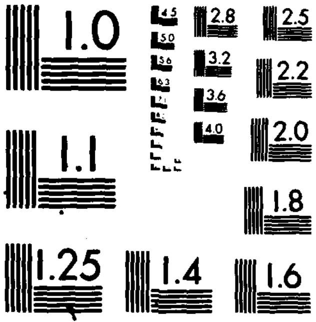
AD-A098 903 NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL S--ETC F/G 9/2
REDUCING LASER PROFILES ON A TABLETOP COMPUTER.(U)
FEB 81 A W LOHANICK
UNCLASSIFIED NORDA-TN-77 NL

1 of 1
AD A098903

END
DATE
FILMED
6-81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NORDA Technical Note 77

See 1473

**Naval Ocean Research
and Development Activity**
NSTL Station, MS 39529

Reducing Laser Profiles on a Tabletop Computer

Alan W. Lohanick

Oceanography Division
Ocean Science and Technology Laboratory

FEBRUARY 1981

ABSTRACT

Standard numerical techniques for removing aircraft motion and discontinuities from airborne laser profiles are adapted to a desk-top computer. Because such a computer is much slower than a larger machine, analog active filters replace the numerical (Hamming) filters normally used. The computer programs are in BASIC, and listings are provided in an Appendix. Several examples of data editing procedures are given.

BACKGROUND

Airborne laser profiling remains by far the most rapid method of obtaining some measure of the "roughness" of a field of ice. This paper will deliberately not deal with the applications of laser profiles, but will describe a specific method of obtaining "clean" profiles from raw data. The reader is referred to Ketchum (1971) for a general description of the profiling system, a Spectra-Physics Geodolite 3A.

Ketchum (1971) introduced airborne laser profiles to the Arctic remote sensing community, and Hibler (1972) devised a computer-based scheme for removing unwanted "aircraft motion" from the raw data. This scheme has been used and modified in subsequent applications (Holyer et al., 1977; Lohanick, 1979), but not greatly changed.

This paper will describe a specific data reduction system written at NORDA, centered around Hewlett-Packard hardware and firmware. It is based on the Hibler three-step filter process, but is accomplished on a mini-computer dedicated to this task, and with some steps which greatly reduce overall data reduction time.

EVALUATION OF RAW PROFILE DATA

Profile data are recorded in the field on FM analog tapes. A time code channel is provided for comparison of the data with annotations on various aircraft logs, and for correlation with other airborne sensors which also record time. These analog tapes must first be converted to digital form (digitized) for handling in the computer. The initially digitized data from the analog field tapes will hereafter be referred to as the "raw" data.

For those unfamiliar with the "look" of raw laser profile data, Figure 1 is provided and annotated to point out relevant features. The vertical exaggeration in this plot is about 300:1.

SAMPLING RATE

The sampling rate (number of points digitized per second) for this type of data has historically given an actual spacing of about 1 m (see Hibler, 1972; Holyer et al., 1977) on the terrain. Hibler's (1972) scheme leaves intact all data that have a wavelength less than about 127 m, although the power spectrum of the topside of sea ice has never specifically been discussed in the short wavelength region (see Hibler and LeSchack, 1972).

AIRCRAFT MOTION

Long-wavelength undulations in the data are due to the motion of the aircraft carrying the laser. As the aircraft slowly rises and falls in altitude, and makes turns, the apparent altitude measured by the profiler varies. This result of aircraft instabilities must be removed if the actual terrain profile is to be recovered.

DISCONTINUITIES

Discontinuities in the apparent profile, such as at points A and B in Figure 1, can be caused either deliberately or accidentally by the profiling system. Causes for these jumps in the apparent profile are discussed in Holyer et al. (1977). Figure 2 shows the results of not removing these discontinuities. Nonexistent terrain features appear in final profile, because the computer recognizes only long-wavelength variations as aircraft motion.

Hibler's 1972 scheme for aircraft motion removal is adequate, and lends itself to numerical or analog processes, as will be discussed below. Removal of discontinuities is essential to the data reduction. But, as pointed out by Holyer et al. (1977), no satisfactory method has yet been devised for reliably eliminating these discontinuities automatically (i.e., without the intervention of a trained analyst). Therefore, some sort of interactive system must be devised to allow a trained operator to choose sections of data visually, as well as to remove undesirable discontinuities. The alternative to interactive graphics is a laborious process involving hand annotation of computer-generated plots and print listings (Lohanick, 1979).

Figure 3 is included to show a data record for which reliable automatic algorithms for locating discontinuities would be difficult to construct. The data were recorded with four times the vertical sensitivity of those in Figure 1. The circled area is referred to in a later section.

RELEVANT CAPABILITIES OF THE COMPUTER HARDWARE

All elements of the hardware system used in the present application are sketched in Figure 4. The computer is a Hewlett-Packard (HP) 9845B with 186 kilobytes of internal memory, a CRT for printing, data display and graphics, an internal thermal printer and two data cartridge (tape) drives. The 9845B has four interface busses for peripherals.

CRT graphics is done on a 560 x 455 dot matrix. Any plot generated on the CRT can be printed on the internal printer with one command. It is also possible to store any plot on a mass-storage device such as a flexible disk or tape cartridge.

The HP2240A Measurement and Control Processor contains both an A/D converter and a D/A converter, whose uses will be described later. Both converters have a resolution of 5 mV in a range of ±10,000 mV (approximate range). The 2240A is directly compatible with the 9845B. The data transfer rate for A/D (12-bit) conversion is about 100 Hz. The HP9885 Flexible Disk drives use seven-inch disks (floppies) with a storage capacity of about 500 K bytes.

Two double Rockland active filters (Models 432 and 452-01) are used between the 2240A D/A output and A/D input to accomplish aircraft motion removal. They replace previously used numerical filters, which are far too slow in this application. A 30000-data point record, which would require 60 hours of time on this computer with the numerical filters, requires about 1.5 hours with active filters.

THE DATA REDUCTION PROCESS

DIGITIZING (PROGRAM "A/D")

Because of hardware and firmware constraints, a maximum of 90,000 regularly-spaced integer data points (16 bits) can be taken from an analog field tape at one time. Since the maximum sampling rate of the system is 100 12-bit samples/second, this amounts to 15 minutes of data, or about 50 nautical miles (at 200 knots aircraft speed).

A data tape is chosen and placed on the Ampex FR-1300 Tape Recorder. A suitable section of the tape is located by reading the time-code channel and comparing with aircraft logs.

The computer program used is named "A/D" and its action is diagrammed in Figure 5 (all programs are in BASIC, and complete listings are provided in Appendix A, with internal documentation). The program enables the HP2240A to read the analog tape, digitize the analog voltages, and make them available for reading by the HP9845B computer. For maximum data transfer rate (100 Hz) the 2240A constantly fills its output buffers with data, while the 9845B asynchronously empties them into internal memory. In this way, time-consuming synchronization is avoided.

Once the 90000 (or fewer) data points are read, digitizing ceases, and the data is immediately stored on a floppy. Because firmware restricts any file to 32767 elements ($2^{15}-1$), a file length of 30000 is chosen. Thus, a full run will fill three files, each containing five minutes of data.

During program execution, the operator presses one key to begin or end the digitizing, and names the files which will store the raw profile data.

REFORMATTING AND PLOTTING (PROGRAM "PLOT")

Some reformatting of data is done to take advantage of special dense data file types which allow storage of eight full files (40 minutes of data) on each floppy. The data in one file are then plotted in small scale on the CRT and stored on disk. A hard copy of the CRT plot is obtained on the thermal printer (see Fig. 6), and is used for some annotation and time-scaling. The stored plot is used for editing the data as described below.

The small scale plot is the visual counterpart of a raw data file, and both are saved permanently.

Program "Plot" requires the operator to name the file which will store the plot.

EDITING (PROGRAM "EDIT")

Program "Edit", takes about 85% of total operator time, since it requires him to make judgments as to the quality and character of the data which has been digitized and plotted. He must also choose one of several editing procedures to remove undesirable discontinuities. At this point, the operator must be a trained analyst who has seen and been guided through, much raw profile data and its peculiarities. The operator's expertise will, of course, have a direct impact on the quality of the final product. Some examples given below will help to point this up.

To begin, the operator names the raw data file and its plot (file). The computer reads these into memory and graphics memory, respectively, and presents the small-scale plot (Fig. 6, again) on the CRT. The operator uses a small cursor (operated from the keyboard) to move to features he wishes to change or to observe more closely. He presses one key, and the CRT provides a large-scale plot of 60 data points in the immediate vicinity of the current point of interest (Fig. 7). This large-scale plot will be called an edit window. The vertical exaggeration in the edit window is about 2:1.

Five different editing procedures are available. Two of these require the operator to mark the beginning and end of the data record (BOR, EOR), which are not defaulted. The three editing procedures which affect the data are:

1. Straight-line tie (see Fig. 8a). This procedure allows the operator to remove a sharp discontinuity and replace it with a horizontal straight line. It is

appropriate when the discontinuity is a "180° phase shift" (an end-of-scale jump back to center scale, provided by the laser hardware to avoid having to record voltages outside a selected range, usually ±1.5 volts), or a spike which may be caused by spurious effects during recording or digitizing.

The operator places the cursor to the immediate left of the discontinuity, presses a key to enter its position, moves the cursor to the present position of the end of the feature, presses the key, moves the cursor to the desired position of this second point (only up and down motion is permitted), and enters it. The CRT shows the result for approval or disapproval (Fig. 8b).

2. Straight-line slope (see Fig. 9a). The operator marks all the points as above. The computer leaves the character of the data in the interval between points 1 and 2 intact, but changes the slope (Fig. 9b).

3. Point-by-point fix (see Fig. 10). Allows the operator to move each point a desired distance up or down. This procedure is appropriate when there are several discontinuities in the edit window, and the data are considered reliable by the operator. Several steps in this process are shown in Figures 10b through 10d.

The product of this program is a tape file of all edit windows for a particular record of raw data (which may be a complete raw data file or portion of it, delineated by the BOR and EOR mentioned above). This program is the only step requiring great skill on the part of the operator.

REMOVING THE DISCONTINUITIES (PROGRAM "DISCON")

The operator provides the raw data file disk and the correct edit window file, and the computer automatically removes all discontinuities as prescribed in the editing procedure above. The operator provides a name for the updated file, which is stored again on a floppy. The original file is not destroyed. Edit window files are generally not saved, since they are intermediate products and are not needed beyond this stage.

AIRCRAFT MOTION REMOVAL (PROGRAM "8-IN-1")

The operator provides the program with the names of the source file(s) (the one(s) created in Program Discon above) and the result file(s). The program is written to handle up to eight pairs of files at a time.

Aircraft motion removal has previously been done with numerical Hamming filters (Hibler, 1972; Holyer et al., 1977; Ketchum, 1971; and Lohanick, 1979). The approximate number of multiplications to be done by one pass of a Hamming filter is the number of points in the record times the number of filter weights. For the low pass filter providing the aircraft motion envelope (see below), the number of filter weights for a sampling rate of 100 points/second is about 250, leading to over six million calculations for a 30000-point record. On the 9845B this would take many hours. In fact, for all three filters, estimated total run time for a five-minute record is about 60 hours, so we must use external active filters.

Data flow through a particular filter is diagrammed in Figure 11. Data are first read into an array in computer memory. Then one point at a time is read into the D/A converter and fed to the appropriate active filter. The output of the filter (connected to the A/D converter) is "simultaneously" read back into computer memory. Data are filtered in pseudo-real-time, since one complete cycle of D/A and A/D conversion is about six times as long as the actual data sampling interval.

Thus, a 30000-point record (5 minutes of data) takes about 30 minutes to filter. Filter cutoff frequencies must be set to about one-sixth of the values calculated for the numerical filters. The entire aircraft motion removal scheme is explained in Hibler (1972).

Since eight files (40 minutes of data) can be handled by the program at one time, a complete run takes about 12 hours, and is run unattended overnight. Operator time is a total of about 15 minutes in setting up. Result files, which are the final terrain profiles, are stored on disk.

One product of Program "8-in-1" is shown in Figure 12. This scaled plot (horizontal dotted lines are at 2 m intervals above the zero line) has a great amount of vertical exaggeration (about 1000:1). The operator uses this plot to look for any unusual features, such as ridges over 10 m high or large spikes below the zero line. Causes for these features must be found. If the cause is faulty editing, then editing must be repeated, but only for this particular error.

The profile is now in its final form on disc, and is ready to be used in any application of sea-ice terrain features such as roughness, ridge counts, ridge height distribution, or power spectral density.

A summary of all steps taken in the data reduction is shown in Figure 13, for a segment about 2 km long.

REFERENCES

Hibler, W. D. III (1972). Removal of Aircraft Altitude Variations from Laser Profiles of the Arctic Ice Pack, J. Geophys. Res., 77(30) p. 7190-7195.

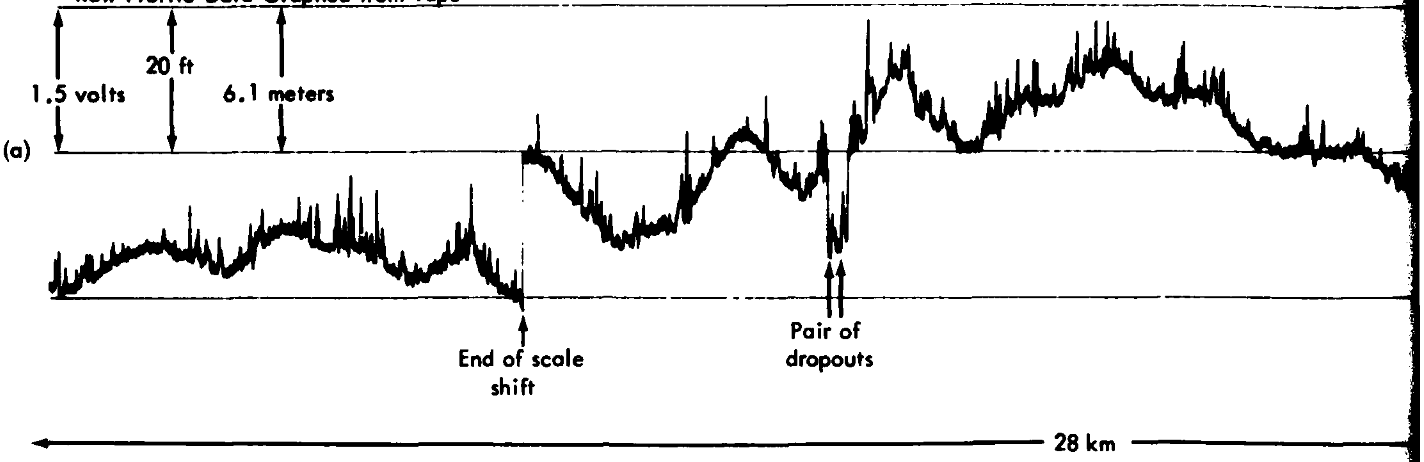
Hibler, W. D. III and L. A. LeSchack (1972). Power Spectrum Analysis of Undersea and Surface Sea Ice Profiles. J. Glaciol. 11, 63, p. 345-356.

Holyer, I. J. J., P. Wadhams and R. T. Lowry (1977). An Interactive Graphics System for the Reduction of Airborne Laser Profiles of Sea Ice. Scott Polar Research Institute Tech. Rept. 77-1, Cambridge, England.

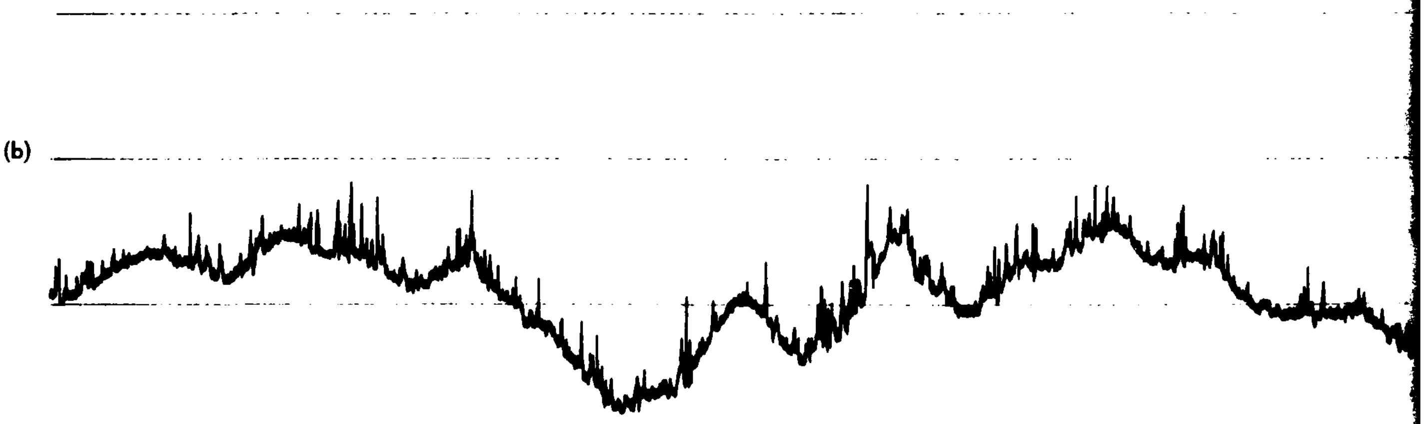
Ketchum, R. D., Jr. (1971). Airborne Laser Profiles of the Arctic Pack Ice, Remote Sensing of Environment 2, p. 41-52.

Lohanick, A. W. (1979). Airborne Laser Sea Ice Profiles Near a Drifting Camp, April 1977, NORDA Technical Note 49.

Raw Profile Data Graphed from Tape



Discontinuities removed



Final profile with aircraft motion removed

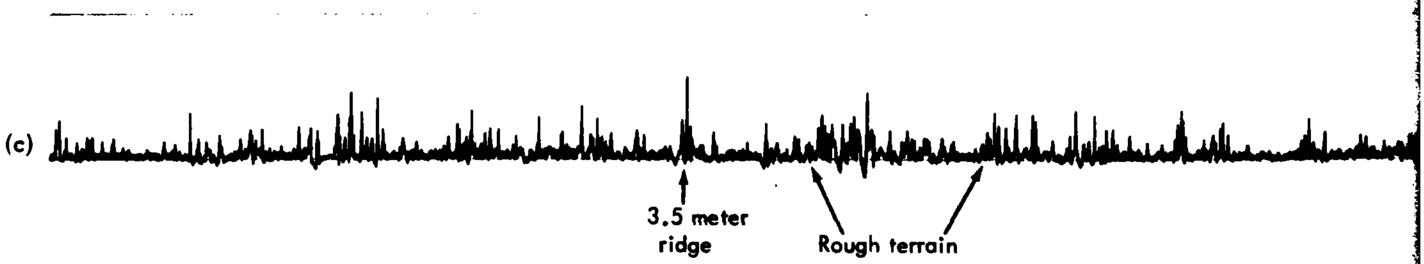


Figure 1. Laser sea-ice profile showing stages of discontinuity
motion removal. Vertical scale exaggeration about

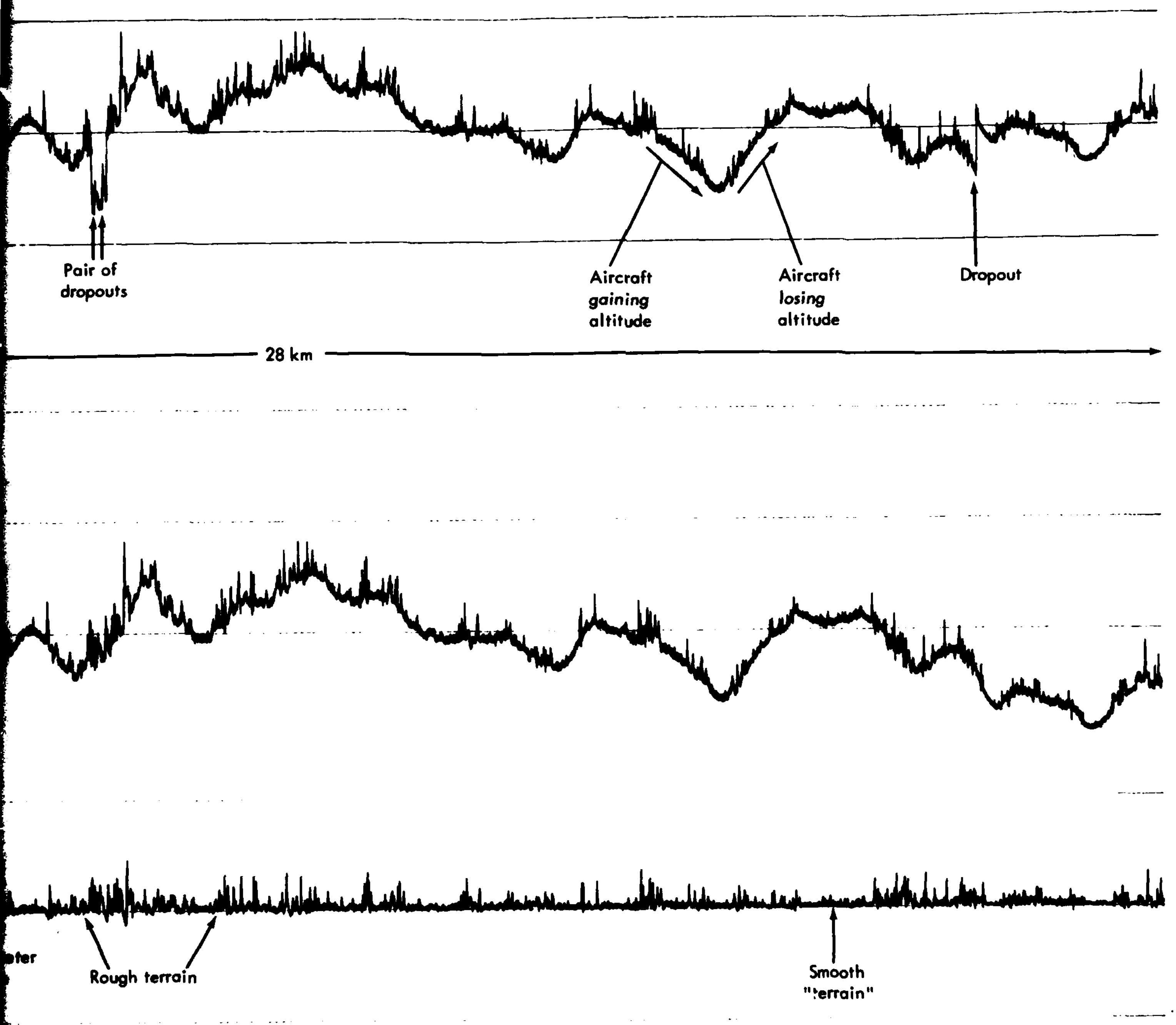
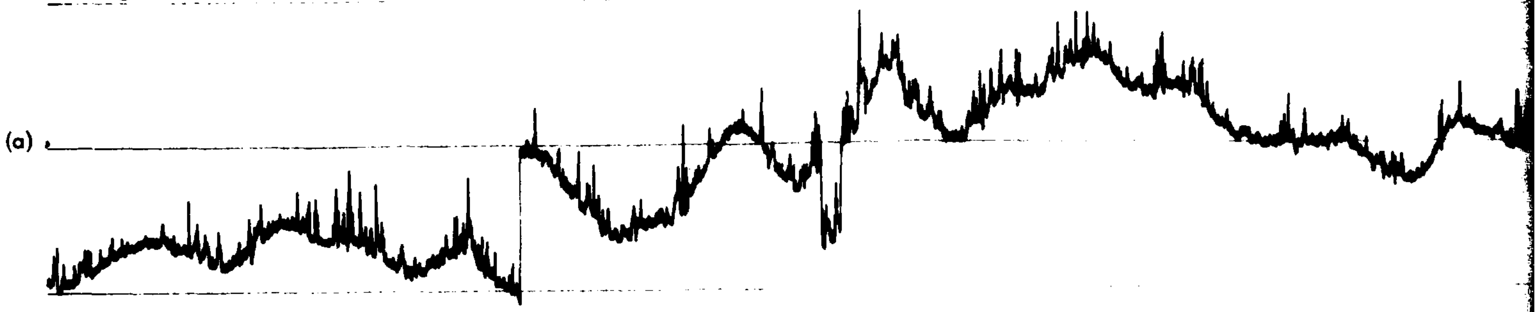


Figure 1. Laser sea-ice profile showing stages of discontinuity and aircraft motion removal. Vertical scale exaggeration about 300:1.

Original Profile



Final Profile with Discontinuities not Removed

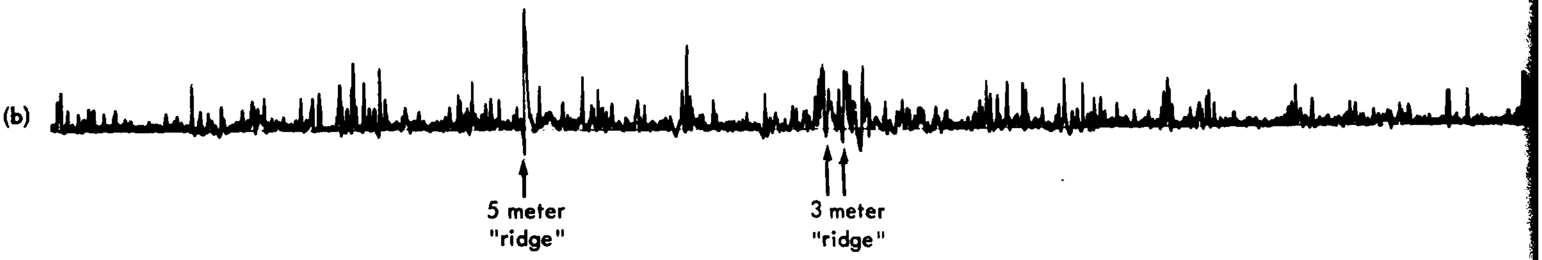


Figure 2. Laser sea-ice profile showing effect of not removing discontinuiti
Vertical exaggeration 300:1. Compare with Figure 1(c).

Laser sea-ice profile showing effect of not removing discontinuities.
Vertical exaggeration 300:1. Compare with Figure 1(c).

2

Figure 3. Laser sea-ice profile with many discontinuities. Circled area is one of several in which two or three discontinuities occur in rapid succession.

3. Laser sea-ice profile with many discontinuities. Circled area is one of several in which two or three discontinuities occur in rapid succession.

2

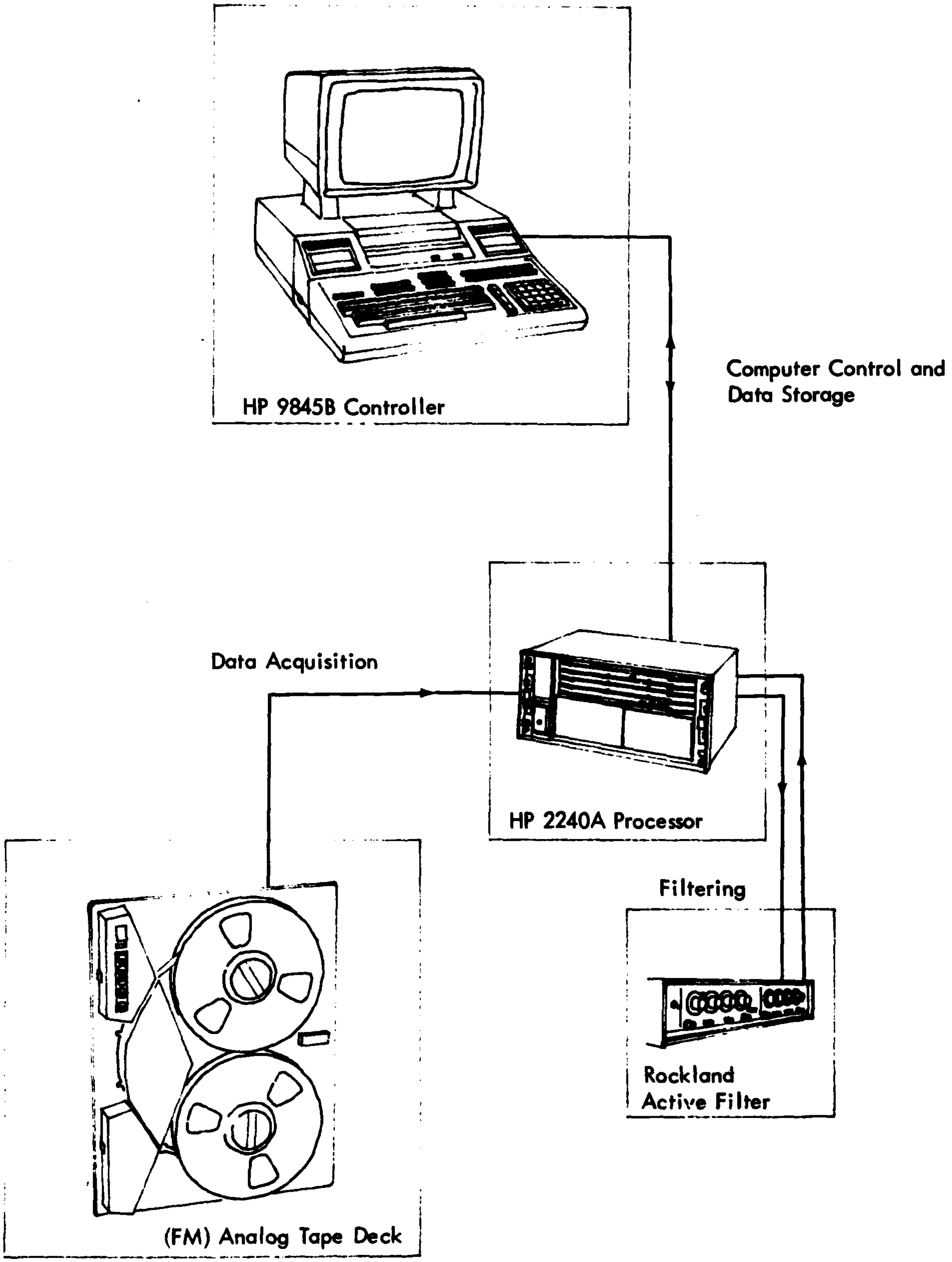


Figure 4. Physical diagram of computer used in laser profile reduction (disc storage not shown)

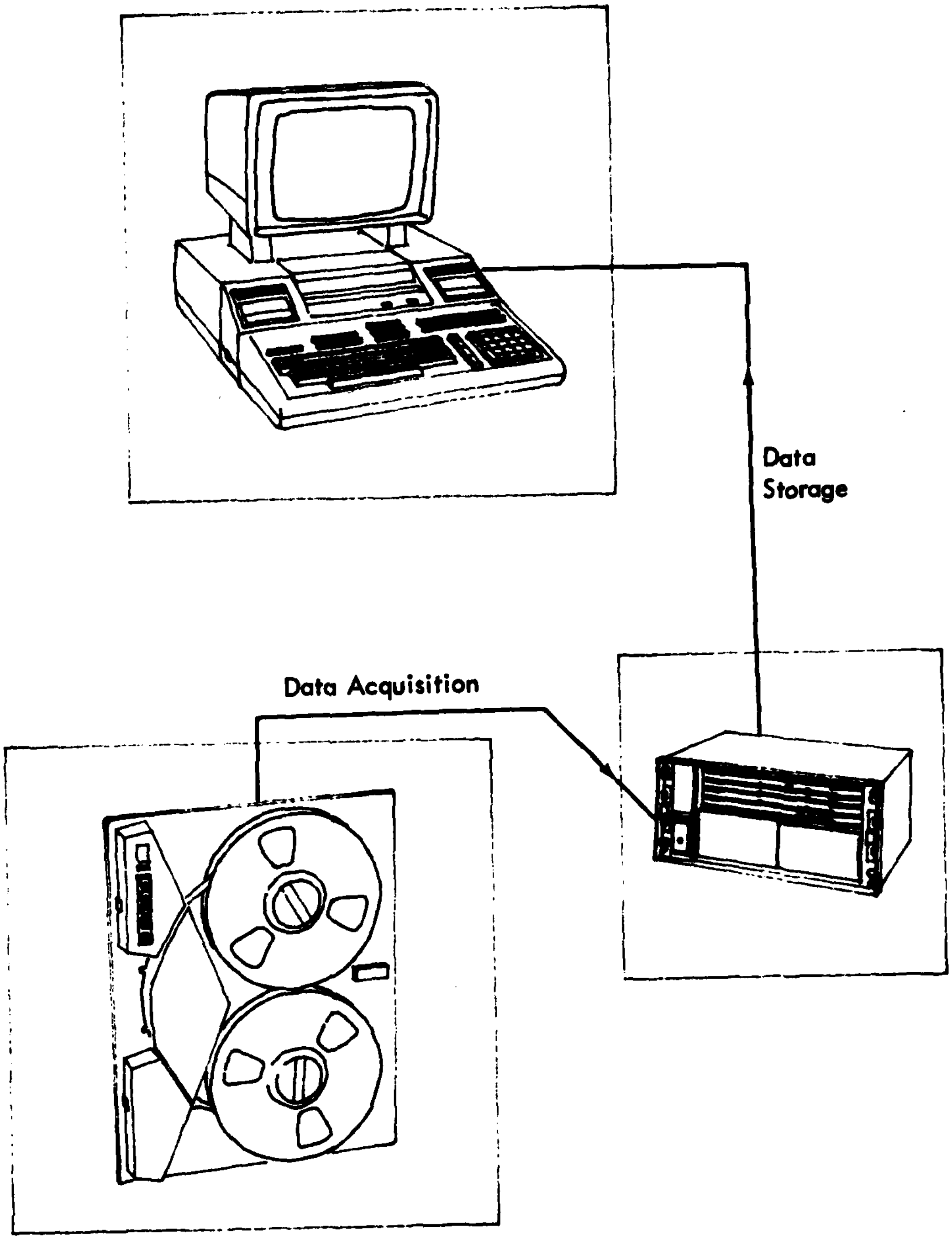


Figure 5. Data acquisition

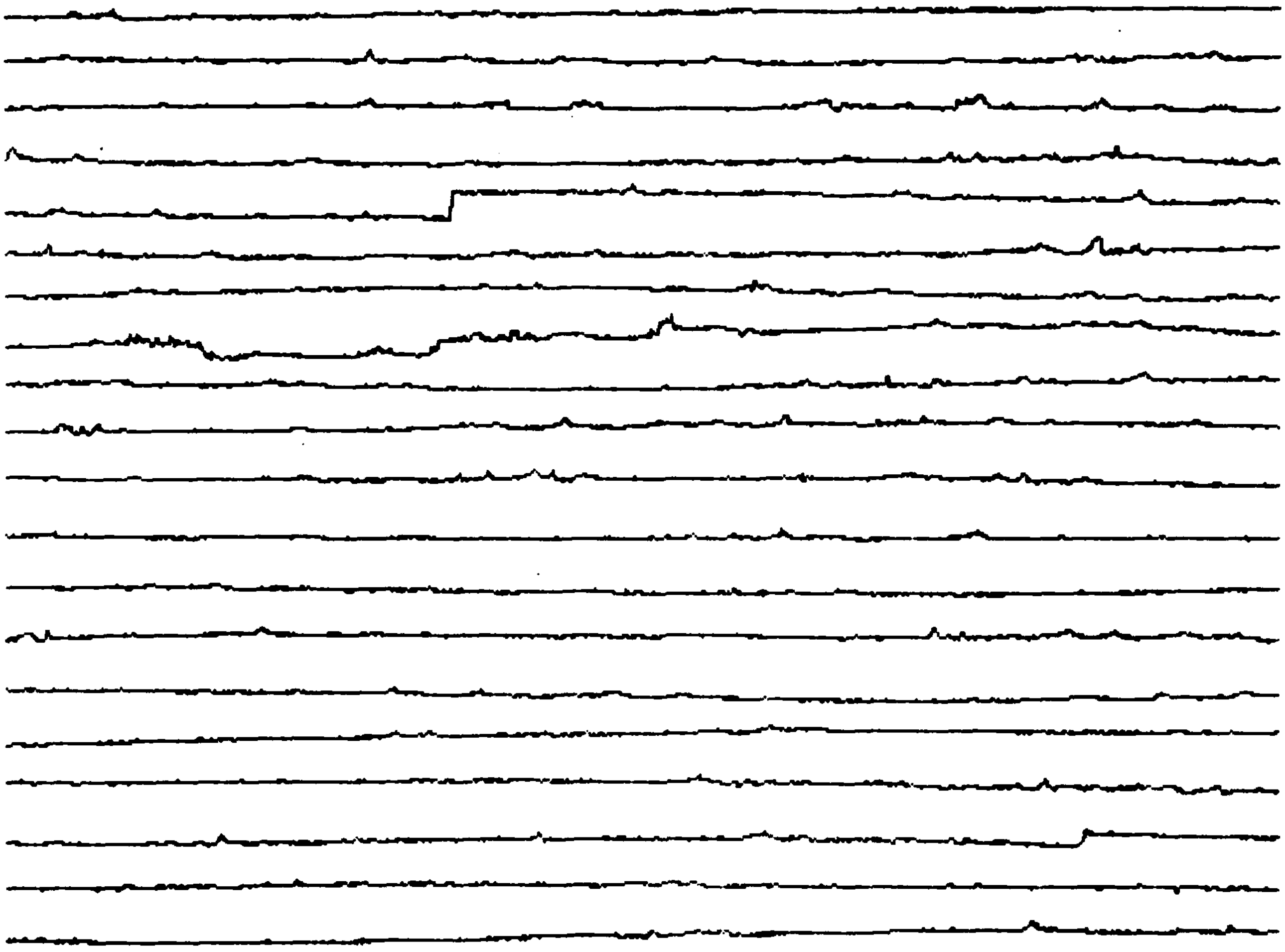


Figure 6. Small-scale CRT plot of 5-minute laser profile record. Shown actual size.

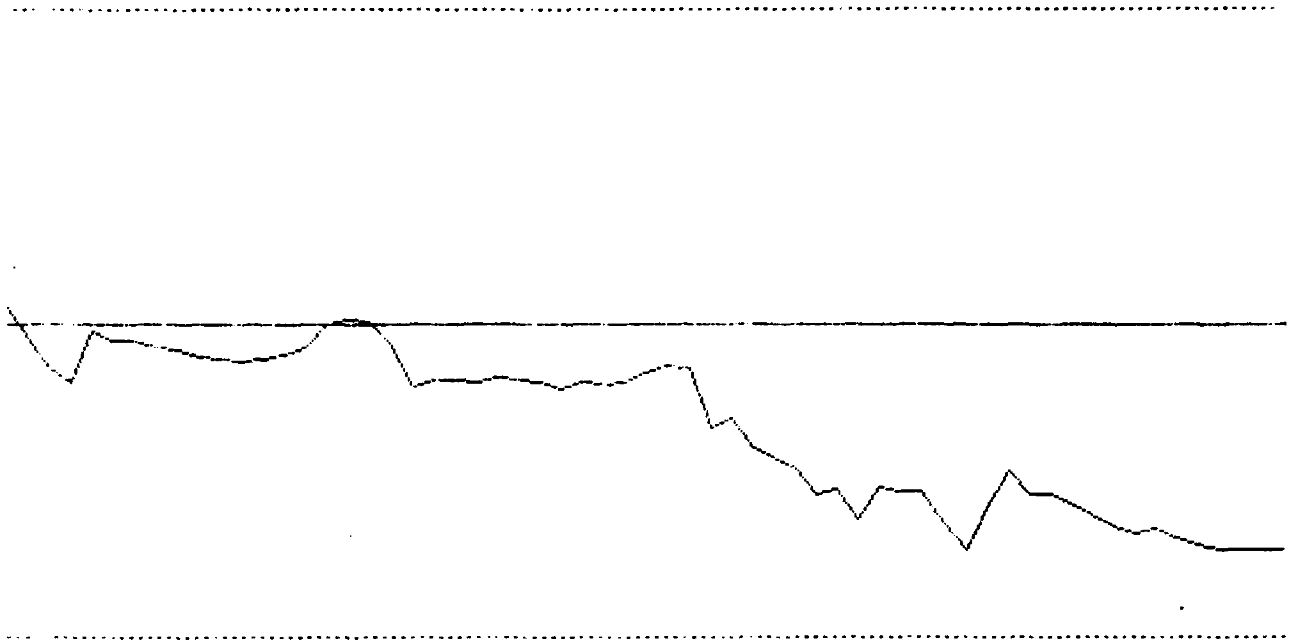
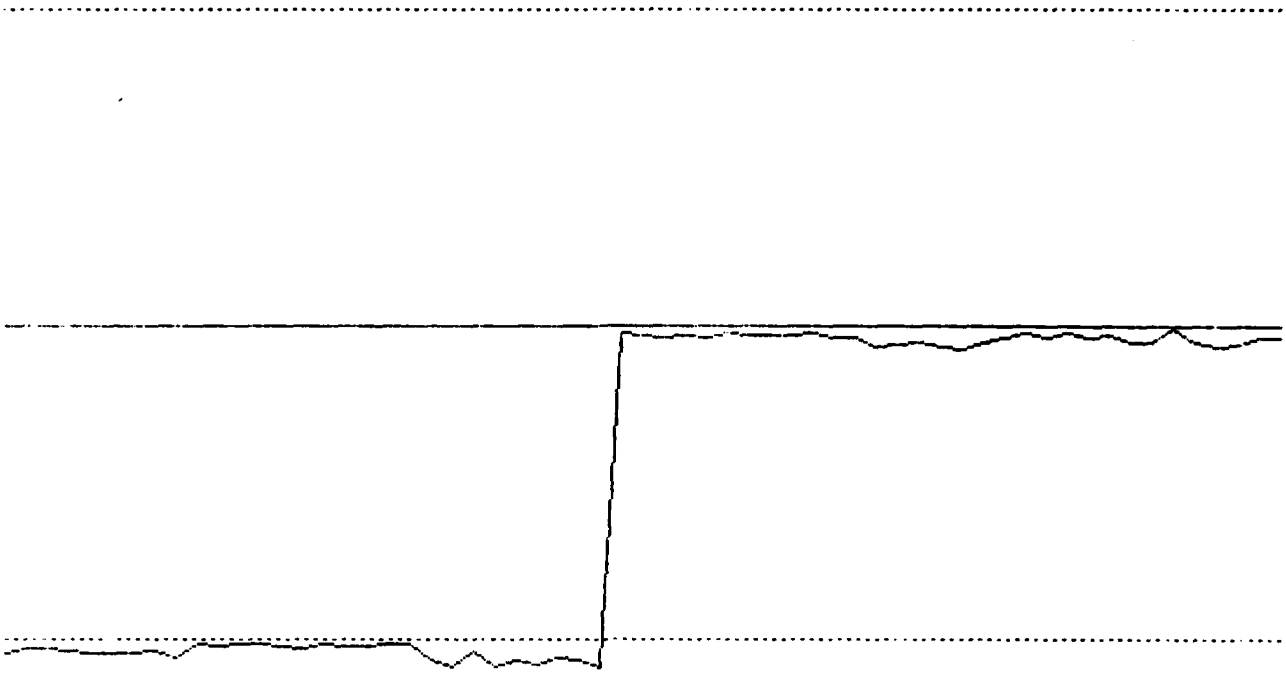
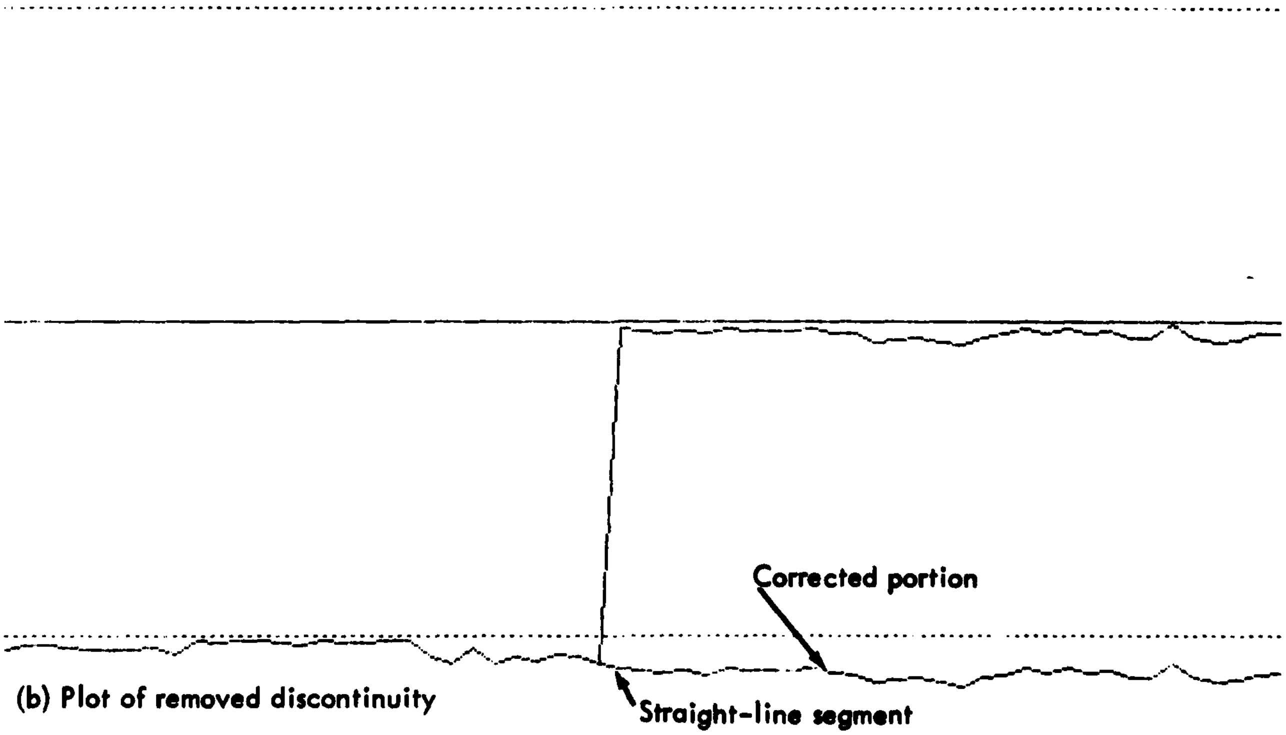


Figure 7. Large-scale plot of 60 points in the vicinity of a profile discontinuity. Shown actual size. Vertical exaggeration 2.25:1.

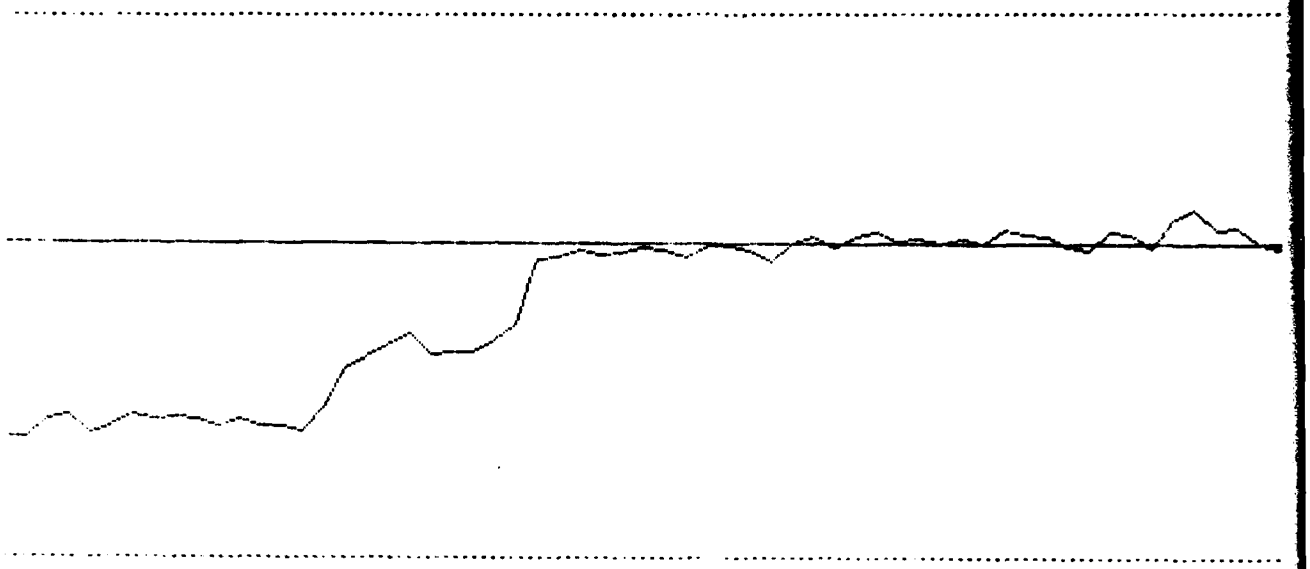


(a) Plot of discontinuity as presented on CRT

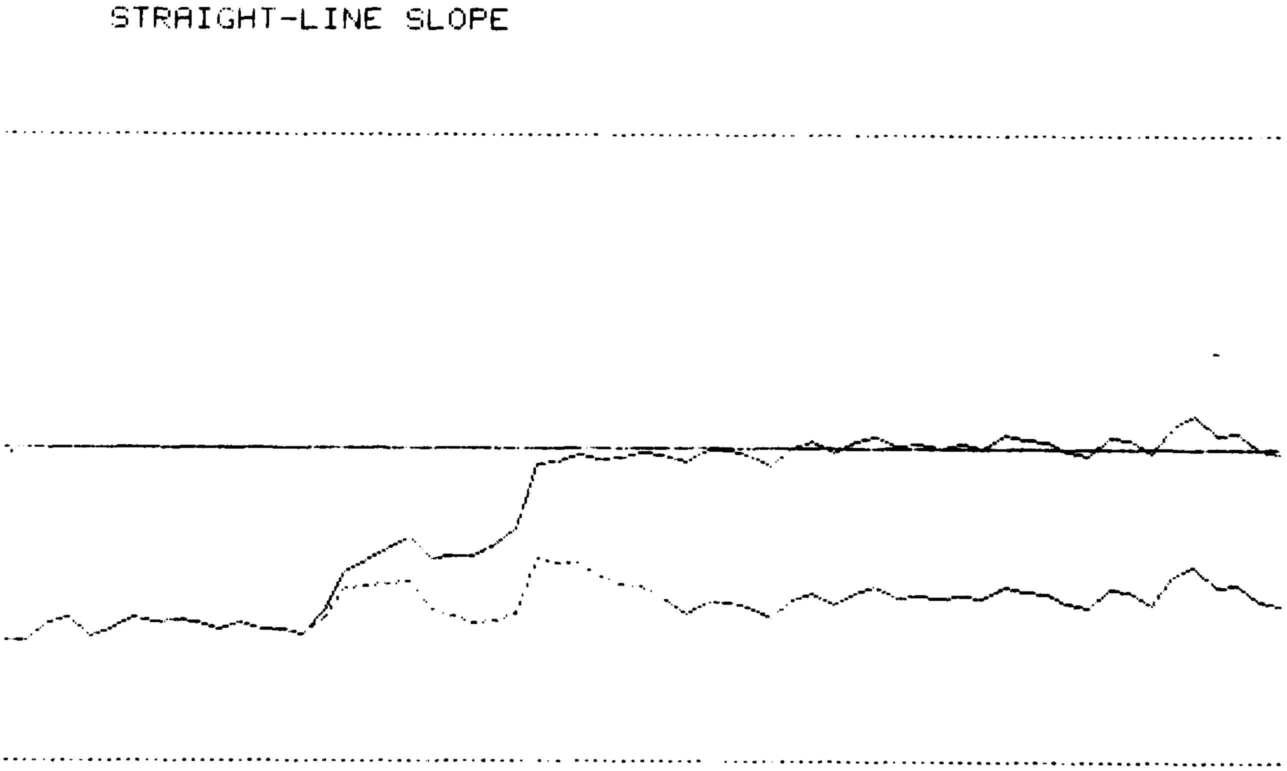


(b) Plot of removed discontinuity

Figure 8. Straight-line editing of a profile discontinuity on the CRT. Shown actual size. Vertical exaggeration of profile 2.25:1.

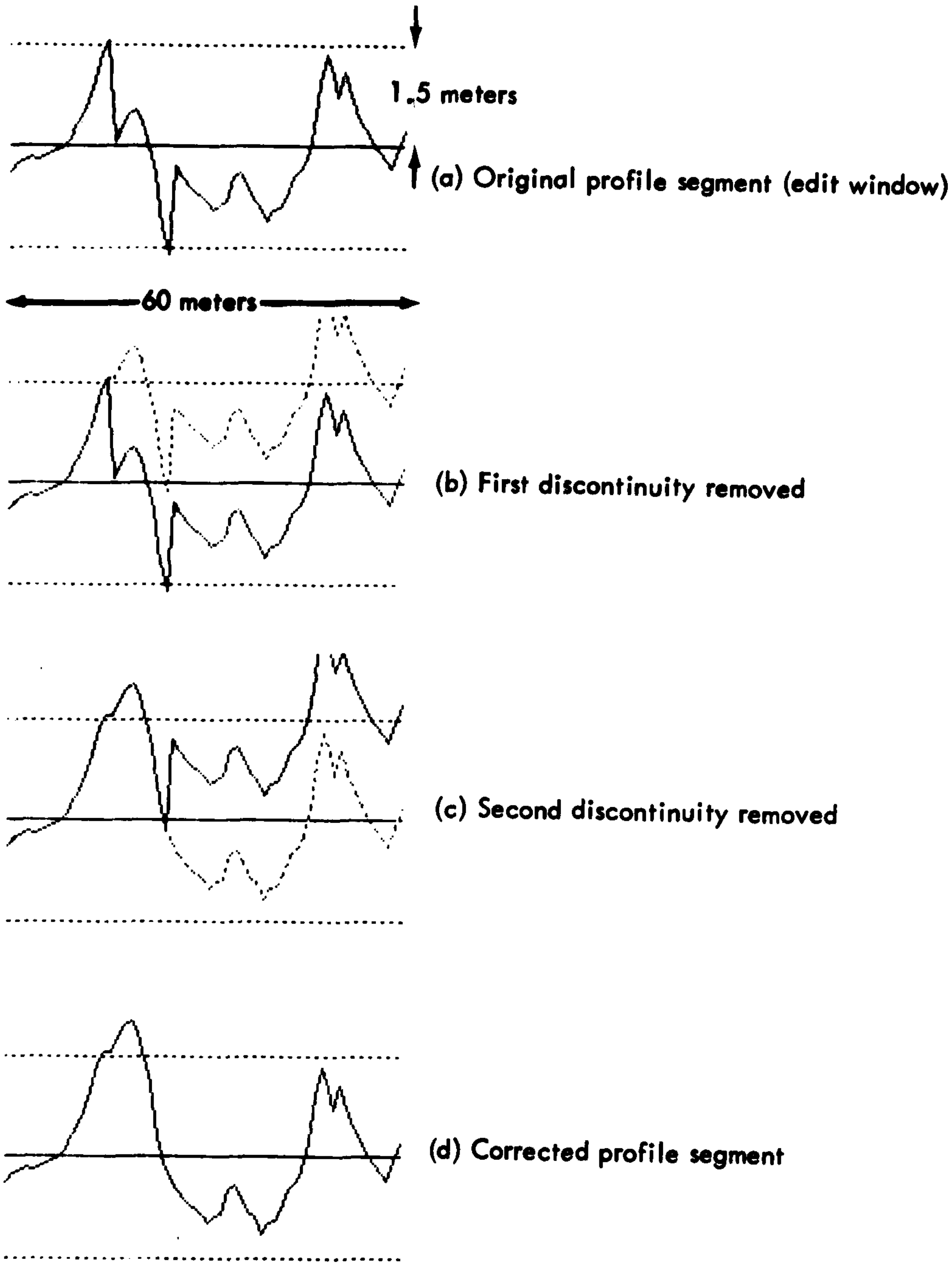


(a) Plot of discontinuity as presented on CRT



(b) Plot of removed discontinuity

Figure 9. Slope removal of a profile discontinuity on the CRT shown actual size. Vertical exaggeration of profile 2.25:1.



(a) Original profile segment (edit window)

(b) First discontinuity removed

(c) Second discontinuity removed

(d) Corrected profile segment

Figure 10. Point-by-point removal of discontinuities from sea-ice profile. Each frame actually fills CRT. Vertical scale exaggeration 10:1.

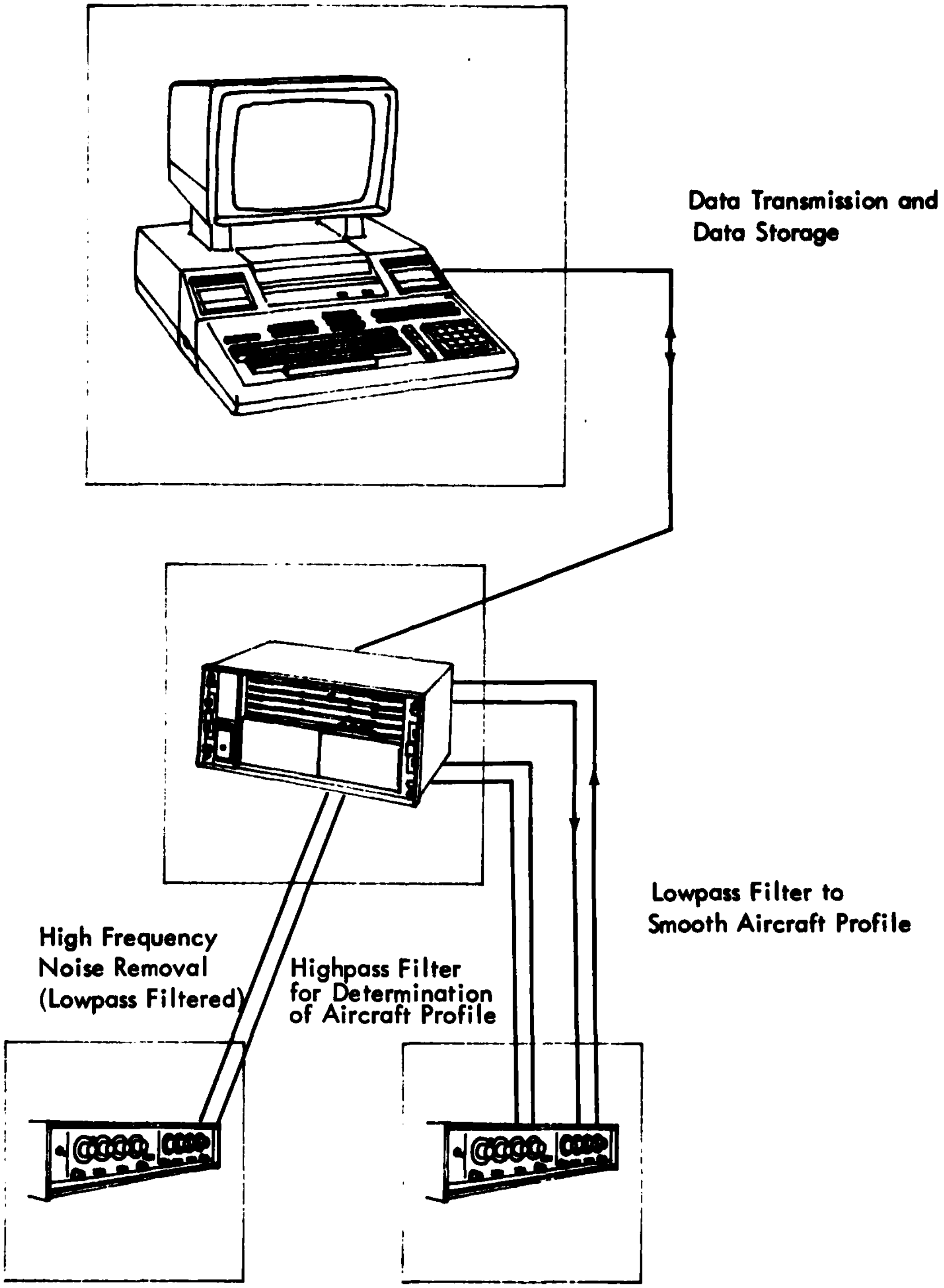


Figure 11. Actively filtering laser profiles

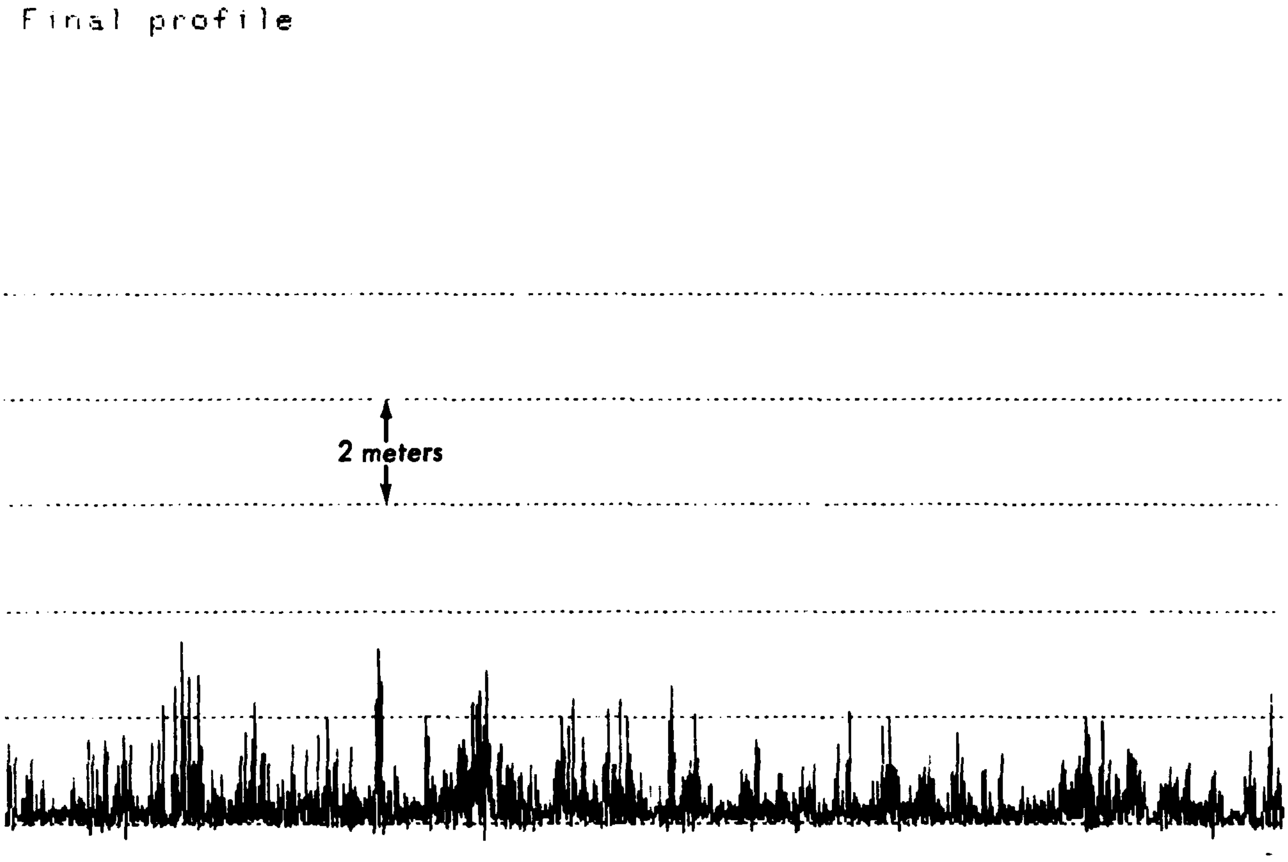
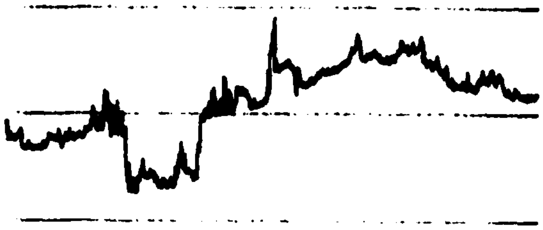
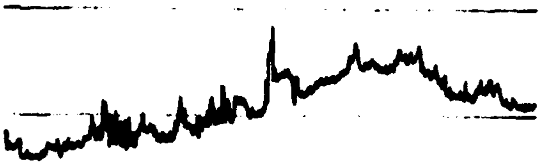


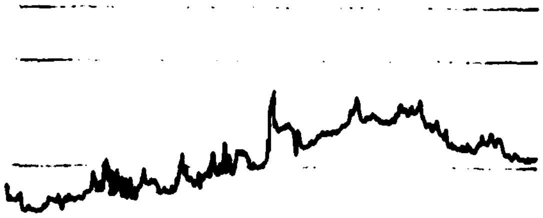
Figure 12. Printout of scaled final sea-ice profile. Vertical scale exaggeration of profile ~1150:1.



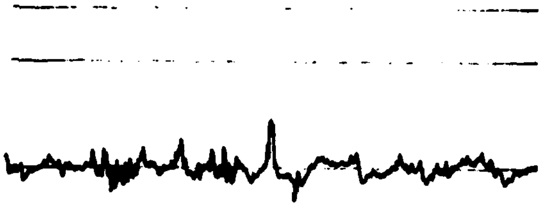
(a) Raw profile data



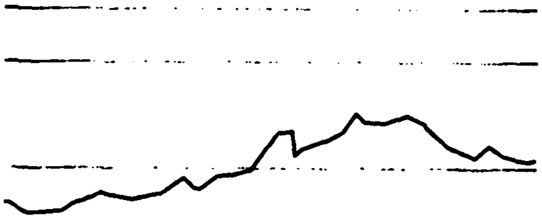
(b) Discontinuities removed



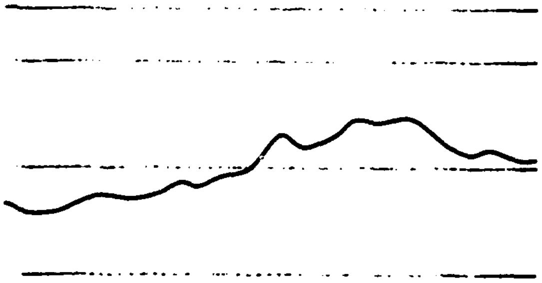
(c) High-frequency noise removed from (b)



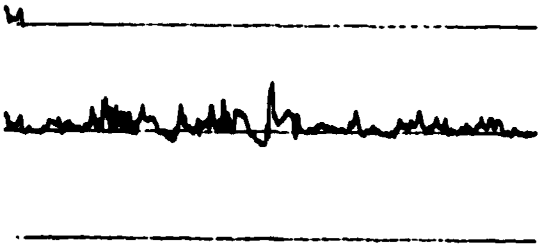
(d) Local low points found by high pass filter of (a)



(e) Low point envelope determined from (a) and (d)



(f) Aircraft motion is lowpass-filtered from low point envelope (e)



(g) Final profile is (c) minus (f)

Figure 13. All steps in aircraft motion removal. Steps (c) through (g) are done automatically by the computer in one program.

APPENDIX

```
1 ! ||||||||||||||||||||||||||  PROGRAM A/D  ||||||||||||||||||||||||||||||||||||||
10    INTEGER U(1:30375),V(1:30375),W(1:30375),D,E,F,File
20    DIM File_name$(3)
21 ! <<<<<<<<<<<<<<<<<<<<<< PROMPTS FOR OPERATOR >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
30    PRINTER IS 16
40    PRINT PAGE
50        PRINT LIN(2)
60        PRINT "  To begin digitizing, press CONT"
70        PRINT LIN(1)
80        PRINT "   To stop digitizing, press PAUSE"
90        PRINT "                        note line number XXX"
100       PRINT "                        type in:   XXX GOTO 260"
110       PRINT "                        press STORE"
120       PRINT "                        press CONT"
130       PAUSE
131 ! <<<<<<<<<<<<<<<<<<< INITIALIZE A/D CONVERTER >>>>>>>>>>>>>>>>>>>>>>>>>>>>>
140   OVERLAP
150   OUTPUT 7,1;"SN;AC,1!"
160   OUTPUT 7,1;"MB,0,82;RP,0;RP,80;WB,8;AI,1,1,1,1;NX;CB;NX!"
161 ! <<<<<<<<<<<<<<<< READ THE RAW PROFILE DATA FROM THE A/D >>>>>>>>>>>>>>>>>>
170   FOR D=1 TO 30375
180   ENTER 7,1 BFHS 2 NOFORMAT;U(D)
190   NEXT D
200   FOR E=1 TO 30375
210   ENTER 7,1 BFHS 2 NOFORMAT;V(E)
220   NEXT E
230   FOR F=1 TO 30375
240   ENTER 7,1 BFHS 2 NOFORMAT;W(F)
250   NEXT F
251 ! <<<<<<<<<<<<<<<<<<< STORE RAW PROFILE DATA ON DISK >>>>>>>>>>>>>>>>>>>>>>>>>
260   IF E=0 THEN Number_of_files=1
270   IF (E<>0) AND (F=0) THEN Number_of_files=2
280   IF F<>0 THEN Number_of_files=3
290       D=D-1
300       E=E-1
310       F=F-1
320   PRINT PAGE
330   PRINT "NAME";Number_of_files;"FILES."
340         IF Number_of_files=1 THEN INPUT " File name ",File_name$(1)
350         IF Number_of_files=2 THEN INPUT " 2 File names",File_name$(1),File_n
ame$(2)
360         IF Number_of_files=3 THEN INPUT " 3 File names",File_name$(1),File_n
ame$(2),File_name$(3)
370   PRINTER IS 0
380   CREATE File_name$(1),D,4
390   ASSIGN #1 TO File_name$(1)
400   REDIM U(1:D)
410   PRINT #1;U(*)
420   PRINT "    ";D;"POINTS IN FILE "&File_name$(1)
430   IF Number_of_files=1 THEN 550
440   CREATE File_name$(2),E,4
450   ASSIGN #2 TO File_name$(2)
460   REDIM V(1:E)
470   PRINT #2;V(*)
480   PRINT "    ";E;"POINTS IN FILE "&File_name$(2)
490   IF Number_of_files=2 THEN 550
500   CREATE File_name$(3),F,4
510   ASSIGN #3 TO File_name$(3)
520   REDIM W(1:F)
530   PRINT #3;W(*)
540   PRINT "    ";F;"POINTS IN FILE "&File_name$(3)
550   END
```

```
10 ! |||||||||||||||||||||||| PROGRAM "Plot" ||||||||||||||||||||||||||||||||||
20    INTEGER Z(1:32767),Picture(0:16380),B,C,D,E,N,Dim,Q,Axis
21 ! <<<<<<<<<<<<<<<<<<<< PROMPTS FOR OPERATOR >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
30    PRINTER IS 16
40    PRINT PAGE
50    PRINT "  This program:"
60    PRINT "            "&CHR$(255)&"  COPIES A DATA FILE FROM DISC INTO MEMORY"
70    PRINT "            "&CHR$(255)&"  INVERTS THE RECORD, AND REMOVES THE ZEROE
S"
80    PRINT "            "&CHR$(255)&"  PURGES THE OLD RECORD FROM DISC"
90    PRINT "            "&CHR$(255)&"  CREATES A BDAT FILE WITH THE SAME NAME ON
 THE SAME DISC"
100   PRINT "            "&CHR$(255)&"  COPIES THE NEW FILE ONTO A DISC"
110   PRINT "           "&CHR$(236)&CHR$(236)&CHR$(236)
120   PRINT "            "&CHR$(255)&"  CREATES A PLOT OF THE NEW FILE "
130   PRINT "            "&CHR$(255)&"  STORES THE PLOT"
140   PRINT LIN(1);"  1. Name the record ( 6 characters : device ), "
150   PRINT "          press CONT."
160   PRINT LIN(1);"  2. Name the file to receive the PLOT,     ( FOR A PLOT ON
LY,"
170   PRINT "                                               CALL THE PLOT "&CHR$(
34)&"X"&CHR$(34)&" [no quotes] )"
180   PRINT "          press CONT."
190   LINPUT " Profile ?",Profile_name$
200   LINPUT " Graph ?",Graph_name$
210   PRINT PAGE
220   PRINT LIN(10);"      DATA NOW BEING READ FROM "&Profile_name$
221 ! <<<<<<<<<<<<<<<<<<<< READ RAW PROFILE DATA FROM DISK >>>>>>>>>>>>>>>>>>>>
230   ASSIGN #1 TO Profile_name$
240   ON END #1 GOTO 280
250       FOR Dim=1 TO 32767
260          READ #1;Z(Dim)
270       NEXT Dim
280   IF Dim>=344 THEN 330
290      PRINT PAGE
300      PRINT LIN(10)
310      PRINT "Record is too small (";Dim;"points )  to use locating algorithm
"
320      STOP
330   PRINT PAGE
340   PRINT "      FILE "&Profile_name$&" CONTAINS";Dim;"POINTS"
350   PRINT LIN(10);"      FILE NOW BEING INVERTED"
360   MAT Z=Z*(-1)          !  INVERT THE RECORD HERE
361 ! <<<<<<<<<<<< REMOVE STATUS BITS FROM RAW PROFILE DATA >>>>>>>>>>>>>>>>>>>>
370   C=0
380   PRINT PAGE
390   PRINT LIN(10);"        ZEROES NOW BEING REMOVED"
400   FOR C=C+1 TO 100
410   IF Z(C)=0 THEN 430
420   NEXT C
430   IF (Z(C+81)=0) AND (Z(C+162)=0) AND (Z(C+243)=0) THEN 480
440          IF C<>101 THEN 470
450             PRINT "ZERO LOCATING ALGORITHM HAS FAILED"
460             STOP
470   GOTO 400
480   D=0
490   !             C IS THE LOCATION OF THE FIRST ZERO
500   FOR E=1 TO Dim
510   IF (E=C) OR ((E-C) MOD 81=0) THEN 540
520   D=D+1
530   Z(D)=Z(E)
540   NEXT E
550   N=D                           !  N IS THE NUMBER OF POINTS REMAINING
560   REDIM Z(1:N)
561 ! <<<<<<<<<<<<<<< RE-STORE RAW PROFILE DATA ON DISK >>>>>>>>>>>>>>>>>>>>>>
570   CALL Redim_and_store(N,Z(*),Profile_name$)
```

```
571 ! <<<<<<<<<<<<<<<<<<<<< CREATE THE RAW DATA GRAPH >>>>>>>>>>>>>>>>>>>>>>>>>>>
580          PLOTTER IS 13,"GRAPHICS"
590   IF Graph_name$="X" THEN 610
600  FCREATE Graph_name$,INT(16381*2/256)+INT(16381*2/65536)+3
610  GRAPHICS
620   GCLEAR
630             FOR B=1 TO N
640             X=B/12.5-120*INT((B-1)/1500)
650             Xaxis=95-4.75*INT((B-1)/1500)
660             Y=Xaxis+Z(B)/600
670             PLOT X,Y
680             IF X=120 THEN PLOT X,Y,2
690             IF X=120 THEN MOVE 0,Xaxis
700             NEXT B
710 !                                                        NUMBER THE ROWS
720          FOR Axis=1 TO 20
730                Xaxis=95.4-4.75*(Axis-1)
740                MOVE 121,Xaxis
750                 PEN 1
760                 CSIZE 15/4.54
770            FOR Q=1 TO 20
780              MOVE 120+Q/5,Xaxis
790              LABEL "|"
800            NEXT Q
810                PEN -1
820                MOVE 120,Xaxis
830                CSIZE 2.2,.5
840                LABEL Axis
850          NEXT Axis
851 ! <<<<<<<<< STORE THE GRAPH ON DISK AND PROVIDE ONE HARD COPY >>>>>>>>>>>>>>
860 !
870  IF Graph_name$="X" THEN 950
880  Picture(0)=-1
890  GSTORE Picture(*)
900  FPRINT Graph_name$,Picture(*)
910  PRINTER IS 0
920  PRINT LIN(5);" This graph of "&CHR$(34)&Profile_name$&CHR$(34)
930  PRINT "         has been stored in "&CHR$(34)&Graph_name$&CHR$(34)
940   PRINT LIN(1);"    "&CHR$(34)&Profile_name$&CHR$(34)&" contains";N;"points."
950  DUMP GRAPHICS
960   PRINT LIN(5)
970  END
980   SUB Redim_and_store(INTEGER M,INTEGER Y(*),F$)
990   INTEGER X(1:M)
1000  MAT X=Y
1010  PURGE F$
1020  FCREATE F$,INT(M*2/256)+INT(M*2/65536)+3
1030  FPRINT F$,X(*)
1040  SUBEND
```

```
10 ! |||||||||!|||||||||||||| PROGRAM "Edit" ||||||||||||||||||||||||||||||||||||
20 !
30    INTEGER Z(1:32767),Zwindow(1:61),Zedit(1:61),A,B,C,D,E,F,G,H,I,Nz,Ndscon,D
elta,In_record,X1,X2,Begin_or_end,X_o,X_aside
40    COM INTEGER Pic(0:16380),Xvalue,Yvalue
50    REAL X,Y
51 ! <<<<<<<<<<<<<<<<<<<<< PROMPTS FOR OPERATOR >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
60    PRINTER IS 16
70    PRINT PAGE
80    PRINT "     This PROGRAM permits editing of edit windows in a file contain
ing "
90    PRINT "     discontinuities."
100   PRINT LIN(1)
110   PRINT "     Resulting corrected windows are stored in a tape file for"
120   PRINT "     use by PROGRAM  Dscon^ ."
130   PRINT LIN(1)
140   PRINT "  ENTER THE FOLLOWING INFORMATION:"
150   PRINT "            "&CHR$(228)&"  name of the file containing the PROFILE"
160   PRINT "            "&CHR$(228)&"  name of the file containing the GRAPH of
the profile"
170   PRINT LIN(1)
180   PRINT "     PLACE A TAPE CARTRIDGE IN :T14, and name the file to record ed
it WINDOWS."
190   Profile$=""
200   Graph$=""
210   Edit$=""
220   LINPUT " PROFILE file name ..... press CONT",Profile$
230       IF Profile$="" THEN 220
240   LINPUT "   GRAPH file name ..... press CONT",Graph$
250       IF Graph$="" THEN 240
260   LINPUT " WINDOWS file name ..... press CONT",Edit$
270       IF Edit$[7,10]<>":T14" THEN 260
280       PRINT PAGE
290       PRINT LIN(8)
300       PRINT Edit$&" now being created.  Takes about 45 seconds."
301 ! <<<<<<<<<<<<<<<<<< CREATE TEMPORARY CORRECTIONS FILE >>>>>>>>>>>>>>>>>>>>>
310   CREATE Edit$,6101,4   ! HOLDS 61 INTEGERS PLUS X_o FOR
320   ASSIGN #1 TO Edit$    ! EACH OF UP TO 100 CORRECTIONS ((61*100)*4)+(1*4))
330   PRINT PAGE
340   PRINT LIN(8)
350   PRINT "   READY TO READ "&Profile$
360   PRINT LIN(1)
370   PRINT "   INSERT CORRECT DISC..... then press CONT."
380   PAUSE
390   FREAD Profile$,Z(*)
400   Nz=ROW(Z)
410   PRINT PAGE
420   PRINT LIN(8)
430   PRINT "   READY TO READ "&Graph$
440   PRINT LIN(1)
450   PRINT "   INSERT CORRECT DISC..... then press CONT."
460   PAUSE
470   FREAD Graph$,Pic(*)
480   PLOTTER IS 13,"GRAPHICS"
490                               ! BEGIN TREATING NEXT DISCONTINUITY
500 Next_d:   !
510     X_aside=X1
520   CALL Locate_dscon(X1,X2)     ! FIND THE DISCONTINUITY
530       IF (X_aside=0) AND (X1=1) THEN 660
540     IF X1>X_aside THEN 660
550         PRINT PAGE
560         EXIT GRAPHICS
570         PRINT LIN(5)
580         PRINT "CURSOR SHOULD BE MOVED FORWARD TO CONTINUE EDITING."
590         BEEP
600         PRINT LIN(1)
```

```
610         PRINT "  to CORRECT the location, enter "&CHR$(34)&"-"&CHR$(34)&" an
d press CONT."
620         PRINT "  to ACCEPT  the location, enter "&CHR$(34)&"+"&CHR$(34)&" an
d press CONT."
630         INPUT "",Qq$
640         IF Qq$="-" THEN 520
650         IF Qq$="+" THEN 660
660   IF X1>=1 THEN 690          !   BE CERTAIN TO STAY WITHIN SUBSCRIPTS !
670         X1=1
680         X2=61
690   IF X2<=ROW(Z) THEN 720
700         X2=ROW(Z)
710         X1=X2-60
720         FOR B=X1 TO X2
730         IF B=X1 THEN I=0
740            I=I+1
750         Zwindow(I)=Z(B)       ! Zwindow(*) is not re-written for a window.
760         NEXT B
770         X_o=X1                ! SETS ASIDE ONLY ONE X TO LOCATE THE WINDOW
780 Next_try:       MAT Zedit=Zwindow
790         !
791 ! <<<<<<<<<<<<<<<<< DRAW LARGE GRAPH OF EDIT WINDOW >>>>>>>>>>>>>>>>>>>>>>>>>
800 Renew:  GRAPHICS
810         GCLEAR
820         LINE TYPE 3
830         MOVE 0,80
840         PLOT 0,80
850         PLOT 120,80
860         MOVE 0,20
870         PLOT 0,20
880         PLOT 120,20
890         LINE TYPE 1
900         MOVE 0,50
910         PLOT 0,50
920         PLOT 120,50
930       FOR C=1 TO 61
940         X=(C-1)*2
950         Y=50+Zedit(C)/50
960         IF C=1 THEN MOVE X,Y
970         PLOT X,Y
980       NEXT C
990 Choose_type:                    PAUSE
1000                                EXIT GRAPHICS
1001 ! <<<<<<<<<<<<<<<<< CHOOSE AN EDITING PROCEDURE >>>>>>>>>>>>>>>>>>>>>>>>>>
1020  PRINTER IS 16
1030  PRINT PAGE
1040  PRINT " BASED ON THE PLOT, CHOOSE ONE OF THE FOLLOWING:"
1050  PRINT LIN(1)
1060  PRINT "             "&CHR$(240)&"      to view the plot again, enter ?"
1070  PRINT "             "&CHR$(240)&"       to skip this location, enter /"
1080  PRINT LIN(1)
1090  PRINT " DISCONTINUITY TYPES :"
1100  PRINT LIN(1)
1110  PRINT "       "&CHR$(240)&" BEGINNING- OR END-OF-RECORD, enter R"
1120  PRINT "       "&CHR$(240)&"        STRAIGHT-SLOPE REMOVAL, enter S"
1130  PRINT "       "&CHR$(240)&"             POINT-BY-POINT FIX, enter P"
1140  PRINT "       "&CHR$(240)&" TIE ( WITH STRAIGHT LINE ) , enter T"
1150  Type$=""
1160  INPUT " ? or / or R or S or P or T ...  then CONT",Type$
1170  IF (Type$<>"?") AND (Type$<>"/") AND (Type$<>"R") AND (Type$<>"S") AND (Ty
pe$<>"P") AND (Type$<>"T") THEN 1160
1180  IF Type$="R" THEN CALL Record(X_o,Zedit(*))
1190  IF Type$="S" THEN CALL Slope(1,Zedit(*))
1200  IF Type$="P" THEN CALL Point(Zedit(*))
1210  IF Type$="T" THEN CALL Slope(0,Zedit(*))
1220  IF Type$<>"?" THEN 1250
1230                GRAPHICS
1240                GOTO Choose_type
1250  IF Type$="/" THEN Next_a
1260  PRINT PAGE
```

```
1270  PRINT LIN(8)
1280  PRINT "    IF SATISFIED WITH EDITING ON THIS WINDOW, enter +"
1290  PRINT "    TO DO EDITING OVER AGAIN FOR THIS WINDOW, enter -"
1300  PRINT "    TO CONTINUE THIS WINDOW WITH ANOTHER TYPE OF CORRECTION, enter
/"
1310  Satisfaction$=""
1320  INPUT "  + or - or /",Satisfaction$
1330  IF (Satisfaction$<>"+") AND (Satisfaction$<>"-") AND (Satisfaction$<>"/")
THEN 1320
1340  IF Satisfaction$="+" THEN Set_aside
1350  IF Satisfaction$="-" THEN Next_try
1360  IF Satisfaction$="/" THEN Renew
1361 ! <<<<<<<<<<<<<<<< WRITE EDIT WINDOW ONTO TAPE CARTRIDGE >>>>>>>>>>>>>>>>>>>>
1370 Set_aside:                      !  Set aside ( 62 points ).
1380        PRINT #1;X_o
1390        PRINT #1;Zedit(*)
1400    GOTO Next_d
1401 ! <<<<<<<<<< SUBROUTINE FOR RECORDING BEGINNING OR END OF FILE>>>>>>>>>>>>>
1410 SUB Record(INTEGER Xzero,INTEGER Z(*))
1420  GRAPHICS
1430 INTEGER X1,X2,X_1,Z_window(1:61),A
1440  REAL U,V
1450 CSIZE 15/4.54
1460 MOVE 20,90
1470 LABEL "RECORD               [] END      [] START"
1480  MOVE 20,85
1490  LABEL "( point , then [] )"
1500 POINTER 10,55,0
1501 ! <<<<<<<<<<<<<<< PROVIDE CURSOR TO LOCATE BOF or EOF >>>>>>>>>>>>>>>>>>>>>>
1510 DIGITIZE X1,Y1
1520  DIGITIZE X2,Y2
1530  IF X1<2 THEN X1=2
1540  IF X1>118 THEN X1=122
1550 X_1=Xzero+INT(X1/2)-1
1560  IF (Y2>89) AND (X2>54) AND (X2<57) THEN Mark=1
1570  IF (Y2>89) AND (X2>76) AND (X2<79) THEN Mark=2
1580  IF Mark=1 THEN Z_window(INT(X1/2))=-32767
1590  IF Mark=2 THEN Z_window(INT(X1/2))=32767
1600 CSIZE 2.5
1610 MOVE X1-10,82
1620 IF Mark=1 THEN LABEL " END   | "&VAL$(X_1)
1630 IF Mark=2 THEN LABEL "STAPT | "&VAL$(X_1)
1640  CSIZE 15/4.54
1650 LINE TYPE 3
1660 MOVE X1,81
1670 PLOT X1,81
1680 PLOT X1,19
1690 LINE TYPE 1
1700 PAUSE
1710 PRINTER IS 16
1720 PRINT PAGE
1730 PRINT LIN(5);"                        If satisfied with the correction, enter +"
1740 PRINT LIN(1);"                                  To re-do the correction, enter -"
1750 Do$=""
1760 INPUT "+ , -",Do$
1770  IF (Do$<>"+") AND (Do$<>"-") THEN 1760
1780                                          IF Do$="-" THEN 1810
1790 Z(INT(X1/2))=Z_window(INT(X1/2))   ! CHANGE ONLY THE VALUE OF ONE Z(*)
1800 SUBEXIT
1801 ! <<<<<<<<<<<<<<<<<< GRAPH OF THE CORRECTED EDIT WINDOW >>>>>>>>>>>>>>>>>>>>
1810  GCLEAR
1820  LINE TYPE 3
1830  MOVE 0,80
1840  PLOT 0,80
1850  PLOT 120,80
1860  MOVE 0,20
1870  PLOT 0,20
1880  PLOT 120,20
1890  MOVE 0,50
1900  LINE TYPE 1
1910  PLOT 0,50
```

```
1920  PLOT 120,50
1930  FOR A=1 TO 61
1940  U=(A-1)*2
1950  V=50+Z(A)/50
1960  IF A=1 THEN MOVE U,V
1970  PLOT U,V
1980  NEXT A
1990 GOTO 1460
2000 SUBEND
2001 ! <<<<<<<<< SUBROUTINE FOR SLOPE OR STRAIGHT-LINE CORRECTION >>>>>>>>>>>>>
2010 SUB Slope(INTEGER Slope_or_tie,INTEGER Z(*))
2020 INTEGER X1,X2,X3,Y1,Y2,Z_window(1:61),A
2030  REAL U,V,Slope_fix_1,Slope_fix_0,Step_value_1,Step_value_0,Y3,Real_z
2040 GRAPHICS
2050 CSIZE 15/4.54
2060         MAT Z_window=Z  ! VALUES OF X(*) NEED NOT BE CHANGED OR NOTED
2070 MOVE 10,90
2080 IF Slope_or_tie=1 THEN LABEL "STRAIGHT-LINE SLOPE"
2090 IF Slope_or_tie=0 THEN LABEL "STRAIGHT-LINE TIE  "
2100 POINTER 10,55,8
2110 DIGITIZE X1,Y1
2120 DIGITIZE X2,Y2
2130 DIGITIZE X3,Y3
2140  IF X3=X2 THEN 2220
2150      PRINT PAGE
2160      EXIT GRAPHICS
2170      PRINT LIN(8)
2180      PRINT "    LAST 2 X's MUST BE EQUAL."
2190      PRINT "    Start over by pressing CONT."
2200      PAUSE
2210      GOTO 2100
2220 Xa=INT(X1/2)+1
2230 Xb=INT(X2/2)+1
2240 Real_z=50*(Y3-50)
2250 IF Xb>Xa THEN 2330
2260 EXIT GRAPHICS
2270 PRINT PAGE
2280 PRINT LIN(5);" Since X(a)=";Xa;" and X(b)=";Xb;", the slope has the wrong s
ign."
2290  PRINT " PLEASE REPEAT, and make X(b) > X(a)."
2300 BEEP
2310 PAUSE
2320 GOTO 2100
2330 Step_value_1=Real_z-Z_window(Xb)
2340 Step_value_0=Real_z-Z_window(Xa)
2350 Slope_fix_1=Step_value_1/(Xb-Xa)
2360 Slope_fix_0=Step_value_0/(Xb-Xa)
2370 FOR A=1 TO 61
2380 IF A<=Xa THEN 2440
2390 IF (Slope_or_tie=1) AND (A>Xa) AND (A<Xb) THEN Z_window(A)=Z_window(A)+Slop
e_fix_1*(A-Xa)
2400 IF (Slope_or_tie=0) AND (A>Xa) AND (A<Xb) THEN Z_window(A)=Z_window(Xa)+Slo
pe_fix_0*(A-Xa)
2410 IF (A>Xa) AND (A<Xb) THEN LINE TYPE 3 ! DOTTED LINE IN FIX AREA
2420 IF A>=Xb THEN Z_window(A)=Z_window(A)+Step_value_1
2430 IF A>=Xb THEN LINE TYPE 1
2440 U=(A-1)*2
2450 V=Z_window(A)/50+50
2460 IF A=1 THEN MOVE U,V
2470 PLOT U,V
2480 NEXT A
2490 PAUSE
2500 PRINT PAGE
2510  PRINT LIN(5)
2520  PRINT "  TO ACCEPT THE CORRECTION, enter +   ...press CONT"
2530  PRINT "  TO CHANGE THE CORRECTION, enter -   ...press CONT"
2540 If$=""
2550 INPUT "    +   or  -",If$
2560  IF (If$<>"+") AND (If$<>"-") THEN 2550
2570 IF If$="-" THEN 2600
```

```
2580 MAT Z=Z_window
2590 SUBEXIT
2591 ! <<<<<<<<<<<<<<<<<< GRAPH OF THE CORRECTED EDIT WINDOW >>>>>>>>>>>>>>>>>>>>
2600  GRAPHICS
2610  GCLEAR
2620       LINE TYPE 3
2630       MOVE 0,80
2640       PLOT 0,80
2650       PLOT 120,80
2660       MOVE 0,20
2670       PLOT 0,20
2680       PLOT 120,20
2690       LINE TYPE 1
2700       MOVE 0,50
2710       PLOT 0,50
2720       PLOT 120,50
2730       FOR A=1 TO 61
2740       U=(A-1)*2
2750       V=50+Z(A)/50
2760       IF A=1 THEN MOVE U,V
2770       PLOT U,V
2780       NEXT A
2790       GOTO 2060
2800 SUBEND
2801 ! <<<<<<<<<<<< SUBROUTINE FOR POINT-BY-POINT CORRECTION >>>>>>>>>>>>>>>>>>
2810 SUB Point(INTEGER Z(*))
2820 INTEGER X1,X2,Z_window(1:61),Z_plot(1:61),A,C,D
2830 REAL Y1,Y2,U,V,Z_real,Delta
2840  CSIZE 15/4.54
2850 Start:  GRAPHICS                                                   !
2860          MAT Z_window=Z
2870 Next:                                                              !
2880             MOVE 10,90
2890             LABEL "POINT FIX"
2900             MOVE 2,5
2910             LABEL "< up or down."
2920             LINE TYPE 3
2930             MOVE 2,100
2940             PLOT 2,100
2950             PLOT 2,0
2960             LINE TYPE 1
2970 Here:                                                               !
2980     POINTER 10,55,8
2990       DIGITIZE X1,Y1
3000       DIGITIZE X2,Y2
3010     IF X2>=2 THEN 3050
3020             X1=X2=0
3030             Delta=(Y2-Y1)*50   ! PROVIDE FOR MOVING ENTIRE WINDOW UP OR DOWN
3040             GOTO Plot
3050     IF X2=X1 THEN Hence
3060  EXIT GRAPHICS
3070  PRINTER IS 16
3080  PRINT PAGE
3090  PRINT LIN(5);"     X's must be equal... press CONT"
3100  PAUSE
3110  GRAPHICS
3120  GOTO Here      ! TRY DIGITIZING AGAIN
3130 Hence:                                                              !
3140           Z_real=50*(Y2-50)
3150           Delta=Z_real-Z_window((X2+2)/2)          ! THE CORRECTION FOR THIS
 Z
3160 Plot:                                                               !
3170     GRAPHICS
3180   MAT Z_plot=Z_window      ! SET UP A TEMPORARY PLOTTING VECTOR
3190             FOR C=1 TO 61
3200                U=(C-1)*2
3210                 IF U>=X1 THEN LINE TYPE 3  ! DOTTED LINE AFTER CORRECTION.
3220                 IF U>=X1 THEN Z_plot(C)=Z_plot(C)+Delta
3230                V=Z_plot(C)/50+50
3240                IF C=1 THEN MOVE U,V
```

```
3250             PLOT U,V
3260          NEXT C
3270             LINE TYPE 1
3280 PAUSE
3290  EXIT GRAPHICS
3300  PRINTER IS 16
3310  PRINT PAGE
3320    PRINT LIN(5);"   If this correction is     acceptable, enter +"
3330    PRINT LIN(2);"   If this correction is NOT acceptable, enter -"
3340    PRINT LIN(2);"      To begin with the original window, enter /"
3350    PRINT LIN(3);"        If all corrections are complete, enter *"
3360 Sat$=""
3370 INPUT "",Sat$
3380  IF (Sat$<>"+") AND (Sat$<>"-") AND (Sat$<>"*") AND (Sat$<>"/") THEN 3370
3390     IF Sat$="*" THEN Final
3400     IF Sat$="-" THEN Minus
3410     IF Sat$="/" THEN Minus
3420     IF Sat$="+" THEN Plus
3430 Plus:                                                               !
3440       MAT Z_window=Z_plot   ! THE PLOT VECTOR IS NOW THE EDIT WINDOW
3450 Minus:                      ! CONTINUE BY FIRST RE-DRAWING THE PLOT
3451 ! <<<<<<<<<<<<<<<<< GRAPH OF THE CORRECTED EDIT WINDOW >>>>>>>>>>>>>>>>>>>
3460       GCLEAR
3470       GRAPHICS
3480       LINE TYPE 3
3490       MOVE 0,80
3500       PLOT 0,80
3510       PLOT 120,80
3520       MOVE 0,20
3530       PLOT 0,20
3540       PLOT 120,20
3550       LINE TYPE 1
3560       MOVE 0,50
3570       PLOT 0,50
3580       PLOT 120,50
3590       FOR A=1 TO 61
3600       U=(A-1)*2
3610          IF (Sat$="-") OR (Sat$="+") THEN V=50+Z_window(A)/50
3620    ! LINE 3240 SHOWS Z_window TO BE THE PRESENT WORKING EDIT WINDOW
3630          IF Sat$="/" THEN V=50+Z(A)/50        ! RETURN TO ORIGINAL Z_window
3640       IF A=1 THEN MOVE U,V
3650       PLOT U,V
3660       NEXT A
3670    IF Sat$="/" THEN Start          ! BEGIN AGAIN BE RE-LOADING Z(*)
3680    IF (Sat$="-") OR (Sat$="+") THEN Next
3690 Final:                     ! ACCEPT EDIT WINDOW                        !
3700         MAT Z_window=Z_plot
3710         MAT Z=Z_window
3720         EXIT GRAPHICS
3730 SUBEND
3731 ! <<<<<< SUBROUTINE FOR LOCATING EDIT WINDOW USING SMALL-SCALE PLOT >>>>>>
3740  SUB Locate_dscon(INTEGER Xzero,INTEGER X100)
3750  INTEGER Correction_at_x
3760  COM INTEGER Pic(*),Xvalue,Yvalue
3770 GLOAD Pic(*)
3780  GRAPHICS
3790 POINTER Xvalue,Yvalue,2   ! POINTER IS PLACED AT LAST CORRECTION POINT
3800                 ! BEGIN CORRECTIONS HERE
3810 DIGITIZE Xvalue,Yvalue
3820 Line_number=INT((102.125-Yvalue)/4.75)
3830 Correction_at_x=1500*Line_number-1499+Xvalue*12.5
3840  Xzero=Correction_at_x-30
3650  X100=Correction_at_x+30
3860 SUBEND
```

```
1 ! ||||||||||||||||||||||||| PROGRAM "Discon" ||||||||||||||||||||||||||||||||||||
2 ! <<<<<<<<<<<<<<<<<<<<<< PROMPTS FOR OPERATOR >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
10    PRINTER IS 16
20    PRINT PAGE;LIN(2)
30    PRINT "        This program uses the edit windows recorded by program Edit"
40    PRINT "            to rewrite a profile, taking out discontinuities."
50    PRINT LIN(2)
60    PRINT "        ENTER the name of the file"
70    PRINT "                    containing the PROFILE    ( a BDAT file ),"
80    PRINT "               containing the edit WINDOWS    ( a DATA file )."
90    PRINT LIN(1)
100   PRINT "        Then press CONT."
110   PRINT LIN(1)
120   PRINT "      ( AS A RECORD IS COMPLETED, ITS NAME WILL BE REQUESTED AS A FI
LE NAME. )"
130     P$=W$="             "
140   LINPUT "            PROFILE ...  then CONT",P$
150     IF P$[8;1]<>"F" THEN 140
160   LINPUT "            WINDOWS ...  then CONT",W$
170   INTEGER Z(1:30100),Window(1:100,0:61),A,Z_size,Delta_z
180   INTEGER No_of_windows,Loop,X,Current_x,Window_x,Next_x,Z_aside
190   ASSIGN #1 TO W$
200   FREAD P$,Z(*)
210   ON END #1 GOTO Continue
220   PRINTER IS 0
221 ! <<<<<<<<<<<<<<< READ EDIT WINDOWS FROM CARTRIDGE >>>>>>>>>>>>>>>>>>>>>>>>>
230   FOR A=1 TO 100
240   FOR B=0 TO 61
250   READ #1;Window(A,B)    ! Window(a,0) is the window starting location.
260   NEXT B
270   PRINT Window(A,0);"|";
280   NEXT A
281 ! <<<<<<<<<<<<< BEGIN CORRECTIONS (all automatic) >>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>
290 Continue: !
300   Z_size=ROW(Z)
310   No_of_windows=A-1
320   Current_x=1
330   ! ___________________________________________________________________________

340   ! Case 1:  First window ( contains a BOR )
350   !               11.  Window(1,0)=1
360   !               12.  Window(1,0)<>1
370   ! Case 2:  Encountering a new window ( Current_x=Next_x ) with:
380   !   .           21.  Window_x=0
390   !                      211.  Loop=No_of_windows  ( contains EOR )
400   !               22.  Window_x<>0   ( overlapping windows )
410   !                      221.  Loop=No_of_windows  ( contains EOR )
420   ! Case 3:  In a window not LAST window( Current_x<Next_x AND Window_x<>0 )
430   !             Encountering a
440   !               31. BOR   ( Window(Loop,Window_x=32767 )
450   !               32. EOR   ( Window(Loop,Window_x=-32767 )
460   ! Case 4:  Leaving a window ( i.e. Window_x=61 )
470   ! Case 5:  Between windows  ( Current_x<Next_x AND Window_x=0 )
480   ! ___________________________________________________________________________

490       IF (Current_x=1) AND (Window(1,0)=1) THEN Case11
500       IF (Current_x=1) AND (Window(1,0) >1) THEN Case12
510 Next_point:!
520       IF (Current_x=Next_x) AND (Window_x=0) THEN Case21
530       IF (Current_x=Next_x) AND (Window_x>0) AND (Window_x<61) THEN Case22
540       IF (Current_x<Next_x) AND (Window_x>0) AND (Window_x<61) THEN Case3
550       IF Window_x=61 THEN Case4
560       IF (Current_x<Next_x) AND (Window_x=0) THEN Case5
570 Case11:
!
```

```
580           Loop=1
590           Next_x=Window(2,0)                  ! ASSUME Next_x occurs AFTER BOR
600           FOR A=1 TO 61
610              IF Window(1,A)<>32767 THEN 670
620                  Print$="ON"
630                  Window(1,A)=Window(1,A+1)
640                  Current_x=A
650                  Window_x=A
660                  GOTO Next_point
670           NEXT A
680 Case12:                                                                     !

690           Next_x=Current_x=Window(1,0)
700           Window(1,1)=Window(1,2)
710         GOTO Next_point
720 Case21:                                                                     !

730                           Loop=Loop+1
740                           IF Loop=No_of_windows THEN Case211
750                           Next_x=Window(Loop+1,0)
760       IF Print$="OFF" THEN Skip21
770           Delta_z=Z(Current_x)+Delta_z-Window(Loop,1)
780       IF ABS(Window(Loop,1)+Delta_z)<30000 THEN 870
790 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
800                       REDIM Z(1:X)                             !
810                       CALL Record_record(Z(*),X)               !
820                       REDIM Z(1:Z_size)                        !
830     IF Window(Loop,1)+Delta_z<-30000 THEN Delta_z=Delta_z+30000!
840      IF Window(Loop,1)+Delta_z>30000 THEN Delta_z=Delta_z-30000!
850                       X=0                                      !
860 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
870           X=X+1
880           Z(X)=Window(Loop,1)+Delta_z
890 Skip21:                                              ! DO NOT RE-COMPUTE Delta_z
900           Window_x=2
910           Current_x=Current_x+1
920         GOTO Next_point
930 Case22:                                                                     !

940                           Loop=Loop+1
950                           IF Loop=No_of_windows THEN Case221
960                           Next_x=Window(Loop+1,0)
970       IF Print$="OFF" THEN Skip22
980           Delta_z=Window(Loop-1,Window_x)+Delta_z-Window(Loop,1)
990       IF ABS(Window(Loop,1)+Delta_z)<30000 THEN 1080
1000 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1010                      REDIM Z(1:X)                             !
1020                      CALL Record_record(Z(*),X)               !
1030                      REDIM Z(1:Z_size)                        !
1040    IF Window(Loop,1)+Delta_z<-30000 THEN Delta_z=Delta_z+30000!
1050     IF Window(Loop,1)+Delta_z>30000 THEN Delta_z=Delta_z-30000!
1060                      X=0                                      !
1070 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1080          X=X+1
1090          Z(X)=Window(Loop,1)+Delta_z
1100 Skip22:                                             ! DO NOT RE-COMPUTE Delta_z
1110          Window_x=2
1120          Current_x=Current_x+1
1130        GOTO Next_point
1140 Case211:                                                                   !

1150          Delta_z=Z(Current_x)+Delta_z-Window(Loop,1)
1160          GOTO Go_on_21
1170 Case221:                                                                   !

1180          Delta_z=Window(Loop-1,Window_x)+Delta_z-Window(Loop,1)
1190 Go_on_21:                                                                  !

1200        FOR A=1 TO 61
1210            IF Window(Loop,A)=-32767 THEN Skip_21
```

```
1220                      REDIM Z(1:X)
1230                      CALL Record_record(Z(*),X)
1240                      PRINT PAGE
1250                      PRINT LIN(5)
1260                      PRINT "PROGRAM COMPLETED"
1270                      STOP
1280 Skip_21:
!
1290        IF ABS(Window(Loop,A)+Delta_z)<30000 THEN 1380
1300 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1310                         REDIM Z(1:X)                                !_
1320                         CALL Record_record(Z(*),X)                  !_
1330                         REDIM Z(1:Z_size)                           !_
1340      IF Window(Loop,A)+Delta_z<-30000 THEN Delta_z=Delta_z+30000!_
1350       IF Window(Loop,A)+Delta_z>30000 THEN Delta_z=Delta_z-30000!_
1360                         X=0                                         !_
1370 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1380                X=X+1
1390                Z(X)=Window(Loop,A)+Delta_z
1400           NEXT A
1410                      REDIM Z(1:X)
1420                      CALL Record_record(Z(*),X)
1430                      PRINT PAGE
1440                      PRINT LIN(5)
1450                      PRINT "   PROGRAM COMPLETED"
1460                      STOP
1470 Case3:

1480        IF Window(Loop,Window_x)=32767 THEN Case31
1490        IF Window(Loop,Window_x)=-32767 THEN Case32
1500          GOTO Case_3_continue
1510 Case31:

1520            Print$="ON"
1530            Window(Loop,Window_x)=Window(Loop,Window_x+1)
1540            Delta_z=0
1550            GOTO Case_3_continue
1560 Case32:

1570            Print$="OFF"
1580            Delta_z=0
1590            REDIM Z(1:X)
1600            CALL Record_record(Z(*),X)
1610            X=0
1620            REDIM Z(1:Z_size)
1630 Case_3_continue:

1640           IF Print$="OFF" THEN No_print_3
1650        IF ABS(Window(Loop,Window_x)+Delta_z)<30000 THEN 1740
1660 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1670                         REDIM Z(1:X)                                    !_
1680                         CALL Record_record(Z(*),X)                      !_
1690                         REDIM Z(1:Z_size)                               !_
1700      IF Window(Loop,Window_x)+Delta_z<-30000 THEN Delta_z=Delta_z+30000!_
1710       IF Window(Loop,Window_x)+Delta_z>30000 THEN Delta_z=Delta_z-30000!_
1720                         X=0                                             !_
1730 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1740                          X=X+1
1750                          Z(X)=Window(Loop,Window_x)+Delta_z
1760 No_print_3:

1770            Current_x=Current_x+1
1780            Window_x=Window_x+1
1790          GOTO Next_point
1800 Case4:

1810          IF Print$="OFF" THEN Skip4
1820            Delta_z=Window(Loop,A)+Delta_z-Z(Current_x)
1830        IF ABS(Z(Current_x)+Delta_z)<30000 THEN 1920
1840 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
1850                       REDIM Z(1:X)                           !
1860                       CALL Record_record(Z(*),X)             !
1870                       REDIM Z(1:Z_size)                      !
1880    IF Z(Current_x)+Delta_z<-30000 THEN Delta_z=Delta_z+30000!
1890    IF Z(Current_x)+Delta_z>30000 THEN Delta_z=Delta_z-30000 !
1900                       X=0                                    !
1910 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1920          X=X+1
1930          Z(X)=Z(Current_x)+Delta_z
1940 Skip4:                                                                   !

1950          Window_x=0
1960          Current_x=Current_x+1
1970        GOTO Next_point
1980 Case5:                                                                   !

1990          FOR A=Current_x TO Next_x-1
2000              IF Print$="OFF" THEN 2120
2010              IF ABS(Z(A)+Delta_z)<30000 THEN 2100
2020 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2030                       REDIM Z(1:X)                           !
2040                       CALL Record_record(Z(*),X)             !
2050                       REDIM Z(1:Z_size)                      !
2060                IF Z(A)+Delta_z<-30000 THEN Delta_z=Delta_z+30000!
2070                IF Z(A)+Delta_z>30000 THEN Delta_z=Delta_z-30000 !
2080                       X=0                                    !
2090 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2100                           X=X+1
2110                           Z(X)=Z(A)+Delta_z
2120          NEXT A
2130             Current_x=Next_x
2140     GOTO Next_point
2150  END
2151 ! <<<<< SUBROUTINE FOR STORING THE DATA WITH DISCONTINUITIES REMOVED>>>>>
2160  SUB Record_record(INTEGER Z_in(*),INTEGER N)
2170  PRINTER IS 16
2180  PRINT PAGE
2190  PRINT LIN(8)
2200  PRINT "  Name the record which will receive this data."
2210  PRINT LIN(1)
2220  PRINT "  LENGTH : ";N;"points."
2230  F$="          "
2240  LINPUT "         XXXXXX:F or XXXXXX:T14",F$
2250  IF (F$[8;1]<>"F") AND (F$[8;1]<>"T") THEN 2240
2260  INTEGER Z(1:N)
2270  MAT Z=Z_in
2280  IF F$[8;1]<>"F" THEN 2320
2290                       FCREATE F$,INT(2*N/256)+INT(2*N/65536)+3
2300                       FPRINT F$,Z(*)
2310                       GOTO 2360
2320  !  DATA file:
2330                       CREATE F$,N/4
2340                       ASSIGN #1 TO F$
2350                       PRINT #1;Z(*)
2360  SUBEND
```

```
1 ! |||||||||||||||||||||||| PROGRAM "8-in-1" ||||||||||||||||||||||||||||||||||||
10       INTEGER Z(1:30800),Z_raw(1:30800),Observations,Nint
20       DIM Raw_in$(8)[12],Result$(8)[12]
21 ! <<<<<<<<<<<<<<<<<<<<<< PROMPTS FOR OPERATOR >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
30   PRINTER IS 16
40   PRINT PAGE
50   PRINT "      PROGRAM 8-in-1 removes aircraft motion from a laser profile"
60   PRINT "        by filtering the data in three steps."
70   PRINT LIN(1)
80   PRINT "      YOU WILL BE REQUESTED TO PROVIDE UP TO 8 PAIRS OF FILES,"
90   PRINT "      EACH CONSISTING OF A RAW DATA FILE FOLLOWED BY A RESULT FILE."
100  PRINT LIN(1)
110  PRINT "      PLEASE MAKE FILE NAMES EXACTLY 6 CHARACTERS LONG."
120  PRINT LIN(1)
130  PRINT LIN(1)
140  PRINT "      TO KEEP TRACK OF THE FILES REQUESTED, "
150  PRINT "      ARRANGE AND NUMBER YOUR LIST OF FILE PAIRS BEFORE CONTINUING."
160  PRINT LIN(1)
170  PRINT "      Press CONT to go on."
180   PAUSE
190   FOR Input=1 TO 8
200      PRINT PAGE
210      PRINT LIN(3)
220      PRINT "NAME YOUR (NEXT) PAIR OF FILES.    (_DEVICE CODE MUST BE "&CHR
$(34)&"F"&CHR$(34)&" )."
230      PRINT LIN(2)
240      PRINT "          Example:      RawDat:F  CONT    Result:F8,1
CONT"
250      PRINT LIN(1)
260      PRINT "        ( WHEN FINISHED NAMING FILES, ENTER NOTHING AND PRESS CO
NT. )"
270      PRINT LIN(3)
280      PRINT "FILE PAIR #";Input
290           LINPUT " Raw data file",Raw_in$(Input)
300           LINPUT "  Result file",Result$(Input)
310           IF (Raw_in$(Input)="") AND (Result$(Input)="") THEN 350
320           IF (Raw_in$(Input)[8;1]<>"F") OR (Result$(Input)[8;1]<>"F") THEN 2
90
330   NEXT Input
340   ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
350   Jmax=Input-1
360               PRINT PAGE
370               PRINT LIN(2)
380               PRINT " YOU WILL BE PROCESSING";Jmax;"FILE(S)."
390               PRINT LIN(1)
400               PRINT "  Your file(s) are:    RAW DATA     RESULT"
410               PRINT LIN(1)
420     FOR Q=1 TO Jmax
430      PRINT "                        "&Raw_in$(Q)&" "&CHR$(240)&" "&Result$(Q)
440     NEXT Q
450               PRINT LIN(1)
460               PRINT " INSERT NEW DISC IF NECESSARY."
470               PRINT " then press CONT"
480             PAUSE
490  CALL Warning                ! CRT warning: "DO NOT DISTURB KEYBOARD"
500   FOR J=1 TO Jmax    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
510                 Raw$=Raw_in$(J)
511 ! <<<<<<<<<<<<<< READ IN DATA WITH DISCONS REMOVED >>>>>>>>>>>>>>>>>>>>>>>>>>
520   FREAD Raw$,Z_raw(*)
530   MAT Z=Z_raw
540   Observations=ROW(Z)
550                 Into$=Result$(J)
551 ! <<<<<<<<<< CREATE A FILE TO ACCEPT THE FINAL PROFILE >>>>>>>>>>>>>>>>>>>>>
560   FCREATE Into$,INT(30800*2/256)+INT(30800*2/65536)+3
570
571 ! <<<<<<<<<< FEED THE RAW DATA THRU THE FIRST ACTIVE FILTER >>>>>>>>>>>>>>>
572 ! <<<<<<<<<<      ( REMOVE THE HIGH-FREQUENCY NOISE )       >>>>>>>>>>>>>>>
```

```
580            CALL Active_filter_1(Z(*),Observations) ! LOWPASS 3 Hz
590                                                                         !
600            MAT Z_raw=Z   ! SINCE HIGH-FREQUENCY NOISE HAS BEEN REMOVED
601 ! <<<<<<<<<< FEED THE RAW DATA THRU THE SECOND ACTIVE FILTER >>>>>>>>>>>>>>>
602 ! <<<<<<<<<<<       < READY DATA FOR LOCATING LOW POINTS >      >>>>>>>>>>>>>>>
610                                                                         !
620            CALL Active_filter_2(Z(*),Observations) ! HIGHPASS .03 Hz
630                                                                         !
640               PRINT PAGE
650               PRINT LIN(8)
660               PRINT " LOW-POINT ROUTINE IN PROGRESS."
662 ! <<<<<<<<< LOCATE LOW POINTS AND DRAW LOW-POINT ENVELOPE >>>>>>>>>>>>>>>>>
670            CALL Low_point(Z_raw(*),Z(*),Observations,50)
680  CALL Warning                ! CRT warning: "DO NOT DISTURB KEYBOARD"
682 ! <<<<<<<<<<<<<<<< SMOOTH THE LOW-POINT ENVELOPE >>>>>>>>>>>>>>>>>>>>>>>>>>>
690            CALL Active_filter_3(Z(*),Observations) ! LOWPASS .13 Hz
700                                                                         !
710        MAT Z=Z_raw-Z                   ! Remove aircraft motion
720        FPRINT Into$,Z(*)               ! Store final profile
730                                                                         !
731 ! <<<<<<<<<<<<<<<< PROVIDE A HARD COPY OF FINAL PROFILE >>>>>>>>>>>>>>>>>>>
740            CALL Graphics(Raw$,Into$,Z(*),Observations)
750                                                                         !
760   NEXT J   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
770                                                                         !
780  END
790  SUB Low_point(INTEGER Z_raw(*),INTEGER Z(*),Observations,Int_length)
800  INTEGER Zmin(1:1000),Xmin(1:1000),Interval,Alow,Ahigh,A,Zlow,No_of_ints,Bst
art,Bstop,B,Z_diminish(1:375),C
810   REAL Delta_x,Delta_z
820   MAT Zmin=(0)
830  Interval=0
840  Flag=0
850  Interval=Interval+1
860  Alow=(Interval-1)*Int_length+1
870  Ahigh=Interval*Int_length
880     IF Ahigh<Observations THEN 910
890              Flag=1
900              Ahigh=Observations
910  FOR A=Alow TO Ahigh
920     IF A=Alow THEN Zlow=0
930     IF Z(A)>=Zlow THEN 960
940        Zlow=Z(A)
950        Xmin(Interval)=A
960    NEXT A
970        IF Zlow<>0 THEN 1010
980               Zmin(Interval)=Z_raw(Ahigh)-Z(Ahigh)
990               Xmin(Interval)=Ahigh
1000              GOTO 1090
1010      IF (Xmin(Interval)-10<1) OR (Xmin(Interval)+10>Observations) THEN 1090
1020             Zlow=Z_raw(Xmin(Interval))
1030                  FOR C=Xmin(Interval)-10 TO Xmin(Interval)+10
1040                        IF Z_raw(C)>=Zlow THEN 1070
1050                                    Zlow=Z_raw(C)
1060                                    Xmin(Interval)=C
1070                  NEXT C
1080             Zmin(Interval)=Z_raw(Xmin(Interval))
1090 IF Flag=0 THEN 850
1100   No_of_ints=Interval
1110 Flag=0
1120 Interval=0
1130 Interval=Interval+1
1140    IF Interval<>1 THEN 1170
1150     Bstart=1
1160     GOTO 1180
1170            IF Interval<=No_of_ints+1 THEN Bstart=Xmin(Interval-1)
1180            IF Interval<No_of_ints+1 THEN Bstop=Xmin(Interval)
1190            IF Interval=No_of_ints+1 THEN Bstop=Observations
1200     IF Bstop=Observations THEN Flag=1
1210  !
```

```
1220  IF Interval=1 THEN Delta_z=Zmin(1)-Z_raw(1)
1230  !
1240  IF (Interval>1) AND (Flag=0) THEN Delta_z=Zmin(Interval)-Zmin(Interval-1)
1250  !
1260  IF Flag=1 THEN Delta_z=Z_raw(Observations)-Zmin(Interval-1)
1270  !
1280 Delta_x=Bstop-Bstart+1
1290 FOR B=Bstart TO Bstop
1300  IF Interval=1 THEN Z(B)=Z_raw(1)+Delta_z/Delta_x*(B-Bstart+1)
1310  IF Interval>1 THEN Z(B)=Zmin(Interval-1)+Delta_z/Delta_x*(B-Bstart+1)
1320 NEXT B
1330 IF Flag=0 THEN 1130
1340 SUBEND
1350  SUB Graphics(Q$,P$,INTEGER Z(*),Obs)
1360  INTEGER A,B
1370  PLOTTER IS "GRAPHICS"
1380  GRAPHICS
1390    LINE TYPE 3
1400    FOR B=10 TO 60 STEP 10
1410      MOVE 0,B
1420      PLOT 0,B
1430      PLOT 120,B
1440    NEXT B
1450      LINE TYPE 1
1460      MOVE 0,85
1470   LABEL " Final profile "&P$&"(from "&Q$&")      ";Obs;"points."
1480                FOR A=1 TO Obs STEP 10
1490                           U=.004*A
1500                           V=Z(A)/38.65+10
1510                           IF A=1 THEN MOVE U,V
1520                           PLOT U,V
1530                NEXT A
1540 DUMP GRAPHICS
1550  PRINTER IS 0
1560  PRINT LIN(5)          !  FEED SOME PAPER WITH PLOT
1570  PRINTER IS 16
1580  SUBEND
1590  SUB Active_filter_1(INTEGER Z(*),Observations)
1600        INTEGER Z_store(1:66),Status,A,B,C,D
1610      OUTPUT 7,1;"AC!"
1620 !  PREPARE STEADY VOLTAGE FOR INPUT
1630      FOR A=1 TO 1000
1640      OUTPUT 7,1;"AB,2,1,1,"&VAL$(Z(1))&"!"
1650      NEXT A
1660 !
1670 !   INPUT PROFILE
1680 !
1690         FOR B=1 TO Observations
1700           OUTPUT 7,1;"AB,2,1,1,"&VAL$(Z(B))&";AI,1,1,1!"
1710           ENTER 7,1 BFHS 2 NOFORMAT;Status
1720           ENTER 7,1 BFHS 2 NOFORMAT;Z(B)
1730         NEXT B
1740 !  CAPTURE POINTS DUE TO PHASE LAG
1750      FOR C=1 TO 66
1760      OUTPUT 7,1;"AB,2,1,1,"&VAL$(Z(Observations))&";AI,1,1,1!"
1770      ENTER 7,1 BFHS 2 NOFORMAT;Status
1780      ENTER 7,1 BFHS 2 NOFORMAT;Z_store(C)
1790      NEXT C
1800      SUBEND
1810  SUB Active_filter_2(INTEGER Z(*),Observations)
1820        INTEGER Z_store(1:66),Status,A,B,C,D
1830      OUTPUT 7,1;"AC!"
1840 !  PREPARE STEADY VOLTAGE FOR INPUT
1850      FOR A=1 TO 1000
1860      OUTPUT 7,1;"AB,2,2,1,"&VAL$(Z(1))&"!"
1870      NEXT A
1880 !
1890 !   INPUT PROFILE
1900 !
1910         FOR B=1 TO Observations
```

```
1920          OUTPUT 7,1;"AB,2,2,1,"&VAL$(Z(B))&";AI,1,2,1!"
1930          ENTER 7,1 BFHS 2 NOFORMAT;Status
1940          ENTER 7,1 BFHS 2 NOFORMAT;Z(B)
1950        NEXT B
1960 !  CAPTURE POINTS DUE TO PHASE LAG
1970      FOR C=1 TO 66
1980      OUTPUT 7,1;"AB,2,2,1,"&VAL$(Z(Observations))&";AI,1,2,1!"
1990      ENTER 7,1 BFHS 2 NOFORMAT;Status
2000      ENTER 7,1 BFHS 2 NOFORMAT;Z_store(C)
2010      NEXT C
2020      SUBEND
2030 SUB Active_filter_3(INTEGER Z(*),Observations)
2040        INTEGER Z_store(1:66),Status,A,B,C,D
2050      OUTPUT 7,1;"AC!"
2060 !  PREPARE STEADY VOLTAGE FOR INPUT
2070      FOR A=1 TO 1000
2080      OUTPUT 7,1;"AB,2,3,1,"&VAL$(Z(1))&"!"
2090      NEXT A
2100 !
2110 !   INPUT PROFILE
2120 !
2130            FOR B=1 TO Observations
2140              OUTPUT 7,1;"AB,2,3,1,"&VAL$(Z(B))&";AI,1,3,1!"
2150              ENTER 7,1 BFHS 2 NOFORMAT;Status
2160              ENTER 7,1 BFHS 2 NOFORMAT;Z(B)
2170            NEXT B
2180 !  CAPTURE POINTS DUE TO PHASE LAG
2190      FOR C=1 TO 66
2200      OUTPUT 7,1;"AB,2,3,1,"&VAL$(Z(Observations))&";AI,1,3,1!"
2210      ENTER 7,1 BFHS 2 NOFORMAT;Status
2220      ENTER 7,1 BFHS 2 NOFORMAT;Z_store(C)
2230      NEXT C
2240   FOR D=1 TO Observations
2250     IF D<=Observations-66 THEN Z(D)=Z(D+66)
2260     IF D>Observations-66 THEN Z(D)=Z_store(D-(Observations-66))
2270   NEXT D
2280      SUBEND
2290 SUB Warning
2300                 PRINT PAGE
2310                 PRINT "   Aircraft motion being removed."
2320                 PRINT LIN(1)
                                            "
2330 PRINT "
2340 PRINT "                                     "
2350 PRINT "                                     "
2360 PRINT "                                     "
2370 PRINT "                                     "
2380 PRINT
2390 PRINT "                                              "
2400 PRINT "                                               "
2410 PRINT "                                              "
2420 PRINT "                                               "
2430 PRINT "                                              "
2440 PRINT
2450 PRINT "

2460 PRINT "
       "
2470 PRINT "

2480 PRINT "
      "
2490 PRINT "
 "
2500  SUBEND
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NORDA Technical Note 77
2. GOVT ACCESSION NO.: AD-A098
3. RECIPIENT'S CATALOG NUMBER: 903
4. TITLE (and Subtitle): (6) Reducing Laser Profiles on a Tabletop Computer.
5. TYPE OF REPORT & PERIOD COVERED: (9) Final rept.
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): (10) Alan W./Lohanick (14) NORDA-TN-77
8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Naval Ocean Research and Development Activity
NSTL Station, Mississippi 39529
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (12) 43
11. CONTROLLING OFFICE NAME AND ADDRESS:
Naval Ocean Research and Development Activity
NSTL Station, Mississippi 39529
12. REPORT DATE: (11) Feb [redacted] 81
13. NUMBER OF PAGES: 38
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Distribution Unlimited
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
Airborne laser profiler
Sea ice surface profile
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

Standard numerical techniques for removing aircraft motion and discontinuities from airborne laser profiles are adapted to a desk-top computer. Because such a computer is much slower than a larger machine, analog active filters replace the numerical (Hamming) filters normally used. The computer programs are in BASIC, and listings are provided in an Appendix. Several examples of data editing procedures are given.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED 393443

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DAT

FILME

6