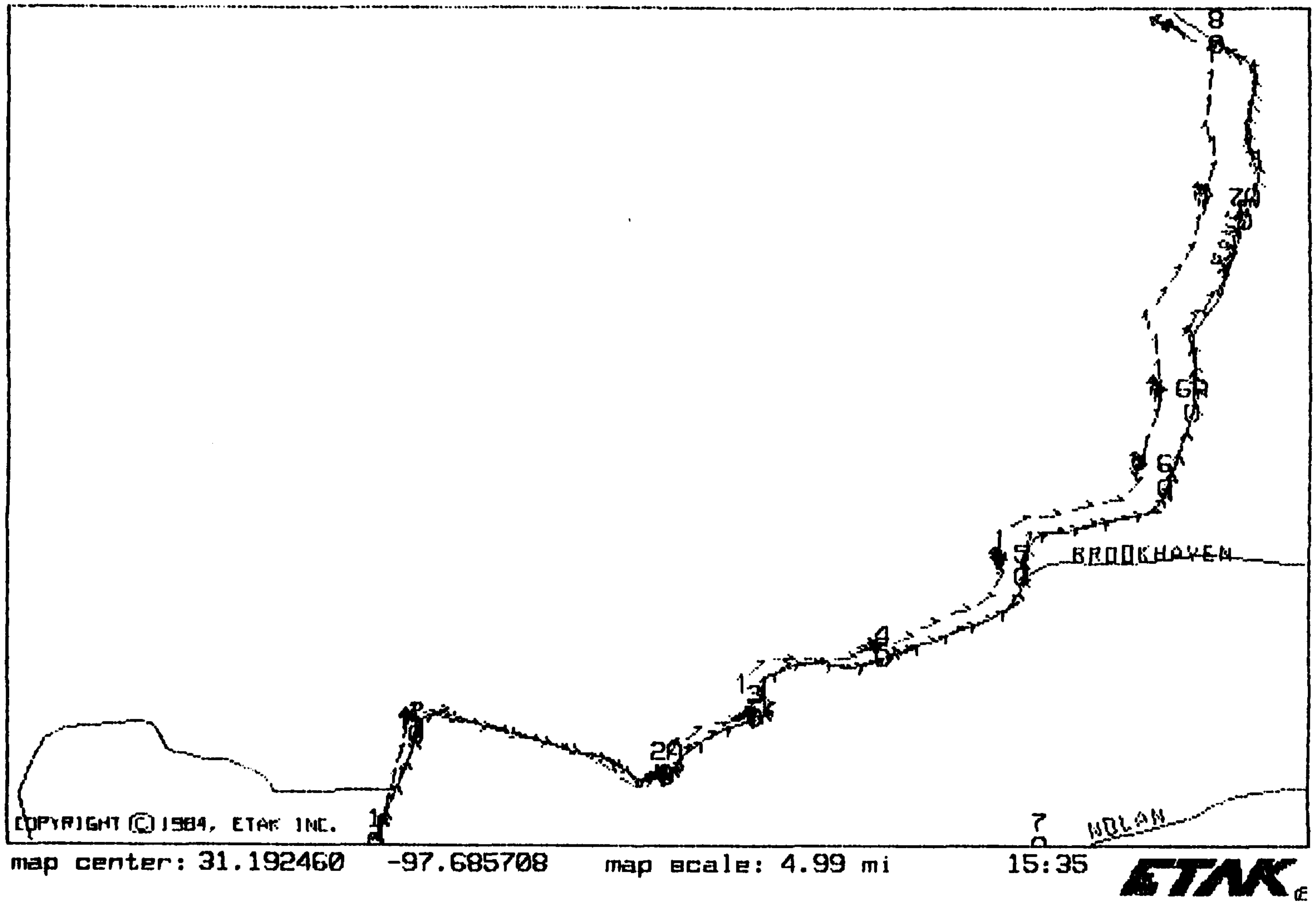

Figure 43. Map Plot for the June SNMM and SNDR Sets

close (within 50 meters) parallel tank trail appearing just to the north of the route. By the western edge of Figure 48 the two runs that had mistakenly updated to the southern tank trail had corrected the error and were again on the proper route. Also at that edge, the northern tank trail is starting to diverge from the true route.

In Figure 49, dead reckoning measurements for the two runs that had updated to the northern tank trail could no longer support map matching to that trail, so the navigation algorithm for those runs went into a phase of dead reckoning only. About one quarter of the way in from the western edge of Figure 49 the map matching algorithm on one of the two errant runs updated to the route. This left a single run off-road and to the north.

In Figure 49, the map matching algorithm on the last remaining errant run updated to the route near checkpoint 6. However, the update corrected its errors in only one dimension and so at the next turn in the road the algorithm again broke away from the route. The run remained to the north of the route until the middle of Figure 51, where the algorithm again placed the vehicle's estimated position on the road. All six runs remained on track to the end of the road at checkpoint 1 in Figure 52. The run that had erred to the north had corrected its position in northing, but because of the map-matching mistake it was still in error in easting.

Figure 53 shows all June map-matching runs at the five-mile scale. The few significant errors are discussed in the sequence above. While the display at this level obscures detail, it gives a very good overview of the accuracy bounds associated with the map-matching algorithm.

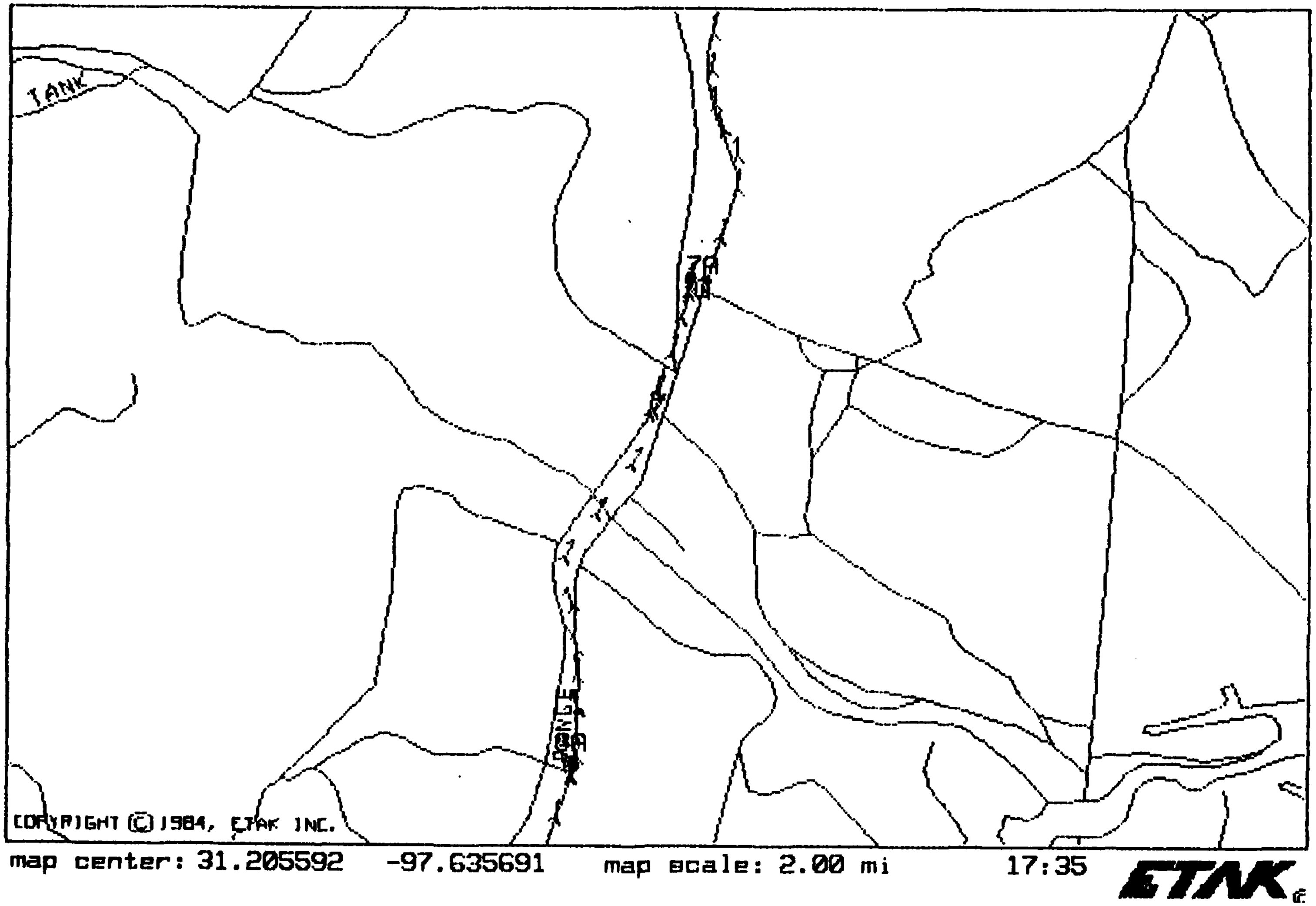

Figure 44. Map Plot at 2-Mile Scale Showing Map Error in Road over Bridge

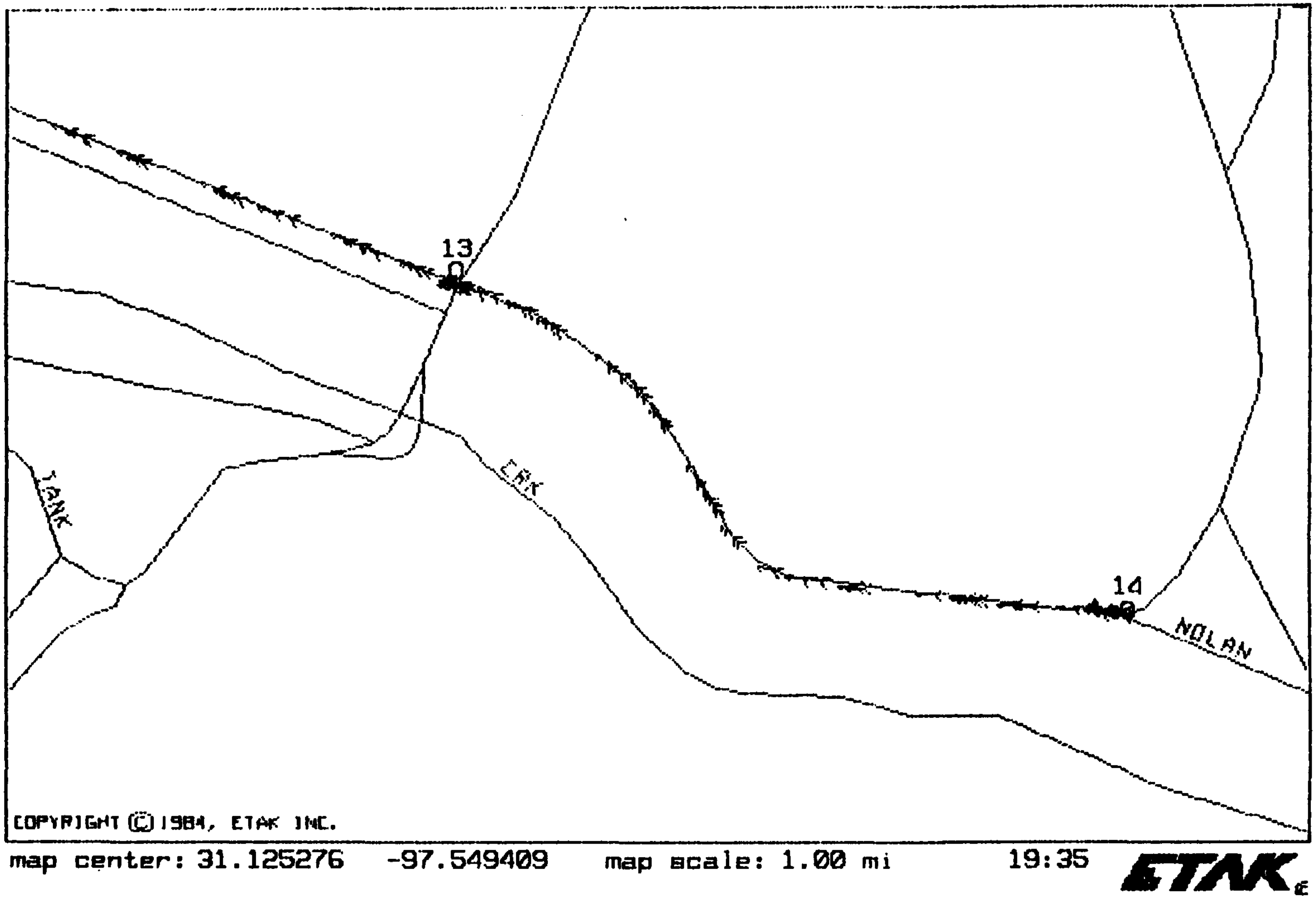

Figure 45. Map Plot at 1-Mile Scale for June EWMM Runs (1 of 8)

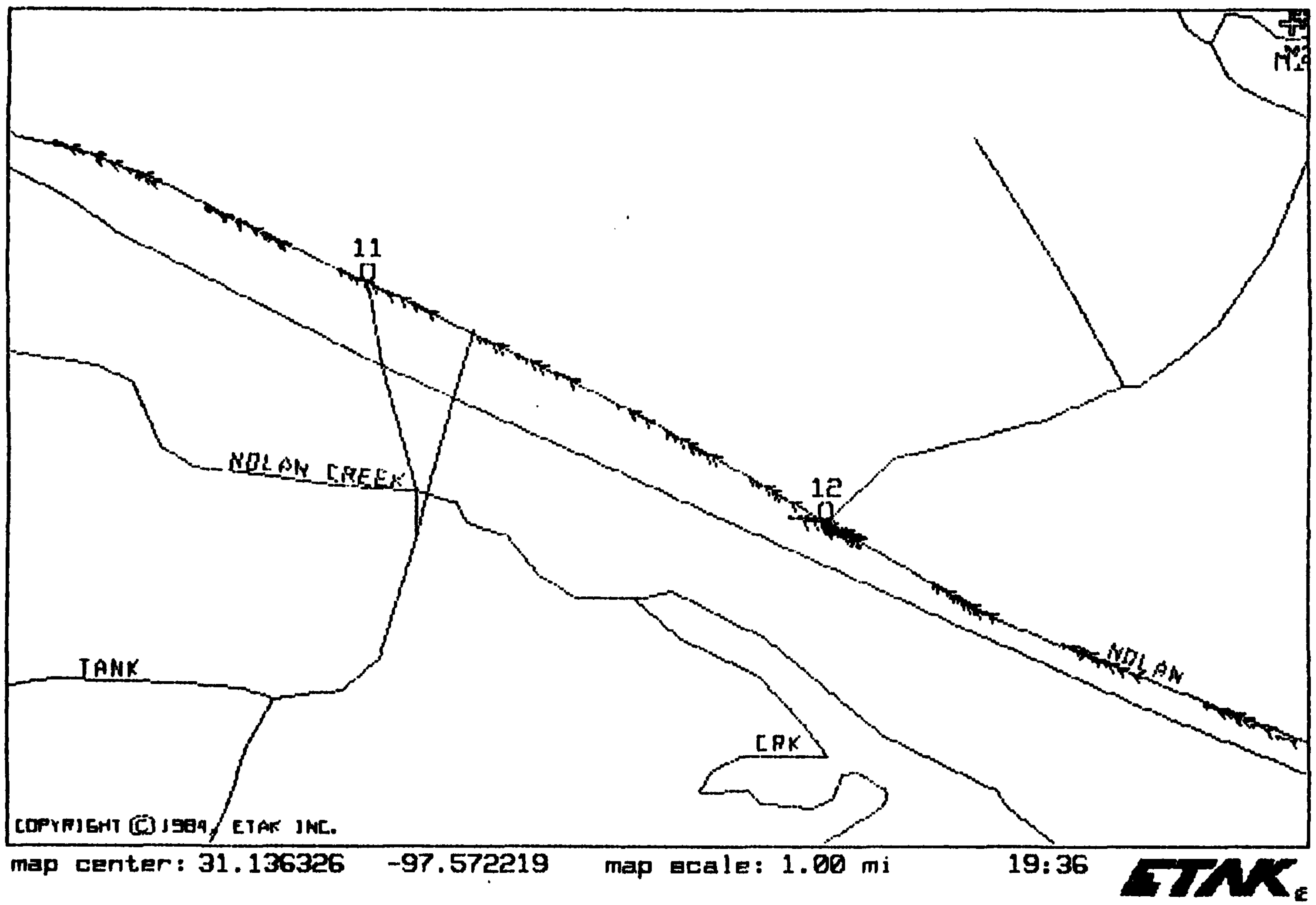

Figure 46. Map Plot at 1-Mile Scale for June EWMM Runs (2 of 8)

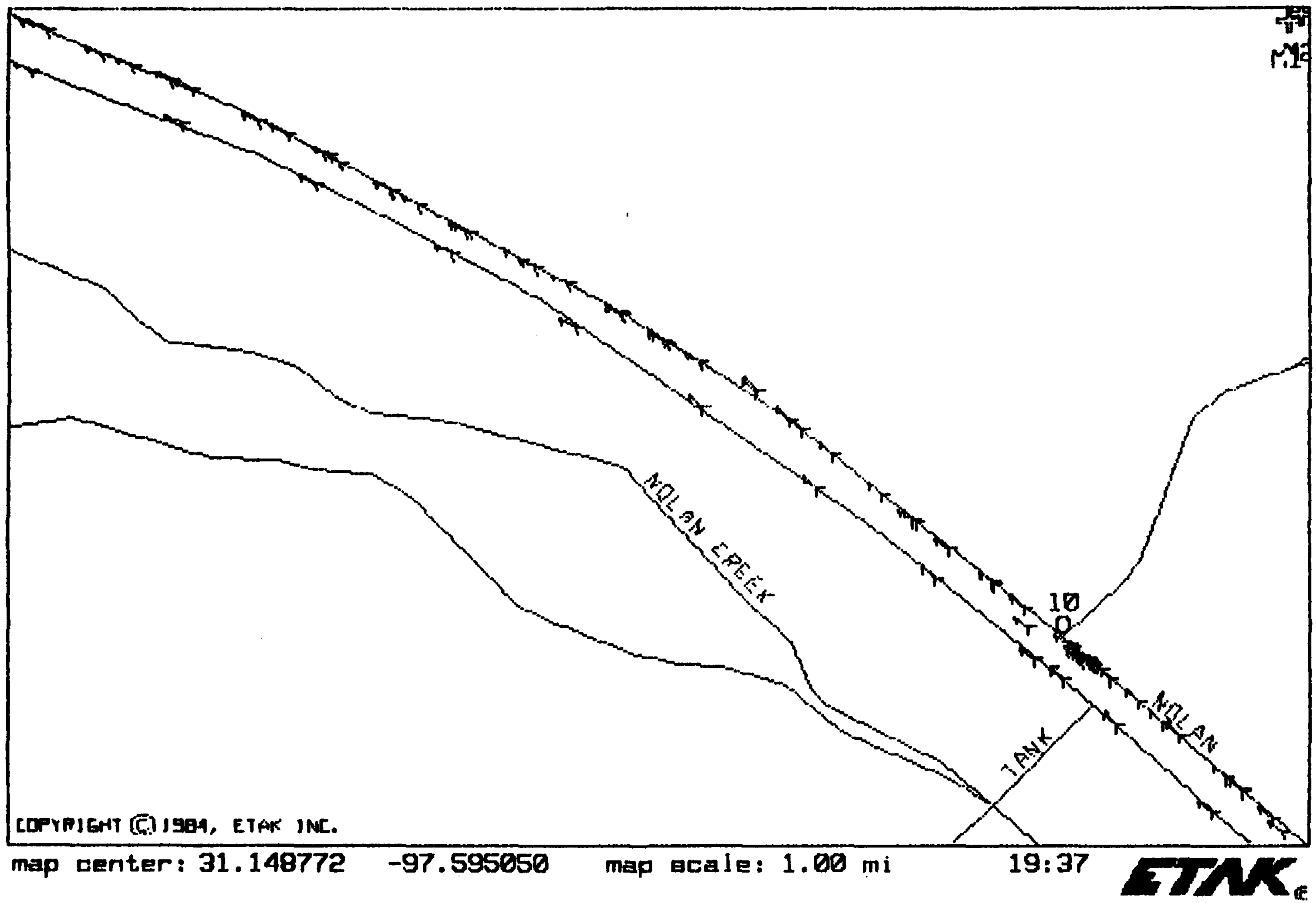

Figure 47. Map Plot at 1-Mile Scale for June EWMM Runs (3 of 8)

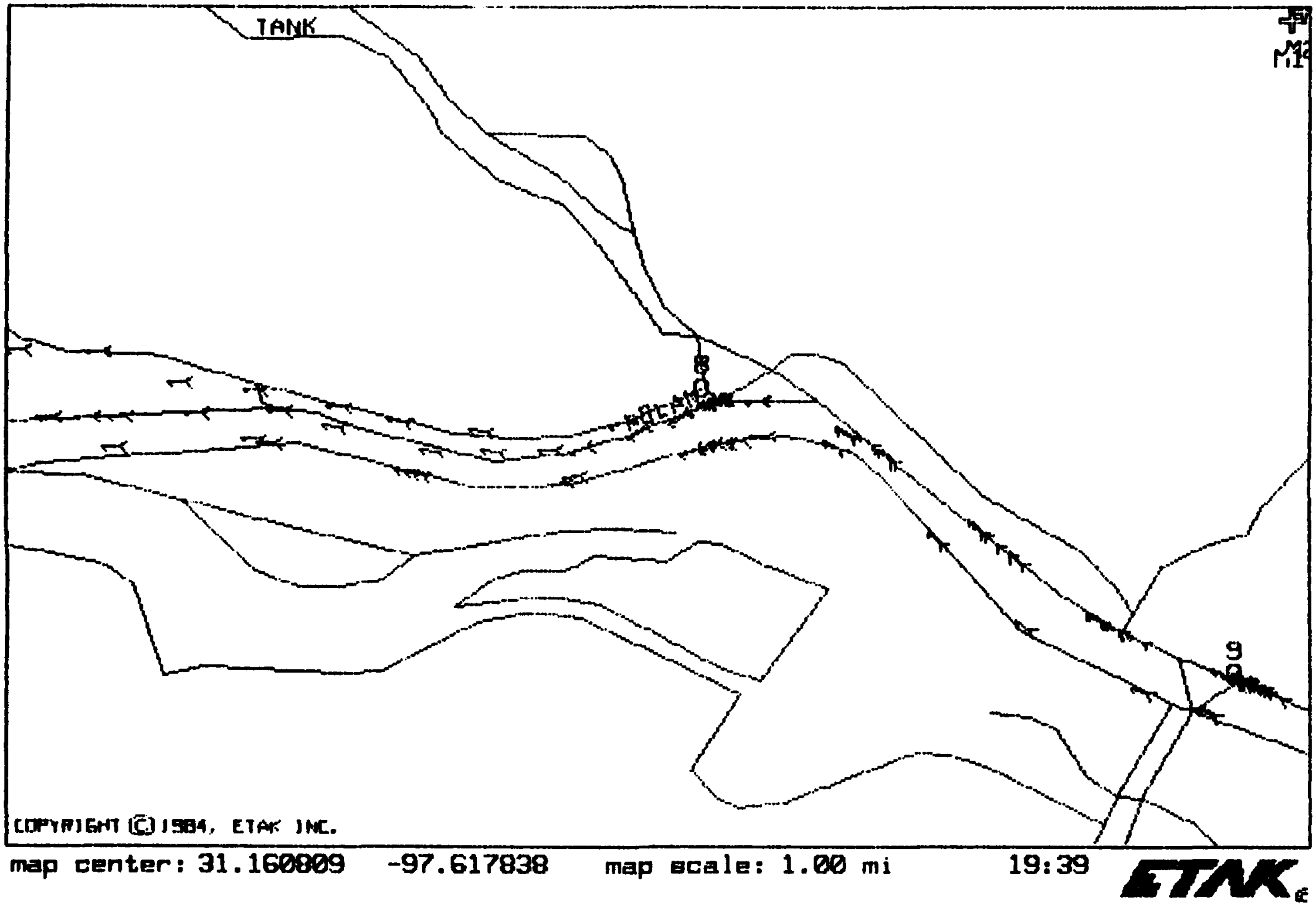

Figure 48. Map Plot at 1-Mile Scale for June EWMM Runs (4 of 8)

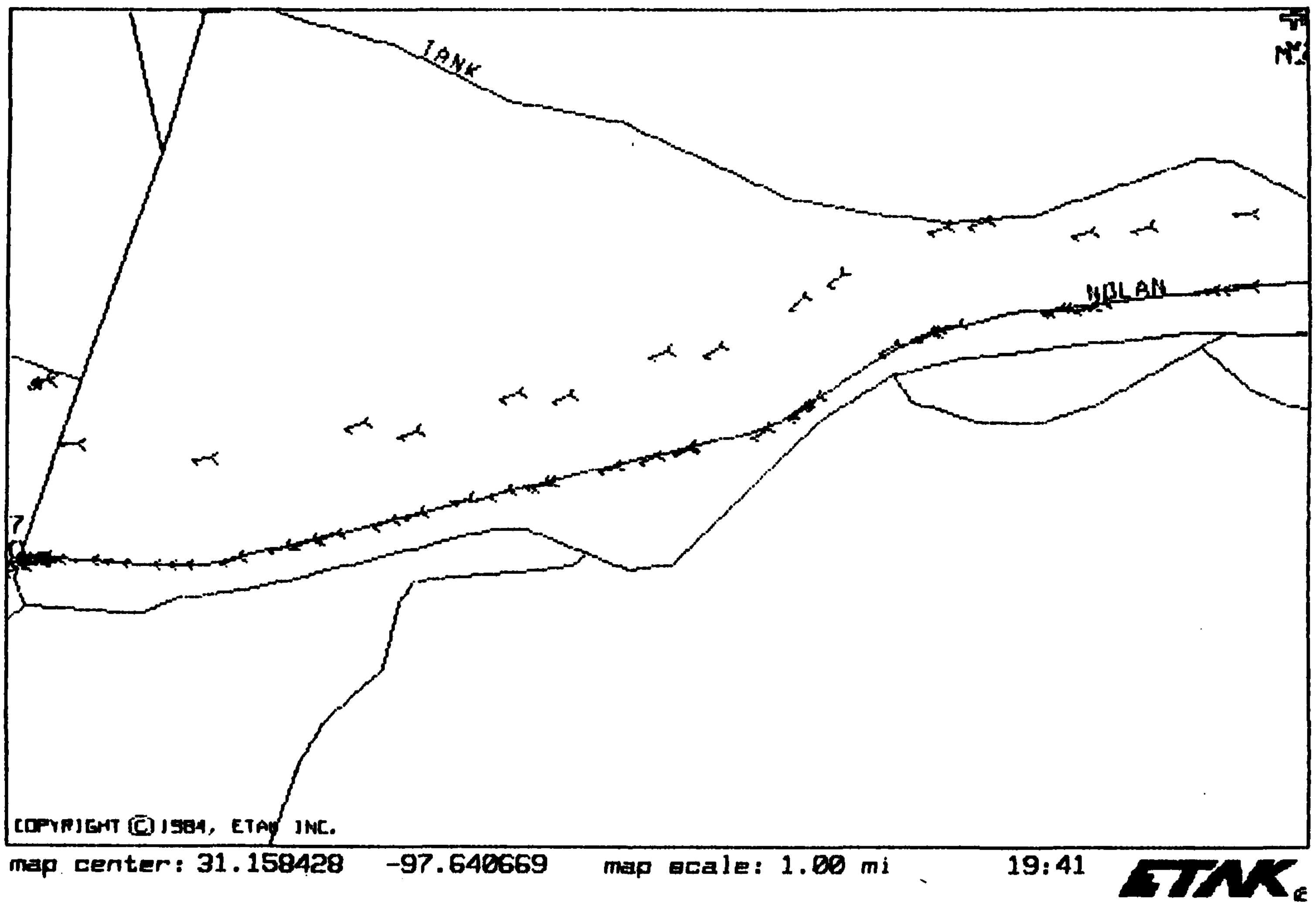

Figure 49. Map Plot at 1-Mile Scale for June EWMM Runs (5 of 8)

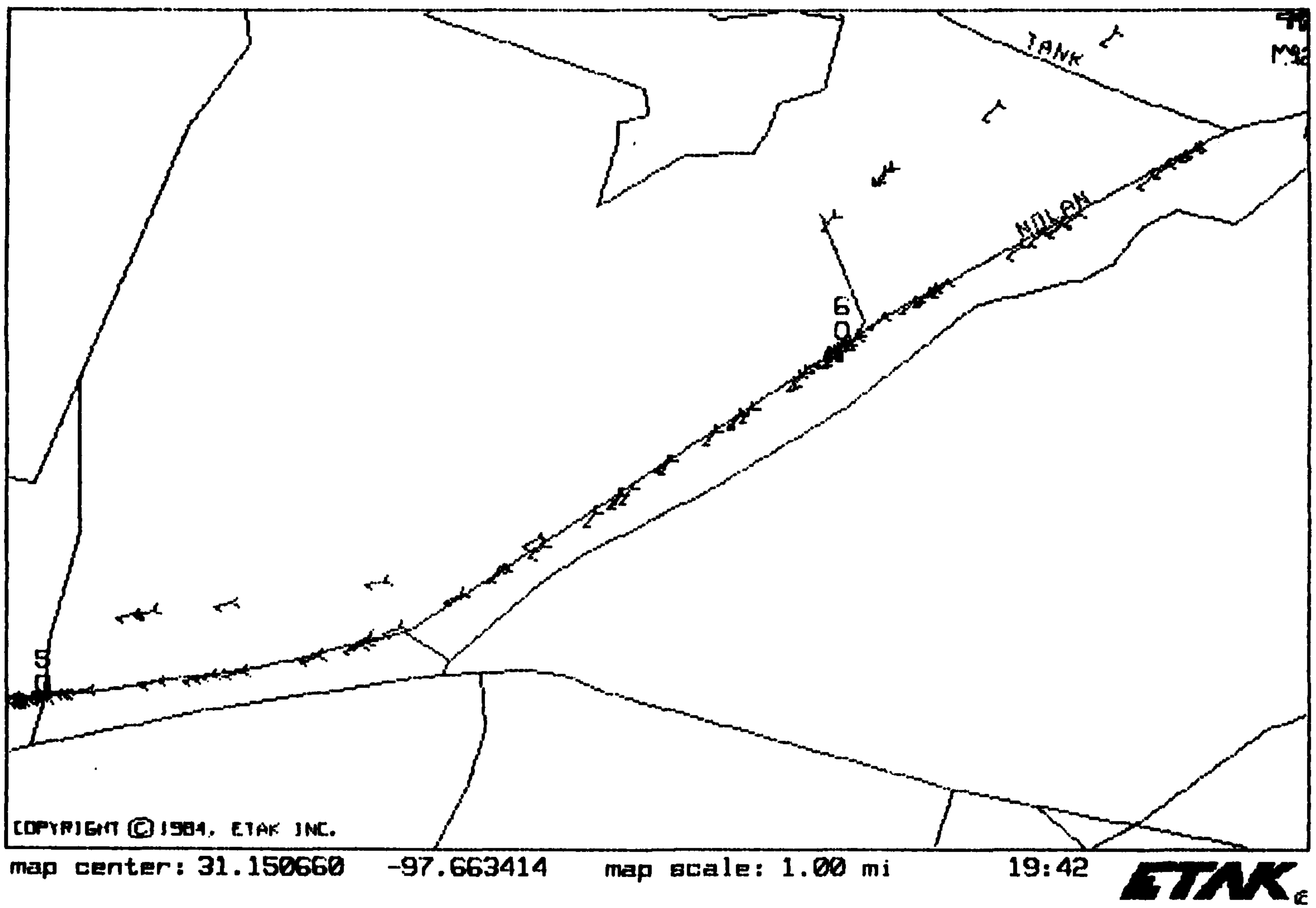

Figure 50. Map Plot at 1-Mile Scale for June EWMM Runs (6 of 8)

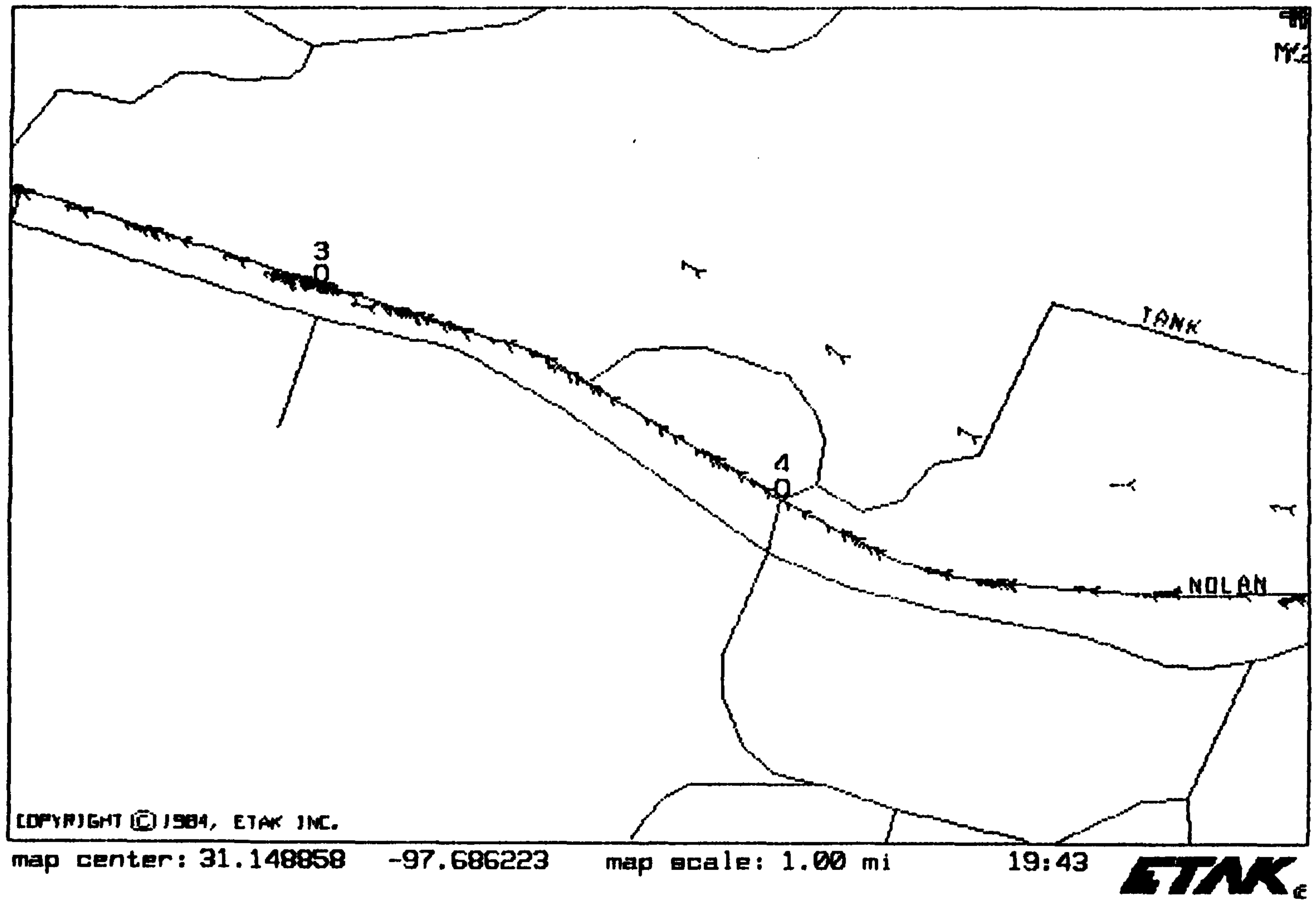

Figure 51. Map Plot at 1-Mile Scale for June EWMM Runs (7 of 8)

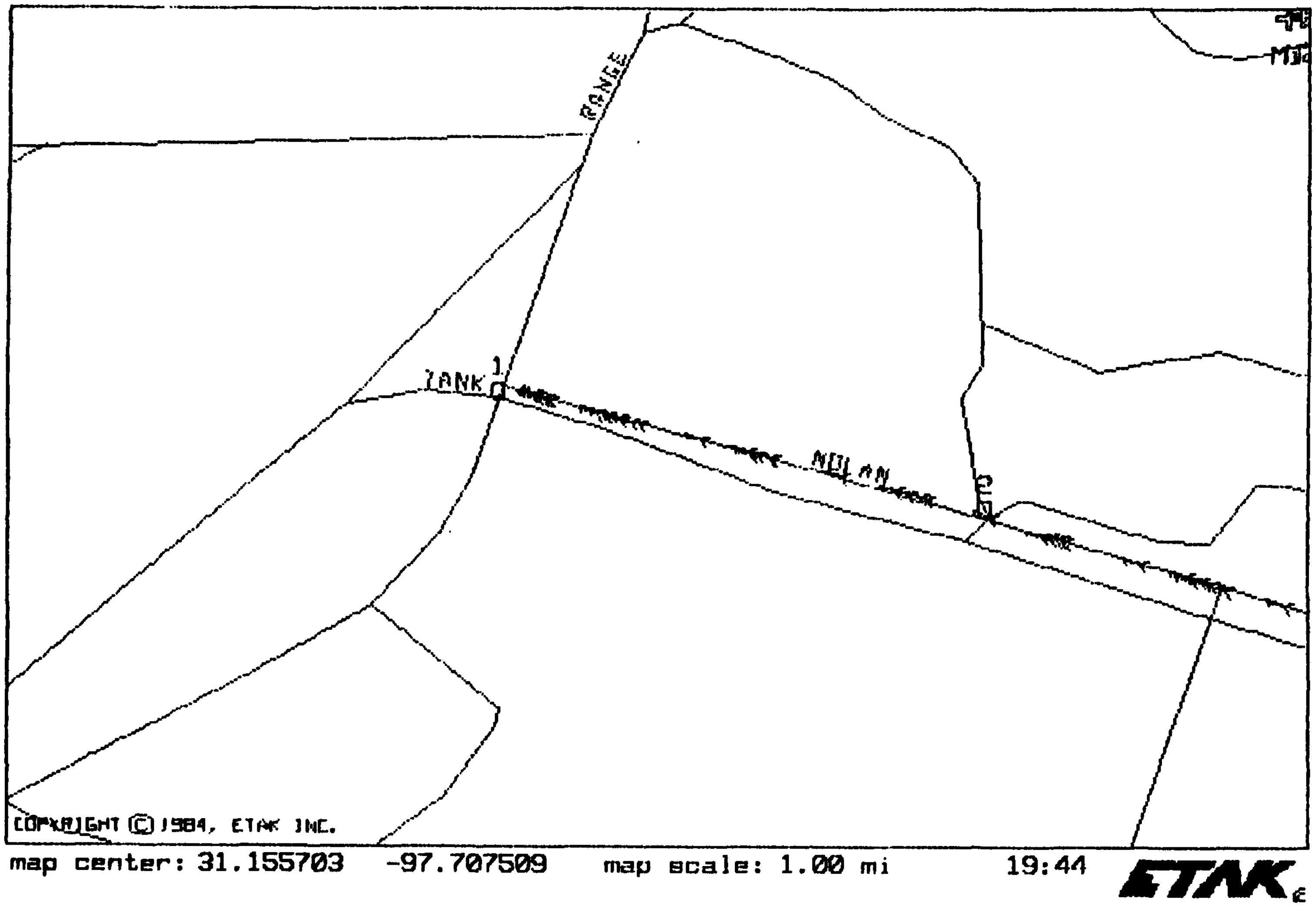

Figure 52. Map Plot at 1-Mile Scale for June EWMM Runs (8 of 8)

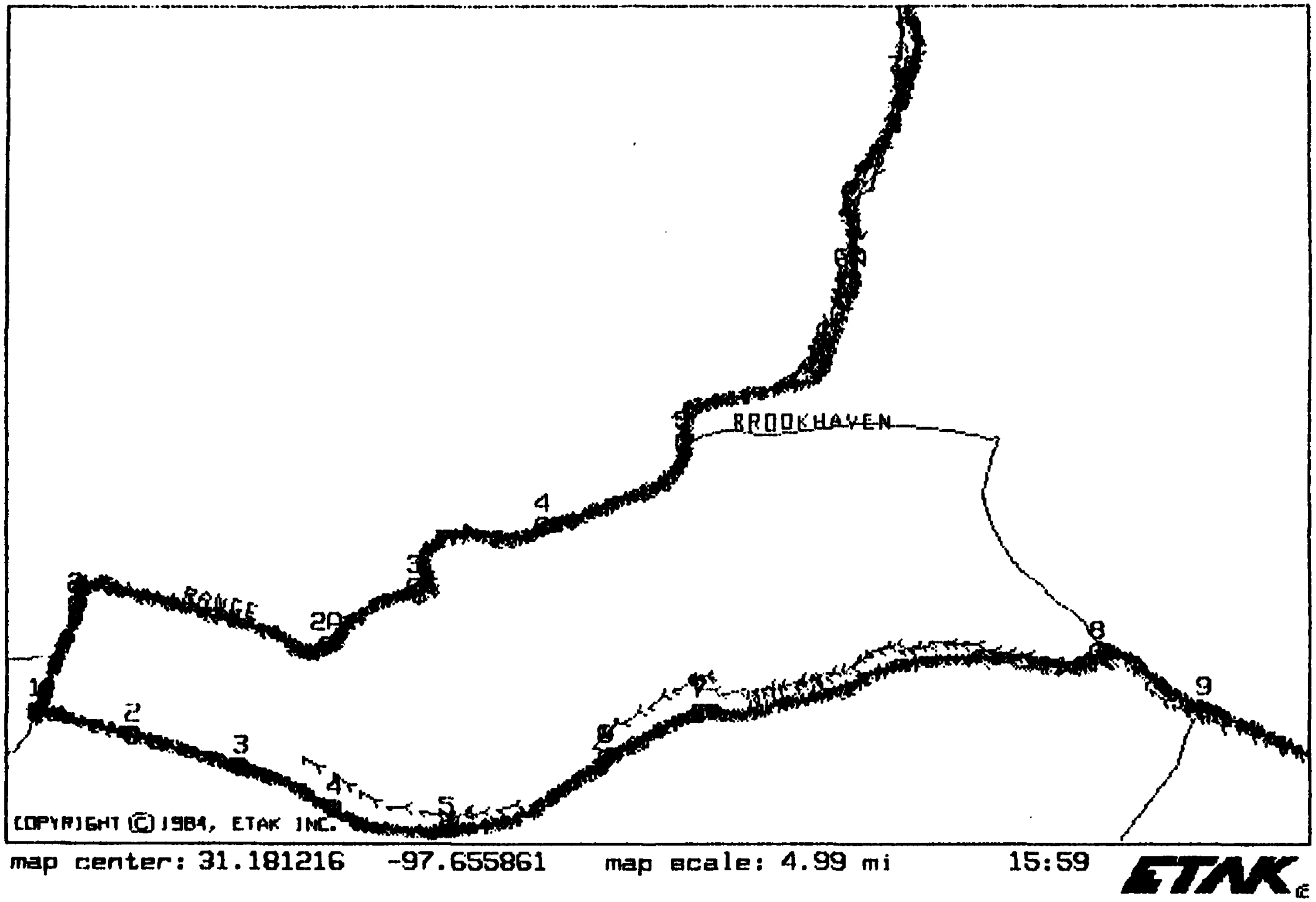

Figure 53 Compendium of June Map-Matching Runs

Chapter 9

Summary of Navigation Results and Discussion

Summary of Dead Reckoning Performance

To summarize the performance of the Navigator using dead reckoning, first the percentage error of each run was computed by Equation (1):

$$P_{ij} = 100 \times \frac{E_{ij}}{D_{ij}} \qquad (1)$$

where:

P_{ij} = percentage error of run i in set j

E_{ij} = magnitude of error at final point of run i in set j (meters)

D_{ij} = length (distance travelled) of run i in set j (meters)

The average percentage error for each set of dead reckoning runs was then computed by Equation (2) and tabulated in the first three columns and four rows of Table 41:

$$P_j = \frac{\sum_{i=1}^{N_j} P_{ij}}{N_j} \qquad (2)$$

where:

P_j = average percentage error for set j

N_j = number of test runs in set j

P_{ij} = percentage error of run i in set j from Equation (1)

An example of a set of runs is the six east-west runs EWDR1 to EWDR6 of June 1990. (Note that j is not shown as such in the tables.)

To summarize errors according to category, an assemblage of seven averages over collections of sets (groups) was made using Equation (3):

$$P_{group} = 100 \times \frac{\sum_{j \text{ in group}} \sum_{i=1}^{N_j} E_{ij}}{\sum_{j \text{ in group}} \sum_{i=1}^{N_j} D_{ij}} \qquad (3)$$

where:

P_{group} is the average percentage error in a selected group of sets

The fourth entry in each row of Table 41 comes from applying Equation (3) to the first three entries of the row. It corresponds to averaging all of the dead-reckoning runs over one route *in one direction* (e.g., all WEDR runs) regardless of date.

The fifth entry in each column of Table 41 comes from applying Equation (3) to the first four entries of the column. It corresponds to averaging all of the dead-reckoning runs of a given series regardless of route and direction.

While the numbers range from the best value of 0.2% (NSDR3, April) to two runs at 3.7% (SNDR3 and NSDR3, June), the weighted averages by date and by route show relatively consistent performance. In addition, an overall average, weighted by distance and combining all of the data, was computed from Equation (4). The weights do not appear explicitly in the equation because they cancel out.

$$P = 100 \times \frac{\sum_{all\,j} \sum_{i=1}^{N_j} E_{ij}}{\sum_{all\,j} \sum_{i=1}^{N_j} D_{ij}} \qquad (4)$$

where:

P = The average weighted percentage error in dead reckoning navigation

This overall average provides a conservative performance estimate of under 2 %.

Table 41. Dead Reckoning Errors (%)

Route	April	June	ETL	Average by Route
West-East	2.1	0.9	2.1	1.6
East-West	1.0	1.8	1.9	1.6
South-North	1.3	3.0	2.0	2.2
North-South	0.3	3.1	2.1	2.1
Average by Date	1.2	2.1	2.0	—
Average Overall	—	—	—	1.9

Analysis of the travel path distance measurements shows that the wheel sensors measure distance within 0.1% or as much as 20 times better than the total navigation error. Clearly, the estimation of heading is the major constraint on navigation accuracy. A heading error of slightly more than 1 degree can account for a 2% error. Etak uses two wheel sensors and a flux gate compass in combination to estimate heading. Many sources of errors, including magnetic dip, magnetic anomalies, quantization errors, and wheel slip can contribute to generate errors. Because of the clustering of the errors from a given set, it appears that over half the error magnitude was due to calibration.

Summary of Map-Matching Performance

Because map matching performance approximates that of an absolute navigation device, performance summaries were computed differently from the dead-reckoning summaries presented above. The measurements from each checkpoint were used to compute an independent measure of position error. All checkpoint navigation errors from a set of runs were then averaged. The results are tabulated in Table 42 and averaged by route and date as before.

A histogram of the map matching position errors is plotted in Figure 54. Sixty-seven percent of the points lie within 50 meters, and ninety-five percent within 88 meters of the correct position.

Table 42. Average Map-matching Errors (meters)

Route	April	June	ETL	Average by Route
West-East	50.0	40.0	—	43.7
East-West	27.7	68.2	—	53.2
South-North	35.6	40.4	—	38.6
North-South	43.3	46.7	—	45.4
Average by Date	39.2	49.6	—	—
Average Overall	—	—	—	45.7

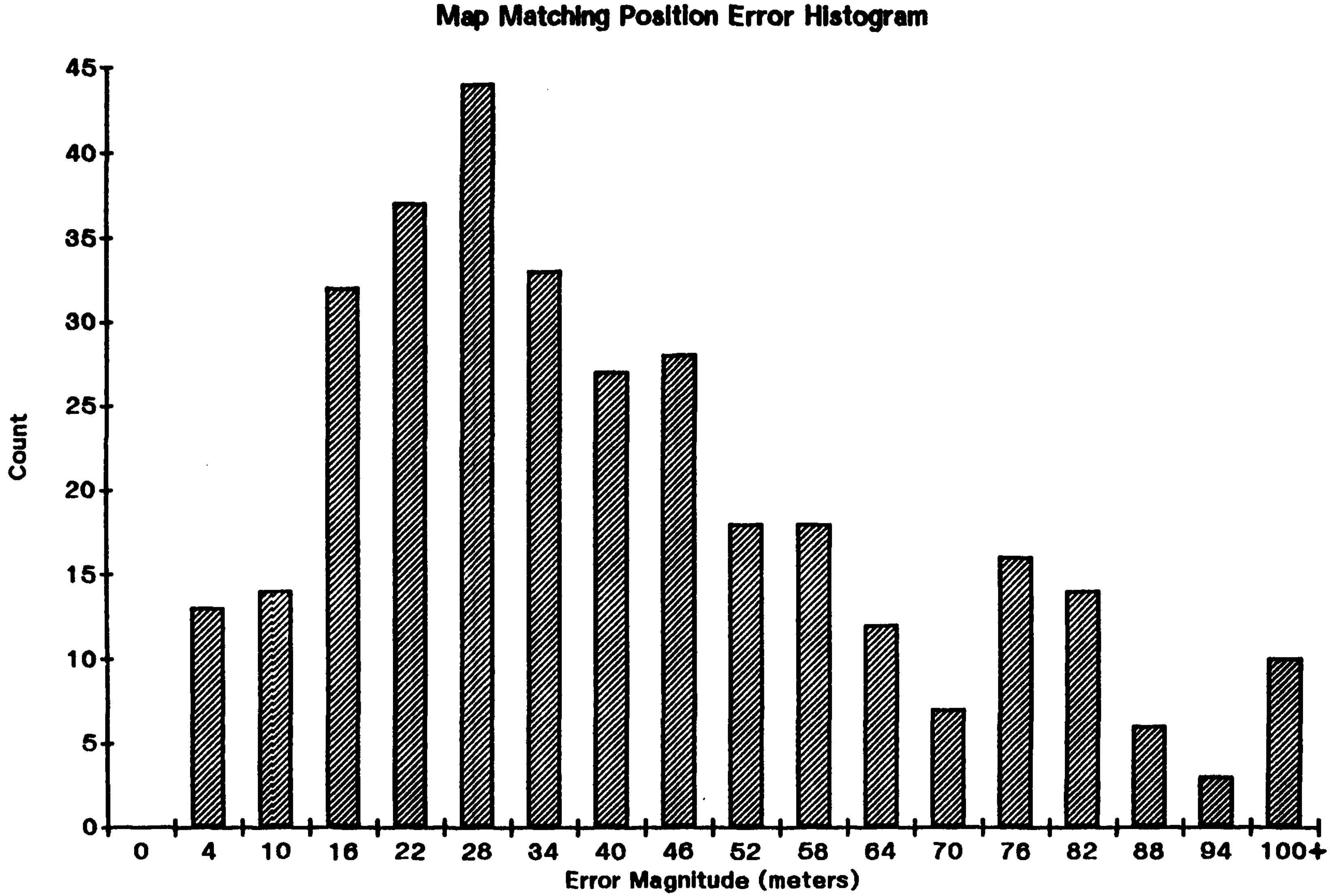

Figure 54. Map Matching Position Error

Chapter 10

Conclusions and Recommendations

Conclusions

The tests show that a navigation quality digital map can be prepared entirely from DMA 1:50,000 scale maps. Selected features such as roads, tank trails, shorelines, and transmission lines can be displayed on an electronic map to aid the driver with his navigational tasks. Overall accuracy was measured at 17.5 m, which is consistent with National Map Accuracy Standards (Reference 5).

The tests further show that a suitably modified Etak Navigator can operate with a user interface working in UTM coordinates. Destinations and waypoints may be entered by appropriate eastings and northings and the vehicle's position and heading may be read out in UTM as well. This user interface should be extremely helpful in operations that require communications and coordination with standard Army maps.

The ability to switch navigation methods from dead reckoning to map matching was demonstrated. This is a powerful tool, allowing the driver to use the more accurate map matching when on the road, while avoiding false map matching when intentionally driving off-road.

Lastly, the tests demonstrate that the overall navigation performance of the Etak Navigator falls within the general guidelines for a low-cost navigation aid. The device provides a dead reckoning base performance of better than 2%, augmented for on-road use by a map matching algorithm which yields an absolute position accuracy of better than 50 meters.

Recommendations

The Etak Navigator has demonstrated navigation capabilities sufficient for use in some Army wheeled vehicles. To improve navigation performance, Etak is working on additional low cost sensors, including an inclinometer and a turn-rate gyro. However, other classes of vehicles, such as tanks and high value treaded vehicles, will need significantly enhanced navigational devices such as INS or GPS. In general, different sets of sensors may be needed to match the cost and performance needs of a variety of vehicle types and mission requirements. It is also possible to standardize the user interface for all navigational aids to a single type of digital map display. In this way the Army can get maximum benefit in training and maximum efficiency in operations.

While the user interface has been developed to accommodate Army operations involving UTM coordinates, tests of the utility of the interface in field operations were beyond the scope of this contract. Etak recommends that the Army equip several units to participate in a variety of field operations. Ideally the Navigators in these units would be linked by radio to a command post where another electronic map display could be used to improve resource deployment. Etak is providing such Fleet Management Systems to public safety fleets, leading to significant improvements in their operations.

Finally, as discussed in Chapters 2 and 4, the map presentation can be greatly improved. With the use of color displays significant enhancements can be made. Etak has been developing improved data storage formats for dealing with enhanced map graphics and recommends that the Army continue its investigation into improving operational navigation by means of digital map displays.

REFERENCES

1. Snyder, John P. Reprinted 1984. *Map Projections Used by the U.S. Geological Survey*. Geological Survey Bulletin 1532, Second Edition. Reston, Virginia: United States Geological Survey.
2. U.S. Department of the Army. April 1973. *Universal Transverse Mercator Grid*. Washington, D.C.: Government Printing Office.
3. Etak, Inc. *Geocoder Workstation User's Guide*. 70-1520, Revision A. Menlo Park, California.
4. Etak, Inc. *Etak Navigator (UTM Version) Operating Instructions for EtakMap Series 1*. 19 June 1990. Menlo Park, California.
5. U.S. Bureau of the Budget. 1947. *United States National Map Accuracy Standards*. Washington, D.C.: Government Printing Office.
6. Corbett, J. P. 1979. *Topological Principles in Cartography*. Washington, D.C.: U.S. Bureau of the Census.
7. White, M. S. 1989. Technical Requirements for a Geographic System. *The American Cartographer*, 11, No. 1, pp. 15-21.
8. Fuchs, Abram, and Grant. Near Real-Time Shaded Display of Rigid Objects. *Computer Graphics*, v. 17, no. 3.

Appendix 1

UTM Conversion Software

Introduction

As called for in the contract, Etak is providing the conversion software, from UTM to geographic and the inverse, to ETL in the form of a stand-alone package.

What Is Provided

Etak has furnished the source code for the conversion programs on a 3.5" DSDD diskette, in addition to the listing in Appendix 2. The root directory contains the source files and supporting files:

```
 Volume in drive B has no label
 Directory of  B:\

DECIMAL       <DIR>      6-07-90   4:17p
DMS           <DIR>      6-07-90   4:17p
FILTER        <DIR>      6-07-90   4:17p
INTMATH  ASM     10042   6-15-90  12:00p
MKDEC    BAT       212   8-07-90  12:00p
MKDMS    BAT       222   8-07-90  12:00p
MKFLTR   BAT       210   8-07-90  12:00p
UTM2GEO            836   6-15-90  12:00p
UTM2GEO  C       34989   8-07-90  12:00p
UTMDECL  H        6244   9-15-89  12:00p
UTMFN    H        1640   4-15-90  12:00p
UTMINIT  H         474  10-15-89  12:00p
       12 File(s)    156672 bytes free
```

The three directories, decimal, dms, and filter, contain the executable code in three versions, as described below. The root files are described briefly here.

File utm2geo.c contains the main program; it is in Microsoft C 5.1. The three header files are utmdecl.h, utmfn.h, and utminit.h. They contain variable and structure definitions and manifest constants; function declarations; and initializers for the spheroids, respectively.

File intmath.asm contains the library of most of the integer math programs used by the conversion software (the others are in utm2geo.c). The integer functions are not actually needed on the PC, since Microsoft C has a complete floating-point library. However, since the Navigator is set up to use only integer arithmetic, the UTM package was written to use the intmath functions. The assembly language module is compatible with Microsoft masm 5.1.

Utm2geo (no filename extension) is a makefile for the package. It is set up to make any one of three sets of executables (see below), and it runs under Microsoft make. Each of the three batch files calls make to create the corresponding set of executables; the names of the batch files are derived from the names of the three directories.

Executables and the User Interface

In the course of developing the UTM transforms and the modified Navigator, three different kinds of interfaces to the transform package were needed, so a program was developed that can be compiled in three forms (or versions). Each version has a distinctive user interface, as described below. In the following discussion, the symbol "↵" means the enter key on a PC keyboard, variously marked "ENTER", "RETURN", or "↵".

The first version of the program, in directory DMS, prompts for user input (angles or UTM coordinates) one field at a time. It accepts angles in degrees minutes seconds, delimited by spaces (if you press ↵ instead of a space, it will still want the next subfield or subfields of the current field). Thus, to enter 32 degrees 47 minutes 31.22 seconds, the user would type:

```
32 47 31.22↵
```

The program outputs angles in the same way, except that unit labels are printed also. For a check during the debugging phase, the program's output, which is quite verbose (fully labelled), gave both the results of the transformation and the results of the inverse transformation on those results (which should always agree with the user's original input, within acceptable limits). It was decided to retain that feature in the final version.

The second version of the program, in directory DECIMAL, differs from the first in only one way: it accepts and prints angles in degrees and decimals, rather than DMS format. Thus, the angle mentioned above would be entered as:

```
32.792006↵
```

This version was needed to facilitate computing transformations on data presented by some of Etak's applications, which use that representation. These first two versions are completely self-explanatory in operation.

The third version of the program, in directory FILTER, is written like a Unix-style filter; that is, it accepts unprompted input from standard in and places terse output on standard out. The input is simply a series of numbers delimited by RETURNs (not by spaces!). The input numbers are either the zone, easting, and northing (in that order), or the longitude and latitude (in that order, and in degrees and decimals). The output is a single line per input set, containing the user's input and the transformed output, delimited by commas. The need for this version of the program arose from the data-analysis phase of the contract. It facilitated converting sets of data using file redirection for the input and output. The comma-delimited output is suitable for importation into most, if not all, spreadsheet programs. Since it does not prompt for input or label its output, this version of the program is not self-explanatory in use.

How the Mode of Operation is Chosen

Program umt2geo converts from UTM coordinates to geographic and from geographic coordinates to UTM coordinates. The latter conversion has an additional mode to allow out-of-zone conversions as well.

The direction or mode of conversion is chosen by using a command-line argument. Acceptable values of the argument are g, u, and z. Case is irrelevant, and only the first letter of argument one matters. There is no minus sign or slash before the argument. If the argument is u, the conversion is from UTM to geographic coordinates (this is the default). If the argument is g, the conversion is from geographic to UTM coordinates.

If the argument is z, the conversion is also from geographic to UTM, but a zone value is required along with the longitude and latitude. When the transformation is made, the UTM coordinates will be forced into the given zone. This form of the program allows the user to make transformations three or more degrees from a zone's central meridian.

Usage examples of the three modes are:

```
utm2geo↵           (convert UTM to geographic)
utm2geo u↵         (convert UTM to geographic)
utm2geo g↵         (convert geographic to UTM)
utm2geo z↵         (geographic to UTM, force zone)
```

Examples of Dialog with the Programs

In the examples below, the system and program output are shown in `normal type` and the user's input is in **bold**.

DMS Version

```
C:> utm2geo↵

To quit, enter a grid zone of 0 at the prompt

To indicate a southern hemisphere value, make the grid
     zone negative.

Enter grid zone: 14↵
Enter easting: 627718↵
Enter northing: 3447084↵

Chosen spheroid is Clarke 1866
Longitude =  97°39'36"403W, Latitude =  31°09'08"770N,
     zone = 14
Inverse xform:  UTM string is  ZN 14 E 627718 N 3447084
Convergence = -0°42'
```

```
Enter grid zone: 0↵

C:> utm2geo g↵

To quit, enter a longitude > 180 0 0.
Enter longitude and latitude in deg min sec
     (eg, -12 27 30).
Fields are separated by spaces.
All three fields must be entered.

Enter longitude: -97 39 36.4↵
Enter latitude:  31 9 8.77↵

Chosen spheroid is Clarke 1866
UTM string is  ZN 14 E 627718 N 3447084
Inverse xform:  Longitude =  97°39'36"403W,
     Latitude =  31°09'08"770N, zone = 14
Convergence = -0°42'

Enter longitude: 999 0 0↵

C:> _
```

Decimal Version

```
C:> utm2geo↵

To quit, enter a grid zone of 0 at the prompt

To indicate a southern hemisphere value, make the grid
     zone negative.

Enter grid zone: 14↵
Enter easting: 639111↵
Enter northing: 3443861↵

Chosen spheroid is Clarke 1866
Longitude = 97.541065W, Latitude = 31.122064N,
     zone = 14
Inverse xform:  UTM string is  ZN 14 E 639111 N 3443861
Convergence = -0°45'

Enter grid zone: 0↵

C:> utm2geo g↵

To quit, enter a longitude > 180.
Enter longitude and latitude in degrees and decimals
     (eg, -12.345...).

Enter longitude: -97.541065↵
Enter latitude:  31.122064↵
```

```
Chosen spheroid is Clarke 1866
UTM string is  ZN 14 E 639111 N 3443861
Inverse xform:  Longitude = 97.541065W,
     Latitude = 31.122064N, zone = 14
Convergence = -0°45'

Enter longitude: 999↵

C:> _
```

Filter Version

```
C:> utm2geo↵
14↵
622718↵
3447984↵
627718,3447084,-97.660112,31.152436
C:> 0↵
C:> utm2geo g↵
-97.660112↵
31.152436↵
-97.660112,31.152436,627718,3447084
C:> 0↵
C:> _
```

Appendix 2. Source Code Listings

Batch Files

MKDEC.BAT

```
@echo off

rem  Make DECIMAL version of geo2utm et al.

rem   Author:  Gene E. Bloch, ETAK, August 1990
rem   Copyright (c) Etak Corp, 1990

if exist utm2geo.obj del utm2geo.obj
make d=/DDECIMAL utm2geo
```

MKDMS.BAT

```
@echo off

rem  Make DMS (degrees minutes seconds) version of geo2utm et al.

rem   Author:  Gene E. Bloch, ETAK, August 1990
rem   Copyright (c) Etak Corp, 1990

if exist utm2geo.obj del utm2geo.obj
make utm2geo
```

MKFLTR.BAT

```
@echo off

rem  Make FILTER version of geo2utm et al.

rem   Author:  Gene E. Bloch, ETAK, August 1990
rem   Copyright (c) Etak Corp, 1990

if exist utm2geo.obj del utm2geo.obj
make d=/DFILTER utm2geo
```

Makefile

UTM2GEO

```
# Make utm2geo using Microsoft C 5.1 and masm 5.1.

# Author:  Gene E. Bloch, ETAK, Sept 1989 - June 1990
# Copyright (c) Etak Corp, 1989,1990

# To make the DMS version, execute
#      "make utm2geo"

# To make the decimal version, execute
#      "make d=/DDECIMAL utm2geo"

# To make the filter version, execute
#      "make d=/DFILTER utm2geo"

# This version is set up for CodeView; to change that,
#   comment out the definitions of cvc and cvl below
#   by inserting a # before each line.

cvc=/Zi /Od                     # Compile with CodeView
cvl=/CO                         # Link with CodeView

utm2geo.obj: utm2geo.c utmdecl.h utminit.h utmfn.h
    cl $(d) /AM /c $(cvc) utm2geo.c

intmath.obj:    intmath.asm
    masm intmath;

utm2geo.exe: utm2geo.obj intmath.obj
    link $(cvl) utm2geo intmath;
```

Header Files

UTMDECL.H

```
/*
**  utmdecl.h - declarations for utm conversions
**
**   Author:  Gene E. Bloch, ETAK, Sept 1989
**   Copyright (c) Etak Corp, 1989
**
*/

typedef unsigned long  UL;

#define BUFSIZE 132

#define HAFM     500000L  /* half million */
#define TENM   10000000L  /* ten million */

#define DHAFM  500000.0  /* half million */

#define EARTH        16777216.0 /* normalize earth radius, arclength, ... */
#define FLAT              512.0 /* normalize reciprocal flatness */
#define FLAT_ONE  0x00800000L
#define FLAT_HALF (FLAT_ONE >> 1)
#define HALF      0x80000000L
#define ALMOST1   0xFFFFFFFFL /* unsigned */
#define ALMOST1S  0x7FFFFFFFL /* signed   */
#define BIG       4294967296.0
#define pi        3.1415926535

#ifdef    SHOW_VALUE
#define PI       ((unsigned long) ((pi/8.0) * BIG))
#define TWOPI    ((unsigned long) ((pi/4.0) * BIG)) /* compiler miscomputes this'n */
#undef  TWOPI
#define TWOPI    ((UL) PI << 1)      /* compiler computes this one ok */
#define PI_DIV_4 (PI >> 2)
#define PI_DIV_2 (PI >> 1)
#else  /* SHOW_VALUE */
#define PI       0x6487ED51
#define TWOPI    0xC90FDAA2
#define PI_DIV_4 0x1921FB54
#define PI_DIV_2 0X3243F6A8
#endif /* SHOW_VALUE */

/* coefficients for sin series, 0 to 45 degrees */
#define C98   ((unsigned long) (BIG/(9.0*8.0)))
#define C76   ((unsigned long) (BIG/(7.0*6.0)))
#define C54   ((unsigned long) (BIG/(5.0*4.0)))
#define C32   ((unsigned long) (BIG/(3.0*2.0)))
/* coefficients for sin series, 45 to 90 degrees */
#define CX9   ((unsigned long) (BIG/(10.0*9.0)))
```

```
#define C87   ((unsigned long) (BIG/(8.0*7.0)))
#define C65   ((unsigned long) (BIG/(6.0*5.0)))
#define C43   ((unsigned long) (BIG/(4.0*3.0)))

#define RAD  (pi/180.0)        /* express angle in radians (from degrees) */
#define DEG  (180.0/pi)        /* express angle in degrees (from radians) */

#define ZRAD (pi/30.0)         /* radians per 6 degree gridzone       */
#define ORAD (183.0*pi/180.0)  /* radian offset, gridzone to Greenwich */

#define ZETAK (UL) 0x22222222L /* 48' in EAU:  converts zones to EAU   */
#define DETAK (UL) 0xB60B60B6L /* 256' in EAU:  converts angles to EAU */
#define EMSEC (UL) 0x4D3F6400L /* 360*60*60*1000:  EAU to milli-arcsec */

#define E80S     (long) (0x471C71C7)
#define E84N     (long) (0xBBBBBBBC)

/* many numeric constants for coefficients, etc */
#define c0ONE      0x10000000L
#define c2ONE      0x10000000L
#define c2THREE    0x30000000L
#define c4TWO      0x00040000L

#define L1ONE      0x01000000L
#define L3ONE      0x01000000L
#define L5TWO      0x00200000L
#define L5THREE    0x00300000L
#define L5FIVE     0x00500000L
#define L5EIGHT    0x00800000L
#define L5_24      0xC0000000L
#define L5_28      0x01C00000L

#define p0ONE      0x20000000L

#define p4THREE    0xc0000000L
#define p4FOUR     0x40000000L
#define p4FIVE     0x02800000L
#define p4NINE     0x04800000L
#define p4TEN      0x05000000L

#ifdef UNDEF
#define p6THREE    0x00006000L
#define p6_45      0x5A000000L
#define p6_61      0x0007A000L
#define p6_90      0x00B40000L
#define p6_252     0x001F8000L
#define p6_298     0x00254000L
#endif /* UNDEF */

#define p6ONE      0x00800000L
#define p6THREE    0x01800000L
#define p6_45      0x16800000L
#define p6_61      0x1E800000L
#define p6_90      0x2D000000L
#define p6_252     0x7E000000L
```

```
#define p6_298      0x95000000L

#define x0ONE       0x01000000L
#define x2ONE       0x01000000L
#define x4FIVE      0x00140000L
#define x4_18       0x12000000L
#define x4_58       0x3A000000L
#define x4_72       0x48000000L

#define y2ONE       0x01000000L
#define y4FIVE      0x05000000L
#define y4NINE      0x09000000L
#define y6_58       0x03A00000L
#define y6_61       0x000F4000L
#define y6_330      0x14A00000L
#define y6_600      0x25800000L

/* fractions (for division by integers) */
#define c4OVER15    0x88888888L
#define L3OVER6     0x2AAAAAABL
#define L5OVER120   0x02222222L
#define p4OVER12    0xAAAAAAABL
#define p6OVER360   0x002D82D8L
#define x2OVER6     0x0AAAAAABL
#define x4OVER120   0x22222222L
#define y4OVER12    0x55555555L
#define y6OVER360   0x5B05B05BL

#define MRAD (0.5/6378206.4)   /* half meter arclength in radians      */

struct ellipse
    {
    int    desig;  /* ellipsoid designator                    */
    double major;  /* major axis                              */
    double rflat;  /* reciprocal flatness                     */
    double scale;  /* scale factor                            */
    char   *name;  /* name of spheroid                        */
    } sphoid[];

/* allow detecting change of spheroid */
int oldsph, newsph;

enum
    {
    NO_SPHEROID,
    CLARKE66, CLARKE80
    }
spheroid;

/* global copy of the current spheroid data */
unsigned long major, rflat, scale;
/* constants for the current spheroid */
unsigned long eccent, eccsq, eccprsq, el, elsq, aesq;

/* constant coefficients for geo ==> UTM */
```

```
unsigned long m0, m2, m4, m6;
/* variable coefficients for geo ==> UTM */
unsigned long N, T, C, A, Asq, M, lambda;
unsigned long x0, x2, x4, y2, y4, y6;
/* trig values for geo ==> UTM */
unsigned long sin2phi, sin4phi, sin6phi, cos2phi, cos4phi;
unsigned long tanphi, sinphi, sinqphi, cosphi, cosqphi;
unsigned long phi;

/* constant coefficients for UTM ==> geo */
unsigned long mu0, mu2, mu4, mu6;
/* variable coefficients for UTM ==> geo */
unsigned long mu, phi1, N1, TAN1, C1, R1, D, Dsq;
unsigned long p2, p4, p6, L3, L5;
/* trig values for UTM ==> geo */
unsigned long sin2mu, sin4mu, sin6mu;
unsigned long sinphi1, sinqphi1, cosphi1, cosqphi1, tanphi1, cosphi1;

/*
**  Place to parse the UTM string.
**  -1 and 99,999,999L represent values not entered.
*/
struct utm_parse
    {
    int gz;                 /* grid zone                 */
    long easting;           /* value of easting          */
    long northing;          /* value of northing         */
    unsigned long conver;   /* convergence of true north */
    }
newUTM, oldUTM, UTMtemp;

/* structures for multi-precision arithmetic */
struct words
    {
    unsigned n0;
    unsigned n16;
    };

union uwo:d
    {
    unsigned long full;
    struct words  half;
    };
```

UTMFN.H

```
/*
**   Function declarations for UTM2GEO.c
**
**   Author:  Gene E. Bloch, ETAK, Sept 1989 - Apr 1990
**   Copyright (c) Etak Corp, 1989,1990
*/

void          main(int, char **);
void          exit(int);
int           progname(char **);
void          geo(int);
void          utm(void);
int           getgeo(UL *, UL *, int *);
int           getutm(UL *, UL *, int *);
int           getDMS(UL *);
int           getUTM(void);
void          clearutm (struct utm_parse *);
int           empty(struct utm_parse *);
int           choose_spheroid_UTM(struct utm_parse *);
int           choose_spheroid_geo(long, long);
int           sph_init(int);
void          geocoeff(long, long);
void          utmcoeff(long, long);
unsigned long converge(void);
void          UTMcoords(long *, long *);
void          geocoords(long *, long *);
long          decimaln(char *, int);
void          utm2geo(struct utm_parse *, UL *, UL *);
void          displaygeo(UL, UL);
void          displayDMS(UL, int, int);
void          displayANG(UL, int, int);
void          displayUTM(struct utm_parse *);
void          displayconv(UL);
void          geo2utm(struct utm_parse *, UL, UL, int);
void          UTMcopy(struct utm_parse *, struct utm_parse *);
long          smul(long,long);
unsigned long psin(unsigned long);
unsigned long pcos(unsigned long);
void          ptrig (UL, UL *, UL *, UL *, UL *);
void          sighandler(int);
void          setsigs (void);

/* found in intmath.asm */
unsigned long mulfrac(UL, UL);
unsigned long divfrac(UL, UL);
unsigned long sqrfrac(UL);
long          smulfrac(long,long);
```

UTMINIT.H

```
/*
**  utminit.h - initializer(s) for utm conversions
**
**   Author:  Gene E. Bloch, ETAK, Sept-Oct 1989
**   Copyright (c) Etak Corp, 1989
*/

struct ellipse sphoid[] =
    {
/*  designator   major          rflat        scale   name             */
    CLARKE66,    6378206.4000, 294.978698, 0.9996, "Clarke 1866",
    CLARKE80,    6378249.1450, 293.465000, 0.9996, "Clarke 1880",

    NO_SPHEROID, 0.0,          0.0,        0.0,    "Unknown spheroid"
    };
```

Program Files

UTM2GEO.C

```
/*
**  utm2geo - convert UTM coords to lat/lon
**       (includes geo2utm, the reverse transformation)
**
**   Author:  Gene E. Bloch, ETAK, Sept 1989 - August 1990
**   Copyright (c) Etak Corp, 1989,1990
**
**   If DECIMAL is defined, I/O is in degrees and decimals,
**       otherwise it is in degreees, minutes, and seconds.
**   If FILTER is defined:
**      1.  DECIMAL is assumed.
**      2.  There are no prompts, and output is terse and comma-delimited.
**          The program is thus suitable for use in pipes as a UNIX-style
**          filter, and its output is suitable for importation into a
**          spreadsheet.
**
**   Usage:
**       utm2geo [u...]
**           will prompt for UTM coords to convert to geographic.
**           (argument optional)
**
**       utm2geo g...
**           will prompt for geographic coords to convert to UTM.
**           (argument required)
**
**       utm2geo z...
**           will prompt for geographic coords to convert to UTM;
**           will also ask for - and enforce - a zone ID.
**           (argument required)
**
**       Only the first letter of the arg (which may be upper or lower
**           case) is scanned.  If there is no argument, the invocation
**           name is used instead (so if a copy of utm2geo.exe called
**           geo2utm.exe exists, "geo2utm" has the same effect as
**           "utm2geo geo" - and "geo2utm utm" is the same as "utm2geo").
**
**   References used:
**       Snyder, John P., "Map Projections Used by the US Geological Survey",
**           Geological Survey Bulletin 1532, Second Edition (1984)
**           See Especially 'Formulas for the Ellipsoid', pp67-69.
**       Defense Mapping Agency (Hager, John W., Fry, Larry L., Jacks,
**           Sandra S., and Hill, David R.), "Datums, Ellipsoids, Grids,
**           and Grid Reference Systems", DMA TM 8358.1 (No date on my
**           photocopy).
**       US Army TM 5-241-8 (I have only a photocopy of some 19 pages of
**           this document, and I don't know the particulars).
*/
```

```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <signal.h>
#include <ctype.h>

#include "utmdecl.h"
#include "utminit.h"
#include "utmfn.h"

static char copr[] = "Copyright (c) Etak Corp, 1989,1990";

/* debug stuff */
static int debug = 0;
#include <time.h>
#include <bios.h>
#define TIMECOUNT 1000
static long starttime, endtime, elapsed;
static int tm, timing = 0;
static char *from, *to;

#ifdef    FILTER
#undef  DECIMAL
#define DECIMAL
#endif /* FILTER */

void
main (argc, argv)
int argc; char **argv;
{
    int i;

    /* debug */
    for (i = 1; i < argc; ++i)
        if (toupper (argv[i][0]) == 'D')
            debug = 1;

    /* debug */
    timing = debug*TIMECOUNT;

    setsigs();

    /* trigger spheroid change */
    newsph = -1;

    /* select if version was chosen on command line */
    for (i = 1; i < argc; ++i)
        switch (toupper (argv[i][0]))
            {
            case 'U':
                utm();
                exit (0);

            case 'G':
                geo(0);
```

```
                exit (0);

            case 'Z':
                geo(1);
                exit (0);

            default:
                break;
            }

    /* otherwise, select according to name used to invoke program */
    switch (progname(argv))
        {
        case 'G':
            geo(0);
            exit (0);

        case 'Z':
            geo(1);
            exit (0);

        case 'U':
        default:
            utm();
            exit (0);
        }
}

/* get first char of name used to invoke program */
progname (argv)
char **argv;
{
    char *pname;

    if ((pname = strrchr (argv[0], '\\')) == NULL)
        pname = strrchr (argv[0], ':');

    return toupper (pname == NULL ?  argv[0][0] : pname[1]);
}

void
geo (zoneforce)
int zoneforce;
{
    UL longitude, latitude;
    int zone;

    from = "geo";
    to   = "UTM";

#ifdef    FILTER
    timing = 0;
#else  /* FILTER */
    tell_how_geo (zoneforce);
#endif /* FILTER */
```

```
    for (;;)
        {
        zone = zoneforce;
        if (getgeo (&longitude, &latitude, &zone) == 0)
            break;

        if (!timing)
            geo2utm (&newUTM, longitude, latitude, zone);
        else
            {
            printf ("Iterating %d times to measure timing ...\n", TIMECOUNT);
            _bios_timeofday (_TIME_GETCLOCK, &starttime);
            for (tm = 0; tm < timing; ++tm)
                geo2utm (&newUTM, longitude, latitude, zone);
            _bios_timeofday (_TIME_GETCLOCK, &endtime);
            elapsed = endtime - starttime; /* ignore 24-hr flag */
            printf ("Average time to convert %s to %s = %.2lfms\n",
                from, to, (double) elapsed/(0.0182*(double) TIMECOUNT));
            }

#ifndef   FILTER
        /* display the output values */
        displayUTM (&newUTM);

        /* make and display the inverse transform */
        utm2geo (&newUTM, &longitude, &latitude);
        printf ("Inverse xform:  ");
        displaygeo (longitude, latitude);
        displayconv (newUTM.conver);
#else  /* FILTER */
        /* display the input & output values */
        displaygeo (longitude, latitude);
        putchar (',');
        displayUTM (&newUTM);
        putchar ('\n');
#endif /* FILTER */
        }
}


void
utm ()
{
    UL longitude, latitude;

    from = "UTM";
    to   = "geo";

#ifndef   FILTER
    if (!debug)
        tell_how_utm();
#endif /* FILTER */

    /*
```

```
    ** Newly-entered values of UTM will replace previous
    **   values of UTM on a field-by-field basis.
    ** We now set up starting values to "14 622731 3443654".
    */
    newUTM.gz       = 14;
    newUTM.easting  = 622731L;
    newUTM.northing = 3443654L;
    newUTM.conver   = -1;

    for (;;)
        {
        UTMcopy (&oldUTM, &newUTM);
        if (getUTM () == 0)
            break;
        if (empty (&newUTM))
            {
            UTMcopy (&newUTM, &oldUTM);
            continue;
            }

        if (!timing)
            utm2geo (&newUTM, &longitude, &latitude);
        else
            {
            printf ("Iterating %d times to measure timing ...\n", TIMECOUNT);
            _bios_timeofday (_TIME_GETCLOCK, &starttime);
            for (tm = 0; tm < timing; ++tm)
                utm2geo (&newUTM, &longitude, &latitude);
            _bios_timeofday (_TIME_GETCLOCK, &endtime);
            elapsed = endtime - starttime; /* ignore 24-hr flag */
            printf ("Average time to convert %s to %s = %.2lfms\n",
                from, to, (double) elapsed/(0.0182*(double) TIMECOUNT));
            }

#ifndef   FILTER
        /* display the output values */
        displaygeo (longitude, latitude);

        /* make and display the inverse transform */
        geo2utm (&newUTM, longitude, latitude, newUTM.gz);
        printf ("Inverse xform:  ");
        displayUTM (&newUTM);
        displayconv (newUTM.conver);
#else  /* FILTER */
        /* display the input & output values */
        displayUTM (&newUTM);
        putchar (',');
        displaygeo (longitude, latitude);
        printf ("\n");
#endif /* FILTER */
        }
}

tell_how_geo (int zoneforce)
{
```

```
    printf ("\nTo quit, enter ");
    if (zoneforce)
        printf ("a zone of 0 or ");
#ifdef    DECIMAL
    printf ("a longitude > 180.\n");
    printf
        ("Enter longitude and latitude in degrees and decimals (eg, -12.345...).\n");
#else  /* DECIMAL */
    printf ("a longitude > 180 0 0.\n");
    printf ("Enter longitude and latitude in deg min sec (eg, -12 27 30).\n");
    printf ("Fields are separated by spaces.\n");
    printf ("All three fields must be entered.\n");
#endif /* DECIMAL */
}

tell_how_utm()
{
    printf ("\nTo quit, enter a grid zone of 0 at the prompt\n");
    printf ("\nTo indicate a southern hemisphere value,");
    printf (" make the grid zone negative.\n");
}

getgeo (longitude, latitude, zone)
UI *longitude, *latitude; int *zone;
{
    UL lon, lat;
    int zonetemp;

#ifndef   FILTER
    putchar ('\n');
#endif /* FILTER */

    if (*zone)
        {
#ifndef   FILTER
        printf ("Enter grid zone: ");
#endif /* FILTER */
        zonetemp = 0;
        scanf ("%d", &zonetemp);
        /* get rid of terminating newline */
        while (getchar () != '\n')
            continue;
        if (zonetemp == 0)
            return 0;
        if (zonetemp < 0)
            zonetemp = -zonetemp;
        }

#ifndef   FILTER
    printf ("Enter longitude: ");
#endif /* FILTER */
    if (getANG (&lon) == 0)
        return 0;

#ifndef   FILTER
```

```
    printf ("Enter latitude:  ");
#endif /* FILTER */
    if (getANG (&lat) == 0)
        return 0;

    if (!*zone)
        {
        if (lon < HALF)
            zonetemp = 30 - (int) ((HALF - lon)/(ZETAK >> 3));
        else
            zonetemp = 31 + (int) ((lon - HALF)/(ZETAK >> 3));
        if (zonetemp == 0)
            zonetemp = 1;
        if (zonetemp == 61)
            zonetemp = 60;
        }

    oldsph = newsph;
    newsph = choose_spheroid_geo (lon, lat);
    if (oldsph != newsph)
        if (!sph_init (newsph))
            return 0;

    *longitude = lon;
    *latitude  = lat;
    *zone      = zonetemp;

    return 1;
}

/* read coords from screen, return Etak angle */
#ifdef DECIMAL
getANG (angle)
UL *angle;
{
    int sign;
    double deg;
    UL temp;

    scanf ("%lf", &deg);
    /* get rid of terminating newline */
    while (getchar () != '\n')
        continue;

    sign = 1;
    if (deg < 0.0)
        {
        deg = -deg;
        sign = -1;
        }
    if (deg > 180.0)
        return 0;

    /* constant is ((2**32)/360) */
    temp  = (UL) (11930464.71*deg + 0.5);
```

```
    *angle = sign == -1 ? HALF - temp : HALF + temp;

    return 1;
}
#else  /* DECIMAL */
getANG (angle)
UL *angle;
{
    int deg, min, sign;
    double sec;
    UL temp;

    scanf ("%d %d %lf", &deg, &min, &sec);
    /* get rid of terminating newline */
    while (getchar () != '\n')
        continue;

    sign = 1;
    if (deg < 0)
        {
        deg = -deg;
        sign = -1;
        }
    if (deg > 180)
        return 0;

    if (min < 0)
        {
        if (deg == 0)
            sign = -1;
        min = -min;
        }
    if (sec < 0.0)
        {
        if (deg == 0  &&  min == 0)
            sign = -1;
        sec = -sec;
        }

    if (((long) deg + ((long) min + sec/60.0)/60.0)  >  180.0)
        return 0;

    temp  = mulfrac ((UL) (1000.0*sec + 0.5) << 16, DETAK);
    temp  = mulfrac (temp, 0x4A90B); /* const is (1 << 8)/3,600,000 */
    temp += mulfrac ((UL) deg << 24, DETAK);
    temp += mulfrac ((UL) min << 24, DETAK)/60;

    *angle = sign == -1 ? HALF - temp : HALF + temp;

    return 1;
}
#endif /* DECIMAL */

getUTM ()
```

```
{
    char UTMbuf[BUFSIZE+1];

    clearutm (&newUTM);

    if (debug)
        newUTM.gz = 30;
    else
        {
#ifndef   FILTER
        printf ("\nEnter grid zone: ");
#endif /* FILTER */
        scanf ("%d", &newUTM.gz);
        /* get rid of terminating newline */
        while (getchar () != '\n')
            continue;
        }
    if (newUTM.gz == 0)
        return 0;

#ifndef   FILTER
    printf ("Enter easting: ");
#endif /* FILTER */
    if (fgets (UTMbuf, BUFSIZE, stdin) == NULL)
        return 1;
    /* pad out to 6 char if needed */
    newUTM.easting = decimaln (UTMbuf, 6) - HAFM;

#ifndef   FILTER
    printf ("Enter northing: ");
#endif /* FILTER */
    if (fgets (UTMbuf, BUFSIZE, stdin) == NULL)
        return 1;
    /* pad out to 7 char if needed */
    newUTM.northing = decimaln (UTMbuf, 7);
    /* deal with southern hemisphere */
    if (newUTM.gz < 0)
        {
        newUTM.northing = newUTM.northing - TENM;
        newUTM.gz = -newUTM.gz;
        }
    newUTM.gz = 1 + ((newUTM.gz - 1) % 60);

    oldsph = newsph;
    newsph = choose_spheroid_UTM (&newUTM);

    return oldsph != newsph ? sph_init (newsph) : 1;
}

/* clear the input structure */
void
clearutm (UTM)
struct utm_parse *UTM;
{
    /* clear the fields to indicate that no data was input yet */
```

```
    UTM->gz       = -1;
    UTM->easting  = 99999999L;
    UTM->northing = 99999999L;
    UTM->conver   = -1;
}

/* check the input structure for absence of new data */
empty (UTM)
struct utm_parse *UTM;
{
    /* check the fields to find out whether any data was input */
    if (UTM->gz       != -1)
        return 0;
    if (UTM->easting  != 99999999L)
        return 0;
    if (UTM->northing != 99999999L)
        return 0;

    return 1;
}

choose_spheroid_UTM (UTM)
struct utm_parse *UTM;
{
    return CLARKE66;
}

choose_spheroid_geo (lon, lat)
long lon, lat;
{
    return CLARKE66;
}

sph_init (spheroid)
int spheroid;
{
    unsigned long tmpl, temp, tempsq;
    int sp;

    for (sp = 0; sphoid[sp].desig != NO_SPHEROID; ++sp)
        if (sphoid[sp].desig == spheroid)
            break;

#ifndef   FILTER
    printf ("Chosen spheroid is %s\n", sphoid[sp].name);
#endif /* FILTER */
    if (sphoid[sp].desig == NO_SPHEROID)
        return 0;

    /*
    **  Move spheroid data to globals
    **  Unsigned
    **  Binary point left of bit 31 (ie, bit 31 = 1/2)
    */
    major = (unsigned long) (BIG*((sphoid[sp].major/EARTH)));
```

```
    rflat = (unsigned long) (BIG*((sphoid[sp].rflat/FLAT)));
    scale = (unsigned long) (BIG*  sphoid[sp].scale);

    /*
    **  Calculate constants of the spheroid.
    **  Formulas have been rearranged for easier integer
    **      calculations with less loss of significance.
    **  Unsigned
    **  Binary point left of bit 31 (ie, bit 31 = 1/2)
    */
    temp    = rflat - FLAT_ONE;
    temp    = divfrac (temp, rflat);
    tempsq  = mulfrac (temp, temp);
    eccsq   = -tempsq;
    eccprsq = divfrac (eccsq, tempsq);
    el      = divfrac (FLAT_HALF, rflat - FLAT_HALF);
    elsq    = mulfrac (el, el);
    aesq    = mulfrac (major, tempsq);

    /*
    **  Calculate constant coefficients
    **  (i.e., coefficients that depend on spheroid but not on coordinates)
    **
    **  1.  geo ==> UTM
    */
/*
**  Coefficients as modified (for computation) from Snyder:
**  m0 = 1.0 - eccsq*(1.0/4.0 + eccsq*((3.0/64.0) + eccsq*(5.0/256.0)));
**  m2 = eccsq*(3.0/8.0 + eccsq*((3.0/32.0) + eccsq*(45.0/1024.0)));
**  m4 = tmpl*(15.0/256.0 + eccsq*(45.0/1024.0));
**  m6 = tmpl*eccsq*(35.0/3072.0);
**  Most of the arithmetic is unsigned.
**  The binary point is usually to the left of bit 31 (ie, bit 31 = 1/2).
**  Note that, when the BP (binary point) is so assigned, '1 - x' is
**      computed by '-x'.
*/
    tmpl = mulfrac (eccsq, eccsq);
    m0  = (3*tmpl)/64;
    m0 += eccsq >> 2;
    m0  = -m0;

    m2  = (3*tmpl) >> 5;
    m2 += (3*eccsq) >> 3;

    m4  = (15*tmpl) >> 8;

    /*
    **  2.  UTM ==> geo
    */
/*
**  Coefficients as modified (for computation) from Snyder:
**  mu0 = 1.0 - eccsq*(1.0/4.0 + eccsq*((3.0/64.0) + eccsq*(5.0/256.0)));
**  mu2 = el*(1.5 - elsq*(27.0/32.0));
**  mu4 = elsq*(21.0/16.0 - elsq*(55.0/32.0));
**  mu6 = el*elsq*(151.0/96.0);
```

```
**  Unsigned
**  Binary point left of bit 31 (ie, bit 31 = 1/2)
*/
    mu0 = m0;
    mu2 = (3*el) >> 1;
    mu4 = (21*elsq) >> 4;

    return 1;
}

/*
** geocoeff expects lon/lat in signed EAU.
** lat runs from the equivalent of -80.0 to +84.0, equator at 0.0.
** lon is centered on the central meridian and usually (not neces-
**     sarily always) runs from the equivalent of -3.0 to 3.0 deg.
**  Mostly unsigned with binary point left of bit 31 (ie, bit 31 = 1/2)
*/
void
geocoeff (lon, lat)
long lon, lat;
{
      long t1, t2, t3;

    if (lon < 0L)
        lon = -lon;
    if (lat < 0L)
        lat = -lat;

    lambda   = mulfrac (lon, TWOPI);
    phi      = mulfrac (lat, TWOPI);

#ifdef    OLDWAY
    sinphi   = psin (phi);

    sinqphi  = mulfrac (sinphi, sinphi);
/*  cosphi   = sinqphi ? sqrfrac (-sinqphi) : ALMOST1;  */
    cosphi   = pcos (phi);
    cosqphi  = mulfrac (cosphi, cosphi);
#else  /* OLDWAY */
    ptrig (phi, &sinphi, &sinqphi, &cosphi, &cosqphi);
#endif /* OLDWAY */

    /*
    **  sin, cos of multiple angles are signed, with binary
    **      point left of bit 30, ie bit 30 = 1/2
    */
    /* sin2phi: '>> 1' to normalize and '<< 1' to mult by 2 */
    sin2phi  = mulfrac (sinphi, cosphi);
    /* sin 2*phi can reach 1, but 1 looks like -0 */
    if ((UL) sin2phi > (UL) ALMOST1S)
        sin2phi = ALMOST1S;
    cos2phi  = (long) (cosqphi >> 1) - (long) (sinqphi >> 1);

    sin4phi  = smulfrac (sin2phi, cos2phi);
/*  sin6phi  = smulfrac (sin4phi, cos2phi) - sin2phi;
```

```
*/

    /*
    **  For tanphi, binary point moved to bit 27, ie bit
    **      28 = 1, bit 31 = 8.  Hence also, for T the point
    **      is at bit 23, and bit 31 = 128.
    */
    tanphi = divfrac (sinphi >> 4, cosphi);

    T = mulfrac (tanphi, tanphi);
    t1 = mulfrac (sinqphi,eccsq);
    t1 = t1 ? sqrfrac (-t1) : ALMOST1;
    N = divfrac (major, t1);
    C = mulfrac (cosphi, eccprsq);
    A = mulfrac (cosphi, lambda);
    Asq = mulfrac (A, A);

/*
**  Coefficients as modified (for computation) from Snyder:
**  x0 = scale*N*A;
**  x2 = (1.0 - T + C)/6.0;
**  x4 = (5.0 + T*(T - 18.0) + 72.0*C - 58.0*eccprsq)/120.0;
*/
    x0 = mulfrac (mulfrac (N, A), scale);
    x2 = smulfrac (x2ONE - T + (C >> 8), x2OVER6);
    x4 = smulfrac (x4FIVE + smulfrac (T, T - x4_18) +
             ((smulfrac (x4_72, C) - smulfrac (x4_58, eccprsq)) >> 8),
             x4OVER120);

/*
**  Coefficients as modified (for computation) from Snyder:
**  y2 = N*tanphi/2.0;
**  y4 = (5.0 - T + C*(9.0 + 4.0*C))/12.0;
**  y6 = (61.0 - T*(58.0 - T) + 600.0*C -330.0*eccprsq)/360.0;
*/
    y2 = mulfrac (N << 1, tanphi) >> 2;
    /* 'C >> 6' is:  '>> 8' to normalize and '<< 2' for 'times 4' */
    y4 = smulfrac (y4OVER12,
             y4FIVE - T +
                 (smulfrac (C, y4NINE + (C >> 6)) >> 2)) >> 4;
    y6 = smulfrac (y6OVER360, y6_61 -
             smulfrac (T, y6_58 - (T >> 4)) +
                 ((smulfrac (y6_600, C) - smulfrac (y6_330, eccprsq)) >> 8)) << 1;
}

void
utmcoeff (east, north)
long east, north;
{
    unsigned long t1, t2, t3;
    unsigned long sinmu, sinqmu, cosmu, cosqmu, cos4phil;
    long cos2mu, sin2mu, sin4mu /*, sin6mu */ ;

unsigned long t4;
```

```
    if (east < 0)
        east = -east;
    if (north < 0)
        north = -north;

    north <<= 5;
    east  <<= 5;

    mu = divfrac (north, mulfrac (major, mulfrac (mu0, scale)));

#ifdef    OLDWAY
    sinmu  = psin (mu);
    sinqmu = mulfrac (sinmu, sinmu);
/*  cosmu  = sinqmu ? sqrfrac (-sinqmu) : ALMOST1; */
    cosmu  = pcos (mu);
    mulfrac (cosmu, cosmu);
    cos2mu = mulfrac (cosmu, cosmu >> 1) - (sinqmu >> 1);
#else  /* OLDWAY */
    ptrig (mu, &sinmu, &sinqmu, &cosmu, &cosqmu);
    cos2mu = ((UL) cosqmu >> 1) - (sinqmu >> 1);
#endif /* OLDWAY */
    /* mult angles treated as signed, binary pt is bit 30 */
    sin2mu = mulfrac (sinmu, cosmu);
    sin4mu = smulfrac (sin2mu, cos2mu);

/*
**  Formula as modified (for computation) from Snyder:
**  phi1 = mu6*sin6mu + mu4*sin4mu + mu2*sin2mu + mu;
*/
    phi1 = ((long) mu << 1) +
                (smulfrac (sin2mu, mu2) >> 3) +
                (smulfrac (sin4mu, mu4) >> 3);
    phi1 >>= 1;

#ifdef    OLDWAY
    sinphi1 = psin (phi1);

    sinqphi1 = mulfrac (sinphi1, sinphi1);
/*  cosphi1  = sinqphi1 ? sqrfrac (-sinqphi1) : ALMOST1; */
    cosphi1  = pcos (phi1);
    cosqphi1 = mulfrac (cosphi1, cosphi1);
#else  /* OLDWAY */
    ptrig (phi1, &sinphi1, &sinqphi1, &cosphi1, &cosqphi1);
#endif /* OLDWAY */
    cos4phi1 = mulfrac (cosqphi1, cosqphi1);

    t3 = -mulfrac (sinqphi1, eccsq);
    if (!t3)
        t3 = ALMOST1;
    N1 = sqrfrac (t3);
    R1 = divfrac (aesq, mulfrac (N1, t3));
    N1 = divfrac (major, N1);
    D  = divfrac (east << 3, mulfrac (N1, scale));
    Dsq = mulfrac (D, D);
```

```
    /*
    **  For tanphil, binary point moved to bit 27, ie bit
    **      28 = 1, bit 31 = 8.  Hence also, for TAN1 the point
    **      is at bit 23, and bit 31 = 128.
    */
    tanphil = divfrac (sinphil >> 4, cosphil);
    TAN1 = mulfrac (tanphil, tanphil); /* was T1, but QuickC barfed (!) */

    C1   = mulfrac (cosqphil, eccprsq);

/*
**  Coefficients as modified (for computation) from Snyder:
**  p2  = N1*tanphil/(R1 + R1);
**  p4  = (5.0 + 3.0*TAN1 + C1*(10.0 - C1*4.0) - 9.0*eccprsq)/12.0;
**  p6  = (61.0 + TAN1*(90.0 + TAN1*45.0) + C1*(298.0 - C1*3.0) - 252.0*eccprsq);
**  p6 /= 360.0;
**
**  Because of problems with loss of significance in integer arithmetic,
**      the formula for p6 has been recast in the form
**   p6  = cosqphil**2*[61 + C1*(298 - 3*C1) - 252*eccprsq] +
**            + 45*sinqphil*(1 + cosqphil);
**   p6 /= 360*cosqphil**2;
*/
    p2  = divfrac (mulfrac (N1, tanphil), R1 << 1);

    /*
    **  Bit 31 = 256, BP at 22
    **  Value of p4OVER12 puts final BP at 25, bit 31 = 32
    */
    p4  = mulfrac (p4OVER12,
                p4FIVE + TAN1 + (TAN1 >> 1) +
                mulfrac (C1, p4TEN - (C1 >> 7)) -
                mulfrac (p4NINE, eccprsq));

/*
**   p6  = cosqphil**2*[61 + C1*(298 - 3*C1) - 252*eccprsq] +
**            + 45*sinqphil*(1 + cosqphil);
**   p6 /= 360*cosqphil**2;
*/
    /* bit 31 = 256, BP at 22 */
    t1 = mulfrac (p6_252, eccprsq);
    t2 = mulfrac (C1, p6_298 - mulfrac (p6THREE, C1));
    t3 = 45*mulfrac (sinqphil, p6ONE + (cosqphil >> 9));
    p6 = t3 + mulfrac (cos4phil, p6_61 + t2 - t1);
    /* p6OVER360 renormalizes:  bit 31 = 1024, BP at 20 */
    p6 = mulfrac (p6OVER360, p6);
    p6 = divfrac (p6, cos4phil);

/*
**  Coefficients as modified (for computation) from Snyder:
**  L3  = (1.0 + 2.0*TAN1 + C1)/6.0;
**  L5  = (5.0 - C1*(2.0 + C1*3.0) + TAN1*(28.0 + TAN1*24.0) + 8.0*eccprsq)/120.0;
*/
    L3  = mulfrac (L3OVER6, L3ONE + (TAN1 << 1) + (C1 >> 8));
```

```
    t1    = mulfrac (L5EIGHT, eccprsq);
    t2    = L5_28 + (mulfrac (L5_24, TAN1) << 1);
    t2    = mulfrac (TAN1, t2 << 4) << 4;
    t3    = mulfrac (C1, L5TWO + mulfrac (L5THREE, C1));
    L5    = L5FIVE - t3 + t2 + t1;
    L5    = mulfrac (L5OVER120, L5);
}

unsigned long
converge ()
{
    unsigned long Csq, c2, c4, lambcos;
    UL t1, t2, t3;

/*
**  Coefficients as modified (for computation) from TM 5-241-8:
**      Csq = C*C;
**      c2  = (1.0 + Csq*(3.0 + 2.0*Csq));
**      c4  = (2.0 - T*T)/15.0;
**      lambcos = lambda*lambda*cosqphi;
*/
    Csq = mulfrac (C, C);
    /* '>> 3' means 'normalize and mult by 2' */
    c2  = c2ONE + mulfrac (Csq, c2THREE + (Csq >> 3));
    c4 = mulfrac (c4OVER15, c4TWO - mulfrac (T << 1, T));
    lambcos = mulfrac (lambda, mulfrac (lambda, cosqphi));

/*
**  Formula as modified (for computation) from TM 5-241-8:
**      return 60.0*DEG*lambda*sinphi*(1.0 + lambcos*(c2 + c4*lambcos));
*/
    t1 = mulfrac (lambcos << 8, c4 << 1) << 5;
    t2 = c0ONE + mulfrac (lambcos << 6, c2 + t1);
    t3 = mulfrac (lambda, mulfrac (sinphi, t2));

    return divfrac (t3 << 4, TWOPI);
}

void
UTMcoords (east, north)
long *east, *north;
{
      long t1, t2, t3, t4;

/*
**  Formula as modified (for computation) from Snyder:
**      *east = N*A*scale*(1.0 + Asq*(x2 + Asq*x4));
*/
#ifdef UNDEF
    *east = mulfrac
             (x0,
                 (x0ONE +
                     smulfrac (Asq << 2,
                         (x2 + smulfrac (Asq << 2, x4 << 2)) << 2)) << 3);
```

```
        t1   = smulfrac (Asq << 2, x4 << 2);
        t1  += x2;
        t2   = smulfrac (Asq << 2, t1 << 2);
        t2  += x0ONE;
        t3   = mulfrac (x0, t2 << 3);
#endif /* UNDEF */

        t1   = (x2 << 3) + smulfrac (Asq << 5, x4 << 2);
        t2   = (x0ONE << 6) + smulfrac (Asq << 7, t1);
        *east = (mulfrac (x0, t2) + 4) >> 3;

/*
**  Formula as modified (for computation) from Snyder:
**      M  = -m6*sin6phi;
**      M += m4*sin4phi;
**      M -= m2*sin2phi;
**      M += m0*phi;
**      M *= major;
*/
    t1 = smulfrac (sin4phi, m4 >> 1) - smulfrac (sin2phi, m2 >> 1);
    M = mulfrac (major, mulfrac (phi, m0) + (t1 >> 3));

/*
**  Formula as modified (for computation) from Snyder:
        *north  = Asq*y2*(1.0 + Asq*(y4 + Asq*y6));
        *north += M;
        *north *= scale;
*/
    t1 = y4 + smulfrac (Asq << 2, y6 << 2);
    t2 = y2ONE + smulfrac (Asq << 2, t1 << 2);
    t3 = mulfrac (Asq << 7, y2 << 3);
    t4 = mulfrac (t2 << 4, t3 << 4);
    *north = M + (t4 >> 3);
    *north = (mulfrac (*north, scale) + 16) >> 5;

    return; /* place for QC to stop */
}

void
geocoords (lon, lat)
long *lon, *lat;
{
    unsigned long t1, t2, t3;

/*
**  Formula as modified (for computation) from Snyder:
**      *lon = D*(1.0 - Dsq*(L3 - L5*Dsq))/cosphi1;
*/
    t1 = L3 - mulfrac (L5 << 2, Dsq << 2);
    t2 = L1ONE - mulfrac (Dsq, t1);
    t3 = mulfrac (D << 2, t2 << 3);
    *lon = divfrac (t3, cosphi1);
    *lon = divfrac (*lon, TWOPI);

/*
```

```
**  Formula as modified (for computation) from Snyder:
**      *lat = -p2*Dsq*(1.0 - Dsq*(p4 - Dsq*p6));
**      *lat += phi1;
*/
    t1 = mulfrac (Dsq, p6) << 5;
    t2 = p0ONE - (mulfrac (Dsq, p4 - t1) << 3);
    t3 = mulfrac (p2, Dsq) << 4;

    *lat = phi1 - mulfrac (t3, t2);
    *lat = divfrac (*lat, TWOPI);

    return;
}

/*
**  convert several ascii characters to decimal while padding
**     to the right with virtual '0' characters
*/
long
decimaln (ascii, n)
char *ascii; int n;
{
    long value;
    int j;

    /* strip leading whitespace */
    while (isspace (*ascii))
        ++ascii;

    /* convert to integer value */
    for (j = 0, value = 0L; isdigit (*ascii)  &&  j < n; ++j)
        value = 10L*value + (long) (*ascii++ - '0');

    /* pad with virtual zeros */
    while (j++ < n)
        value = 10L*value;

    return value;
}

void
utm2geo (UTM, longitude, latitude)
struct utm_parse *UTM;
UL *longitude, *latitude;
{
    UL zone;

    utmcoeff  (UTM->easting, UTM->northing);
    geocoords (longitude, latitude);

    if (UTM->gz > 30)
        zone = HALF + (((UL) UTM->gz - 30L)*(ZETAK >> 3)) - (ZETAK >> 4);
    else
        zone = HALF - ((31L - (UL) UTM->gz)*(ZETAK >> 3)) + (ZETAK >> 4);
```

```
    *longitude = (UTM->easting >= 0L)
        ? zone + *longitude
        : zone - *longitude;

    *latitude = (UTM->northing >= 0L)
        ? HALF + *latitude
        : HALF - *latitude;
}

void
displaygeo (longitude, latitude)
UL longitude, latitude;
{
#ifndef   FILTER
    printf ("Longitude = ");
    displayANG (longitude, 'E', 'W');
    printf (", Latitude = ");
    displayANG (latitude, 'N', 'S');
    printf (", zone = %d", newUTM.gz);
    putchar ('\n');
#else  /* FILTER */
    displayANG (longitude, 'E', 'W');
    putchar (',');
    displayANG (latitude, 'N', 'S');
#endif /* FILTER */
}

#ifdef    DECIMAL
void
displayANG (angle, flag, other)
UL angle; int flag, other;
{
    double deg;
    UL thou;

    if (angle >= HALF)
        thou = angle - HALF;
    else
        {
        flag = other;
        thou = HALF - angle;
        }
    thou = mulfrac (thou + 2, EMSEC + 1);
    if (!thou)
        flag = ' ';

    deg  = (double) thou/3600000.0;

#ifndef   FILTER
    printf ("%03.6lf%c", deg, flag);
#else  /* FILTER */
    if (flag == other)
        deg = -deg;
    printf ("%03.6lf", deg);
#endif /* FILTER */
```

```
}
#else  /* DECIMAL */
void
displayANG (angle, flag, other)
UL angle; int flag, other;
{
    long min, deg, sec;
    UL thou;

    if (angle >= HALF)
        thou = angle - HALF;
    else
        {
        flag = other;
        thou = HALF - angle;
        }
    thou = mulfrac (thou + 2, EMSEC + 1);
    if (!thou)
        flag = ' ';

    deg  = thou/(60L*60L*1000L);
    min  = (thou/(60L*1000L)) % 60L;
    sec  = (thou/1000L) % 60L;
    /* A jug of wine, a book of verses, and ... */
    thou = thou % 1000L;

    printf ("%3ld\xF8%02ld'%02ld\"%03ld%c", deg, min, sec, thou, flag);
}
#endif /* DECIMAL */

void
displayUTM (UTM)
struct utm_parse *UTM;
{
    long east, north;

#ifndef   FILTER
    printf ("UTM string is ");

    if (UTM->gz != -1)
        printf (" ZN %d", UTM->northing >= 0 ? UTM->gz : -UTM->gz);

    if (UTM->easting != 99999999L  &&  UTM->northing != 99999999L)
        {
        east  = UTM->easting + HAFM;
        north = UTM->northing >= 0L ? UTM->northing : UTM->northing + TENM;
        printf (" E %06ld N %07ld", east, north);
        }

    printf ("\n");
#else  /* FILTER */
    if (UTM->easting != 99999999L  &&  UTM->northing != 99999999L)
        {
        east  = UTM->easting + HAFM;
        north = UTM->northing >= 0L ? UTM->northing : UTM->northing + TENM;
```

```
        printf ("%06ld,%07ld", east, north);
        }
#endif /* FILTER */
}

void
displayconv (conv)
UL conv;
{
    long min, deg;
    UL thou;
    char *flag;

    flag = conv >= HALF ? "" : "-";
    thou = conv >= HALF  ? conv - HALF : HALF - conv; /* abs(conv) */
    thou = mulfrac (thou + 2, EMSEC + 1);      /* adjusted for accuracy */
    if (!thou)
        flag = "";

    thou += 30L*1000L;                /* round to nearest arcminute */
    deg  = thou/(60L*60L*1000L);
    min  = (thou/(60L*1000L)) % 60L;

    printf ("Convergence = %s%ld\xF8%02ld'\n", flag, deg, min);
}

/*
**  geo2utm expects:
**     A UTM structure to write to.
**     Normal geographic lon and lat in EAU
**          (corresponding to limits -180.0 to 180.0 degrees and
**          -80.0 to 84.0 degrees).
**     A gridzone number.  We can't use the mod function to
**          generate lambda because then a longitude such as
**          -120.0 deg would always generate lambda = -3.0 (in
**          gridzone 11), even when we want +3.0 (in 10).
*/
void
geo2utm (UTM, ulon, ulat, gz)
struct utm_parse *UTM;
UL ulon, ulat;
int gz;
{
    long deast, dnorth;
    long lon, lat, zone;

    UTM->gz = gz;

    if (gz > 30)
        zone = HALF + (((UL) UTM->gz - 30L)*(ZETAK >> 3)) - (ZETAK >> 4);
    else
        zone = HALF - ((31L - (UL) UTM->gz)*(ZETAK >> 3)) + (ZETAK >> 4);
    lon = ulon - zone;

    if (ulat > (UL) E84N)
```

```
        ulat = (UL) E84N;
    if (ulat < E80S)
        ulat = E80S;
    lat = ulat ^ HALF;

    geocoeff (lon, lat);
    newUTM.conver = converge();
    if ((lon > 0) ^ (lat < 0))
        newUTM.conver = HALF - newUTM.conver;
    else
        newUTM.conver = HALF + newUTM.conver;
    UTMcoords (&deast, &dnorth);

    if (lat < 0)
        dnorth = -dnorth;
    if (lon < 0)
        deast = -deast;

    UTM->easting  = deast;
    UTM->northing = dnorth;

    return; /* helps in QuickC */
}

void
UTMcopy (dest, srce)
struct utm_parse *dest, *srce;
{
    dest->gz       = srce->gz;
    dest->easting  = srce->easting;
    dest->northing = srce->northing;
    dest->conver   = srce->conver;
}

unsigned long
psin (x)
unsigned long x;
{
    UL xsq, temp;

    x <<= 2; /* partially renormalize x */
    if (x <= PI_DIV_4)
        {
        x <<= 1; /* finish renormalizing */
        xsq = mulfrac (x, x);
        temp = mulfrac (xsq, C98);
        temp = mulfrac (mulfrac (xsq, C76), -temp);
        temp = mulfrac (mulfrac (xsq, C54), -temp);
        temp = mulfrac (mulfrac (xsq, C32), -temp);
        return mulfrac (x,                  -temp);
        }
    else
        {
        x = PI_DIV_2 - x;
        if (x == 0)
```

```
            return ALMOST1;
        x <<= 1; /* finish renormalizing */
        xsq = mulfrac (x, x);
        temp  = mulfrac (xsq, CX9);
        temp  = mulfrac (mulfrac (xsq, C87), -temp);
        temp  = mulfrac (mulfrac (xsq, C65), -temp);
        temp  = mulfrac (mulfrac (xsq, C43), -temp);
        return -mulfrac (        (xsq >> 1), -temp);
        }
}

unsigned long
pcos (x)
unsigned long x;
{
    UL xsq, temp;

    x <<= 2; /* partially renormalize x */
    if (x > PI_DIV_4)
        {
        x = PI_DIV_2 - x;
        x <<= 1; /* finish renormalizing */
        xsq = mulfrac (x, x);
        temp = mulfrac (xsq, C98);
        temp = mulfrac (mulfrac (xsq, C76), -temp);
        temp = mulfrac (mulfrac (xsq, C54), -temp);
        temp = mulfrac (mulfrac (xsq, C32), -temp);
        return mulfrac (x,                  -temp);
        }
    else
        {
        if (x == 0)
            return ALMOST1;
        x <<= 1; /* finish renormalizing */
        xsq = mulfrac (x, x);
        temp  = mulfrac (xsq, CX9);
        temp  = mulfrac (mulfrac (xsq, C87), -temp);
        temp  = mulfrac (mulfrac (xsq, C65), -temp);
        temp  = mulfrac (mulfrac (xsq, C43), -temp);
        return -mulfrac (        (xsq >> 1), -temp);
        }
}

/*
**  For a quantity x whose binary point is just left of bit 31,
**      1.0 - x is well represented by -x, but here I choose to
**      to represent it by ~x so that 1.0 - 0.0 is near 1 rather
**      than 0.0.
*/
#define ONE_MINUS(x) ~(x)

void
ptrig (x, sinx, sinqx, cosx, cosqx)
unsigned long x, *sinx, *sinqx, *cosx, *cosqx;
{
```

```
    UL xsq, temp, tempsq, other, othersq;
    int octant;

    octant = 1;
    if (x <= PI_DIV_4)
        octant = 0;
    else
        x = PI_DIV_2 - x;

    x <<= 3; /* renormalize x */
    if (x == 0L)
        {
        temp  = tempsq  = 0L;
        other = othersq = ALMOST1;
        }
    else
        {
        xsq = mulfrac (x, x);
        temp = mulfrac (xsq, C98);
        temp = mulfrac (mulfrac (xsq, C76), ONE_MINUS(temp));
        temp = mulfrac (mulfrac (xsq, C54), ONE_MINUS(temp));
        temp = mulfrac (mulfrac (xsq, C32), ONE_MINUS(temp));
        temp = mulfrac (x,                  ONE_MINUS(temp));
        tempsq = mulfrac (temp, temp);
        other  = sqrfrac (ONE_MINUS(tempsq));
        othersq = mulfrac (other, other);
        }

    if (!octant)
        {
        *sinx  = temp;
        *sinqx = tempsq;
        *cosx  = other;
        *cosqx = othersq;
        }
    else
        {
        *sinx  = other;
        *sinqx = othersq;
        *cosx  = temp;
        *cosqx = tempsq;
        }
}

void
sighandler (signal)
int signal;
{
    printf ("\nTerminated at operator's request.\n");

    exit (0);
}
```

```
void
setsigs ()
{
    signal (SIGINT, sighandler);
}
```

INTMATH.ASM

```
;   Integer math routines

;   G. E. Bloch, Oct 1989-Jun 1990
;   Copyright (c) Etak Corp, 1989,1990

copyright       segment byte public 'code'
copr    db      'Copyright (c) Etak Corp, 1989,1990',0
copyright       ends

arg1    equ     word ptr  6[bp] ; first arg
arg2    equ     word ptr 10[bp] ; second arg

prod0   equ     word ptr 0[bx]  ; build product in these four words
prod1   equ     word ptr 2[bx]
prod2   equ     word ptr 4[bx]
prod3   equ     word ptr 6[bx]

;  mulfrac

;    Multiply two 32-bit unsigned fractions, returning a
;        32-bit unsigned fraction.  An "unsigned fraction" is a
;        number whose binary point is at the left end of the word,
;        so that the MSB has value 1/2.

;   Invocation:  x = mulfrac (a, b);

;       entry:   two unsigned longs on stack

;       return:  unsigned long product in dx:ax

mulfrac         segment byte public 'code'
        assume  cs:mulfrac,ds:nothing
        public  _mulfrac

_mulfrac        proc    far
        push    bp
        mov     bp,sp
        push    bx
        push    cx
        push    ds

        xor     dx,dx           ; create high-order zero in case ...
        mov     ax,arg1         ; look for zero args
        or      ax,arg1+2
        jz      mulend
        mov     ax,arg2
        or      ax,arg2+2
        jz      mulend

        ; adjust stack pointer and clear work area
```

```
        xor     ax,ax
        push    ax
        push    ax
        push    ax
        push    ax

        mov     ax,ss           ; get argument addressability
        mov     ds,ax
        mov     bx,sp

        mov     ax,arg1         ; generate low-order "digit" of product
        mul     arg2            ;   (these digits are 16 bits long,
        mov     prod0,ax        ;    ie, base = 65536)
        mov     prod1,dx

        mov     ax,arg1         ; generate half of second "digit"
        mul     arg2+2
        add     prod1,ax
        adc     prod2,dx
        adc     prod3,0

        mov     ax,arg1+2       ; generate other half of second digit
        mul     arg2
        add     prod1,ax
        adc     prod2,dx
        adc     prod3,0

        mov     ax,arg1+2       ; generate third and fourth digits
        mul     arg2+2
        add     prod2,ax
        adc     prod3,dx

        pop     ax              ; clear workarea
        pop     ax
        pop     ax              ; get result into registers
        pop     dx

mulend:
        pop     ds
        pop     cx
        pop     bx
        pop     bp
        ret

_mulfrac        endp
mulfrac         ends

divfrac         segment byte public 'code'
        assume  cs:divfrac,ds:nothing
        public  _divfrac

_divfrac        proc    far
        push    bp
        mov     bp,sp
        push    bx
```

```
        push    cx
        push    si

        mov     ax,arg1         ; get arg1
        mov     dx,arg1+2       ;     arg1 is in ax:dx

        cmp     dx,arg2+2       ; avoid num >= denom
        jb      div_ok
        ja      div_ng
        cmp     ax,arg2
        jb      div_ok
div_ng:
        xor     ax,ax           ; bad situation, return (almost) 1.0
        not     ax
        mov     dx,ax
        jmp     end_div

div_ok:
        xor     si,si           ; set up high-order (5th byte) of numerator
        clc                     ; normalize numerator
        rcl     ax,1
        rcl     dx,1
        rcl     si,1

        call    divhalf         ; get high-order word of quotient
        push    bx              ; save same
        call    divhalf         ; get low-order word of quotient
        push    bx              ; save same
        ; next call understands how previous call leaves the regs
        call    round           ; see if round up is needed
        pop     ax              ; set up return value
        pop     dx
        adc     ax,0            ; apply the rounding
        adc     dx,0

end_div:
        pop     si              ; return
        pop     cx              ; return
        pop     bx              ; return
        pop     bp
        ret

divhalf proc    near
        xor     bx,bx           ; clear quotient
        mov     cx,8000h        ; loop count and quotient bit

divloop:
        or      si,si           ; see if numerator is big
        jnz     got_bit         ; it is (HO denominator is implied 0)
        cmp     dx,arg2+2
        jb      loopend
        ja      got_bit
        cmp     ax,arg2
        jb      loopend
```

```
got_bit:
        or      bx,cx           ; got a bit, put it in high bit of quotient
        sub     ax,arg2         ; subtract denominator
        sbb     dx,arg2+2
        sbb     si,0

loopend:
        clc                     ; shift divisor left
        rcl     ax,1
        rcl     dx,1
        rcl     si,1

        or      ax,ax           ; can quit when num is zero
        jnz     final
        or      dx,dx
        jnz     final
        or      si,si           ; see if numerator is big
        jz      endhalf

final:
        clc                     ; loop control
        rcr     cx,1
        or      cx,cx
        jnz     divloop

endhalf:
        ret

round   proc    near

        or      si,si           ; see if numerator is big
        jnz     setround        ; it is (HO denominator is implied 0)
        cmp     dx,arg2+2
        jb      clrround
        ja      setround
        cmp     ax,arg2
        jb      clrround

setround:
        stc
        ret

clrround:
        clc
        ret

round   endp

divhalf endp

_divfrac        endp
divfrac         ends

sqrfrac         segment byte public 'code'
        assume  cs:sqrfrac,ds:nothing
```

```
        public  _sqrfrac

_sqrfrac        proc far
        push    bp
        mov     bp,sp
        push    bx
        push    cx
        push    di

        mov     ax,arg1         ; get argument
        mov     dx,arg1+2

        mov     bx,ax           ; test for zero arg
        or      bx,dx
        jz      endsqr          ; arg zero, answer is zero

        cmp     dx,0FFFFh       ; arguments near "1.0" get
        jne     mainsqr         ;     special treatment
        cmp     ax,0FFA5h
        jb      mainsqr
        stc                     ;     namely  (1.0 - x) = 1.0 - x/2.0
        rcr     dx,1            ; when binary pt is left of MSB,
        rcr     ax,1            ;     -y is equivalent to 1.0 - y
        jmp     endsqr

mainsqr:
        xor     cx,cx           ; count zeros before the first 1 bit
        clc
normloop:
        inc     cx
        rcl     ax,1            ; carry is always clear here
        rcl     dx,1
        jnc     normloop        ; this can't fail because we know arg != 0

guessloop:
        rcr     dx,1            ; rotate back but only half as far
        rcr     ax,1            ; always rotates 0 into carry
        dec     cx
        loopnz  guessloop       ; decrement cx again

newton:
        mov     di,4            ; this many iterations

newtonloop:
        mov     bx,arg1         ; get arg into call
        mov     cx,arg1+2
        push    dx              ; denominator is latest guess
        push    ax
        push    cx              ; numerator is argument
        push    bx
        call    _divfrac        ; get x   guess in ax:dx
        pop     bx              ; pop old x-value ...
        pop     bx              ;   ... so as to enable us to ...
        pop     bx              ;   ... get previous guess
        pop     cx
```

```
        add     ax,bx           ; next guess is (guess + x/guess)/2
        adc     dx,cx
        rcr     dx,1            ; divide sum by two
        rcr     ax,1

        dec     di              ; loop control
        jnz     newtonloop


endsqr:
        pop     di              ; return
        pop     cx
        pop     bx
        pop     bp
        ret

_sqrfrac        endp
sqrfrac         ends

smulfrac        segment byte public 'code'
        assume  cs:smulfrac,ds:nothing
        public  _smulfrac

_smulfrac       proc far
        push    bp
        mov     bp,sp
        push    bx

        xor     dx,dx           ; create high-order zero in case ...
        mov     ax,arg1         ; look for zero args
        or      ax,arg1+2
        jz      smulend
        mov     ax,arg2
        or      ax,arg2+2
        jz      smulend

        mov     ax,arg1         ; get first arg
        mov     dx,arg1+2
        mov     bx,dx           ; all I want here is the sign bit
        test    dx,8000h        ; if positive, push arg1 as is
        jz      smularg1
        not     dx              ; take abs of arg1
        neg     ax
        cmc
        adc     dx,0
smularg1:
        clc                     ; double arg1 before calling mulfrac
        rcl     ax,1
        rcl     dx,1
        push    dx
        push    ax

        mov     ax,arg2         ; get second arg
        mov     dx,arg2+2
        xor     bx,dx           ; I still only care about the sign bit
```

```
        test    dx,8000h        ; if positive, push arg2 as is
        jz      smularg2
        not     dx              ; take abs of arg2
        neg     ax
        cmc
        adc     dx,0
smularg2:
        clc                     ; double arg2 before calling mulfrac
        rcl     ax,1
        rcl     dx,1
        push    dx
        push    ax

        call    _mulfrac        ; do the mult
        add     sp,8            ; clear args from stack

        test    bx,8000h        ; if negative, change sign of result
        jz      smulend

        not     dx              ; take abs of result
        neg     ax
        cmc
        adc     dx,0

smulend:
        pop     bx              ; return
        pop     bp
        ret

_smulfrac       endp
smulfrac        ends

        end
```

Appendix 3. Raw Digitization Data

This appendix presents the raw data for the analysis of digital map accuracy described in Chapter 6. Tables 43 through 47 show the sample point UTM coordinates measured and interpolated from the UTM grids on the map sheets as "paper map data," and the data reported by the computer directly from the digitized data as "digitized map data (raw)." Since the computer gives longitude and latitude in decimal degrees, program utm2geo was used to convert the positions into UTM coordinates, shown as "digitized map data (derived)."

Tables 48 and 49 present the measured positions of intersections of UTM grid lines in the same format. The adopted values are just the UTM *coordinates of the* chosen grid intersections, and the measured coordinates are their longitude and latitude as read directly from the map sheets by measurement and interpolation.

Finally, Tables 50 through 54 show the same paper map data as Tables 43 through 47, but the data are compared to coordinates of the test points read from the Navigator screen. These data, labelled "Navigator Map Data," were read by carefully positioning the Navigator's car cursor on each test point in turn and reading the resulting UTM coordinates directly from the Navigator's Map Information Screen (see Figure 5).

Table 43. Raw Digitization Data - Purmela

Digitization Errors						
	Paper Map Data (Interpolated)		Digitized Map Data (Raw)		(Derived)	
Point	Easting	Northing	Longitude	Latitude	Easting	Northing
A1	596507	3484822	-97.983803	31.495745	596511	3484806
B1	604979	3484781	-97.894454	31.494587	604999	3484760
C1	610839	3484623	-97.832849	31.492546	610852	3484595
D1	617468	3484092	-97.763062	31.487135	617488	3484067
A2	597506	3474600	-97.974188	31.403363	597520	3474576
B2	604186	3475407	-97.903926	31.409973	604193	3475373
C2	609295	3475463	-97.850183	31.410054	609302	3475434
D2	617621	3476317	-97.762323	31.416927	617646	3476286
A3	596903	3467804	-97.981238	31.342031	596913	3467771
B3	602445	3467990	-97.922974	31.343186	602454	3467952
C3	610190	3466774	-97.841589	31.331568	610210	3466743
D3	617411	3467547	-97.765690	31.337840	617424	3467517
A4	596895	3458413	-97.981972	31.257513	596929	3458403
B4	603769	3458823	-97.910039	31.260466	603775	3458796
C4	610352	3458378	-97.840937	31.255802	610360	3458346
D4	617328	3458897	-97.767712	31.259842	617328	3458869

Table 44. Raw Digitization Data - Gatesville

Digitization Errors						
	Paper Map Data (Interpolated)		Digitized Map Data (Raw)		(Derived)	
Point	Easting	Northing	Longitude	Latitude	Easting	Northing
E1	620787	3484089	-97.728335	31.486966	620787	3484086
F1	627567	3484631	-97.656989	31.491086	627558	3484624
G1	634411	3483757	-97.585102	31.482420	634399	3483749
H1	641410	3484168	-97.511226	31.485330	641413	3484165
E2	620565	3476240	-97.731548	31.416097	620572	3476228
F2	627402	3477132	-97.659401	31.423439	627421	3477123
G2	634581	3477481	-97.584087	31.425746	634576	3477468
H2	641463	3476628	-97.511737	31.417149	641467	3476606
E3	619227	3467678	-97.746686	31.338974	619230	3467663
F3	627264	3467358	-97.662246	31.335211	627269	3467340
G3	634373	3467815	-97.587540	31.338599	634372	3467804
H3	641498	3467785	-97.512666	31.337371	641498	3467762
E4	620562	3459642	-97.733650	31.266580	620563	3459653
F4	627364	3460180	-97.662132	31.270638	627367	3460183
G4	635913	3460525	-97.572353	31.272708	635912	3460520
H4	641660	3460420	-97.511909	31.271188	641669	3460427

Table 45. Raw Digitization Data - Fort Hood

Digitization Errors						
Paper Map Data (Interpolated)			Digitized Map Data (Raw)		(Derived)	
Point	Easting	Northing	Longitude	Latitude	Easting	Northing
A5	596420	3456686	-97.987667	31.242203	596403	3456701
B5	603947	3456459	-97.908610	31.239274	603935	3456449
C5	611479	3456944	-97.829337	31.243142	611480	3456955
D5	618367	3456412	-97.757288	31.237580	618348	3456413
A6	596842	3448690	-97.983930	31.169923	596832	3448694
B6	603863	3448868	-97.910278	31.170709	603851	3448848
C6	611555	3448462	-97.829539	31.166538	611551	3448464
D6	617055	3448479	-97.771966	31.165975	617039	3448461
A7	596931	3440617	-97.983743	31.097068	596924	3440619
B7	603806	3439202	-97.911766	31.083686	603804	3439201
C7	611785	3440148	-97.827853	31.091519	611799	3440151
D7	618348	3439691	-97.759183	31.086790	618355	3439698
A8	597190	3430646	-97.982256	31.007080	597157	3430647
B8	603507	3431132	-97.915869	31.010866	603491	3431127
C8	612304	3431538	-97.823509	31.013654	612305	3431525
D8	618291	3431747	-97.760907	31.014949	618280	3431734

Table 46. Raw Digitization Data - Killeen

Digitization Errors						
Paper Map Data (Interpolated)			Digitized Map Data (Raw)		(Derived)	
Point	Easting	Northing	Longitude	Latitude	Easting	Northing
E5	621164	3456351	-97.727607	31.236585	621177	3456335
F5	628387	3456253	-97.651886	31.235075	628390	3456253
G5	635052	3456948	-97.581837	31.240549	635055	3456943
H5	641253	3456478	-97.516808	31.235614	641255	3456478
E6	620269	3448513	-97.737826	31.166107	620292	3448512
F6	626853	3448290	-97.668927	31.163256	626863	3448273
G6	634505	3448195	-97.588808	31.161552	634502	3448179
H6	641604	3448167	MISSING	MISSING	MISSING	MISSING
E7	620601	3439557	-97.735632	31.085161	620604	3439543
F7	627569	3439569	-97.662559	31.084518	627576	3439553
G7	634462	3439870	-97.590439	31.086301	634453	3439836
H7	641289	3440693	-97.518640	31.093029	641292	3440671
E8	620232	3431021	-97.740601	31.008324	620226	3431021
F8	627248	3431162	-97.666992	31.008780	627253	3431154
G8	633684	3431203	-97.599590	31.008349	633689	3431185
H8	640562	3432323	-97.527511	31.017821	640557	3432324

Table 47. Raw Digitization Data - Test Roads

Digitization Errors						
	Paper Map Data (Interpolated)		Digitized Map Data (Raw)		(Derived)	
E/W Run						
Point	Easting	Northing	Longitude	Latitude	Easting	Northing
2	623751	3447244	-97.701624	31.154160	623758	3447228
3	624663	3446943	MISSING	MISSING	MISSING	MISSING
4	625448	3446547	-97.684041	31.147548	625443	3446515
5	626383	3446381	-97.674164	31.145921	626387	3446346
6	627718	3447084	-97.659908	31.152284	627738	3447067
7	628493	3447525	-97.651880	31.156184	628498	3447509
8	631816	3448141	-97.616720	31.161303	631842	3448118
9	632716	3447603	-97.607573	31.156387	632721	3447584
10	634615	3446379	-97.587892	31.145075	634613	3446354
11	635630	3445705	-97.577325	31.138892	635629	3445681
12	636409	3445251	-97.569182	31.134705	636412	3445227
13	637974	3444504	-97.552874	31.127731	637977	3444474
14	639111	3443861	-97.541148	31.121923	639103	3443845
N/S Run						
Point	Easting	Northing	Longitude	Latitude	Easting	Northing
1	622930	3447463	-97.710266	31.156220	622932	3447447
2	623256	3448502	-97.706750	31.165579	623255	3448488
2A	625365	3448148	-97.684679	31.162119	625363	3448129
3	626093	3448703	-97.676940	31.166992	626094	3448678
4	627156	3449307	-97.665726	31.172369	627156	3449287
5	628321	3450104	-97.653246	31.179456	628336	3450087
6	629495	3450969	-97.640911	31.187090	629501	3450947
6A	629707	3451696	-97.638577	31.193646	629715	3451677
7A	630122	3453553	-97.634038	31.210443	630124	3453544
8	629872	3455257	-97.636377	31.225877	629880	3455252

Table 48. Raw Map Data - Purmela

UTM Grid Errors						
Adopted Data UTM Points Geographic			Measured Coords Derived Values			
Point	Easting	Northing	Longitude	Latitude	Easting	Northing
A1	597000	3485000	-97.978601	31.497374	597003	3484992
B1	604000	3485000	-97.904869	31.496714	604007	3484986
C1	610000	3485000	-97.841673	31.496109	610010	3484981
D1	617000	3485000	-97.767878	31.495442	617020	3484983
A2	597000	3477000	-97.979392	31.425235	597003	3476995
B2	604000	3477000	-97.905709	31.424565	604007	3476988
C2	610000	3477000	-97.842524	31.423951	610013	3476982
D2	617000	3477000	-97.768798	31.423265	617022	3476982
A3	597000	3468000	-97.980304	31.344036	596999	3467995
B3	604000	3468000	-97.906666	31.343378	604005	3467989
C3	610000	3468000	-97.843591	31.342817	610007	3467988
D3	617000	3468000	-97.769953	31.342187	617013	3467994
A4	597000	3459000	-97.981182	31.262707	596999	3458980
B4	604000	3459000	-97.907653	31.262064	604001	3458975
C4	610000	3459000	-97.844594	31.261484	610006	3458972
D4	617000	3459000	-97.771055	31.260839	617009	3458976

Table 49. Raw Map Data - Killeen

UTM Grid Errors						
Adopted Data UTM Points Geographic			Measured Coords Derived Values			
Point	Easting	Northing	Longitude	Latitude	Easting	Northing
E5	621000	3456000	-97.729544	31.233624	620996	3456005
F5	627000	3456000	-97.666511	31.232971	627000	3456003
G5	634000	3456000	-97.593045	31.232170	633999	3456001
H5	641000	3456000	-97.519569	31.231395	640999	3456007
E6	621000	3448000	-97.730404	31.161474	621006	3448007
F6	627000	3448000	-97.667496	31.160833	627003	3448006
G6	634000	3448000	-97.594123	31.160270	633998	3448030
H6	641000	3448000	-97.520749	31.159264	640993	3448010
E7	621000	3440000	-97.731413	31.089237	621001	3439999
F7	627000	3440000	-97.668545	31.088587	626999	3439997
G7	634000	3440000	-97.595224	31.087795	633994	3439996
H7	641000	3440000	-97.521897	31.086993	640991	3439998
E8	621000	3431000	-97.732555	31.008030	620995	3430997
F8	627000	3431000	-97.669693	31.007416	626997	3430999
G8	634000	3431000	-97.596375	31.006618	633998	3430997
H8	641000	3431000	-97.523094	31.005816	640996	3430999

Table 50. Raw Navigator Data - Purmela

Navigator Errors				
Paper Map Data (Interpolated)			Navigator Map Data (From Map Info Screen)	
Point	Easting	Northing	Easting	Northing
A1	596507	3484822	596516	3484811
B1	604979	3484781	604999	3484762
C1	610839	3484623	610851	3484599
D1	617468	3484092	617481	3484068
A2	597506	3474600	597516	3474576
B2	604186	3475407	604194	3475373
C2	609295	3475463	609298	3475432
D2	617621	3476317	617636	3476282
A3	596903	3467804	596912	3467770
B3	602445	3467990	602455	3467954
C3	610190	3466774	610209	3466743
D3	617411	3467547	617425	3467523
A4	596895	3458413	596933	3458402
B4	603769	3458823	603776	3458800
C4	610352	3458378	610361	3458349
D4	617328	3458897	617331	3458874

Table 51. Raw Navigator Data - Gatesville

Navigator Errors				
Paper Map Data (Interpolated)			Navigator Map Data (From Map Info Screen)	
Point	Easting	Northing	Easting	Northing
E1	620787	3484089	620781	3484094
F1	627567	3484631	627559	3484629
G1	634411	3483757	634404	3483749
H1	641410	3484168	641414	3484165
E2	620565	3476240	620579	3476230
F2	627402	3477132	627421	3477127
G2	634581	3477481	634577	3477471
H2	641463	3476628	641469	3476606
E3	619227	3467678	619232	3467665
F3	627264	3467358	627268	3467341
G3	634373	3467815	634372	3467808
H3	641498	3467785	641499	3467768
E4	620562	3459642	620559	3459655
F4	627364	3460180	627363	3460187
G4	635913	3460525	635916	3460522
H4	641660	3460420	641668	3460425

Table 52. Raw Navigator Data - Fort Hood

Navigator Errors				
Paper Map Data (Interpolated)			Navigator Map Data (From Map Info Screen)	
Point	Easting	Northing	Easting	Northing
A5	596420	3456686	596403	3456708
B5	603947	3456459	603937	3456454
C5	611479	3456944	611481	3456952
D5	618367	3456412	618350	3456416
A6	596842	3448690	596838	3448695
B6	603863	3448868	603856	3448852
C6	611555	3448462	611546	3448464
D6	617055	3448479	617040	3448466
A7	596931	3440617	596927	3440618
B7	603806	3439202	603806	3439207
C7	611785	3440148	611800	3440154
D7	618348	3439691	618349	3439690
A8	597190	3430646	597159	3430650
B8	603507	3431132	603491	3431131
C8	612304	3431538	612305	3431537
D8	618291	3431747	618283	3431740

Table 53. Raw Navigator Data - Killeen

Navigator Errors				
	Paper Map Data (Interpolated)		Navigator Map Data (From Map Info Screen)	
Point	Easting	Northing	Easting	Northing
E5	621164	3456351	621176	3456339
F5	628387	3456253	628388	3456255
G5	635052	3456948	635055	3456950
H5	641253	3456478	641255	3456482
E6	620269	3448513	620290	3448513
F6	626853	3448290	626865	3448278
G6	634505	3448195	634496	3448182
H6	641604	3448167	MISSING	MISSING
E7	620601	3439557	620601	3439544
F7	627569	3439569	627574	3439552
G7	634462	3439870	634465	3439849
H7	641289	3440693	641294	3440680
E8	620232	3431021	620225	3431020
F8	627248	3431162	627251	3431163
G8	633684	3431203	633688	3431185
H8	640562	3432323	640553	3432330

Table 54. Raw Navigator Data - Test Roads

Navigator Errors				
	Paper Map Data (Interpolated)		Navigator Map Data (From Map Info Screen)	
E/W Run				
Point	Easting	Northing	Easting	Northing
2	623751	3447244	623752	3447228
3	624663	3446943	624666	3446920
4	625448	3446547	625442	3446522
5	626383	3446381	626383	3446355
6	627718	3447084	627741	3447068
7	628493	3447525	628496	3447513
8	631816	3448141	631842	3448120
9	632716	3447603	632720	3447586
10	634615	3446379	634616	3446354
11	635630	3445705	635631	3445682
12	636409	3445251	636404	3445221
13	637974	3444504	637978	3444476
14	639111	3443861	639105	3443848
N/S Run				
Point	Easting	Northing	Easting	Northing
1	622930	3447463	622931	3447451
2	623256	3448502	623256	3448490
2A	625365	3448148	625365	3448134
3	626093	3448703	626095	3448685
4	627156	3449307	627160	3449292
5	628321	3450104	628332	3450089
6	629495	3450969	629500	3450950
6A	629707	3451696	629714	3451678
7A	630122	3453553	630122	3453548
8	629872	3455257	629878	3455256

Appendix 4. Raw Navigation Data

Appendix 4 presents the raw data from all the test runs of the modified Navigator made at Fort Hood, Texas, in April and June of 1990, as discussed in Chapters 7 and 8. All twenty tables are in the same format. Each table has a tabular presentation of the reference data for the run, labeled "reference data" for the April runs and, to emphasize the source of the data, "GPS data" (for "Global Positioning System data") for the June runs. This tabulation is followed by a tabulation of the positions reported by the Navigator at each check point, labelled by the naming conventions described below. The checkpoints are labelled by numbers and letters, showing — for April — the results of changes as field experience showed that some original points were not useful, and that some new points had to be added for better coverage of the route.

The April reference data were obtained by interpolation from the paper DMA maps, using an estimate by eye of the position of each test point. This was itself subject to errors, since there was no practical way to estimate the position of the two edges of the road, and since in many cases the crossing road was rather vaguely defined in practice. For example, some roads jog where they cross the main road, and some curve just where they meet the main road (these two problems sometimes occur together). The June reference data were much more consistent, since the test points were defined by ETL personnel, who drove nails into the pavement at the center of small (30-50 cm) painted triangles, one on each side of the road at each checkpoint, and then measured the UTM coordinates of each directly by means of GPS equipment, using differential techniques for an accuracy of some 20 cm. It will be seen that the GPS coordinates are different for the two directions of travel on each route, since such accuracy allows meaningful measurements on each side of a road.

The Navigator position data were read by computer, using the RS-232C port and software built into the Navigator. A laptop computer was programmed to interrogate the Navigator for UTM coordinates and other information every 15 seconds during a test drive. At each test point, the CUCV was stopped at the marked position and the Navigator was interrogated for the same data by pressing a function key on the laptop. (The checkpoint samples were labelled as such by the laptop's software for ease in subsequent retrieval.) It is the UTM coordinates read from the Navigator at these checkpoints which are presented in the tables below.

The test runs are labelled by route, direction, and map-matching status. The first two letters of the name of a set of runs are "WE", "EW", "SN", or "NS", corresponding to a run in each direction on each route, the east-west and north-south routes. The last two letters of the name are "MM" or "DR", corresponding to map matching or dead-reckoning (that is, map matching off). An individual run is named by adding a number to the four-letter name of its set name. The April runs were numbered 0, 1, and 2. The first set of runs in June were numbered from 1 to 5 or 6 (because some extra runs were made on some sets). The final set of June runs, made by ETL personnel alone, were numbered 10 through 14 to distinguish them from the prior runs. Thus, Table 55 is WEMM, the raw data for the west-to-east runs (in April) and it contains data for runs WEMM0, WEMM1, and WEMM2.

Table 55. Raw Test Run Data - WEMM April

Reference Data		
Point	Easting	Northing
2	623747	3447243
3	624661	3446940
4	625445	3446543
5	626385	3446380
6	627712	3447082
7	628490	3447523
8	631813	3448142
9	632717	3447602
10	634613	3446376
11	635621	3445700
12	636405	3445247
13	637969	3444504
14	639111	3443860

	WEMM0 Data		WEMM1 Data		WEMM2 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
2	623791	3447181	623793	3447181	623789	3447178
3	624710	3446885	624709	3446883	624705	3446885
4	625483	3446480	625482	3446485	625477	3446487
5	626393	3446326	626395	3446327	626392	3446330
6	627744	3447015	627744	3447018	627750	3447024
7	628491	3447463	628491	3447464	628494	3447468
8	631839	3448085	631837	3448083	631825	3448079
9	632693	3447591	632696	3447591	632700	3447592
10	634633	3446354	634633	3446355	634633	3446357
11	635599	3445717	635599	3445718	635604	3445715
12	636392	3445253	636390	3445257	636397	3445256
13	637953	3444496	637948	3444498	637952	3444495
14	639099	3443828	639101	3443837	639110	3443831

Table 56. Raw Test Run Data - EWMM April

Reference Data		
Point	Easting	Northing
14	639111	3443860
13	637969	3444504
12	636405	3445247
11	635621	3445700
10	634613	3446376
9	632717	3447602
8	631813	3448142
7	628490	3447523
6	627712	3447082
5	626385	3446380
4	625445	3446543
3	624661	3446940
2	623747	3447243

	EWMM0 Data		EWMM1 Data		EWMM2 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
14	639142	3443825	639144	3443823	639140	3443825
13	638021	3444465	638017	3444466	638012	3444468
12	636453	3445215	636451	3445217	636451	3445224
11	635663	3445685	635660	3445683	635658	3445682
10	634681	3446311	634689	3446306	634685	3446306
9	632753	3447561	632763	3447556	632761	3447557
8	631897	3448090	631902	3448090	631903	3448089
7	628545	3447472	628556	3447472	628548	3447472
6	627772	3447032	627773	3447033	627769	3447029
5	626409	3446327	626401	3446332	626382	3446325
4	MISSING	MISSING	625477	3446490	625469	3446499
3	624708	3446886	624701	3446885	624691	3446890
2	623791	3447183	623785	3447183	623776	3447185

Table 57. Raw Test Run Data - SNMM April

Reference Data		
Point	Easting	Northing
14	639111	3443860
13	637969	3444504
12	636405	3445247
11	635621	3445700
10	634613	3446376
9	632717	3447602
8	631813	3448142
7	628490	3447523
6	627712	3447082
5	626385	3446380
4	625445	3446543
3	624661	3446940
2	623747	3447243

	EWMM0 Data		EWMM1 Data		EWMM2 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
14	639142	3443825	639144	3443823	639140	3443825
13	638021	3444465	638017	3444466	638012	3444468
12	636453	3445215	636451	3445217	636451	3445224
11	635663	3445685	635660	3445683	635658	3445682
10	634681	3446311	634689	3446306	634685	3446306
9	632753	3447561	632763	3447556	632761	3447557
8	631897	3448090	631902	3448090	631903	3448089
7	628545	3447472	628556	3447472	628548	3447472
6	627772	3447032	627773	3447033	627769	3447029
5	626409	3446327	626401	3446332	626382	3446325
4	MISSING	MISSING	625477	3446490	625469	3446499
3	624708	3446886	624701	3446885	624691	3446890
2	623791	3447183	623785	3447183	623776	3447185

Table 58. Raw Test Run Data - NSMM April

Reference Data		
Point	Easting	Northing
8	629869	3455257
7A	630123	3453552
6A	629709	3451694
6	629496	3450965
5	628319	3450102
4	627151	3449302
3	626091	3448698
2A	625367	3448168
2	623258	3448499
1	622927	3447469

	NSMM0 Data		NSMM1 Data		NSMM2 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
8	629909	3455235	629907	3455235	629911	3455232
7A	630153	3453553	630156	3453564	630155	3453557
6A	629736	3451625	629727	3451595	629728	3451600
6	629531	3450941	629520	3450912	629521	3450916
5	628338	3450016	628345	3450011	628344	3450012
4	627164	3449251	627167	3449246	627170	3449246
3	626134	3448655	626125	3448651	626136	3448655
2A	625347	3448065	625355	3448066	625357	3448068
2	623297	3448457	623298	3448460	623297	3448448
1	622972	3447430	622973	3447425	622972	3447421

Table 59. Raw Test Run Data - WEDR April

Reference Data		
Point	Easting	Northing
2	623747	3447243
3	624661	3446940
4	625445	3446543
5	626385	3446380
6	627712	3447082
7	628490	3447523
8	631813	3448142
9	632717	3447602
10	634613	3446376
11	635621	3445700
12	636405	3445247
13	637969	3444504
14	639111	3443860

	WEDR0 Data		WEDR1 Data		WEDR2 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
2	623793	3447181	623747	3447243	623791	3447183
3	624718	3446905	624661	3446940	624717	3446920
4	625512	3446546	625445	3446543	625497	3446539
5	626438	3446362	626385	3446380	626413	3446326
6	627792	3447097	627712	3447082	627766	3447054
7	628518	3447587	628490	3447523	628491	3447542
8	631862	3448242	631813	3448142	631832	3448158
9	632726	3447795	632717	3447602	632687	3447700
10	634680	3446595	634613	3446376	634608	3446452
11	635663	3445990	635621	3445700	635580	3445831
12	636466	3445546	636405	3445247	636372	3445368
13	638055	3444845	637969	3444504	637939	3444621
14	639220	3444264	639111	3443860	639082	3444001

Table 60. Raw Test Run Data - EWDR April

Reference Data		
Point	Easting	Northing
14	639111	3443860
13	637969	3444504
12	636405	3445247
11	635621	3445700
10	634613	3446376
9	632717	3447602
8	631813	3448142
7	628490	3447523
6	627712	3447082
5	626385	3446380
4	625445	3446543
3	624661	3446940
2	623747	3447243

	EWDR0 Data		EWDR1 Data		EWDR2 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
14	639144	3443823	639142	3443823	639138	3443827
13	638028	3444512	638024	3444503	637995	3444470
12	636465	3445269	636457	3445247	636414	3445181
11	635673	3445736	635660	3445705	635612	3445628
10	634705	3446357	634685	3446317	634625	3446218
9	632798	3447649	632767	3447591	632684	3447452
8	631953	3448173	631919	3448110	631828	3447953
7	628605	3447618	628574	3447535	628484	3447381
6	627862	3447153	627838	3447057	627743	3446912
5	626526	3446413	626512	3446301	626406	3446169
4	625625	3446633	625611	3446512	625501	3446379
3	624853	3447035	624834	3446907	624720	3446766
2	623931	3447314	623911	3447183	623797	3447039

Table 61. Raw Test Run Data - SNDR April

Reference Data		
Point	Easting	Northing
1	622927	3447469
2	623258	3448499
2A	625367	3448168
3	626091	3448698
4	627151	3449302
5	628319	3450102
6	629496	3450965
6A	629709	3451694
7A	630123	3453552
8	629869	3455257

	SNDR0 Data		SNDR1 Data		SNDR2 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
1	622964	3447397	622962	3447394	622964	3447397
2	623220	3448464	623171	3448460	623256	3448447
2A	625254	3448062	625149	3448054	625359	3448022
3	626014	3448616	625892	3448619	626128	3448573
4	627080	3449233	626782	3449239	627167	3449203
5	628244	3450015	627963	3450017	628346	3449979
6	629395	3450939	629141	3450940	629515	3450908
6A	629574	3451631	629327	3451633	629702	3451600
7A	629995	3453502	629718	3453526	630132	3453479
8	629806	3455187	629518	3455212	629962	3455167

Table 62. Raw Test Run Data - NSDR April

Reference Data		
Point	Easting	Northing
8	629869	3455257
7A	630123	3453552
6A	629709	3451694
6	629496	3450965
5	628319	3450102
4	627151	3449302
3	626091	3448698
2A	625367	3448168
2	623258	3448499
1	622927	3447469

	NSDR0 Data		NSDR1 Data		NSDR2 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
8	629907	3455242	629903	3455239	629913	3455237
7A	630055	3453611	630047	3453609	630058	3453602
6A	629604	3451757	629715	3451733	629614	3451731
6	629391	3451076	629499	3451051	629410	3451045
5	628213	3450151	628318	3450126	628235	3450124
4	627045	3449370	627148	3449341	627079	3449336
3	626047	3448719	626137	3448691	626067	3448691
2A	625297	3448155	625390	3448115	625319	3448114
2	623254	3448549	623314	3448482	623251	3448505
1	622956	3447490	623010	3447423	622940	3447445

Table 63. Raw Test Run Data - WEMM June

GPS Data		
Point	Easting	Northing
A	622985	3447485
3	624656	3446961
5	626373	3446406
6	627725	3447126
7	628502	3447561
8	631819	3448128
9	632707	3447613
10	634634	3446346
12	636411	3445256
13	637959	3444506
14	639108	3443873

	WEMM1 Data		WEMM2 Data		WEMM3 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
A	622978	3447458	622978	3447458	622978	3447458
3	624643	3446931	624642	3446930	624646	3446930
5	626341	3446350	626339	3446349	626348	3446349
6	627681	3447024	627674	3447017	627684	3447025
7	628448	3447495	628444	3447495	628475	3447501
8	631770	3448083	631792	3448092	631788	3448091
9	632666	3447612	632674	3447605	632680	3447606
10	634595	3446367	634603	3446361	634610	3446354
12	636371	3445264	636377	3445251	636386	3445246
13	637900	3444508	637911	3444498	637925	3444496
14	639063	3443849	639069	3443848	639071	3443841
	WEMM4 Data		WEMM5 Data		WEMM6 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
A	622978	3447458	622976	3447458	622986	3447457
3	624645	3446929	624643	3446931	624649	3446928
5	626348	3446349	626340	3446352	626344	3446349
6	627683	3447024	627685	3447026	627685	3447026
7	628453	3447496	628455	3447496	628455	3447496
8	631783	3448089	631791	3448091	631777	3448086
9	632673	3447607	632679	3447602	632678	3447604
10	634603	3446360	634611	3446353	634607	3446357
12	636376	3445246	636384	3445245	636387	3445245
13	637918	3444500	637924	3444493	637924	3444496
14	639074	3443841	639076	3443843	639075	3443841

Table 64. Raw Test Run Data - EWMM June

GPS Data		
Point	Easting	Northing
14	639112	3443881
13	637961	3444512
12	636415	3445262
10	634639	3446351
9	632709	3447619
8	631818	3448134
7	628502	3447569
6	627720	3447132
5	626373	3446412
3	624660	3446969
A	622986	3447494

	EWMM1 Data		EWMM2 Data		EWMM3 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
14	639103	3443848	639073	3443859	639103	3443851
13	637986	3444468	637975	3444473	637982	3444469
12	636438	3445200	636429	3445208	636438	3445209
10	634663	3446301	634660	3446305	634611	3446275
9	632752	3447572	632749	3447575	632679	3447515
8	631850	3448114	631883	3448039	631853	3448035
7	628541	3447850	628521	3447507	628526	3447506
6	627784	3447386	627739	3447067	627723	3447056
5	626561	3446522	626366	3446349	626363	3446349
3	624796	3446875	624655	3446921	624658	3446924
A	623136	3447410	622994	3447453	622992	3447453
	EWMM4 Data		EWMM5 Data		EWMM6 Data	
Point	Easting	Northing	Easting	Northing	Northing	Northing
14	639077	3443852	639079	3443854	639099	3443858
13	637983	3444470	637977	3444472	637978	3444472
12	636434	3445205	636430	3445210	636432	3445209
10	634677	3446294	634654	3446309	634651	3446315
9	632762	3447569	632739	3447580	632735	3447582
8	631848	3448115	631834	3448112	631850	3448115
7	628525	3447505	628538	3447504	628561	3447503
6	627718	3447056	627722	3447053	627741	3447070
5	626360	3446348	626361	3446349	626385	3446350
3	624654	3446925	624653	3446925	624676	3446917
A	622989	3447454	622988	3447455	623012	3447447

Table 65. Raw Test Run Data - SNMM June

GPS Data		
Point	Easting	Northing
C	622945	3447554
2	623254	3448554
2A	625326	3448157
3	626093	3448713
4	627136	3449334
5	628316	3450113
6	629490	3451000
6A	629713	3451729
7A	630134	3453606
8	629897	3455268

	SNMM1 Data		SNMM2 Data		SNMM3 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
C	622953	3447506	622957	3447507	622954	3447528
2	623247	3448497	623246	3448499	623249	3448519
2A	625314	3448101	625314	3448101	625327	3448108
3	626074	3448670	626061	3448663	626077	3448669
4	627148	3449282	627160	3449287	627139	3449276
5	628314	3450047	628315	3450044	628312	3450038
6	629491	3450924	629491	3450923	629488	3450916
6A	MISSING	MISSING	629701	3451649	629700	3451640
7A	630070	3453522	630117	3453533	630070	3453534
8	629955	3455193	629937	3455206	629925	3455215

	SNMM4 Data		SNMM5 Data	
Point	Easting	Northing	Easting	Northing
C	622954	3447530	622964	3447530
2	623250	3448519	623254	3448522
2A	625325	3448107	625328	3448108
3	626074	3448664	626065	3448650
4	627148	3449281	627147	3449281
5	628312	3450029	628312	3450038
6	629489	3450919	629489	3450919
6A	629702	3451643	629698	3451644
7A	630074	3453543	630116	3453531
8	629928	3455212	629934	3455208

Table 66. Raw Test Run Data - NSMM June

GPS Data		
Point	Easting	Northing
8	629893	3455261
7A	630126	3453608
6A	629705	3451730
6	629484	3451003
5	628309	3450116
4	627132	3449341
3	626090	3448720
2A	625323	3448164
2	623246	3448555
C	622938	3447555

	NSMM1 Data		NSMM2 Data		NSMM3 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
8	629884	3455237	629882	3455239	629884	3455239
7A	630127	3453567	630127	3453565	630128	3453570
6A	629618	3451717	629708	3451656	629617	3451705
6	629394	3451004	629494	3450932	629379	3450993
5	628310	3450011	628312	3450028	628308	3450014
4	627126	3449271	627135	3449278	627121	3449269
3	626088	3448676	626077	3448671	626078	3448673
2A	625307	3448113	625306	3448097	625316	3448103
2	623254	3448503	623257	3448513	623259	3448520
C	622949	3447512	622952	3447522	622955	3447533
	NSMM4 Data		NSMM5 Data			
Point	Easting	Northing	Easting	Northing		
8	629882	3455241	629882	3455237		
7A	630126	3453562	630127	3453564		
6A	629607	3451709	629618	3451716		
6	629386	3450994	629402	3450994		
5	628315	3450035	628310	3450008		
4	627139	3449277	627121	3449271		
3	626082	3448674	626090	3448677		
2A	625316	3448102	625315	3448102		
2	623255	3448507	623255	3448506		
C	622956	3447535	622950	3447515		

Table 67. Raw Test Run Data - WEDR June

GPS Data		
Point	Easting	Northing
A	622985	3447485
3	624656	3446961
5	626373	3446406
6	627725	3447126
7	628502	3447561
8	631819	3448128
9	632707	3447613
10	634634	3446346
12	636411	3445256
13	637959	3444506
14	639108	3443873

	WEDR1 Data		WEDR2 Data		WEDR3 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
A	622986	3447457	622988	3447456	622986	3447457
3	624658	3446937	624650	3446926	624636	3446887
5	626369	3446333	626350	3446310	626326	3446256
6	627716	3447052	627670	3447059	627650	3446995
7	628468	3447530	628403	3447559	628393	3447481
8	631791	3448097	631704	3448151	631690	3448081
9	632693	3447630	632599	3447672	632580	3447602
10	634626	3446384	634518	3446418	634501	3446345
12	636400	3445304	636284	3445331	636263	3445254
13	637951	3444567	637827	3444590	637801	3444500
14	639103	3443953	638981	3443978	638941	3443866
	WEDR4 Data		WEDR5 Data		WEDR6 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
A	622986	3447457	622986	3447457	622986	3447457
3	624632	3446876	624633	3446883	624646	3446911
5	626318	3446236	626319	3446236	626351	3446303
6	627638	3446983	627642	3446975	627680	3447046
7	628375	3447478	628384	3447462	628423	3447535
8	631676	3448065	631685	3448047	631732	3448131
9	632565	3447584	632574	3447568	632628	3447653
10	634479	3446323	634491	3446309	634563	3446415
12	636236	3445222	636254	3445219	636334	3445332
13	637774	3444470	637791	3444465	637885	3444600
14	638914	3443836	638938	3443845	639040	3443988

Table 68. Raw Test Run Data - EWDR June

GPS Data		
Point	Easting	Northing
14	639112	3443881
13	637961	3444512
12	636415	3445262
10	634639	3446351
9	632709	3447619
8	631818	3448134
7	628502	3447569
6	627720	3447132
5	626373	3446412
3	624660	3446969
A	622986	3447494

	EWDR1 Data		EWDR2 Data		EWDR3 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
14	639099	3443858	639099	3443858	638943	3443877
13	637936	3444450	637957	3444487	637798	3444496
12	636358	3445119	636407	3445208	636244	3445210
10	634550	3446141	634635	3446281	634462	3446268
9	632595	3447360	632723	3447556	632540	3447527
8	631687	3447851	631834	3448071	631650	3448042
7	628372	3447250	628550	3447454	628359	3447442
6	627626	3446763	627816	3446956	627623	3446945
5	626309	3445999	626523	3446169	626323	3446167
3	624588	3446560	624822	3446759	624618	3446748
A	622902	3447030	623159	3447285	622950	3447258

	EWDR4 Data		EWDR5 Data		EWDR6 Data	
Point	Easting	Northing	Easting	Northing	Northing	Northing
14	639099	3443858	639099	3443858	639099	3443858
13	637949	3444474	637949	3444473	637947	3444473
12	636392	3445182	636384	3445162	636381	3445166
10	634606	3446233	634594	3446205	634587	3446209
9	632676	3447481	632656	3447443	632645	3447445
8	631782	3447990	631762	3447952	631753	3447960
7	628495	3447375	628468	3447357	628461	3447333
6	627759	3446879	627729	3446867	627726	3446833
5	626458	3446108	626425	3446096	626424	3446056
3	624755	3446680	624725	3446692	624717	3446642
A	623086	3447190	623057	3447200	623044	3447146

Table 69. Raw Test Run Data - SNDR June

GPS Data		
Point	Easting	Northing
C	622945	3447554
2	623254	3448554
2A	625326	3448157
3	626093	3448713
4	627136	3449334
5	628316	3450113
6	629490	3451000
6A	629713	3451729
7A	630134	3453606
8	629897	3455268

	SNDR1 Data		SNDR2 Data		SNDR3 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
C	622952	3447520	622952	3447520	622952	3447520
2	623155	3448538	623156	3448537	623146	3448539
2A	625187	3448105	625186	3448107	625178	3448132
3	625932	3448679	625930	3448681	625921	3448708
4	626958	3449328	626952	3449331	626929	3449373
5	628120	3450125	628096	3450143	628070	3450187
6	629242	3451047	629214	3451064	629181	3451121
6A	629417	3451782	629384	3451799	629347	3451857
7A	629774	3453665	629774	3453664	629701	3453739
8	629545	3455345	629546	3455345	629446	3455418
	SNDR4 Data		SNDR5 Data			
Point	Easting	Northing	Easting	Northing		
C	622952	3447520	622952	3447520		
2	623144	3448541	623150	3448537		
2A	625177	3448115	625184	3448129		
3	625921	3448688	625933	3448697		
4	626940	3449339	626965	3449349		
5	628085	3450150	628114	3450158		
6	629200	3451077	629230	3451085		
6A	629375	3451811	629407	3451818		
7A	629740	3453692	629780	3453692		
8	629520	3455377	629546	3455368		

Table 70. Raw Test Run Data - NSDR June

GPS Data		
Point	Easting	Northing
8	629893	3455261
7A	630126	3453608
6A	629705	3451730
6	629484	3451003
5	628309	3450116
4	627132	3449341
3	626090	3448720
2A	625323	3448164
2	623246	3448555
C	622938	3447555

	NSDR1 Data		NSDR2 Data		NSDR3 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
8	629884	3455239	629884	3455239	629884	3455239
7A	630103	3453602	630087	3453599	630108	3453606
6A	629724	3451734	629676	3451751	629787	3451724
6	629536	3451003	629485	3451021	629596	3450993
5	628425	3450066	628367	3450090	628472	3450066
4	627299	3449222	627245	3449246	627351	3449224
3	626331	3448541	626274	3448567	626379	3448549
2A	625612	3447943	625551	3447977	625661	3447951
2	623558	3448325	623498	3448354	623611	3448320
C	623320	3447315	623260	3447344	623371	3447311

	NSDR4 Data		NSDR5 Data	
Point	Easting	Northing	Easting	Northing
8	629884	3455239	629884	3455239
7A	630096	3453601	630108	3453608
6A	629709	3451730	629749	3451729
6	629505	3451003	629549	3451002
5	628372	3450088	628422	3450082
4	627239	3449258	627292	3449252
3	626264	3448581	626320	3448577
2A	625537	3447996	625596	3447988
2	623489	3448380	623545	3448363
C	623247	3447371	623306	3447354

Table 71. Raw Test Run Data - WEDR June (ETL)

GPS Data		
Point	Easting	Northing
A	622985	3447485
3	624656	3446961
5	626373	3446406
6	627725	3447126
7	628502	3447561
8	631819	3448128
9	632707	3447613
10	634634	3446346
12	636411	3445256
13	637959	3444506
14	639108	3443873

	WEDR10 Data		WEDR11 Data		WEDR12 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
A	622986	3447485	622986	3447485	622986	3447485
3	624652	3446959	624652	3446959	624653	3446965
5	626361	3446375	626360	3446374	626361	3446374
6	627659	3447169	627664	3447155	627674	3447139
7	628384	3447683	628391	3447663	628401	3447648
8	631673	3448356	631679	3448345	631693	3448314
9	632570	3447887	632577	3447877	632593	3447855
10	634508	3446658	634521	3446657	634536	3446638
12	636286	3445592	636308	3445605	636322	3445584
13	637846	3444880	637871	3444903	637884	3444883
14	639001	3444278	639018	3444285	639035	3444273
	WEDR13 Data		WEDR14 Data			
Point	Easting	Northing	Easting	Northing		
A	622986	3447485	622986	3447485		
3	624653	3446968	624644	3446940		
5	626360	3446384	626342	3446330		
6	627668	3447160	627670	3447077		
7	628396	3447668	628417	3447558		
8	631686	3448342	631725	3448146		
9	632583	3447871	632618	3447671		
10	634520	3446644	634547	3446430		
12	636303	3445589	636319	3445356		
13	637866	3444889	637872	3444633		
14	639019	3444293	639025	3444023		

Table 72. Raw Test Run Data - EWDR June (ETL)

GPS Data		
Point	Easting	Northing
14	639112	3443881
13	637961	3444512
12	636415	3445262
10	634639	3446351
9	632709	3447619
8	631818	3448134
7	628502	3447569
6	627720	3447132
5	626373	3446412
3	624660	3446969
A	622986	3447494

	EWDR10 Data		EWDR11 Data		EWDR12 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
14	639113	3443882	639113	3443882	639113	3443882
13	637954	3444487	637952	3444487	637949	3444481
12	636386	3445179	636387	3445179	636377	3445157
10	634588	3446217	634588	3446214	634571	3446176
9	632639	3447446	632636	3447435	632624	3447402
8	631739	3447950	631736	3447934	631726	3447906
7	628428	3447397	628432	3447361	628425	3447317
6	627673	3446925	627680	3446887	627678	3446835
5	626349	3446180	626360	3446138	626361	3446085
3	624639	3446759	624650	3446716	624649	3446655
A	622966	3447261	622975	3447215	622975	3447151

	EWDR13 Data		EWDR14 Data	
Point	Easting	Northing	Easting	Northing
14	639113	3443882	639113	3443882
13	637951	3444476	637958	3444489
12	636380	3445157	636386	3445166
10	634575	3446177	634582	3446186
9	632616	3447384	632632	3447407
8	631712	3447873	631730	3447895
7	628411	3447289	628439	3447271
6	627666	3446801	627695	3446788
5	626355	3446046	626386	3446021
3	624645	3446621	624675	3446579
A	622971	3447117	622999	3447065

Table 73. Raw Test Run Data - SNDR June (ETL)

GPS Data		
Point	Easting	Northing
C	622945	3447554
2	623254	3448554
2A	625326	3448157
3	626093	3448713
4	627136	3449334
5	628316	3450113
6	629490	3451000
6A	629713	3451729
7A	630134	3453606
8	629897	3455268

	SNDR10 Data		SNDR11 Data		SNDR12 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
C	622946	3447554	622946	3447554	622946	3447554
2	623169	3448565	623158	3448566	623143	3448571
2A	625204	3448170	625187	3448166	625169	3448159
3	625956	3448737	625943	3448727	625922	3448723
4	627007	3449373	626997	3449359	626965	3449358
5	628173	3450162	628158	3450144	628119	3450159
6	629314	3451066	629290	3451051	629238	3451072
6A	629497	3451797	629485	3451779	629407	3451808
7A	629898	3453662	629910	3453641	629763	3453689
8	629692	3455344	629705	3455320	629521	3455366
	SNDR13 Data		SNDR14 Data			
Point	Easting	Northing	Easting	Northing		
C	622946	3447554	622946	3447554		
2	623160	3448567	623153	3448568		
2A	625184	3448150	625179	3448147		
3	625936	3448715	625936	3448706		
4	626979	3449352	626989	3449335		
5	628124	3450152	628142	3450124		
6	629252	3451059	629270	3451029		
6A	629436	3451790	629454	3451760		
7A	629840	3453650	629851	3453626		
8	629621	3455333	629631	3455309		

Table 74. Raw Test Run Data - NSDR June (ETL)

GPS Data		
Point	Easting	Northing
8	629893	3455261
7A	630126	3453608
6A	629705	3451730
6	629484	3451003
5	628309	3450116
4	627132	3449341
3	626090	3448720
2A	625323	3448164
2	623246	3448555
C	622938	3447555

	NSDR10 Data		NSDR11 Data		NSDR12 Data	
Point	Easting	Northing	Easting	Northing	Easting	Northing
8	629892	3455261	629892	3455261	629892	3455261
7A	630085	3453632	630106	3453634	630071	3453626
6A	629688	3451770	629749	3451758	629662	3451759
6	629486	3451044	629549	3451031	629449	3451034
5	628354	3450127	628420	3450116	628310	3450119
4	627218	3449309	627287	3449291	627173	3449305
3	626248	3448637	626320	3448611	626200	3448634
2A	625529	3448033	625603	3448007	625473	3448037
2	623481	3448406	623558	3448379	623419	3448397
C	623218	3447402	623298	3447375	623154	3447395
	NSDR13 Data		NSDR14 Data			
Point	Easting	Northing	Easting	Northing		
8	629892	3455261	629892	3455261		
7A	630064	3453628	630066	3453629		
6A	629671	3451754	629653	3451762		
6	629460	3451030	629443	3451038		
5	628318	3450120	628296	3450131		
4	627169	3449314	627146	3449328		
3	626196	3448643	626166	3448657		
2A	625464	3448055	625435	3448066		
2	623413	3448420	623389	3448445		
C	623138	3447418	623113	3447445		