UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

RESEARCH ON URANIUM RESOURCE MODELS,

A PROGRESS REPORT:

PART V, LDIGIT--A COMPUTER PROGRAM TO DIGITIZE GRAPHICS DATA FOR SPATIAL DATA ANALYSIS

by

Robert M. Turner, William A. Scott, and Richard B. McCammon

Work done in cooperation with the U.S. Department of Energy

U.S. Geological Survey
Open-File Report 82 843

1982

CONTENTS

ILLUSTRATIONS

APPENDIX

Introduction

I Spatial data analysis is becoming more important in regional mineral resource studies as both the methods of investigation and the types of data collected during large-scale field investigations increase in number and variety. (Botbol and others, 1978; McCammon and others, 1979, 1983; Sinding-Larsen and others, 1979, 1981) Some types of analyses require integrated digital data sets formed by combining data from many different sources. This requires the ability to digitize graphic data readily and easily using a common grid. Such data include points, lines, and areas from such sources as maps, charts, profiles, or photographs. Each data set, moreover, has its own built-in resolution, making it necessary to re-sample each data set in a way that insures approximately equal representation within a specified geographic area.

LDIGIT is a computer program to digitize graphic data in a form amenable to spatial data analysis. The program is flexible in that a user can select any desired scale and sampling interval for a particular data source. The program is written in ANSI FORTRAN IV using Tektronix Distributed Graphics Support Subroutines (DGSS). It is designed for use on the Tektronix 4081 Graphics System.

The purpose of this paper is to describe both the system components and the procedure for digitizing using LDIGIT, and to provide an example using the program. This information should enable others to use the program with a minimum of delay.

II System Hardware

The 4081 System consists of components for data input, data output, and process control.

Two digitizing tables (12" x 12" or 30" x 40") are available for data input.

Materials to be digitized are placed on the table surface, and a cursor with crosshair and pushbutton is used to trace and select the desired lines or points for input.

The resulting data file, which consists of "x" and "y" coordinate pairs, together with labeling information, is loaded onto disk files for editing and further processing or subsequent transmittal elsewhere for processing and storage.

The input and output operations are controlled by a 64k byte minicomputer equipped with a Cathode Ray Tube (CRT) screen for data and graphics display and a keyboard for entering controlling instructions or supplementary data.

III System Software

General Function The LDIGIT program resides in the microprocesser memory to handle the collection of data and to provide a command system allowing the operator to control the collecting process.

The program obtains the necessary information and commands from user responses to questions displayed on the CRT. It performs user-selected calculations on the data and displays status information useful for editing and processing.

Provisions There are several provisions that the program makes for the digitizing process. Some are modifiable in response to queries

from the program (indicated by "*" in the following discussion), while others can be changed only by program modification. All have default values which can be used. The provisions are listed below in the order they appear in the program.

1) Large or Small Graphics Tablet. Currently, two separate programs (LDIGIT and SDIGIT) with separate output files are used for the large and small tablets, respectively.

2) Labeled output files on disk. An output file is opened on the disk and the rotated and transformed data are written out. The file is closed upon NORMAL termination of the program. The default disk is the user disk (USR:), and the default file names are DIGIT1.NEW (for SDIGIT) and DIGIT2.NEW (for LDIGIT). The default file size is 500 blocks (11,000 points). These values are displayed on the screen. An additional file, DIGIT2.EDT (file size 50 blocks) is created to contain a copy of the screen prompts, answers, and other data useful for later editing. These files MUST BE RENAMED after a normal closing of the program; otherwise they will be overwritten and their contents lost during the next run of the program. (This feature avoids the keeping of large unneeded files, but requires careful attention by the operator!)

3) Optional blocking of points to output file. This feature is currently disabled. It allows points to be saved in a buffer and output to disk in 256 character blocks. This reduces the space required and allows more points in a file, but causes display peculiarities. It is preferable, therefore, simply to increase the size of the output file.

4) * Time and Distance filters for accepting input data points.

Defaults: Time = 20 milliseconds

Distance = 100 port units (~ 1.0mm.)

These are used during continuous (line) recording to provide for the separation of successive points (sample interval) and to reduce the number of points to be processed and stored. The time filter is used to ignore points generated within the given interval after the preceeding stored point. The distance filter is used to ignore points generated within the given distance from the preceeding accepted point. The time and distance filters are applied in that order.

These values should be selected according to the speed of the cursor travel, the size of the tablet used, and the fineness of detail to be digitized.

5) * Window for input data. This allows the operator to select a window area by digitizing the corners and assigning minimum and maximum x and y values in the coordinate system desired (decimal latitude/longitude, U.T.M., etc.) to the lower left and upper right corners, respectively (fig. 1).

Defaults: Lower Left (x,y) = Ø.,Ø.

Upper Right (x,y) = 1ØØ.,1ØØ.

These are the absolute coordinate values to be equivalenced to the port coordinates of the window corners.

The entire CRT screen is set to display the window area; thus only points selected within the window will be displayed, and the displayed image may appear enlarged or reduced.

6) Rotation correction. This constructs x and y axes through the window corners and generates a rotation transform. This is applied to each point, and effectively rotates the x and y axes to be mutually perpendicular and to coincide with the axes of the table. This corrects for any misalignment of the input documents and makes their placement on the table noncritical.

7) Scale correction. This uses the window points and values to establish an absolute coordinate grid (fig. 1). The cursor position as returned from the table in port coordinates is converted to an absolute coordinate position on this grid. The grid is projected beyond the window and can encompass points digitized anywhere on the active table surface though they will not be displayed.

Maps of differing scales can be adjusted such that equivalent points have the same coordinate values, and coordinates can be adjusted to coincide with the cell of a sampling grid.

It is important to note that the current program constructs a grid which is a parallelogram, therefore, only rectangular projections are accurately reproduced. Trapezoidal projections, especially in higher latitudes, must be limited to small areas so that convergence is minimized.

8) Line or point labeling. There is provision for a two digit label to be attached to each point or series of points, allowing 100 different categories of lines. This label, as well as the number of points in the line, is displayed on the screen and written to the output file (fig. 2).

9) Display of coordinates and point numbers of initial points. The coordinates and point number of the initial point of each segment of a line (this includes single points) are displayed on the screen.

These are also copied to the editing file and saved to aid in editing multi-segment lines by allowing individual segments to be identified for separate correction or removal (fig. 2).

10) Display of digitized points. Digitized points which fall within the selected window are drawn on the screen to show the operator what has been done.

11) End of line/end of session characters. Each line digitized is ended by typing any character other than an asterisk. An asterisk (*) is typed to end input for the last line and to terminate the session.

Operating Procedure

Initial setup

The procedures prior to the actual digitizing are primarily those of simple "housekeeping."

The disk files are checked to make sure that the work file from a previous session has been renamed if it is to be retained. If it is not wanted, it can be overwritten; otherwise, there must be space available for a new file (500 Blocks) to be created by the program.

The table should be located where the operator can see the display screen and reach the keyboard with a minimum of effort. The table itself should be checked for proper bias. The table locates the cursor position by means of magnetostrictive pulses along a grid of wires. The bias field of this grid is easily distorted by metal or magnetic objects placed upon it. The bias is easily checked by running the digitizing program. The operator sets the window to include the entire table and uses the cursor

with a wood or plastic straight-edge to draw a cross-hatched pattern of lines while watching the screen. Any distortion will show as offsets or "spikes" on the displayed lines. When the program is reinitialized, it will destroy the test file in the process of digitizing the desired data.

The materials to be digitized are placed on the table surface (avoiding the electrically "dead" areas along the edges of the table) and taped to prevent shifting. The following conventions are used in the discussion below:

==========	= computer prompt or message
[ ]	= response typed by operator
<cr>	= carriage return (understood to terminate each line of input)
nn	= any number

Starting the program

[ipl sys:ipldgs] The digitizing session is started by loading the System Controller program.

GOS Version 4 Level 02 (ipldgs)

#[char 2] Select character size desired (default = 3)

#[run ldigit] Load and start the digitizing program. The program commences a dialogue to establish the operating parameters for the session, as follows:

Output file name = usr:digit2.new (file is on user disk)

file size = 500 (blocks) (approximately 11,000 points)

edit file name = usr:digit2.edt

file size = 50 (blocks)

RENAME File after each close!!! (warning to avoid data loss)

USE DECIMAL POINTS with the following parameters (ie 20.)	To be certain that values are properly justified.
Enter TIME FILTER.(Default = 20) [nn.<cr>] or <cr>	
Enter DISTANCE FILTER.(Default = 100.) [nnn.<cr>] or <cr>	
Time = 20 Dist = 100	The values to be used (defaults if others were not specified) are displayed.
Select Window Coordinates: Lower Left (L.L.) x y; Upper Right (U.R.) x y To select Default values, hit "cr"	Routine to enter the absolute coordinate values to be assigned to the window corners.
Default Window = Ø.,Ø.,100.,100. Enter L.L. x coord = [nn.n<cr>] or <cr> Enter L.L. y coord = [nn.n<cr>] or <cr> Enter U.R. x coord = [nn.n<cr>] or <cr> Enter U.R. y coord = [nn.n <cr> or <cr>	
Window = 0.00000 0.00000 100.00 100.00	The values to be used are displayed. The operator now uses the table cursor to digitize the window corners as requested.

digitize lower left corner

The first (minimum) absolute x and y window values selected above are equivalenced to this point.

-nnnnn -nnnnn FFT

The unprocessed port coordinates of the point and the setting of three cursor logic switches are listed for each corner.

digitize upper right corner

The second (maximum) absolute x & y window values are equivalenced to this point.

nnnnn nnnnn FFT

digitize lower right corner

nnnnn -nnnnn FFT

The program now calculates the conversion, scaling, and rotation factors and displays the absolute coordinates of the window center point (invariant point).

invar pt = nn.nnn nn.nnn

This prompt signals the end of the parameter entry phase.

The digitizing process

The program signals the start of the digitizing phase by prompting for the two digit identification number for the next line to be digitized.

enter 2 character Line I.D.

[Ø1 <cr>]

Entering the <cr> activates the table for cursor input. The operator now traces the desired map features. Points traced while the cursor button is pressed are transmitted, processed, and displayed on the screen. In addition, the absolute coordinates and a

point number are displayed for the initial point of each segment. If digitizing is momentarily interrupted (by releasing the button or lifting the cursor), the next point digitized is considered to be an initial point. A long line can thus be separated into identifiable segments by interrupting to record coordinates at identifiable locations (Figs. 2 and 3).

[n] (& lift cursor)

Input for a line is terminated by typing any alpha character except an asterisk (*) and lifting the cursor from the table.

As each line is terminated, a final record is written at the end of the coordinate list. This record consists of a flag (two alpha characters; currently "aa" in columns 1 and 2), the identifier assigned to the line, and the total number of points recorded for the line (Fig. 3).

nn nnn

These numbers are also displayed.

Enter 2 character Line I.D.

The program then prompts for the next line identification.

[nn <cr>]

File closing and handling

[*] (& lift cursor)

nn nnn

End of linefile

Exit

The session is ended by typing an asterisk (*) to terminate the last line. The program then closes all files, acknowledges by a prompt, and ends.

__RENAME files before restarting program.__

PROGRAM LISTING

Page 1

```
.ine No.
    1               program ldigit
    2               implicit integer*2 (a-z)
    3               integer*1 char
    4               dimension p(2),screen(2),ixy(2),ivxy(2)
    5               real inw(6),outw(4),t(2,3),pos(2),scrnxy(2)
    6               logical s(3),out,away,down
    7               equivalence (s(1),out),(s(2),away),(s(3),down)
    8               logical avail,switch
    9           c initialize output file
   10                 call bfnew (2,'usr:digit2.new ',500)
   11           c initialize graphics
   12                 call bfdev (15, 'dc: ')
   13                 call gonew (15)
   14           c
   15           c
   16           c set up points buffer, if desired.
   17           c (.t. = yes; .f. = no)
   18           c
   19                 call buff (.f.)
   20           c
   21           c
   22           c initialize locator GIN
   23                 call bfdev (14, 'tb1: ')
   24                 call giloc (14)
   25           c
   26           c output data file parameters
   27           c
   28           14    format (i3)
   29                 write (1,*)'output file name   =  usr:digit2.new'
   30                 write (1,*)'       file size   =  500 (blocks)'
   31                 write (1,*)'RENAME File after each close !!!'
   32           c
   33           c Initialize values
   34           c
   35                 outw(1)=0.
   36                 outw(2)=0.
   37                 outw(3)=0.
   38                 outw(4)=0.
   39                 time=0
   40                 dist=0
   41           c
   42                 write (1,*)'USE DECIMAL POINTS with '
   43                 write (1,*)'the Following Parameters  (ie 20. )'
   44           c Query for Filters
   45           c
   46                 write (1,*)'Enter TIME FILTER..(Default=20.)'
   47                 read (5,14)time
   48                 write (1,*)'Enter DISTANCE FILTER..(Default=100.)'
   49                 read (5,14)dist
   50                 if (time.ne.0)go to 17
```

```
                time=20
          17    write (1,*)'Time =',time
                if (dist.ne.0)go to 18
                dist=100
          18    write (1,*)'Dist =',dist
          c
          c Query for window
          c
          15    format (f5.2)
                write (1,*)'Select Window Coordinates:'
                write (1,*)'Lower Left (L.L.) X & Y ; Upper Right (U.R.) X &
                write (1,*)'To select Default values, hit "CR"'
                write (1,*)'Default Window = 0. , 0. ; 100. , 100. '
          c
                write (1,*)'Enter L.L. X coord. = '
                read (5,15)outw(1)
                write (1,*)'Enter L.L. Y coord. = '
                read (5,15)outw(2)
                write (1,*)'Enter U.R. X coord. = '
                read (5,15)outw(3)
                write (1,*)'Enter U.R. Y coord. = '
                read (5,15)outw(4)
          c
          c get window
          c
                if (outw(1).ne.0)go to 21
                outw(1)=0.
          21    if (outw(2).ne.0)go to 22
                outw(2)=0.
          22    if (outw(3).ne.0)go to 23
                outw(3)=100.
          23    if (outw(4).ne.0)go to 24
                outw(4)=100.
          24    continue
                write (1,*)'window = ',outw(1),outw(2),outw(3),outw(4)
                call wind (outw)
          c
          c
          c get map location (for skew corr.)
          c
          6     format (10a4)
          7     format (i10,2x,i10)
          11    write (1,*)'digitize lower left corner'
                call qgiloc (p,s)
                write (1,*)p,s
                call p2w (p,inw(1))
          12    write (1,*)'digitize upper right corner'
                call qgiloc (p,s)
                write (1,*)p,s
                call p2w (p,inw(3))
```

```
101          13     write (1,*)'digitize lower right corner'
102                 call qgiloc (p,s)
103                 write (1,*)p,s
104                 call p2w (p,inw(5))
105                 pos(1)=(outw(1)+outw(3))/2
106                 pos(2)=(outw(2)+outw(4))/2
107                 write (1,*)'invar pt = ',pos(1),pos(2)
108                 call ipos (pos)
109          c
110          c get unskew transform
111          c
112                 call unskew (t,inw,outw)
113          c
114          c
115          c --- LOOP POINT -- for Line routine --
116          c --- (Line may contain several segments)
117          c --- (initial point coordinates and point number
118          c ---  of each line segment are written to the screen.)
119          c
120          5      write (6,*)'enter 2 character Line I.D.'
121                 read (5,3) lineid
122          3      format (i2)
123                 count=0
124          c trace drawing
125                 call gitra (14,10)
126          c
127          c
128                 call gitif (time)
129                 call gidif (dist)
130                 call giact
131          c
132          c get FIRST point - for MOVE
133          c
134          10     call qgitra (p,s)
135                 call cnget (char,avail)
136                 if (avail) go to 40
137                 call p2w (p,outw)
138                 call apply (t,outw(1),outw(3))
139          c
140          c
141          c Check if pen Down
142                 if (.not.s(3)) go to 10
143          c
144                 count=count+1
145                 write (1,*)outw(3),outw(4),count
146                 call wmove (outw(3))
147                 write (2,*)outw(3),outw(4)
148          c
149          c get NEXT points - for DRAW
150          c
```

```
151          20     call qgitra (p,s)
152                 call cnget (char,avail)
153                 if (avail) go to 40
154                 if (.not.s(3))go to 10
155                 call p2w (p,outw)
156                 call apply (t,outw(1),outw(3))
157          c
158          c
159                 call wdraw (outw(3))
160                 count=count+1
161                 write (2,*)outw(3),outw(4)
162          c
163          c check for line/record end
164          c
165          c check for keyboard input
166          c
167                 call cnget (char,avail)
168                 if (avail) go to 40
169                 go to 20
170          c
171          c --- end of line routine ---
172          c
173          40     write (1,*)char
174                 if (char.eq.42) go to 45
175                 write (2,4) lineid,count
176          4      format ('aa',2x,i2,2x,i10)
177                 write (1,*)lineid,count
178                 call giclo
179                 go to 5
180          c
181          c --- end of file routine ---
182          c
183          45     write (2,4) lineid,count
184                 write (1,*) lineid,count
185                 write (1,*)'end of linefile'
186                 call giclo
187                 call goclo
188                 call bfclo (2)
189          c
190          c skip to end
191          c
192                 go to 99
193          c
194          c
195          c
196          c Create cross for GIN cursor
197          c
198          49     call open (100)
199                 ixy(1)=0
200                 ixy(2)=1
```

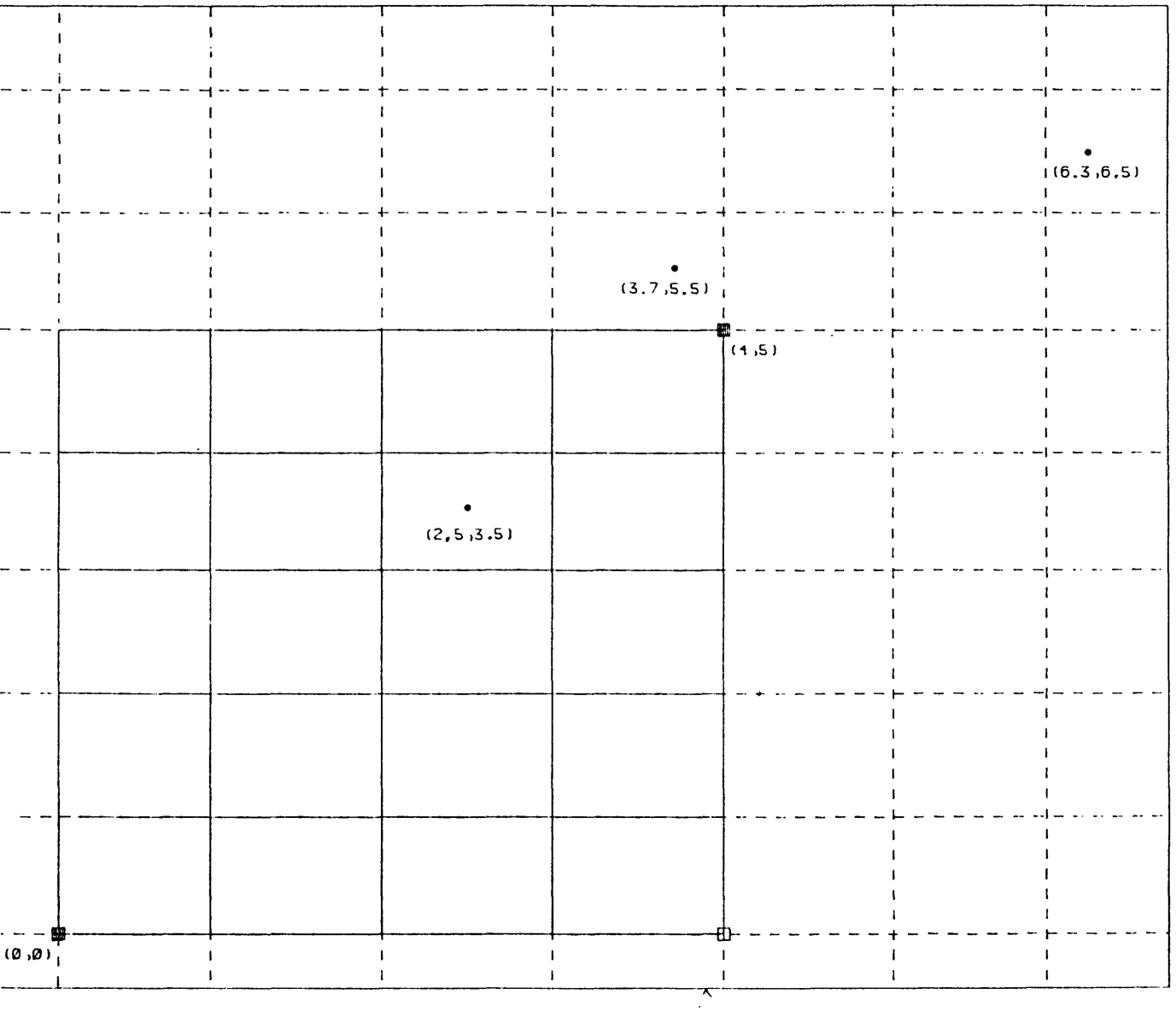
```
201          50     call vmove (ixy)
202          51     call vabs (.f.)
203                 ixy(1)=8000
204                 call vmove (ixy)
205                 ixy(1)=-16000
206                 call vdraw (ixy)
207                 ixy(1)=8000
208                 ixy(2)=8000
209                 call vmove (ixy)
210                 ixy(1)=0
211                 ixy(2)=-16000
212                 call vdraw (ixy)
213                 call vabs (.t.)
214                 call close
215                 call post (100)
216          c
217          c return to program
218          c
219          c      go to 52
220          c
221          99     continue
222                 end
```

-------- LIST OF PROGRAM LABELS ----------

AVAIL	Logical Variable
AWAY	Logical Variable
CHAR	Integer*1 Variable
COUNT	Integer*2 Variable
DIST	Integer*2 Variable
DOWN	Logical Variable
INW	Real Variable Array (6)
IVXY	Integer*2 Variable Array (2)
IXY	Integer*2 Variable Array (2)
LINEID	Integer*2 Variable
OUT	Logical Variable
OUTW	Real Variable Array (4)
P	Integer*2 Variable Array (2)
POS	Real Variable Array (2)
S	Logical Variable Array (3)
SCREEN	Integer*2 Variable Array (2)
SCRNXY	Real Variable Array (2)
SWITCH	Logical Variable
T	Real Variable Array (2,3)
TIME	Integer*2 Variable
10	Statement Label
11	Statement Label
12	Statement Label
13	Statement Label
17	Statement Label
18	Statement Label
20	Statement Label
21	Statement Label
22	Statement Label
23	Statement Label
24	Statement Label
40	Statement Label
45	Statement Label
49	Statement Label
5	Statement Label
50	Statement Label
51	Statement Label
99	Statement Label
14	Format Label
15	Format Label
3	Format Label
4	Format Label
6	Format Label
7	Format Label
F	External Subroutine
H	External Subroutine
%U	External Subroutine
%V	External Subroutine
APPLY	External Subroutine

BFCLO	External Subroutine
BFDEV	External Subroutine
BFNEW	External Subroutine
BUFF	External Subroutine
CLOSE	External Subroutine
CNGET	External Subroutine
GIACT	External Subroutine
GICLO	External Subroutine
GIDIF	External Subroutine
GILOC	External Subroutine
GITIF	External Subroutine
GITRA	External Subroutine
GOCLO	External Subroutine
GONEW	External Subroutine
IPOS	External Subroutine
OPEN	External Subroutine
P2W	External Subroutine
POST	External Subroutine
QGILOC	External Subroutine
QGITRA	External Subroutine
UNSKEW	External Subroutine
VABS	External Subroutine
VDRAW	External Subroutine
VMOVE	External Subroutine
WDRAW	External Subroutine
WIND	External Subroutine
WMOVE	External Subroutine

Figure 1



• Digitized Point.

□ Window Corner.

■ Window Corner designated as minimum / maximum point.

—— Grid constructed inside window.

– – Grid projected beyond window.

(,) Coordinates of a Point.

View of table surface, showing how selection of window points and assignment of values to them creates a grid and projects it over the entire surface. The coordinates of all points are recorded, but only points within the window are displayed.

```
output file name    =  usr:digit2.new
      file size     =  5000 (blocks)
editing file name   =  usr:digit2.edt
      file size     =  50 (blocks)
RENAME Files after each close !!!
USE DECIMAL POINTS with
the Following Parameters  (ie 20. )
Enter TIME FILTER..(Default=20.)
Enter DISTANCE FILTER..(Default=100.)
Time =     20
Dist =    100
Select Window Coordinates:
Lower Left (L.L.) X & Y ; Upper Right (U.R.) X & Y
To select Default values, hit "CR"
Default Window = 0.  , 0. ; 100.  , 100.
Enter L.L. X coord. =
Enter L.L. Y coord. =
Enter U.R. X coord. =
Enter U.R. Y coord. =
window =   0.000000  0.000000   100.00   100.00
digitize lower left corner
-21509 -14438 F F T
digitize upper right corner
  4883   3344 F F T
digitize lower right corner
  4773 -14304 F F T
inver pi =      50.000    50.000
  55.800    37.573         1
  54.521    55.436        27
  67.086    55.807        54
  68.349    38.158        85
  52.806    27.117       110
 48
     1       138
  78.503    52.907         1
 48
     2        32
  54.535    43.728         1
 48
     3        38
  68.178    44.548         1
 42
     4        28
end of linefile
#
```

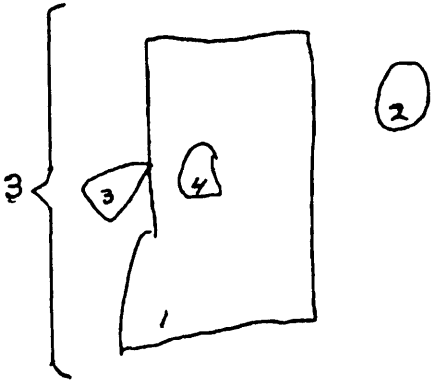


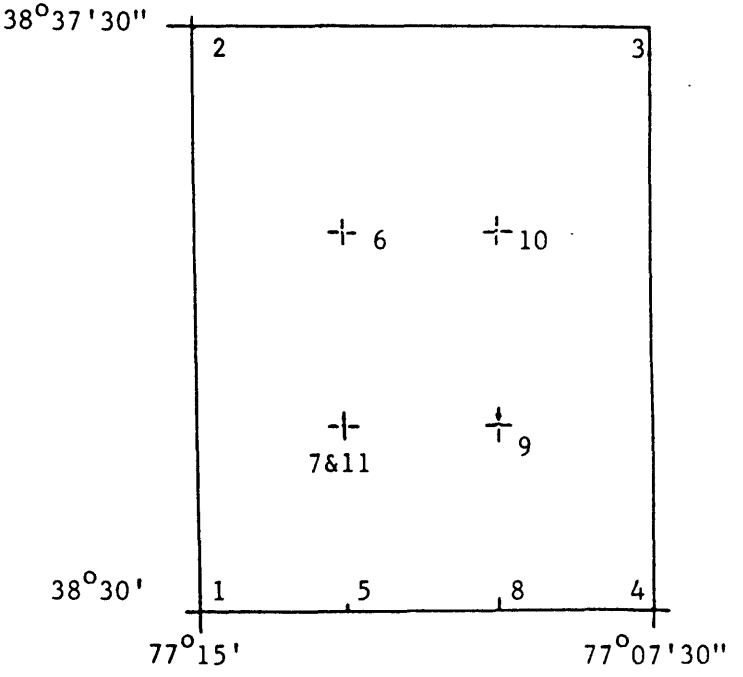
Figure 2. Screen display during digitizing session. Shown are:
1) Session parameters, questions, and responses.
2) Coordinates of the initial points of each line segment.
3) Trace of all lines drawn within the chosen window area.

```
type digit2.new
   56.899    37.573    68.205    53.485    54.848    27.857    76.266    53.053    59.786    44.208
   56.898    38.230    68.323    52.656    63.658    27.773 aa  2 32        59.785    44.789
   56.841    38.819    68.382    50.997    53.101    27.606    54.535    43.720 aa  1 28
   54.987    39.569    68.383    49.999    52.053    27.778    54.255    43.148  *
   54.819    40.220    68.384    49.882    52.200    27.613    53.922    42.568
   54.818    40.979    68.385    48.884    52.906    27.117    53.642    41.988
   54.761    41.568    68.338    47.901    52.906    27.777    53.389    41.488
   54.783    42.225    68.274    46.893    52.838    28.383    52.919    40.828
   54.783    42.886    68.222    45.188    52.837    29.023    52.625    40.320
   54.845    43.387    68.166    44.267    52.894    29.689    52.135    39.824
   54.844    43.968    68.167    43.353    52.893    30.198    51.745    39.324
   54.844    44.554    68.168    42.448    52.202    30.934    51.355    38.992
   54.843    45.213    68.169    41.442    52.316    31.599    50.984    39.162
   54.842    45.903    68.113    40.529    52.424    32.179    50.573    39.588
   54.841    46.877    68.114    39.616    52.534    32.788    50.125    40.077
   54.840    47.874    68.115    38.618    52.647    33.425    49.734    40.659
   54.586    48.783    68.117    37.541    52.756    34.085    49.286    41.326
   54.585    49.786    68.118    36.543    52.869    34.591    49.084    41.987
   54.584    50.366    68.119    35.638    53.035    35.250    49.585    42.655
   54.583    50.947    68.177    34.886    53.201    35.836    50.088    42.818
   54.583    51.528    68.178    34.851    53.367    36.416    50.455    43.085
   54.582    52.188    68.178    33.222    53.591    36.996    50.982    43.317
   54.524    53.191    68.179    32.642    53.814    37.577    51.293    43.488
   54.523    53.772    68.180    32.061    54.261    37.903    51.748    43.642
   54.522    54.085    68.238    31.311 aa  1 138     37.992    52.138    43.810
   54.522    55.266    68.291    30.738    76.593    52.987    52.521    43.973
   54.521    55.435    68.349    30.158    76.198    53.137    52.915    44.056
   55.879    55.433    68.349    30.158    75.698    53.138    53.386    44.148
   56.469    55.596    67.958    29.981    75.251    53.055    53.696    44.139
   56.917    55.595    67.454    29.983    74.868    52.723    54.087    44.138
   56.421    55.593    67.804    29.984    74.624    52.143    54.477    44.052
   56.868    55.588    66.673    29.985    74.247    51.584 aa  3 38      54.868    43.967
   57.259    55.428    66.282    29.986    73.967    50.984    58.176    44.548
   57.786    55.427    65.836    29.988    73.881    50.319    59.952    45.121
   58.154    56.426    65.445    29.989    73.688    49.738    59.617    45.783
   58.712    55.348    64.716    29.989    73.636    49.158    59.223    45.867
   59.269    55.254    64.269    29.991    73.698    48.571    58.832    45.868
   59.717    55.168    63.825    29.992    73.884    47.990    58.385    45.786
   60.274    55.251    63.431    29.914    74.138    47.489    57.995    45.374
   60.888    55.258    62.987    29.915    74.529    47.168    57.662    44.718
   61.449    55.333    62.548    29.747    75.091    46.995    57.439    44.138
   61.958    55.332    62.893    29.663    75.481    47.073    57.273    43.485
   62.397    55.416    61.645    29.581    75.928    47.156    57.187    42.885
   62.844    55.493    61.198    29.417    76.376    47.319    57.187    42.219
   63.349    55.578    60.751    29.334    76.766    47.487    57.218    41.638
   63.966    55.738    60.193    29.257    77.156    47.819    57.778    41.058
   64.577    55.821    59.689    29.089    77.683    48.314    58.337    41.134
   65.248    55.984    59.189    29.005    77.826    48.978    58.838    41.133
   65.915    55.983    58.575    29.007    77.935    49.559    59.285    41.218
   66.587    55.985    57.981    28.845    77.992    50.139    59.675    41.388
   67.281    55.984    57.399    28.762    77.991    50.804    60.066    41.299
   67.591    55.898    56.786    28.688    77.933    51.385    60.468    41.298
   67.986    55.897    56.228    28.517    77.823    51.966    60.468    41.878
   67.986    55.897    55.614    28.278    77.541    52.553    60.235    42.468
   67.938    55.317    55.053    28.823    77.158    52.988    59.811    43.041
   68.897    54.482    54.496    27.948    76.768    53.136    59.843    43.622
```

Figure 3. Contents of the output data file (digit2.new). The two character alpha "flag" (aa) identifies the label line for the editing program.

```
output file name = usr:digit2.new
       file size = 500 (blocks)
editing file name = usr:digit2.edt
       file size = 50 (blocks)
RENAME files after each close !!
USE DECIMAL POINTS with
the Following Parameters (ie 20. )
Enter TIME FILTER. . (Default=20.)
20.
Enter DISTANCE FILTER. . (Default=100.)
100.
Time =  20
Dist = 100
Select Window Coordinates:
Lower Left (L.L.) X & Y ; Upper Right (U.R.)-X & Y
To select Default values, hit "CR"
Enter L.L. X coord. =-77.25
Enter L.L. Y coord. =38.5
Enter U.R. X coord. =-77.125
Enter U.R. Y coord. =38.625
window = -77.250 38.500 -77.125 38.625
digitize lower left corner
-21186 -12906 F F T
digitize upper right corner
  4862  20714 F F T
digitize lower right corner
  5127 -12744 F F T
invar pt =  -77.188  38.563
enter 4 character lineid
0
-77.250    38.500      1
-77.250    38.625      2
-77.125    38.625      3
-77.125    38.500      4
-77.208    38.500      5
-77.208    38.583      6
-77.208    38.542      7
-77.167    38.500      8
-77.167    38.542      9
-77.167    38.583     10
-77.208    38.542     11
42
    0     11
end of linefile
```

a) display of prompts and answers, point coordinates in decimal degrees, and point numbers.



b) diagram of map area showing number and location of each digitized point.

Figure 4. Example of digitizing session using decimal latitude and longitude.

- The area used is the Indian Head Md.-Va. 7 1/2' Quadrangle.
- The numbers on the diagram are those of the points digitized on the 2' 30" hatchures.
- West Longitudes (and South Latitudes) are assigned negative prefixes.

```
          3.366     58.023
          3.403     57.576
          3.849     57.480
          4.294     57.829
          4.391     58.275
          4.170     58.719
          4.267     59.163
          4.711     59.195
          5.156     59.034
          4.956     59.479
          4.324     59.862
          4.061     60.022
          3.635     59.609
          3.256     60.056
          2.810     60.120
          2.366     60.058
          1.922     60.026
          1.477     59.805
          1.349     59.359
          1.791     59.609
          2.234     58.881
          2.679     58.563
          3.026     58.118
**   1              25
          1.024     58.304
          1.025     58.526
          0.994     58.748
          1.185     58.526
          1.151     58.303
**   1               5
          1.036     70.093
          1.229     70.285
          1.523     70.062
          1.195     69.840
          1.067     69.616
          1.005     69.841
          0.774     70.063
**   1               7
```

```
1   25
    3.366     58.023     3.403     57.576     3.849     57.480     4.294     57.829     4.391     58.275
    4.170     58.719     4.267     59.163     4.711     59.195     5.156     59.034     4.956     59.479
    4.324     59.862     4.061     60.022     3.635     59.609     3.256     60.056     2.810     60.120
    2.366     60.058     1.922     60.026     1.477     59.805     1.349     59.359     1.791     59.609
    2.234      .881      2.679     58.563     3.026     58.118
1    5
    1.024     58.304     1.025     58.526     0.994     58.748     1.185     58.526     1.151     58.303
1    7
    1.036     70.093     1.229     70.285     1.523     70.062     1.195     69.840     1.067     69.616
    1.005     69.841     0.774     70.063
```

Figure 5. Examples of data as output from the LDIGIT program and as reformatted by TEKM.

APPENDIX

TEKM program for data file reformatting. TEKM is an interactive program written in PL1 to run on the Honeywell MULTICS computer system. This program reads the output data file from the LDIGIT digitizing program and reformats the data for input to the current version of the graphics cell program (Hanley, 1982). In this application, each digitized line defines a closed polygon to be used in creating graphic cells.

The LDIGIT data file consists of records of two format types (Fig. 5).

Type 1: Format = 2F11.3

Contents = x and y coordinate pair

A type 1 record is generated for each point selected during digitizing. A line (polygon) contains a varying number of records.

Type 2: Format = A2, 2X, I2, 2X, I10

Contents = "aa" (flag), Line ID, Number of x-y pairs

A type 2 record is generated whenever a line (series of points) is terminated during digitizing. It contains a "flag" to identify the record type, the line identifier entered by the operator, and the number of x-y pairs (Type 1 records) composing the line. It is the last record for each line. TEKM reads this file, creating temporary buffers for checking correct coordinates and for making counters.

This data is re-formatted and three different records are written to an output file - created as an input file for the grid cell construction program (Fig. 5).

Record 1: format - i4

number of polygons in file

This record is written only once and always as the FIRST record of the file.

Record 2: format - 2i4

formation code,

number of vertices in this polygon.

This record is written as the first record of each of the polygons in the file.

Record 3: format - 20f11.3

x-y coordinate pairs (max: 10 pairs per line)

A sample TEKM session follows:

```
tekm
Enter name of 4081 input file:  wcnk.dat
Enter unique name for MULTICS output file:  mfile
tekm began
r 10:32 4.131 283
```

```
Line Number
     1              (stringrange):
     2              (subscriptrange):
     3     tekm:
     4              proc;
     5              dcl (sysin,sysprint) file;
     6              dcl (tekf)    file;
     7              dcl (cvtfile) file;
     8              dcl (countf)  file;
     9              dcl (multf)   file;
    10              dcl tkname  char (16)  varying init ("");
    11              dcl mtname  char (16)  varying init ("");
    12              dcl tekr    char (80)  varying init ("");
    13              dcl aatest  char (4)   varying init ("");
    14              dcl tfield  char (24)  varying init ("");
    15              dcl xfield  char (24)  varying init ("");
    16              dcl yfield  char (24)  varying init ("");
    17              dcl xyfield char (48)  varying init ("");
    18              dcl fvfield char (24)  varying init ("");
    19              dcl fcode   char (8)   varying init ("");
    20              dcl vnum    char (8)   varying init ("");
    21              dcl fvline  char (16)  varying init ("");
    22              dcl xyline  char (280) varying init ("");
    23              dcl (achar,lchar,recnum,polynum) fixed bin;
    24              dcl (pflag,fvflag,aaflag,teflag) fixed bin;
    25              dcl rename entry (char (*), char (*));
    26              dcl delete entry options (variable);
    27              dcl new_proc entry;
    28              dcl quit    cond;
    29              dcl endfile cond;
    30              dcl error   cond;
    31              on error begin;
    32              put list ("error detected") skip;
    33              close file (tekf);
    34              close file (cvtfile);
    35              close file (multf);
    36              go to endtm;
    37              end;
    38              on quit begin;
    39              call new_proc;
    40              end;
    41              on endfile (tekf)    go to tekend;
    42              on endfile (cvtfile) go to tend;
    43              on endfile (countf)  go to tend;
    44              achar,lchar,recnum,polynum = 0;
    45              tekr,aatest,tkname,mtname,yfield,xyfield,xyline,tfield = "";
```

```
                ENTER TEKTRONIX INPUT FILE NAME

46      entertm:
47              put list ("Enter name of 4081 input file:  ") skip;
48              get list (tkname);
49              if tkname = "" then go to entertm;

                ENTER MULTICS OUTPUT FILE NAME

50      enterm:
51              put list ("Enter unique name for Multics output file:  ") skip;
52              get list (mtname);
53              if mtname = "" then go to enterm;
54              put list ("tekm began") skip;
55              put list ("") skip;
56              call rename (tkname, "tekf");

                OPEN INPUT/OUTPUT FILES

57              open file (tekf) environment (stringvalue) record input
58              title ("record_stream_ -target vfile_ tekf");
59              open file (cvtfile) stream output;
60              open file (countf)  stream output;
61              polynum,fvflag,recnum = 0;

                READ INPUT FILE UNTIL END OF POLYGON IS REACHED

62      rdrec:
63              read file (tekf) into (tekr);
64              if tekr = "" then go to rdrec;
65              recnum = recnum + 1;
66              aatest = (substr (tekr,1,2));
67              if aatest ^= "aa" then go to sameln;
68              polynum = polynum + 1;
69              put file (cvtfile) edit (tekr) (a);
70              put file (cvtfile) skip (1) edit ("") (a);
71              put file (countf) edit (tekr) (a);
72              put file (countf) skip (1) edit ("") (a);
73              recnum = 0;
74              go to rdrec;
75      sameln:
76              fvflag = 1;
77              achar  = 3;
78              lchar  = 11;
79      rdfield:
80              tfield = (substr (tekr,achar,lchar));
81              if achar ^= 3 then go to yjump;
82              xfield = tfield;
83              tfield = "";
84              achar  = 14;
85              go to rdfield;
86      yjump:
87              yfield  = tfield;
88              xyfield = xfield || yfield;
```

```
                STORE DATA IN A TEMPORARY CONVERT FILE OF 10 X-Y PAIRS

 89             put file (cvtfile) edit (xyfield) (a);
 90             put file (cvtfile) skip (1) edit ("") (a);
 91             if recnum ^= 10 then go to rdrec;
 92             recnum = 0;
 93             put file (cvtfile) edit ("TEN") (a);
 94             put file (cvtfile) skip (1) edit ("") (a);
 95             go to rdrec;
 96     tekend:
 97             put file (cvtfile) skip (1) edit ("") (a);

                CLOSE TEMPORARY FILES

 98             close file (cvtfile);
 99             close file (countf);

                OPEN TEMPORARY  FILES AS INPUT

100     tjmp:
101             open file (cvtfile) environment (stringvalue) record input
102             title ("record_stream_ -target vfile_ cvtfile");
103             open file (countf) environment (stringvalue) record input
104             title ("record_stream_ -target vfile_ countf");

                OPEN OUTPUT FILE

105             open file (multf) output linesize (688);
106             pflag,aaflag,teflag,fvflag = 0;
107             xyfield,aatest,xyline,fvfield,fcode,vnum,fvline = "";

                READ X-Y PAIRS

108     rdcvt:
109             read file (cvtfile) into (xyfield);
110             if xyfield = "" then go to rdcvt;
111             aatest = (substr (xyfield,1,2));
112             aaflag, teflag = 0;
113             if aatest = "aa" then aaflag = 1;
114             if aatest = "aa" then fvflag = 0;
115             if aatest = "TE" then teflag = 1;
116             if (aaflag = 1 | teflag = 1) then go to writln;
117             xyline = xyline || xyfield;
118             go to rdcvt;
119     writln:
120             if pflag ^= 0 then go to rdcount;
121             put file (multf) edit (substr (polynum,7,4)) (a);
122             pflag = 1;
123     rdcount:
124             if fvflag ^= 0 then go to skhdr;
125             read file (countf) into (fvfield);
126             if fvfield = "" then go to rdcount;
127             fcode = (substr (fvfield,3,4));
128             vnum = (substr (fvfield,15,4));
129             fvline = fcode || vnum;
130             if xyline = "" then go to blankl;
```

```
                WRITE DATA TO MULTICS FILE

131             if teflag = 0 then put file (multf) skip (1) edit (xyline) (a);
132     blankl:
133             if teflag = 0 then put file (multf) skip (1) edit (fvline) (a);
134             else put file (multf) skip (1) edit (fvline) (a);
135             fvflag = 1;
136     skhdr:
137             if teflag ^= 0 then put file (multf) skip (1) edit (xyline) (a)
138             xyfield, xyline = "";
139             go to rdcvt;

                CLOSE FILES

140     tend:
141             close file  (tekf);
142             call rename ("tekf", tkname);
143             put file (multf) skip (1) edit (xyline) (a);
144             put file (multf) skip (1) edit ("") (a);
145             close file (multf);
146             close file (cvtfile);
147             close file (countf);

                RENAME OUTPUT FILE TO USER GIVEN NAME

148             call rename ("multf", mtname);

                DELETE TEMPORARY FILES

149             call delete ("countf");
150     endtm:
151             call delete ("cvtfile");
152             end tekm;
```

REFERENCES CITED

Botbol, J.M., Sinding-Larsen, R., McCammon, R.B., and Gott, G.B., 1978, A regionalized multivariate approach to target selection in geochemical exploration: Econ. Geol., v. 73, no. 4, p. 534-546.

Hanley, J.T., 1982, The graphic cell, a new look at digitizing geologic maps, Computers and Geosciences, in press.

McCammon, R.B., Botbol, J.M., McCarthy, Jr., J.H., and Gott, G.B., 1979, Drill-site favorability for concealed porphyry copper prospect, Rowe Canyon, Nevada based on characteristic analysis of geochemical anomalies: Soc. Min. Engr. Fall Mtg., Preprint No. 79-311, 7 p.

McCammon, R.B., Botbol, J.M., Sinding-Larsen, R., and Bowen, R.W., 1983, Characteristic Analysis-1981. Final Program and a possible Discovery: Math. Geology, in press.

Sinding-Larsen, R., Botbol, J.M., and McCammon, R.B., 1979, Use of weighted characteristic analysis as a tool in resource assessment: Eval. of Uranium Res. Proc. Advis. Grp. Mtg., Internat. Atomic Energy Agency, Vienna, p. 275-285.

Sinding-Larsen, R. and Strand, G., 1981, Quantitative integration of mineral exploration data. A case duty of the Grong mining region of Norway: Episodes, v. 1981, no. 1, p. 9-12.