1984/47. Use of the Department of Mines digitiser

R.G. Richardson

Abstract

The digitiser is used for converting areal data to computer readable format. Once converted the data may be used for plotting with changed scale, calculation of perimeters or areas and any other calculation implemented. Software is described for performing deskewing and scaling to user co-ordinates and an example of a program for calculating perimeters and areas is given.

INTRODUCTION

The Department of Mines has a Complot 7048 digitiser with a tablet dimension of 36" by 48" and a resolution of 0.001". Data is output from the digitiser to the computer in blocks containing four digitised points to reduce overheads. Subroutine DIGTRD (Appendix 2) provides a simple user interface providing one cursor button number and one pair of X and Y values each time it is called.

USING THE PROGRAMS

Perimeter and area measurement

The program is run by typing PLANIM. The digitiser should at this stage be put in point mode.

The program then requests:

MEASURE OR END - M to make a measurement
- E to end the run

The co-ordinates of three digitised points, the first two on the same horizontal (user) line and the third not on the same line.

The figure to be measured is then digitised using either point mode or switch-stream incremental. Any button except * may be used. When the figure has been closed ensure the digitiser is in point mode and push the * button once. The area and perimeter will then be calculated in user units and printed on the terminal.

The user may then make another measurement or finish the run. If another measurement is to be made on the same plan without moving the plan on the digitiser, scaling is not required for the second and subsequent measurements.

Co-ordinate input

To read one digitiser co-ordinate call DIGTRD (IBTN, X, Y), where IBTN is the button number pushed on the cursor and X and Y are the digitiser co-ordinates in inches. Normally X and Y will need to be deskewed and scaled to user co-ordinates. IBTN has values -4 and 0 to 11 inclusive. A value of 10 corresponds to * and 11 to #. If IBTN = -4 the digitiser is operating in stream mode.

THE PROGRAMS

PLANIM (Appendix 1)

The program performs three point scaling and deskewing of the plan. The perimeter is calculated by summing the distance between co-ordinate pairs and the area is calculated using the formula given on lines 118 and 119 of the program. Scaling is performed by digitising two known points on a horizontal (user) line and a third point not on the same line. If the first point has digitiser co-ordinates XO, YO and user co-ordinates XOMAP, YOMAP, the second point has digitser co-ordinates XC, YC and user co-ordinates XCMAP, and the third co-ordinate has digitiser co-ordinates XV, YV and user co-ordinate YVMAP, then the X scale is ABS (XCMAP - XOMAP)/ SQRT ((XC-XO) **2+ (YC-YO) **2) the skew is θ = ARCTAN (YC-YO/XC-XO).

For the third point if XD = XV-XO
and YD = YV-YO
then the Y distance in digitiser units after deskewing is
YR = -XD * SIN (θ) + YD * COS (θ)
and the Y scale is ABS ((YVMAP - YOMAP)/YR).

Any point having digitiser co-ordinates XD, YD may be converted to user co-ordinates as follows:

```
XD = XD-XO )
YD = YD-YO )  Subtract digitiser off-set

XR = XD * COS (θ) + YD * SIN (θ)   )
YR = -XD * SIN (θ) + YD * COS (θ)  )  Deskew

XMAP = XR + XOMAP )
YMAP = YR + YOMAP )  Add user off-set
```

DIGTRD (Appendix 2)

This subroutine reads one block of four sets of data from the digitiser and passes the values to the user one set at a time. In the event of the digitiser filling a block with dummy data only the real data values are returned to the user. The user must declare common block IO in the calling program and set LUNO and LUNI to the logical unit number of the console.

[11 July 1984]

Program PLANIM

```
$TITL  PLANIM.FTN - AREA AND PERIMETER MEASUREMENT
C PLANIM.FTN
C USES THE DIGITISER TO CALCULATE THE PERIMETER AND AREA OF A
C POLYGON HAVING MORE THAN TWO VERTICES.
C FOR MOST CALCULATIONS THE POLYGON SHOULD BE CLOSED
      LOGICAL SCALED
C USED TO CHECK IF ALREADY BEEN THROUGH SCALING SECTION
      COMMON/IO/LUNO,LUNI
C FOR COMPATIBILITY WITH DIGTRD
      OPEN(UNIT=5,FILE='CON:')
C OPEN THE CONSOLE AND, BY DEFAULT, THE DIGITISER AS LOGICAL UNIT 5
      LUNI=5
      LUNO=5
      SCALED=.FALSE.
C START OFF AS NOT SCALED
   10 WRITE(5,100)
  100 FORMAT(' PERIMETER AND AREA MEASUREMENT PROGRAM'/
     .       ' ENTER (M)EASURE OR (E)ND')
      READ(5,101) ANS
  101 FORMAT(A1)
      IF (ANS .EQ. 'E') STOP
      IF (ANS .NE. 'M') GOTO 10
C
C SCALE ONLY IF NOT PREVIOUSLY SCALED OR THE USER HAS A NEW
C DRAWING TO WORK ON
      IF (SCALED) THEN
      WRITE(5,102)
  102 FORMAT(' WORKING ON THE SAME PLAN AS THE LAST MEASUREMENT?[Y]')
      READ(5,101) ANS
      IF (ANS .EQ. 'N') GOTO 20
C NEED TO SCALE AGAIN
      GOTO 30
C OTHERWISE USE OLD SCALING
      ENDIF
C
   20 WRITE(5,103)
  103 FORMAT(' ENSURE THAT THE DIGITISER IS IN POINT MODE'/
     .       ' DIGITISE TWO KNOWN POINTS ON A HORIZONTAL LINE'/
     .       ' AND THEN A THIRD KNOWN POINT NOT ON THE SAME LINE')
      CALL DIGTRD(IBTN,X0,Y0)
      CALL DIGTRD(IBTN,XC,YC)
      CALL DIGTRD(IBTN,XV,YV)
C HAVE NOW READ ENOUGH POINTS TO ALLOW CALCULATION OF X AND Y SCALES
      WRITE(5,105)
  105 FORMAT(' USER COORDINATES OF FIRST POINT AS X,Y?')
      READ(5,*) X0MAP,Y0MAP
      WRITE(5,106)
  106 FORMAT(' USER X COORDINATE OF SECOND POINT?')
      READ(5,*) XCMAP
      WRITE(5,107)
  107 FORMAT(' USER Y COORDINATE OF THIRD POINT?')
      READ(5,*) YVMAP
C NOW CALCULATE THE SKEWING
      XD=XC-X0
      YD=YC-Y0
      XDIGIT=SQRT(XD*XD+YD*YD)
C DISTANCE ALONG HORIZONTAL LINE IN DIGITISER UNITS
      THETA=ATAN(YD/XD)
C THE SKEW OF THE USER GRID (RADIAN)
```

```
      CTHETA=COS(THETA)
      STHETA=SIN(THETA)
C NEEDED FOR COORDINATE TRANSFORMATION
C USE TRANSFORM ASSUMING A COMMON ORIGIN BY
C     REMOVING DIGITISER OFFSET  X0,Y0
C     ROTATING
C           XR=XD*COS(THETA)+YD*SIN(THETA)
C           YR=-XD*SIN(THETA)+YD*COS(THETA)
C     THE ROTATED COORDINATES ARE THEN SCALED USING XSCALE,YSCALE
C     AND IF DESIRED MADE RELATIVE TO THE USER ORIGIN BY ADDING
C     X0MAP,Y0MAP
C NOW NEED THE VERTICAL DISTANCE BETWEEN THE HORIZONTAL LINE AND THE
C THIRD POINT
      XD=XV-X0
      YD=YV-Y0
C REMOVE DIGITISER OFFSET
      YR=-XD*STHETA+YD*CTHETA
C NOW HAVE DESKEWED AND GOT Y DISTANCE IN DIGITSER UNITS
      DMAPX=ABS(XCMAP-X0MAP)
      DMAPY=YVMAP-Y0MAP
C THE KNOWN MAP DISTANCES
      XSCALE=DMAPX/XDIGIT
      YSCALE=ABS(DMAPY/YR)
C
C NOW HAVE ALL PARAMETERS FOR GETTING USER COORDINATES
C BUT
C IN THIS CASE THERE IS NO NEED TO GET TRUE USER COORDINATES
C AS WE ARE ONLY USING RELATIVE CALCULATIONS
      SCALED=.TRUE.
C REMEMBER THAT WE HAVE NOW SCALED
   30 CONTINUE
C NOW PROMPT FOR COORDS WHICH ARE PUT STRAIGHT TO DISK TO SAVE TIME
      OPEN(UNIT=4,FILE='DIGTAREA.DGT',RECL=16)
C OPEN A SCRATCH FILE FOR OUTPUT
      WRITE(5,108)
  108 FORMAT(' START DIGITISING'/
     .       ' USE * BUTTON TO STOP AFTER LAST COORDINATE')
   35 CALL DIGTRD(IBTN,X,Y)
      IF (IBTN .EQ. 10) GOTO 40
C IF * BUTTON GO AND CALCULATE
      WRITE(4,200) X,Y
  200 FORMAT(2F8.4)
      GOTO 35
C AND BACK ROUND
C
C AND THEN COME TO HERE TO START CALCULATING
   40 NUMPNT=0
C COUNTER FOR NUMBER OF POINTS DIGITISED
      A1=0.
      A2=0.
C INITIALISE AREA SUBTOTALS
      PERIM=0.
C INITIALISE PERIMETER TOTAL
C
C THE METHOD
C    PERIMETER - SUMS THE DISTANCE BETWEEN COORDINATE PAIRS
C    PLANIMETER - THE AREA OF A POLYGON BOUNDED BY N (X,Y) COORDINATE
C                 PAIRS MAY BE DETERMINED USING:
C      ( (X(1)*Y(2)+X(2)*Y(3)+ . .  X(N-1)*Y(N)+X(N)*Y(1)) -
C      ( (Y(1)*X(2)+Y(2)*X(3)+ . .  Y(N-1)*X(N)+Y(N)*X(1)) )/2
```

```
C                  THIS EQUATION ASSUMES THAT THE FIGURE IS CLOSED AND
C                  CALCULATES THE AREA BASED ON A CLOSED FIGURE EVEN IF
C                  THE USER DOES NOT START AND STOP DIGITISING AT THE SAME
C                  POINT.  IT SHOULD BE NOTED THAT IF THERE ARE ANY LINES
C                  IN THE FIGURE THAT CROSS, THE AREAS OF THE SEPARATE
C                  CLOSED FIGURES WILL BE CALCULATED AND SUBTRACTED
C                  FROM EACH OTHER BY THE ABOVE EQUATION.  FOR ACCURATE
C                  AREA CALCULATIONS OF FIGURES WITH LINES THAT CROSS,
C                  THE USER SHOULD CALCULATE OF EACH CLOSED PART OF THE
C                  FIGURE AND ADD THE VALUES TO DETERMINE THE TOTAL AREA.
      REWIND 4
      READ(4,200,END=46) XF,YF
      NUMPNT=NUMPNT+1
C INCREMENT POINT COUNT
C
C N.B.  IF ACCURACY IS A PROBLEM USE DOUBLE PRECISION
C
      XD=XF-X0
      YD=YF-Y0
C TAKE OFF THE DIGITISER OFFSET
      XF=XD*CTHETA+YD*STHETA
      YF=-XD*STHETA+YD*CTHETA
C ROTATE OUT ANY SKEW
      XF=XF*XSCALE
      YF=YF*YSCALE
C SCALE TO USER UNITS
      XP=XF
      YP=YF
C SAVE THE FIRST POINT
   45 READ(4,200,END=46) XC,YC
      NUMPNT=NUMPNT+1
      XD=XC-X0
      YD=YC-Y0
C TAKE OUT THE DIGITISER OFFSET
      XC=XD*CTHETA+YD*STHETA
      YC=-XD*STHETA+YD*CTHETA
C DESKEW
      XC=XC*XSCALE
      YC=YC*YSCALE
C SCALE TO USER UNITS
      A1=A1+(XP*YC)
C THE FIRST HALF OF THE EQUATION
C (X(1)*Y(2)+X(2)*Y(3)+. .+X(N-1)*Y(N)+X(N)*Y(1))
      A2=A2+(YP*XC)
C THE SECOND HALF OF THE EQUATION
C (Y(1)*X(2)+Y(2)*X(3)+. .+Y(N-1)*X(N)+Y(N)*X(1))
      PERIM=PERIM+SQRT((XP-XC)*(XP-XC)+(YP-YC)*(YP-YC))
C GET THE DISTANCES BETWEEN THE VERTICES AND SUM THEM
      XP=XC
      YP=YC
      GOTO 45
C READ NEXT VALUE
   46 IF (NUMPNT .GT. 2) GOTO 48
      WRITE(5,109) NUMPNT
  109 FORMAT(' ONLY',I2,' POINTS - NO SUMS TO DO')
      GOTO 10
C
   48 CONTINUE
      A1=A1+(XP*YF)
      A2=A2+(YP*XF)
```

6/7

```
C PERFORM THE FINAL CALCULATION
C
C NOW SUBTRACT THE ACCUMULATED VALUES AND DIVIDE BY 2
      AREAIO=ABS(A1-A2)*0.5
      CLOSE(UNIT=4,STATUS='DELETE')
C GET RID OF THE SCRATCH FILE
      WRITE(5,110) PERIM,AREAIO
  110 FORMAT(' PERIMETER IS',F12.5/
     *       ' AREA IS',F12.5,' BOTH IN USER UNITS')
      GOTO 10
      END
$INCLUDE DIGTRD.FTN,-
```

APPENDIX 2

Subroutine DIGTRD

```
      SUBROUTINE DIGTRD(IBTN,X,Y)
C FOR READING ONE NUMBER AT A TIME FROM THE DIGITISER ALTHOUGH
C THE DIGITISER PUTS OUT BLOCKS OF 4
C THE VALUES OF X AND Y ARE IN INCHES
C IBTN IS THE CURSOR BUTTON PRESSED 0 1 2 3 4 5 6 7 8 9  *  #
C VALUE                             0 1 2 3 4 5 6 7 8 9 10 11
C IBTN=-4 IMPLIES STREAM MODE (AS DISTINCT FROM SWITCHSTREAM).
      INTEGER*4 IBTN,IIX(4),IIY(4),IIBTN(4),POINT
      REAL*4 X,Y
C UNBLOCKED AND BLOCKED X,Y AND BUTTON VALUES
      DATA NREAD/5/,POINT/1H./
      SAVE NREAD,IIX,IIY,IIBTN
C NREAD IS THE NEXT NUMBER OF DATA VALUES OF THE FOUR TO READ
C SET TO 5 TO FORCE READ TO START WITH
      COMMON /IO/ LUNO,LUNI
C COMMON BLOCK IS USED FOR COMPATABILITY WITH DIGITISER SOFTWARE
C AND SHOULD BE SET PRIOR TO CALLING THIS ROUTINE
C LUNI=LUNO=LOGICAL UNIT NUMBER OF DIGITISER
   10 IF (NREAD .EQ. 5) THEN
C READ A FULL BLOCK OF FOUR OR FIRST TIME
      READ(LUNI,100) (IIBTN(I),IIX(I),IIY(I),I=1,4)
  100 FORMAT(4(A1,I5,I5))
C READ ONE BLOCK OF FOUR RECORDS
      NREAD=1
      ENDIF
C
      IF (IIBTN(NREAD) .EQ. POINT) THEN
      NREAD=5
      GOTO 10
      ENDIF
C REMAINDER IS FILLED BLOCK SO GET A NEW BLOCK
C NOW GET THE VALUES
      X=IIX(NREAD)*0.001
      Y=IIY(NREAD)*0.001
C CONVERTED TO INCHES HERE
C NOW GET THE RIGHT BUTTON NUMBER
      CALL ILBYTE(IBTN,IIBTN(NREAD),0)
C PICK UP THE CURSOR BUTTON AS A DECIMAL BUTTON
C AND
      IBTN=IBTN-48
C SUBTRACT 48 (THE DECIMAL VALUE OF THE CHARACTER 0) TO GET THE BUTTON
C NUMBER
      NREAD=NREAD+1
      RETURN
      END
```