EDITOR MULTITEMPORAL SYSTEM

by

Martin Ozga
Sue R. Faerman
Richard S. Sigman

United States Department of Agriculture
Economics, Statistcis, and Cooperative Service
Statistical Research Division

November, 1979

TABLE OF CONTENTS

EDITOR MULTITEMPORAL SYSTEM
INTRODUCTION

INTRODUCTION

There is now available in EDITOR a method of creating multitemporal LANDSAT digital data sets. A multitemporal LANDSAT digital data set consists of pixel values from two different scenes taken at different dates but covering the same ground area. The present system in EDITOR has been tried for only a few pairs of scenes so far and, although the results seem promising, more experience with a wide variety of scenes is needed to determine the utility of the system.

When creating a multitemporal digital data set, one of the original two scenes is selected as the primary scene and the other as the secondary scene. The resultant multitemporal data set will have the same mapping of pixel row and column to the ground as the primary scene. The overlay procedure then consists of selecting, for each pixel in the primary scene, the pixel in the secondary scene most nearly representing the same area on the ground.

Several operations are required to use the EDITOR multitemporal system. First, an approximate overlay of the two scenes is created. This is done by digitizing control points on photos or transparencies of the two scenes and then using the control points to generate least squares polynomials representing the approximate overlay. Next, using the approximate overlay, the list of blocks to be read from the two scenes is generated. These blocks are evenly spaced on a grid covering the area common to both scenes. The first two steps are done using the following EDITOR command sequence:

```
! REGISTRATION AND DIGITIZATION FUNCTIONS
2! SCENE TO SCENE REGISTRATION
3! EDIT AND EVALUATE CORRESPONDING POINTS FILE
```

Next, the blocks are read from tape. This is done using the EDITOR command sequence:

```
! TAPE READING TO CREATE WINDOW FILES
2! CORRELATION BLOCKS READ FROM TAPE
```

The blocks thus created represent data for one channel only. The user may select the channel to be used with either channel 2 or channel 4 being the most likely choices. The block size is 64 by 64 for the primary scene and 32 by 32 for the secondary scene. The correlation process is performed on the ILLIAC-IV and consists of applying a correlation function between the 32 by 32 secondary block and each 32 by 32 primary subblock. The shift value returned is the position yielding the highest correlation value. It should be noted that in the EDITOR application, the (single-channel) gradient at each pixel position is used rather than the pixel value. The ILLIAC-IV job to perform the correlation is submitted and checked for correct completion using the EDITOR command:

```
! ILLIAC IV ANALYSIS
```

Finally, after the correlation has been performed, the results must be analyzed to see how well the blocks correlated. Typically, there will be a large number of block pairs which did not correlate well and these will be deleted immediately. Others are then deleted until the maximum residual for all points remaining when applying least squares polynomials is below some value selected by the user. If this condition occurs and there is a sufficiently large number of block pairs (control points) remaining and they are reasonably well distributed over the area common to both scenes, the correlation has succeeded. Otherwise, it has failed and it is then not possible to make a multitemporal data set using the correlation results. This analysis, which is somewhat heuristic, is done using the EDITOR command sequence:

```
! REGISTRATION AND DIGITIZATION FUNCTIONS
2! SCENE TO SCENE REGISTRATION
3! EXAMINE CORRELATION OUTPUT
```

If the correlation was successful, the last operation, creation of a multitemporal data set, is performed. This operation consists of mapping pixels from the secondary to the primary scene using the least squares polynomials from the analysis of the results. It is done outside of EDITOR using the program OVER3 on the IBM 370. The resampling performed on the pixels of the secondary scene is simply the nearest neighbor method; that is, selection of the pixel from the secondary scene closest to the location indicated by applying the polynomials to the row and column of the pixel in the primary scene. The result is an eight-channel data set, the first four channels of each pixel being from the primary scene and the second four from the secondary scene.

MANUAL SELECTION OF CORRESPONDING POINTS

The first step in creation of a multitemporal tape is to use LANDSAT photographic products to manually select corresponding points, i.e., physical features which appear in both of the scenes being overlaid. Such points are used to determine an initial scene-to-scene registration between the scenes to be overlaid. For a first order initial registration only six to ten corresponding points are usually required.

The locations of corresponding points should be unambiguous and not change between the dates of the two scenes. Candidates for corresponding points include small ponds, bends in narrow rivers, highway intersections, and river-road intersections.

The EDITOR Multitemporal Corresponding Points Editor is used to digitize the locations of corresponding points, to determine an initial overlay polynomial, and to build a list of coordinates for the the correlation blocks. It is invoked by the command sequence

```
! REGISTRATION AND DIGITIZATION FUNCTIONS
2! SCENE TO SCENE REGISTRATION
3! EDIT AND EVALUATE CORRESPONDING POINTS FILE.
```

In the example, the ADD CP PAIRS option is used to digitize the locations of corresponding points. When the LANDSAT images are in EDIPS format, they are calibrated to digitizer coordinates by digitizing the 1/4 inch registration marks located in the corners of the border area around each image. When images are in X-format the corners of the actual data portion of the image are digitized to calibrate the image to digitizer coordinates. After each image is calibrated, the calibration residuals in pixels are printed. Whenever one or more of the residuals exceeds two pixels, the image should be recalibrated. After the corresponding points are digitized, the initial overlay polynomial can be determined, as described below, or the corresponding points can be saved in a file with the WRITE UNDELETED CPS TO A FILE option. Saved corresponding points can be used in a future EDITOR session by using the READ OLD FILE OF CPS AND ADD THEM TO CURRENT LIST option.

EDITOR MULTITEMPORAL SYSTEM
MANUAL SELECTION OF CORRESPONDING POINTS (EXAMPLE)

```
@<SRS>EDITOR

EDITOR VERSION 5.35, SEPTEMBER 14, 1979

TODAY IS Wednesday, October 3, 1979 17:34:25-EDT

!REGISTRATION AND DIGITIZATION FUNCTIONS
2!SCENE TO SCENE REGISTRATION
3!EDIT AND EVALUATE CORRESPONDING POINTS FILE
Multitemporal corresponding points editor...Version 1.1

+<ADD CP PAIRS
DIGITIZER TYPE? (? FOR LIST) USDA
MARK MEANS DIGITIZE A POINT ON THE TABLET.

Images greyscales or full scenes? (G or S) S
Tranparency in edips format (annotated with "usgs-edc") (Y or N)? Y
SCENES OVERLAYED OR SIDE BY SIDE? (O or S)S

Place images side-by-side on the digitizer tablet.
For image 1:
Digitize REGISTRATION MARKS of the SCENE, starting at upper-left corner,
   working clockwise:    UL CORNER: UR CORNER: LR CORNER: LL CORNER:
        R-SQUARE
LINE   1.000000  RESIDUALS(UL,UR,LR,LL):  0.025 -0.025  0.025 -0.025
COLUMN 1.000000  RESIDUALS(UL,UR,LR,LL):  0.779 -0.779  0.780 -0.780
Calibration satisfactory? (Y or N) Y
For image 2:
Digitize REGISTRATION MARKS of the SCENE, starting at upper-left corner,
   working clockwise:    UL CORNER: UR CORNER: LR CORNER: LL CORNER:
        R-SQUARE
LINE   1.000000  RESIDUALS(UL,UR,LR,LL): -0.773  0.772 -0.772  0.773
COLUMN 1.000000  RESIDUALS(UL,UR,LR,LL):  0.348 -0.347  0.348 -0.348
Calibration satisfactory? (Y or N) Y
Digitize corresponding points on SCENES
   ("ERROR" to begin a new pair of SCENES, "QUIT" to stop).
Image 1:   Image 2:
Image 1:   Image 2:
Image 1:   Image 2:
Image 1:   Image 2:
Image 1:   Image 2:
Image 1:   Image 2:
Image 1:   Image 2:
Image 1:   Image 2:
Image 1:   Image 2:
Image 1:
```

CREATION OF INITIAL OVERLAY PARAMETER FILE

The ANALYZE VIA LEAST-SQUARES option of the Multitemporal Corresponding Points Editor creates an initial overlay parameter file. This file is an input file to the GENERATE LIST OF BLOCKS FILE option and also to the CALCULATE WINDOW IN SECONDARY IMAGE option.

The initial overlay parameter file contains the coefficients of a bivariate polynomial which transforms primary scene coordinates into secondary scene coordinates. The coefficients are calculated by a least-squares regression analysis of the corresponding points. Also in the initial overlay parameter file are the primary and secondary scene coordinates for the largest window common to the user-defined primary and secondary scene windows of interest.

In the example, only a portion of both the primary and secondary scenes was of interest because of clouds in both images. The window of interest in each scene was determined prior to multitemporal registration processing by means of the EDITOR command sequence

```
! REGISTRATION AND DIGITIZATION FUNCTIONS
2! FIND A POINT ON A MAP OR SCENE
3! SCENE FIND A POINT ON A SCENE
```

When an entire scene overlay is desired, however, the following windows should be specified:

EDIPS format: 1, 1, 2983, 3548

EDITOR MULTITEMPORAL SYSTEM
CREATION OF INITIAL OVERLAY PARAMETER FILE

X-format reformatted to USDA (non-deskewed) format:

LANDSAT 1: 1, 1, 2340, 3240
LANDSAT 2: 1, 1, 2340, 3264
LANDSAT 3: 1, 1, 2340, 3192

EDITOR MULTITEMPORAL SYSTEM
CREATION OF INITAL OVERLAY PARAMETER FILE (EXAMPLE)

```
+<ANALYZE VIA LEAST-SQUARES
USE A GENERAL POLYNOMIAL? (Y OR N) Y
DEGREE? (0-4) 1
ROOT MEAN SQUARE ERRORS ARE:      2.1672      2.5837
MAX ABS ERRORS ARE:      2.9667      3.1996
AND OCCUR FOR CPS           1           3

+<<O

Containing window in primary scene (l,c,l,c) = 1084,1,2983,3548
Containing window in secondary scene (l,c,l,c) = 1090,756,2983,3548
EVALUATING INVERSE
ROOT MEAN SQUARE ERRORS ARE:      2.1677      2.5844
MAX ABS ERRORS ARE:      2.9661      3.1890
AND OCCUR FOR CPS           1           3
Window coordinates are...
1092,536,2981,3327  (primary)
1090,756,2983,3548  (secondary)
The overlay parameter output file is MO-MT.CP-OVERLAY-PARAMS [New file]

+<<Q
```

GENERATE LIST OF BLOCKS FILES

After creating the initial overlay parameter file, the user must create the list of blocks files. This is done using the EDITOR command sequence:

```
! REGISTRATION AND DIGITIZATION FUNCTIONS
2! SCENE TO SCENE REGISTRATION
3! EDIT AND EVALUATE CORRESPONDING POINTS FILE
+<GENERATE LIST OF BLOCKS FILES
```

Since the blocks are presumed to be on a grid, the user is first asked for the number of blocks per row and the number of rows of blocks. The total number of blocks, i.e. the number of blocks per row times the number of rows of blocks may not exceed 340. Next the user is asked for the initial overlay parameter file. The program reads this file and uses the information in it to check that the primary window block coordinates will not overlap, either horizontally or vertically. In this case an appropriate message is written out and the user is given a chance to enter new values for the number of blocks per row and rows of blocks.

Next the user is questioned about the tape and transparency format. This is necessary so that the program can check that the generated secondary window block coordinates are indeed valid block coordinates. First the user is asked if the transparency is in EDIPS format. If so the program will generate appropriate check values. If not the user is asked whether or not the tapes are in USDA format. If the tapes are deskewed the user should answer negatively to this question. The user will then be asked for the coordinates of the secondary scene so that these values may be used for check values. If the tapes are in USDA format, i.e. non-deskewed, the user is asked which LANDSAT mission the data is from. The program will generate appropriate check values using this information.

The program then calculates the coordinates for the list of blocks files. Note that two files are created, a primary file with 64 by 64 block coordinates and a secondary file with 32 by 32 block coordinates. If any block is outside the coordinates of the check area, the user is notified that the block has been dropped from both files. Similarly, if an entire row has been dropped an appropriate message will be issued. If either an entire row or an entire column has been dropped the user might want to check with a programmer for a possible error.

In the example the user has chosen to use 20 blocks per row and 17 rows of blocks. The overlay parameter file is specified as MO-MT.CP-OVERLAY-PARAMS which was the file created in the previous step. The transparency is in EDIPS format. The two output files are MO-MT.PRIMARY-WINS/WEST and MO-MT.SECONDARY-WINS/WEST. These files will be used during the next step of the analysis.

EDITOR MULTITEMPORAL SYSTEM
GENERATE LIST OF BLOCKS FILES (EXAMPLE)

```
+<G


 NUMBER OF BLOCKS PER ROW = 20
 NUMBER OF ROWS OF BLOCKS = 17
 OVERLAY PARAMETER input file is MO-MT.CP-OVERLAY-PARAMS;1 [Old version]
 TRANSPARENCY IN EDIPS FORMAT  (ANNOTATED WITH "USGS-EDC") (Y OR N)? Y
 PRIMARY output file is MO-MT.PRIMARY-WINS/WEST [New file]
 SECONDARY output file is MO-MT.SECONDARY-WINS/WEST [New file]

+<Q

CPU TIME: 3.92  ELAPSED TIME: 3:50.52
NO EXECUTION ERRORS DETECTED
```

READING BLOCKS FROM TAPE

Once the lists of block coordinates have been built, the blocks must be read from tape. This is done using the EDITOR command sequence:

```
! TAPE READING TO CREATE WINDOW FILES
2! CORRELATION BLOCKS READ FROM TAPE
```

The 64 by 64 blocks must be read from the tape containing the data for the primary scene and the 32 by 32 blocks must be read from the tape containing the data for the secondary scene. It is the user's responsibility to be sure that the correct tapes are used; the program does not check this.

Also, the user should be sure to answer 'Y' to the question

CREATE BLOCKS FOR GRADIENT CORRELATION (Y OR N)?

since the program will then insert some extra edge data so that a true full size (64 by 64 or 32 by 32 as the case may be) gradient image may later be created for each block. Finally, the channel to be used must be selected. The same channel must be selected for both the primary and secondary scenes. The channel is a number between 1 and 4 in the usual EDITOR manner.

In the example, the operator is asked to mount the tape containing the primary scene. After the tape is mounted, the appropriate EDITOR command sequence is used to enter the program to read the blocks from tape. The input parameter file is specified as MO-MT.PRIMARY-WINS/WEST which has been created to contain the coordinates of the 64 by 64 blocks. Channel 4 is specified as the channel from which the data is to be taken and the blocks are to be created for the gradient correlator. The output file of blocks is specified as MO-MT.CHNL-4/PRIMARY-BLOCKS/WEST.

Once tape reading starts, one dot is printed for each tape record (image line) read. All blocks intersecting that image line receive data from it so that no tape record is read more than once. The assumption is made, however, that the rows of blocks are nearly horizontal with respect to the coordinate system (rows and columns) of the scene.

When tape reading is completed and the file of 64 by 64 blocks has been created, the procedure is repeated for the file of 32 by 32 blocks.

After both files of blocks have been created, the user is ready to submit the correlation job to the ILLIAC-IV.

EDITOR MULTITEMPORAL SYSTEM
READING BLOCKS FROM TAPE (EXAMPLE)

```
@LINK OPERATOR

LINK FROM SIGMAN, JOB 31, TTY 70
!;HI.  CAN YOU MOUNT TAPE 21654-16100 ON A 9 TRACK DRIVE WITH OUT A RING?
@;ONE MINUTE
!;OK ALL SET ON MTA2:
!;THANKS. BYE.
@BREAK (LINKS)
@ASSIGN MTA2:
@MOUNT MTA2:
@<SRS>EDITOR

EDITOR VERSION 5.35, SEPTEMBER 14, 1979

TODAY IS Thursday, October 4, 1979 13:22:02-EDT

!TAPE READING TO CREATE WINDOW FILES
2!CORRELATION BLOCKS READ FROM TAPE
BLOCK PARAMETER FILE=MO-MT.PRIMARY-WINS/WEST
 [Old version]
USING 64 BY 64 BLOCKS
ENTER CHANNEL TO USE: 4
CREATE BLOCKS FOR GRADIENT CORRELATION (Y OR N)? Y
TAPE DRIVE=MTA2:
 [OK]

.
TAPE WINDOW=1,1,2983,3548
OUTPUT FILE OF 64 BY 64 BLOCKS=MO-MT.CHNL-4/PRIMARY-BLOCKS/WEST
 [New file]
................................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**..............................................................................
**............................

USED 2:26 IN 44:21
End of SAIL execution
2!Q
!Q
USED 2:29 IN 50:11
@UNLOAD MTA2:
```

EDITOR MULTITEMPORAL SYSTEM
READING BLOCKS FROM TAPE (EXAMPLE)

```
@LINK OPERATOR

LINK FROM SIGMAN, JOB 31, TTY 70
@;YES
!
!;HI. I AM FINISHED WITH THE TAPE ON MTA2:.  CAN YOU
@;MOUNT ANOTHER TAPE FOR ME ON MTA2:?  IT HAS BBN
@;NUMBER 3496
@;1 SEC
@;ALL SET ON MTA2:
!;THANKS. BYE.
@BREAK (LINKS)
@MOU MTA2:
@<SRS>EDITOR

EDITOR VERSION 5.35, SEPTEMBER 14, 1979

TODAY IS Thursday, October 4, 1979 14:28:53-EDT

!TAPE
2!CORRELATION BLOCKS READ FROM TAPE
BLOCK PARAMETER FILE=MO-MT.SECONDARY-WINS/WEST;1 [Old version]
USING 32 BY 32 BLOCKS
ENTER CHANNEL TO USE: 4
CREATE BLOCKS FOR GRADIENT CORRELATION (Y OR N)? Y
TAPE DRIVE=MTA2:
 [OK]
.
TAPE WINDOW=1,1,2983,3548
OUTPUT FILE OF 32 BY 32 BLOCKS=MO-MT.CHNL-4/SECONDARY-BLOCKS/WEST
 [New file]
.......................................................................................
**.....................................................................................
**.....................................................................................
**.....................................................................................
**.....................................................................................
**.....................................................................................
**.....................................................................................
**.....................................................................................
**...............

USED 50 IN 9:57
End of SAIL execution
2!Q
!Q
USED 0:53 IN 29:51
@UNLOAD MTA2:
@DEAS MTA2:
```

CORRELATION ON THE ILLIAC-IV

After the files of blocks have been read from tape, the correlation must be performed. This is done using the ILLIAC-IV since it is a job requiring a large amount of computation and fits the architecture of the ILLIAC-IV reasonably well.

The ILLIAC-IV batch job to do the correlations is submitted using the top-level EDITOR command

! ILLIAC IV ANALYSIS

Upon completion of the job, a message will be received in the user's directory from I4-TENEX, the front-end computer for the ILLIAC-IV. The same top-level EDITOR command (ILLIAC IV ANALYSIS) is then used to check that the job was completed correctly. Submitting ILLIAC-IV correlation jobs is very similar to submitting other ILLIAC-IV jobs, such as classify and cluster. The two input files which must be specified are the file of 64 by 64 blocks from the primary scene and the file of 32 by 32 blocks from the secondary scene as read from their respective tapes. The output is the file of correlation values and shifts for the block pairs.

The correlation algorithm used on the ILLIAC-IV has been adapted from the correlation algorithm developed by the Laboratory for Applications of Remote Sensing (LARS) at Purdue University [1],[2]. In the ILLIAC-IV algorithm, each block pair is handled individually. First, both the 64 by 64 and 32 by 32 blocks are converted to gradient form where the single-channel gradient of a pixel at row I and column J is defined to be

```
SQRT((X[I+1,J]-X[I-1,J])^2-(X[I,J+1]-X[I,J-1])^2)
```

Note that when the blocks were read from tape, extra edge information was read so that a complete gradient image is made for each block.

Next, the correlation function is applied to the 32 by 32 secondary scene block and each 32 by 32 subblock of the 64 by 64 primary scene block. There are 1089 such subblocks (33 columns and 33 rows) and the correlation value returned is that with highest absolute value found. The correlation function used is

```
N^2*(SXY-SX*SY)/SQRT((N^2*SX2-S2X)*(N^2*SY2-S2Y))
```

where N=32 and other symbols are explained below. Let Y refer to gradient values in the 32 by 32 subblock of the the 64 by 64 block from the primary scene and X refer to gradient values in the 32 by 32 block from the secondary scene. Then, if we let SUM(Z) indicate a sum of all elements in a 32 by 32 block and X*Y denote multiplication of corresponding components in two 32 by 32 blocks,

```
SXY=SUM(X*Y)
SX=SUM(X)
SY=SUM(Y)
SX2=SUM(X^2)
S2X=(SUM(X))^2
SY2=SUM(Y^2)
S2Y=(SUM(Y))^2
```

Two shift values are generated for the 32 by 32 subblock with the highest correlation value found, a horizontal and a vertical shift. These shifts are the displacement from the 32 by 32 subblock centered in the 64 by 64 block. The horizontal shift is positive to the left (west) and negative to the right (east). The vertical shift is positive upwards (north) and negative downwards (south).

The final step is to do an interpolation on the shifts, the interpolation being done separately on the horizontal and vertical shifts. The interpolated value of the horizontal shift is

IH=H-(CL-CR)/(4*C-2*CL-2*CR)

where IH is the interpolated horizontal shift, H is the horizontal shift before interpolation, C is the maximum correlation value found, CL is the correlation value immediately to the left of the position of C, and CR is the correlation value immediately to the right of the position of C. A similar interpolation is done on the vertical shift. No interpolation is done if the maximum correlation value lies at an edge of the 33 by 33 correlation surface, that is, if the absolute value of either the horizontal or vertical shift is 16.

In the example, the ILLIAC-IV job is submitted using MO-MT.CHNL-4/PRIMARY-BLOCKS-WEST as the file of 64 by 64 primary blocks and MO-MT.CHNL-4/SECONDARY-BLOCKS/WEST as the file of 32 by 32 secondary blocks. The output file is to be MO-MT.CHNL-4/CORR/WEST. Since CPYNET is used for both input and output files, the output file will appear in the user's directory upon completion of the ILLIAC-IV job. The INQUIRE feature of ILLIAC IV ANALYSIS (not shown in the example) will still be needed to check for correct completion and to delete files at I4-TENEX.

EDITOR MULTITEMPORAL SYSTEM
CORRELATION ON THE ILLIAC-IV (EXAMPLE)

```
@<SRS>EDITOR

EDITOR VERSION 5.35, SEPTEMBER 14, 1979

TODAY IS Thursday, October 4, 1979 15:19:06-EDT

!ILLIAC
SUBMIT AND RETRIEVE ILLIAC JOBS, VERSION 3.15, (21-Sep-79)

ENTER USER CODE AT I4-TENEX:ERTS
ENTER PASSWORD FOR ERTS AT I4-TENEX:
ENTER ACCOUNT NUMBER FOR ERTS AT I4-TENEX:17
#CPYNET (USE BATCH FTP AT I4)
CPYNET OF FILES: BOTH INPUT AND OUTPUT FILES
SEND ALL LOCAL INPUT FILES USING CPYNET (Y OR N)? Y
RETRIEVE ALL OUTPUT FILES USING CPYNET (Y OR N)? Y
#FILE DIRECTORY (USE A DIFFERENT DIRECTORY AT I4 FOR INPUT/OUTPUT FILES)
FILE DIRECTORY FOR: BOTH INPUT AND OUTPUT FILES
DIRECTORY TO USE AT I4 FOR INPUT AND OUTPUT FILES=I4DM-TAPE
#SEL
SELECT JOBS TO BE RUN
%CORRELATE
INPUT 64 BY 64 BLOCKS FILE=MO-MT.CHNL-4/PRIMARY-BLOCKS/WEST
LOCATION OF FILE MO-MT.CHNL-4/PRIMARY-BLOCKS/WEST:LOCAL
INPUT 32 BY 32 BLOCKS FILE=MO-MT.CHNL-4/SECONDARY-BLOCKS/WEST
LOCATION OF FILE MO-MT.CHNL-4/SECONDARY-BLOCKS/WEST:LOCAL
NAME OF OUTPUT CORRELATION WHEN RETRIEVED FROM I4=MO-MT.CHNL-4/CORR/WEST
DELETE OUTPUT CORRELATION AT I4 IF CPYNET IS SUCCESSFUL (Y OR N)? Y

THIS JOB WILL REQUIRE 1 BAND OF I4 DISK

YOUR I4 JOB NUMBER IS 809

ENTER PASSWORD FOR SIGMAN AT BBN-TENEXB:
ENTER ACCOUNT NUMBER FOR SIGMAN AT BBN-TENEXB:112421
WARNING, DO NOT DELETE THE FOLLOWING FILES:
<SIGMAN>MO-MT.CHNL-4/PRIMARY-BLOCKS/WEST;1
<SIGMAN>MO-MT.CHNL-4/SECONDARY-BLOCKS/WEST;1
UNTIL THE STATUS OF THE ILLIAC JOB IS 'I4.WAITING' OR 'COMPLETED'
DIRECTORY TO WHICH OUTPUT FILES ARE TO BE SENT AT BBN-TENEXB: SIGMAN
THE FILE MO-MT.CHNL-4/CORR/WEST
WILL APPEAR IN DIRECTORY SIGMAN AT BBN-TENEXB ON COMPLETION OF JOB 809

DISK FILE SUMMARY
FILE NAME AT I4   (FILE NAME AT BBN-TENEXB)

INPUT FILES:
<I4DM-TAPE>CO809.B64   (<SIGMAN>MO-MT.CHNL-4/PRIMARY-BLOCKS/WEST;1)*
<I4DM-TAPE>CO809.B32   (<SIGMAN>MO-MT.CHNL-4/SECONDARY-BLOCKS/WEST;1)*
OUTPUT FILES:
<I4DM-TAPE>CO809.OUT   (MO-MT.CHNL-4/CORR/WEST)*
```

EDITOR MULTITEMPORAL SYSTEM
CORRELATION ON THE ILLIAC-IV (EXAMPLE)

```
* INDICATES CPYNET USEAGE
FILE TRANSMISSION TO I4 IN PROGRESS, WAIT....

CL809.PIF;1
700  <ERTS>CL809.PIF;1 SUBMITTED ILLIAC  4-OCT-79 12:33 <ERTS>CO809.POF

!
I4 JOB 809 SUCCESSFULLY SUBMITTED
#Q

USED 7 IN 14:13
End of SAIL execution
!Q
USED 0:12 IN 14:52
@
```

ANALYSIS OF CORRELATION OUTPUT

After the correlation job has been run on the ILLIAC-IV, the output must be examined and analyzed to determine the quality of the results and their suitability for creation of a multitemporal digital data set. This is done using the EDITOR command sequence

! REGISTRATION AND DIGITIZATION FUNCTIONS
2! SCENE TO SCENE REGISTRATION
3! EXAMINE CORRELATION OUTPUT

The shifts determined for each block pair subtracted from the row and column of the center of the primary block determine the primary scene coordinates of each control point. The secondary scene coordinates are the center row and column of the secondary block. Thus, the output of the ILLIAC-IV job constitutes a collection of control points to relate the two scenes, with each control point having a row and column of the primary scene and the corresponding row and column of the secondary scene. The basic idea of the analysis is to arrive at a subset of the original control points for which the maximum residual for polynomials generated by the control points and then evaluated at the same control points is less than some value. Two polynomials are generated, one for rows and the other for columns. These are least-square polynomials and may be second or third order as desired. If the program OVER3 is to be used to create the multitemporal data set, third order polynomials must be used. The polynomials generate row and column values in the secondary scene as a function of row and column values in the primary scene. Of course, for the subset of control points to be useful, the number of control points must be sufficiently large and the control points must be reasonably well distributed over the area covered by the two scenes. The polynomials then provide a good fit of the undeleted control points and, presumably, also provide a good mapping of the secondary scene to the primary scene in the parts of both scenes to be used in creation of the multitemporal data set.

The procedure used consists of deleting control points, generating least-squares polynomials using the remaining control points, and evaluating the polynomials at these control points until the maximum residual from the evaluation is less than a user-specified value. This value is in pixels and a suitable value might be .5 pixel. First, deletions are made to get rid of "obviously bad" control points. These are control points with small correlation values or large shifts. What constitutes small correlation values and large shifts will vary somewhat with the data.

Therefore, the first thing one might do is use the 'V' command to view the correlation output from the ILLIAC-IV. Then, control points with correlation values smaller than most should be deleted using the 'DC' command. Certainly, control points with correlation values less than .15 should be deleted. For some data sets with a large number of high correlation values, this threshold might be raised. Next, the 'DS' command should be used to delete control points for which the absolute value of the shifts is high. Control points with the absolute value of the shift in either horizontal or vertical direction exceeding the value specified will be deleted. Certainly, control points with the absolute value of either shift greater than 15 should be deleted since the maximum absolute value of a shift is 16. If most shift values are low, this threshold for deletion by shifts may be set somewhat lower.

After those deletions have been completed, the remaining control points will probably still not meet the requirement for the maximum residual when evaluating the polynomials. The final deletion step is to then successively delete the control point with the highest residual, recompute the polynomials using the remaining control points, and evaluate the polynomials at the control points until the maximum residual falls below the value specified. This is done using the 'AUTO' command. Once the maximum residual has reached the desired value, the number of remaining control points is printed so that a check may be made to see if this number is sufficiently large. Also, a display of the distribution of the remaining control points may be printed using the 'M' command. If the number of control points remaining is 100 or more and they appear to be well-distributed, the correlation is said to be successful and the user may proceed to create a multitemporal data set. Naturally, the higher the number of remaining control points, the more confidence one may have in the quality of the multitemporal data set to be created. In looking at the distribution, it should be noted that the correlation will fail in areas where there is cloud cover in either scene or areas where there are large bodies of water.

If the correlation is successful, the 'O' command is used to write to disk the coefficients of the polynomials. This final overlay parameter file can be transmitted to an IBM 370 and used as the input parameter file to program OVER3 to create a multitemporal data set.

If the correlation failed in one channel, another channel might be tried. Also, to improve the results, the 'CO' command may be used to combine control points from two different channels. For the 'CO' command to work, the ILLIAC-IV correlation job must have been run on the same block pairs but for different channels. The output file for one of these channels is read in and some control points are deleted using the 'DS', 'DC', and possibly the 'AUTO' command. The control points, with the deletions marked, are then saved on a disk file. This is done using the 'CP' command to copy the control points to the output buffer and the 'W' command to write the output buffer to disk. The file created by the ILLIAC-IV for the other channel is then read in and some deletions are performed. Finally, the 'CO' command is used to combine the two. In the combined resulting list of control points, if a control point is undeleted in both inputs, the control point values corresponding to the highest correlation value are chosen. If a control point is deleted in one input and undeleted in the other, the undeleted control point values are used. If a control point is deleted in both inputs, it remains deleted. The resultant control points may then be analyzed as described above.

In the exmaple, the input file is MO-MT.CHNL-4/CORR/WEST. This file was created by an ILLIAC-IV correlation job. Third order polynomials are to be used. All control points with horizontal or vertical absolute shift values greater than 12 are deleted using the 'DS' command. This deletes 106 control points (referred to as blocks by the program since they represent block pairs). The 'DC' command is used to delete control points with correlation values less than .15. Since there are no remaining control points with such correlation values, none are deleted. Finally, the 'AUTO' command is used with maximum residual allowed set to .5. A dot is printed for each phase of the 'AUTO' command. This causes 115 control points to be deleted, with 119 remaining. Using the 'M' command, it is seen that the remaining control points seem to be well-distributed, so that the correlation results are acceptable. Therefore, it is not necessary to use the 'CO' command to combine the control points from the correlation of the same blocks on another channel. In this particular example, it was revealed by examination of the transparencies for the two scenes that there were some clouds on the left where undeleted control points seem somewhat sparse.

The 'V' command is used to display the correlation values, shifts, and residuals associated with the undeleted control points. Finally, the 'O' command is used to write the final overlay parameter file.

EDITOR MULTITEMPORAL SYSTEM
ANALYSIS OF CORRELATION OUTPUT (EXAMPLE)

```
@<SRS>EDITOR

EDITOR VERSION 5.38, OCTOBER 18, 1979

TODAY IS Thursday, October 18, 1979 13:49:44-EDT

!REGISTRATION AND DIGITIZATION FUNCTIONS
2!SCENE TO SCENE REGISTRATION
3!EXAMINE CORRELATION OUTPUT

INPUT XFORM PARM FILE=MO-MT.CHNL-4/CORR/WEST;1

DEGREE OF POLYNOMIAL? (2 OR 3) : 3

FILE TYPE=11, OUTPUT FROM CORRELATION FOR 340 BLOCKS RUN ON ILLIAC-IV

TYPE "?" FOR HELP,
WARNING--ONLY FIRST 256 PAIRS USED.


#DS

DELETE CORRELATIONS WITH ABSOLUTE VALUE OF SHIFT(S) GREATER THAN: 12

106 BLOCKS DELETED
POLYNOMIAL COEFFICIENTS ARE:
  2030.5620    887.3360       1.5288       0.6944       1.5214      -1.2891
     2.0547     -4.2993      -1.4871       0.2072

  1805.7672      2.7504    1485.2721      -0.1509       0.9419       1.8519
    - .7486      0.6487      -1.8842      -0.4862

ROOT MEAN SQUARE ERRORS ARE:       3.6522       3.9093
MAX ERRORS ARE:       12.8141      13.2524


#DC

DELETE CORRELATIONS WITH VALUE LESS THAN: .15

  0 BLOCKS DELETED
POLYNOMIAL COEFFICIENTS ARE:
  2030.5620    887.3360       1.5288       0.6944       1.5214      -1.2891
     2.0547     -4.2993      -1.4871       0.2072

  1805.7672      2.7504    1485.2721      -0.1509       0.9419       1.8519
    -0.7486      0.6487      -1.8842      -0.4862

ROOT MEAN SQUARE ERRORS ARE:       3.6522       3.9093
MAX ERRORS ARE:       12.8141      13.2524
```

EDITOR MULTITEMPORAL SYSTEM
ANALYSIS OF CORRELATION OUTPUT (EXAMPLE)

```
#AUTO

MAXIMUM ABSOLUTE ACCEPTABLE ERROR? (>0.0) .5
.............................................................................................
**..................................

115 BLOCKS DELETED, MAX ERROR NOW    0.497

 119 CORRELATIONS REMAIN OUT OF  340 OR  35.0%

#M

DISPLAY OF BLOCKS
    1 .*....*...*......*..
    2 ....**...*...******.
    3 ......**......**..*.
    4 .*..*.**.**...*.....
    5 ...*.....*.***.*..*.
    6 ........******.*.**.
    7 ...*...**..*.**.....
    8 ....*.*...****.**...
    9 ..*....*..*.**..*.*.
   10 ...****.**.*.******.
   11 ......*..*.*......*.
   12 ....*.....*.**.*....
   13 .*....**.*.*.*....*.
   14 .*...***.**.*****...
   15 .......*..***.*.....
   16 .....*..***.***.....
   17 ......*.*.....**.*..


"*" = UNDELETED
"." = DELETED

#V

COR #  VALUE  X-SHIFT  Y-SHIFT      64 BY 64 WINDOW      32 BY 32 WINDOW
    X-ERROR Y-ERROR

   2   0.245   -1.149    2.823  1122, 227,1185, 290  1135, 463,1166, 494
       0.33   -0.29
   7    .310   -3.101    1.833  1122,1052,1185,1115  1136,1288,1167,1319
       0.05    0.12
  11   0.347   -3.131    0.431  1122,1712,1185,1775  1137,1948,1168,1979
       0.10    0.24
  18   0.162   -2.828   -2.787  1122,2867,1185,2930  1139,3103,1170,3134
      -0.50   -0.10
  25   0.334   -2.965    1.681  1232, 722,1295, 785  1246, 958,1277, 989
      -0.24   -0.18
  26   0.512   -3.090    1.650  1232, 887,1295, 950  1246,1123,1277,1154
```

EDITOR MULTITEMPORAL SYSTEM
ANALYSIS OF CORRELATION OUTPUT (EXAMPLE)

	-0.12	-0.14			
30	0.326	-3.213	0.023	1232,1547,1295,1610	1247,1783,1278,1814
	0.05	-0.39			
34	0.328	-3.045	-0.865	1232,2207,1295,2270	1248,2443,1279,2474
	- .17	0.16			
35	0.344	-3.026	-1.133	1232,2372,1295,2435	1248,2608,1279,2639
	-0.30	-0.01			
36	0.212	-2.647	-1.126	1232,2537,1295,2600	1248,2773,1279,2804
	-0.06	0.09			
37	0.199	-2.025	-1.228	1232,2702,1295,2765	1248,2938,1279,2969
	0.42	0.08			
38	0.393	-2.037	-2.142	1232,2867,1295,2930	1249,3103,1280,3134
	0.28	0.24			
39	0.886	-2.213	-2.189	1232,3032,1295,3095	1249,3268,1280,3299
	-0.00	0.25			
47	0.331	-3.144	0.285	1342,1052,1405,1115	1357,1288,1388,1319
	-0.12	-0.45			
48	0.242	-3.144	0.998	1342,1217,1405,1280	1357,1453,1388,1484
	-0.02	0.32			
55	0.706	-2.189	-0.751	1342,2372,1405,2435	1358,2608,1389,2639
	0.44	0.12			
56	0.208	-2.614	-1.099	1342,2537,1405,2600	1358,2773,1389,2804
	-0.13	-0.17			
59	0.571	-2.088	-2.492	1342,3032,1405,3095	1359,3268,1390,3299
	0.02	-0.43			
62	0.249	-0.967	2.095	1452, 227,1515, 290	1466, 463,1497, 494
	0.17	0.29			
65	0.219	-2.746	2.227	1452, 722,1515, 785	1466, 958,1497, 989
	-0.32	0.38			
67	0.237	-2.552	0.942	1452,1052,1515,1115	1467,1288,1498,1319
	0.29	0.14			
68	0.307	-2.873	0.929	1452,1217,1515,1280	1467,1453,1498,1484
	0.07	0.17			
70	0.379	-2.528	1.102	1452,1547,1515,1610	1467,1783,1498,1814
	0.45	0.44			
71	0.324	-2.876	0.299	1452,1712,1515,1775	1467,1948,1498,1979
	0.05	-0.31			
75	0.228	-2.149	-0.960	1452,2372,1515,2435	1468,2608,1499,2639
	0.30	-0.37			
84	0.289	-2.174	1.834	1562, 557,1625, 620	1576, 793,1607, 824
	-0.35	-0.01			
90	0.361	-3.002	0.535	1562,1547,1625,1610	1577,1783,1608,1814
	- .28	-0.29			
92	0.240	-2.793	-0.080	1562,1877,1625,1940	1578,2113,1609,2144
	-◡.20	0.15			
93	0.370	-2.116	-0.588	1562,2042,1625,2105	1578,2278,1609,2309
	0.36	-0.34			
94	◡.359	-2.583	-0.055	1562,2207,1625,2270	1578,2443,1609,2474
	-0.24	0.22			
96	0.494	-2.131	-0.234	1562,2537,1625,2600	1578,2773,1609,2804
	-0.07	0.06			

EDITOR MULTITEMPORAL SYSTEM
ANALYSIS OF CORRELATION OUTPUT (EXAMPLE)

99	0.384	-1.888	-1.205	1562,3032,1625,3095	1579,3268,1610,3299
	-0.20	0.04			
109	0.256	-2.523	0.817	1672,1382,1735,1445	1687,1618,1718,1649
	-0.10	-0.17			
110	0.413	-2.785	1.441	1672,1547,1735,1610	1687,1783,1718,1814
	- .37	0.45			
111	0.328	-2.272	1.125	1672,1712,1735,1775	1687,1948,1718,1979
	0.10	0.13			
112	0.378	-2.466	0.173	1672,1877,1735,1940	1688,2113,1719,2144
	-0.18	0.17			
113	0.206	-2.408	0.270	1672,2042,1735,2105	1688,2278,1719,2309
	-0.24	0.27			
114	0.247	-1.936	0.124	1672,2207,1735,2270	1688,2443,1719,2474
	0.11	0.11			
116	0.336	-1.800	-0.087	1672,2537,1735,2600	1688,2773,1719,2804
	-0.04	-0.13			
118	0.373	-1.320	-1.151	1672,2867,1735,2930	1689,3103,1720,3134
	0.17	-0.28			
119	0.258	-1.345	-0.870	1672,3032,1735,3095	1689,3268,1720,3299
	0.04	-0.06			
124	0.242	-1.141	2.130	1782, 557,1845, 620	1796, 793,1827, 824
	- .01	0.26			
128	0.233	-2.113	1.086	1782,1217,1845,1280	1797,1453,1828,1484
	-0.10	-0.01			
129	0.228	-2.159	0.831	1782,1382,1845,1445	1797,1618,1828,1649
	-0.10	-0.31			
132	0.279	-1.994	0.352	1782,1877,1845,1940	1798,2113,1829,2144
	-0.06	0.12			
134	0.229	-1.723	-0.192	1782,2207,1845,2270	1798,2443,1829,2474
	-0.03	-0.49			
135	0.265	-1.173	0.255	1782,2372,1845,2435	1798,2608,1829,2639
	0.38	-0.08			
145	0.441	-1.030	1.830	1892, 722,1955, 785	1906, 958,1937, 989
	0.03	-0.15			
147	0.203	-1.290	1.183	1892,1052,1955,1115	1907,1288,1938,1319
	0.21	0.04			
151	0.462	-1.762	0.157	1892,1712,1955,1775	1908,1948,1939,1979
	-0.14	-0.24			
152	0.242	-1.900	0.201	1892,1877,1955,1940	1908,2113,1939,2144
	-0.36	-0.25			
153	0.454	-1.225	0.498	1892,2042,1955,2105	1908,2278,1939,2309
	0.21	-0.01			
154	0.524	-1.072	0.760	1892,2207,1955,2270	1908,2443,1939,2474
	0.24	0.18			
156	0.313	-0.873	-0.040	1892,2537,1955,2600	1909,2773,1940,2804
	0.16	0.23			
157	0.336	-0.686	-0.406	1892,2702,1955,2765	1909,2938,1940,2969
	0 21	-0.23			
163	0.389	-0.048	2.234	2002, 392,2065, 455	2016, 628,2047, 659
	-0.21	0.49			
168	0.309	-1.596	1.130	2002,1217,2065,1280	2017,1453,2048,1484

EDITOR MULTITEMPORAL SYSTEM
ANALYSIS OF CORRELATION OUTPUT (EXAMPLE)

	-0.41	-0.19			
171	0.516	-1.078	1.064	2002,1712,2065,1775	2018,1948,2049,1979
	0.13	0.49			
173	0.317	-1.111	0.631	2002,2042,2065,2105	2018,2278,2049,2309
	-0.08	-0.12			
174	0.263	-0.760	0.653	2002,2207,2065,2270	2018,2443,2049,2474
	0.14	-0.19			
177	0.466	-0.734	0.481	2002,2702,2065,2765	2019,2938,2050,2969
	-0.24	0.31			
179	0.235	-0.580	0.735	2002,3032,2065,3095	2019,3268,2050,3299
	-0.31	0.29			
184	0.299	-1.024	1.659	2112, 557,2175, 620	2126, 794,2157, 825
	-0.20	-0.21			
185	0.359	-1.050	1.876	2112, 722,2175, 785	2126, 959,2157, 990
	0.13	-0.14			
186	0.351	-1.432	0.753	2112, 887,2175, 950	2127,1124,2158,1155
	0.00	-0.40			
187	0.710	-0.334	1.421	2112,1052,2175,1115	2127,1288,2158,1319
	0.29	0.14			
189	0.290	-0.879	1.450	2112,1382,2175,1445	2127,1618,2158,1649
	-0.07	-0.07			
190	0.267	-0.775	1.852	2112,1547,2175,1610	2127,1783,2158,1814
	0.04	0.22			
192	0.452	-1.050	0.860	2112,1877,2175,1940	2128,2113,2159,2144
	-0.35	0.01			
194	0.340	-0.155	1.293	2112,2207,2175,2270	2128,2443,2159,2474
	0.33	0.21			
195	0.413	-0.344	1.139	2112,2372,2175,2435	2128,2608,2159,2639
	0.01	-0.08			
196	0.274	-0.560	0.470	2112,2537,2175,2600	2129,2773,2160,2804
	-0.35	0.12			
197	0.309	-0.152	0.319	2112,2702,2175,2765	2129,2938,2160,2969
	-0.07	-0.18			
198	0.230	0.058	0.464	2112,2867,2175,2930	2129,3103,2160,3134
	0.02	-0.20			
199	0.460	0.007	0.954	2112,3032,2175,3095	2129,3268,2160,3299
	-0.13	0.12			
207	0.492	-0.883	1.081	2222,1052,2285,1115	2237,1289,2268,1320
	0.30	-0.23			
210	0.427	-0.890	1.311	2222,1547,2285,1610	2237,1784,2268,1815
	0 49	-0.42			
212	0.424	-0.050	1.126	2222,1877,2285,1940	2238,2113,2269,2144
	0.23	0.11			
219	0.329	1.038	1.162	2222,3032,2285,3095	2239,3268,2270,3299
	0.49	-0.04			
225	0.211	-0.413	0.996	2332, 722,2395, 785	2347, 959,2378, 990
	-0.15	0.06			
231	0.323	-0.588	1.134	2332,1712,2395,1775	2348,1949,2379,1980
	0 33	0.16			
233	0.298	-0.305	1.121	2332,2042,2395,2105	2348,2279,2379,2310
	0.45	-0.19			

EDITOR MULTITEMPORAL SYSTEM
ANALYSIS OF CORRELATION OUTPUT (EXAMPLE)

234	0.325	-0.970	1.120	2332,2207,2395,2270	2348,2444,2379,2475
	-0.33	-0.36			
236	0.326	-0.394	1.201	2332,2537,2395,2600	2349,2774,2380,2805
	-0.01	0.34			
242	0.296	1.929	0.921	2442, 227,2505, 290	2456, 464,2487, 495
	0.36	-0.19			
247	0.585	0.039	1.063	2442,1052,2505,1115	2457,1289,2488,1320
	0.35	-0.18			
248	0.340	-0.366	1.474	2442,1217,2505,1280	2457,1454,2488,1485
	0.08	0.03			
250	0.204	-0.340	1.281	2442,1547,2505,1610	2458,1784,2489,1815
	0.19	0.45			
252	0.280	-0.469	1.367	2442,1877,2505,1940	2458,2114,2489,2145
	-0.02	0.15			
254	0.376	-0.073	1.199	2442,2207,2505,2270	2458,2444,2489,2475
	0.17	-0.42			
259	0.316	0.265	1.923	2442,3032,2505,3095	2459,3269,2490,3300
	-0.06	0.12			
262	0.185	1.641	0.837	2552, 227,2615, 290	2566, 464,2597, 495
	-0.35	-0.05			
266	0.255	0.606	1.083	2552, 887,2615, 950	2567,1124,2598,1155
	0.31	0.17			
267	0.222	0.296	0.945	2552,1052,2615,1115	2567,1289,2598,1320
	0.21	-0.20			
268	0.245	-0.180	1.295	2552,1217,2615,1280	2567,1454,2598,1485
	-0.13	-0.07			
270	0.303	-0.287	0.586	2552,1547,2615,1610	2568,1784,2599,1815
	-0.14	-0.22			
271	0.389	-0.038	1.093	2552,1712,2615,1775	2568,1949,2599,1980
	0.09	0.07			
273	0.287	-0.371	1.709	2552,2042,2615,2105	2568,2279,2599,2310
	-0.39	0.24			
274	0.203	-0.179	1.697	2552,2207,2615,2270	2568,2444,2599,2475
	-0.30	-0.00			
275	0.686	0.063	0.857	2552,2372,2615,2435	2569,2609,2600,2640
	-0.18	-0.08			
276	0.283	0.184	1.483	2552,2537,2615,2600	2569,2774,2600,2805
	-0.18	0.29			
277	0.342	0.154	1.705	2552,2702,2615,2765	2569,2939,2600,2970
	-0.33	0.25			
288	0.316	0.400	1.230	2662,1217,2725,1280	2677,1454,2708,1485
	0.09	0.00			
291	0.383	0.556	1.281	2662,1712,2725,1775	2678,1949,2709,1980
	0.34	0.32			
292	0.279	-0.011	1.101	2662,1877,2725,1940	2678,2114,2709,2145
	-0.28	-0.11			
293	0.389	0.049	1.565	2662,2042,2725,2105	2678,2279,2709,2310
	-0.30	0.10			
295	0.500	0.330	0.789	2662,2372,2725,2435	2679,2609,2710,2640
	-0.24	-0.20			
306	0.296	0.616	0.740	2772, 887,2835, 950	2787,1124,2818,1155

```
       -0.37   0.27
309     0.289     0.124     0.035  2772,1382,2835,1445  2788,1619,2819,1650
       -0.41  -0.26
310     0.208     0.122     0.388  2772,1547,2835,1610  2788,1784,2819,1815
       -0.38  -0.18
311     0.254     0.787     1.120  2772,1712,2835,1775  2788,1949,2819,1980
        0.28   0.28
313     0.280     0.656     0.972  2772,2042,2835,2105  2788,2279,2819,2310
        0.02  -0.42
314     0.278     0.432     2.140  2772,2207,2835,2270  2788,2444,2819,2475
       -0.30   0.46
315     0.339     1.068     0.978  2772,2372,2835,2435  2789,2609,2820,2640
        0.22   0.01
327     0.382     1.092     0.397  2882,1052,2945,1115  2897,1289,2928,1320
        0.07  -0.04
329     0.342     1.031     0.354  2882,1382,2945,1445  2898,1619,2929,1650
        0.26   0.32
335     0.293     1.345     0.928  2882,2372,2945,2435  2899,2609,2930,2640
        0.29   0.06
336     0.401     1.075     0.722  2882,2537,2945,2600  2899,2774,2930,2805
       -0.09  -0.48
338     0.455     1.869     1.779  2882,2867,2945,2930  2899,3104,2930,3135
        0.50  -0.14

#O
 output file is MO-MT.CHNL-4/OVERLAY-PARAMS [New file]

DO YOU WISH TO CONTINUE (Y OR N)? N

3!Q
2!Q
!Q
USED 7:21 IN 30:10
@
```

FORMATTING OF EIGHT-CHANNEL TAPE

The final step in the creation of a multitemporal tape is to use the PL/I program OVER3 to format an eight-channnel tape on the IBM 370. The input to OVER3 consists of two four-channel EDITOR tapes, one for the primary scene and one for the secondary scene, and 32 overlay parameters which appear in the SYSIN input stream.

The OVER3 SYSIN input data is specified in PL/I data-directed format. The following is an example of OVER3 SYSIN data:

```
N=1092,W=1827,S=2981,E=3327
F1A=-2.306323,F1B=.001134772,F2A=-1.183024,F2B=.0007119611
C1(1) = 2032.945,C1(2) = 882.5245,C1(3) = .7143793,
C1(4) = -.6600814,C1(5) = 1.851665,C1(6) = -.01500387,
C1(7) = -.6804507,C1(8) = -.03415236,C1(9) = .08915249,
C1(10) = .2149834,
C2(1) = 1880.406,C2(2) = 3.412875,C2(3) = 1404.912,
C2(4) = .4726703,C2(5) = -.1991254,C2(6) = 1.454399,
C2(7) = -.9167777,C2(8) = -.0817734,C2(9) = .04173463,
C2(10) = -.8367744,
PR = 2002, PC = 1712, SR = 2002, SC = 1930;
```

The four parameters N, W, S, and E specify the primary scene coordinates of the window for which an eight-channel tape is to be created. The remaining 28 parameters are obtained from the final overlay parameter file, where they appear in the same data-directed format as is required for OVER3 input. This permits rapid preparation of the OVER3 input stream if some type of file transfer capability is available between BBN and the IBM computer on which OVER3 resides. For example, the following procedure uses a Texas Instrument (TI), Model 765 terminal, with bubble memory storage, to create SYSIN input for OVER3:

1. At BBN use program <SRS>TIXMIT to record the final overlay parameter file in the TI bubble memory.
2. Logout of BBN and logon to TSO (Time Share Option) at the IBM computer.
3. Enter the TSO text editor and playback the stored data from the TI bubble memory.
4. Edit in N, W, S, and E values and then save text on a disk file data set.

For a given pair of primary and secondary scenes, two eight-channel tapes instead of one will have to be made if either of the following situations occurs:

1. the number of columns in the window which is common to the user-defined primary and secondary scene windows of interest exceeds 2048, or
2. the amount of tape required to write the eight-channel data exceeds the length of a single tape reel.

The number of feet of tape required to write to tape a window of eight-channel data is given by the following formula:

FEET = [(E - W + 1)*8/DENSITY + .6] * (S - N + 1) / 12

where DENSITY is the tape recording density in bits per inch. For tapes sent to BBN, DENSITY = 1600. The amount of tape usable for data storage on a single tape reel is 2380 feet for a standard length tape (e.g. Scotch 700) and 3180 feet for a long tape (e.g. Scotch 701).

When two eight-channel tapes have to be created, two OVER3 jobs are submitted. The two jobs differ only in their N, W, S, and E values and in the VOL/SER number of their respective output tapes. The two sets of N, W, S, and E values define in primary scene coordinates two, usually overlapping, windows of eight-channel data whose union is the common window of desired multitemporal data. Typically, one eight-channel tape is created for the left side of the multitemporal window and the second eight-channel tape for the right side.

The TSO c-list RAD06.CCT.CLIST(OVER3) prepares JCL for OVER3 and submits the JCL for background execution. The following example illustrates the use of the c-list and lists the generated JCL.

EDITOR MULTITEMPORAL SYSTEM
FORMATTING OF EIGHT-CHANNEL TAPE (EXAMPLE)

```
EX 'RAD06.CCT.CLIST(OVER3)'
THIS PROC CREATES AN 8-CHANNEL EDITOR TAPE
THE "CARD" INPUT FILE MUST HAVE THE FOLLOWING ATTRIBUTES:
        LRECL(80)
        BLKSIZE(80)
        RECFM(FB)
DO YOU WANT TO SEE DEFAULTS (Y OR N)? Y
DEFAULT VALUES USED ARE:
    WCC JOB NAME = JOBNAME(RADOVER)
    WCC PRIORTY USED = PRTY(1)
    WCC CLASS = CLASS(N)
    WCC TIME = TIME(10)
    JCL MESSAGE PRINT = MSGCLAS(A)
    SYSPRINT FILE = SYSOUT(A)
IF YOU DON'T LIKE THESE VALUES, BREAK AND USE THE
TSO EXECUTE COMMAND PARAMETER OVERRIDE:
    EXEC CCT.CLIST(OVER3) 'PARM1(XX) PARM2(XX)'

SOME PARAMETERS MUST BE ENTERED FROM THE TERMINAL:
  THE PROJECT CODE FOR THIS TAPE =4166771889
IS PRIMARY INPUT TAPE A LIBRARY TAPE? Y
THE VOL/SER NUMBER OF PRIMARY INPUT TAPE =W01925
THE DSN OF PRIMARY INPUT TAPE =RAD06.EDITOR
IS SECONDARY INPUT TAPE A LIBRARY TAPE? (Y OR N)Y
THE VOL/SER NUMBER OF SECONDARY INPUT TAPE =0155448
THE DSN OF SECONDARY INPUT TAPE =RAD06.EDITOR
THE DSN OF "CARDS" FILE FOR OVERLAY PARAMETERS =RAD06.RSS.FB.DATA(ROVER)
  THE VOL/SER OF OUTPUT USER REEL TAPE =UR0787
  THE DSN ON OUTPUT USER REEL =RAD06.OUT
  THE PROGRAMMER NAME ( <10 CHARS) =SIGMAN
WOULD YOU LIKE TO MAKE A COPY OF YOUR OUTPUT TAPE? (Y OR N)Y
THE DSNAME ON COPY TO BE MADE=RAD06.OUT
IS COPY TO A USER REEL? (Y OR N)N
TIME-10:44:11 CPU-00:00:02 SERVICE-5187 SESSION-00:20:32 OCTOBER 26,1979
THERE ARE 099 TSO USERS ACTIVE.
JOB RADOVER(JOB06875) SUBMITTED  **  FREE ALL FILES  **
 SIGMAN.CNTL
00010 //RADOVER JOB (4166771889,    <ESCS-SRD SIGMAN>
00020 //       TS000,,,,,,,0),'SIGMAN X73131',  <TSO SUBMIT>
00030 //       MSGCLASS=A,NOTIFY=RAD06,
00040 //       MSGLEVEL=(2,1),TIME=10,PRTY=1,
00050 //       CLASS=N
00060 /*ROUTE  PRINT RMT9
00070 //* CREATE 8-CHANNEL EDITOR TAPE
00080 //S1 EXEC SRSRLGO,P=OVER3,REGION=350K
00100 //STEPLIB DD DSN=SYS1.OPT.PL1BASE,DISP=SHR
00110 //        DD DSN=SYS1.OPT.PL1BASE,DISP=SHR
00120 //        DD DSN=RAD01.PLIB.LOAD,DISP=SHR
00130 //GO.SYSPRINT DD SYSOUT=A
00140 //GO.PL1DUMP DD DUMMY
00150 //GO.SYSIN DD DSN=RAD06.RSS.FB.DATA(ROVER),DISP=SHR
```

EDITOR MULTITEMPORAL SYSTEM
FORMATTING OF EIGHT-CHANNEL TAPE (EXAMPLE)

```
//GO.P   DD DCB=OPTCD=C,UNIT=2400,VOL=SER=W01925,
//          LABEL=(,SL),DSN=RAD06.EDITOR
//GO.S DD DCB=OPTCD=C,UNIT=2400,VOL=SER=0155448,
//         LABEL=(,SL),DSN=RAD06.EDITOR
//GO.O DD DCB=OPTCD=C,UNIT=2400,VOL=(,RETAIN,SER=UR0787),
//         LABEL=(,NL),DSN=RAD06.OUT
//GO.OUT DD UNIT=AFF=O,VOL=(,RETAIN,SER=UR0787),
//           DCB=(OPTCD=C,DEN=3),LABEL=(,NL),DISP=(MOD,PASS),
//           DSN=RAD06.OUT
/*
//S2 EXEC SRSRLGO,P=ERTSDUPE,REGION=350K,PARM='/ISA(5K),R'
//STEPLIB DD DSN=SYS1.OPT.PL1BASE,DISP=SHR
//         DD DSN=SYS1.OPT.PL1LIB,DISP=SHR
//         DD DSN=RAD01.PLIB.LOAD,DISP=SHR
//GO.SYSPRINT DD SYSOUT=A
//GO.PL1DUMP DD DUMMY
//I DD DCB=OPTCD=C,UNIT=2400,VOL=(,RETAIN,REF=*.S1.GO.OUT),
//       DSN=*.S1.GO.OUT
//O  DD DCB=(OPTCD=C,DEN=4),UNIT=2400,
//       VOL=(,RETAIN),LABEL=(,SL),
//        DSN=RAD06.OUT
//OUT DD UNIT=AFF=O,VOL=(,RETAIN,REF=*.O),
//       LABEL=(,SL),DISP=(MOD,PASS),
//       DSN=*.O,DCB=(OPTCD=C,DEN=4)
//S3 EXEC PGM=IEFBR14
//DUMS DD DSN=*.S2.O,UNIT=2400,VOL=(PRIVATE,RETAIN,
//  REF=*.S2.O),DISP=(OLD,KEEP,KEEP)
//
ENTRY (A) RAD06.SIGMAN.CNTL DELETED
JOB RADOVER(JOB06875) WAITING FOR EXECUTION, IN HOLD STATUS
READY
```

EDITOR MULTITEMPORAL SYSTEM
PROGRAM DESCRIPTION

PROGRAM DESCRIPTION

The EDITOR multitemporal system is made up of five programs, MCPEDT, GBLOCK, GCORR, EDITCR, and OVER3. MCPEDT, GBLOCK, and EDITCR are directly callable in EDITOR and thus run on TENEX systems. GCORR runs on the ILLIAC-IV. OVER3 runs on IBM 370 systems.

MCPEDT is used to get the inital overlay by digitizing the photographic products associated with LANDSAT images. MCPEDT then uses this initial overlay to generate the block coordinates. GBLOCK reads the blocks from tape. GCORR performs the correlation on the ILLIAC-IV. EDITCR is used to analyze the correlation output and create the final parameter file if the correlation was successful. OVER3 uses the final parameter file to create the multitemporal tape.

All of these programs will be described below. Source listings of the programs are available from the authors of this paper.

MCPEDT

MCPEDT stands for multitemporal corresponding points editor. It is called using the EDITOR command sequence

```
!  REGISTRATION AND DIGITIZATION FUNCTIONS
2! SCENE TO SCENE REGISTRATION
3! EDIT AND EVALUATE CORRESPONDING POINTS FILE
```

The program MCPEDT is written in RATFOR. It consists of a main program and 26 subroutines. Of these only the major routines will be discussed here. In addition, there are 12 object modules which must be loaded with MCPEDT when creating the "SAV" version of the program. These routines fall mainly into one of the following categories : 1) digitizer functions, 2) plotter functions, or 3) I/O or string handling functions, and will therefore not be discussed in this paper.

MAIN PROGRAM

Upon entering the program, there are 18 commands immediately available to the user. These commands are:

```
ADD CP PAIRS
ANALYSE VIA LEAST-SQUARES
BACKUP NOW
BR..BACKUP RESTORE AFTER CRASH
CALCULATE WINDOW IN SECONDARY IMAGE
CONTAINING WINDOW FOR CPS
DELETE CPS
EXPUNGE DELETED POINTS FOREVER
GENERATE LIST OF BLOCKS FILES
LIST CPS
LD..LIST DELETED CPS
NO BACKUP DESIRED
QUIT
```

READ OLD FILE OF CPS AND ADD THEM TO CURRENT LIST
RESTART PROGRAM
SUMMARIZE
UNDELETE CPS
WRITE UNDELETED CPS TO A FILE

Most of these commands require calls to other procedures to carry out the specified function. A few commands are carried out within the main program. These are commands to delete, expunge or undelete corresponding points.

GOVRLY

Subroutine GOVRLY is used to digitize the corresponding points from two greyscales or transparencies. It is entered through the command ADD CP PAIRS. After entering GOVRLY, subroutines INITDIG and BUTCAL are called to set the proper digitizer variables. Next the user is asked whether the images to be digitized are greyscales or full scenes. If the images are full scenes the user is asked for the transparency format, i.e. whether or not the scenes are in EDIPS format, and if not which LANDSAT mission the data is from. Next the user is asked whether the images are overlaid or side-by-side. Subroutine CALIB4 is then called in order to calibrate the two images. Within subroutine CALIB4, arrays BLINE1, BCOL1, BLINE2, and BCOL2 are set. These arrays are used to transform the digitized values into values which correspond to the coordinates of the image. If the images are overlaid, the user is then asked to digitize each point on the overlaid images. If the images are side-by-side, the user is asked to digitize the two corresponding points on the primary and secondary images. The digitized values are then transformed into image coordinates and these values are checked to make sure that they are within the appropriate image. The row and column coordinates for each pair of points are stored in arrays LINE1, COL1, LINE2 and COL2. These point coordinates can later be written out to a file using the WRITE UNDELETED POINTS TO A FILE command.

<u>LSQEDIT</u>

Subroutine LSQEDIT performs various operations associated with regression analysis of multitemporal control points. These operations include calculation of regression coefficients, sorting and listing of residuals, writing coefficients to disk, and plotting of control points and residuals. LSQEDIT is called from the main program through the command ANALYZE VIA LEAST SQUARES. Processing options in LSQEDIT are the following:

AUTO DELETE UNTIL LSQ ERROR <= A VALUE
DELETE CPS WITH LSQ ERROR > A VALUE
EVALUATE LSQ POLYNOMIAL
OUTPUT POLYNOMIAL TO OVERLAY PARAMETER FILE
PLOT CPS
PRINT LSQ POLYNOMIAL COEFFICIENTS
QUIT
S...SORT UNDELETED CPS BY MAXIMUM LSQ ERROR
SA..SORT ALL CPS BY LSQ ERROR
SPECIFY TERMS OF LSQ POLYNOMIAL
SUMMARIZE

The majority of these options are performed by subroutines called by LSQEDIT. The major subroutines among these are the following:

POLY, LSQ1

POLY sets up arguments for LSQ1, which performs bivariate regression analysis. When POLY is called from LSQEDIT via the option AUTO DELETE UNTIL LSQ ERROR <= A VALUE or the option DELETE CPS WITH LSQ ERROR > A VALUE or the option EVALUATE LSQ POLYNOMIAL, then POLY calls LSQ1 to compute the transformation from primary scene to secondary scene. When POLY is called from GETWIN, POLY calls LSQ1 to compute the inverse transformation.

LSQ1EVAL LSQ1EVAL evaluates a bivariate polynomial.

PLOTCP PLOTCP is called from LSQEDIT via the option PLOT CPS and creates a plot file for a plot of corresponding point locations. The resulting plot locates corresponding points with respect to both primary and secondary scenes.

GETWIN GETWIN computes the coordinates of the overlap window with respect to both the primary and secondary scenes.

RESID RESID is called by LSQEDIT via the option GRAPH RESIDUALS. The independent variable is either primary-scene line or primary-scene column value. The dependent variable is either line residual or column residual. Residuals for deleted points are not graphed.

GETTERMS, IGETTERMS

GETTERMS is called by LSQEDIT when LSQEDIT is entered for the first time and via the option SPECIFY TERMS OF LSQ POLYNOMIAL. GETTERMS asks the question 'USE A GENERAL POLYNOMIAL?'. If the user responds affirmatively, the degree of the polynomial is queried and the number of terms and the term exponent values are set accordingly. If the user responds negatively, IGETTERMS is called in order that the user can specify each individual term desired.

GENBLK

Subroutine GENBLK is written in FORTRAN but is included within the RATFOR source code. The routine is used in the second step of creating the multitemporal imagery and is called through the GENERATE LIST OF BLOCKS FILES command. Upon entering the subroutine, variables IFLAG and IFLAGH are set to zero. IFLAG is an indicator of whether or not the initial overlay parameter file has been read. IFLAGH is an indicator that a change has been made to the header of the primary output file. Next the user is asked for the number of blocks per row and the number of rows of blocks. These values are multiplied to make sure that their product doesn't exceed 340, the maximum number of blocks which the ILLIAC-IV correlation program can handle. Next the user is asked for the initial overlay parameter file. The containing window coordinates for the primary and secondary windows are read from this file. A boundary of ten pixels in each direction is used on the primary scene so that none of the blocks gets too close to the edge of the scene. Also, the scaling factors and the least squares polynomials are read from this file. Next the subroutine CALCPQ is called to generate the least squares polynomial exponents. Flag IFLAG is then set to a value of one, indicating that the initial overlay parameter file has been read. The block centers for the primary file are then generated and stored in arrays PTL and PTC, for the rows and columns respectively. A check is made that the blocks do not overlap. If the blocks do overlap, either horizontally or vertically, the user is asked to reenter values for both the number of blocks per row and the number of rows of blocks. Next the user is asked for the tape format. This is done so that check values can be generated for the secondary scene data. First the user is asked if the transparencies (tapes) are in EDIPS format. If so, appropriate check values are generated. If not, the user is asked if the tapes are in USDA format, i.e. non-deskewed. If so, the user is asked which LANDSAT mission the data is from. The program uses this information to generate appropriate check values. If the tapes are not USDA format, i.e. deskewed, the user is asked to enter check values for the secondary scene. In each case, the check values are adjusted so that again there is a boundary of ten pixels in each direction.

The primary and secondary output files are then opened and initial headers are written out. The program then iterates over the rows, evaluating the secondary block centers from each primary block center and adding appropriate constants to generate upper left and lower right block coordinates. The secondary values are checked for scene containment. If a block lies outside the scene, that block is dropped from both files. For non-deskewed tapes, horizontal and vertical constants are added to the block coordinates to adjust for the tape format. When either a single block or an entire row of blocks is dropped the user is notified. After iterating over the rows, the primary and secondary headers are rewritten when necessary.

CALCWIN2

Subroutine CALCWIN2 implements the CALCULATE WINDOW IN SECONDARY IMAGE command. This subroutine, like GENBLK, is written in FORTRAN. CALCWIN2 is similar to GENBLK in that it reads the initial overlay parameter file and evaluates secondary row and column coordinates from the primary values. The primary values, however, are not computer generated values, they are digitized. The output windows from this routine can be used in conjunction with a greyscaling program to check if the least squares polynomial will create a good overlay or not. CALCWIN2 is generally not required to create a multitemporal digital data set.

Upon entering CALCWIN2, subroutines INITDIG and BUTCAL are called to set the proper digitizer variables. Then the user is asked for the initial overlay parameter file. The program reads this file, ignoring the containing window information and reading only the values of the scaling factors and the coefficients of the least squares polynomials. Subroutine CALCPQ is then called to generate the least squares exponents. The user is then asked about the tape format, i.e. whether or not the tapes are in EDIPS format, and if not, which LANDSAT mission the data is from. If the tape is not in EDIPS format, the user is also asked if the tape is an old reformatted tape. If the tape is non-deskewed the user should answer positively to this question. This will set appropriate horizontal and vertical constants to be added to the image coordinates to adjust for the tape format. Next the user is asked to place the image on the digitizer tablet and subroutine CALIB4 is called to calibrate the image. The user then digitizes the individual features. For each feature, the digitized values are transformed into image coordinates and these values are checked to make sure that they are within the image. Subroutine LSQEVAL is called to evaluate the secondary point coordinates and both the primary and secondary row and column coordinates are printed out.

CALCPQ

Subroutine CALCPQ is called by subroutines GENBLK and CALCWIN2 to generate the exponents for the least squares polynonmials. The parameter, NC, is the number of cooeficients in the polynomial, e.g. a second degree polynomial has six coefficients.

EDITOR MULTITEMPORAL SYSTEM
PROGRAM DESCRIPTION (GBLOCK)

GBLOCK

GBLOCK is the program used to read the blocks for correlation from tape to create the files of blocks which are the input for the ILLIAC-IV correlation program GCORR. GBLOCK is called using the EDITOR command sequence

```
! TAPE READING TO CREATE WINDOW FILES
2! CORRELATION BLOCKS READ FROM TAPE
```

GBLOCK is executed once for the creation of the file of 64 by 64 primary scene blocks and again for the creation of the file of 32 by 32 secondary scene blocks.

GBLOCK has one main source file, GBLOCK.SAI, written in SAIL. In addition, COMS.SAI is required for some I/O functions. GBLOCK is made up of a long main program and a few procedures. Both the main program and the procedures will be described.

MAIN PROGRAM

The user is first asked for the block parameter file, this being the file of block coordinates created by MCPEDT. If, instead of a file name, "*ENTER" is entered, the user will be expected to input the block coordinates from the terminal in the usual format for specifying a window in EDITOR. Assuming a file of block coordinates is used, the header of the file is read and checked to be sure that the file is indeed a file of block coordinates, that the block size is either 64 by 64 or 32 by 32, and that no row of blocks has more than the maximum number of blocks, currently 60.

The user is asked to select the channel from which the data will be taken for each pixel. The value entered is checked to be an integer from 1 to 4. The shift distance needed to get the value for that channel to the rightmost part of a word is computed. The user is also asked if the blocks are to be made suitable for use with the gradient correlation program by adding extra edge data.

The tape input file is opened and the first record is read and checked to be sure the tape is a 4-channel, raw data, EDITOR-compatible tape. The skeleton of the header of the output file of blocks is built and written to the output file as a place holder.

For each row of blocks, the data from the input tape is read and the blocks from that row are written to the output file. The minimum and maximum image lines for that row of blocks are read from the input file of block coordinates and stored in RW1 and RW3. The image line and column coordinates for each block in the row are also read and stored in the arrays W1, W2, W3, and W4. If the blocks are to be used with the gradient correlation program, one is subtracted from the first image line of each block as stored in W1 and also from the first image line of the row of blocks as stored in RW1 to account for the extra upper edge. Similarly, one is added to the last image line in each block as stored in W3 and the last image line of the row of blocks as stored in RW3 to account for the lower edge. The pointers BKPT and EPT, used to store the main part of the block data and the left and right edge data respectively, are set to point into BLOCKS and EDGES at the proper places for the blocks. Finally, the tape is spaced to the record for the first image line of the row of blocks. Since the input tape is an EDITOR-compatible tape, there is one record for each image line.

For each row of blocks, all records for which there is data to be inserted in any block are read. As each record is read, a check is made of each block to see if any data from that record should be placed in that block by determining if the image line for the record lies between the first and last image lines for the block. For all blocks taking data from a record, data from the appropriate areas of the record are deposited using the pointer BKPT for the main part of the block and EPT for the left and right edges. After all records for the row of blocks have been read, the block data for that row is written to the output file of blocks and the procedure is repeated for the next row of blocks. When all blocks have been written, the array HEADER, which now contains all block coordinates, is written to the header of the output file and the output file is closed.

TAPERR

Procedure TAPERR prints out status information in case of a tape reading error.

SPACET

Procedure SPACET spaces the tape, forward or backward, to the record specified by the parameter PLACE. CURREC is assumed to contain the current position of the tape.

GETCOORD

Boolean procedure GETCOORD is used to read coordinates entered by the user if the program is operating in the mode in which the user enters block coordinates from the terminal. The procedure returns TRUE if some coordinates were entered and FALSE if a carriage return was entered to indicate that no more coordinates will be entered.

PRINCIPAL DATA AREAS

TJFN,BJFN,CBJFN are the JFNs for the input tape file, the input file of block coordinates, and the output file of blocks respectively.

CURREC is the record number of the next record to be read from tape given the current position of the tape.

RW1,RW2,RW3,RW4 describe the containing window for the current row of blocks.

TW1,TW2,TW3,TW4 describe the containing window for the entire tape.

UCHAN,CSHIFT are the channel being used and the shift required to bring the pixel value for that channel to the rightmost part of a word.

BLKBUF is an array used as a buffer while reading the input file of block coordinates.

EPT is an array of pointers, one for each block in the current row of blocks, into the left and right edge data describing where the next edge data should be placed for the block.

EDGES is a two-dimensional array of data buffers for the left and right edge data for all blocks in the current row of blocks.

BKPT is an array of pointers, one for each block in the current row of blocks, describing where the next data for the main part of the block should be inserted.

BLOCKS is a two-dimensional array of data buffers for the block data, exclusive of left and right edges, for all blocks in the current row of blocks.

HEADER is an array holding the header of the output file of blocks. HEADER is built as the block coordinates are read from the input file of block coordinates.

BUF is the input buffer for reading the tape and holds a single tape record.

GCORR

GCORR is the program which performs the correlation of the block pairs on the ILLIAC-IV. It is written in ASK, the ILLIAC-IV assembly language, and runs mostly in 32-bit mode for greater speed. The program requires two input files, the file of 64 by 64 blocks from the primary scene and the file of 32 by 32 blocks from the secondary scene. There is one output file which contains the correlation values and shifts for all block pairs.

GCORR calls only one subroutine, which takes the square root of its argument. It is rather standard and will not be described. The description will cover the main program, which does the correlation. GCORR has only one source file, GCORR.ASK.

INPUT AND CONVERSION

The first ILLIAC disk page of each of the two input files is read to check the header information. The check is to see that the file types are correct, that the number of blocks does not exceed the maximum allowed (340), and that the number of blocks in the two files is the same. If an error is found, an error indicator is put into register TRO and also into the HALT instruction used to terminate execution and execution is terminated immediately.

Assuming that the files are correct, the header of the file of 64 by 64 blocks is transferred to OUTBUF to be used, with slight modification, as the header of the output file. When reading for the first time, 36 blocks from the primary file and

20 blocks from the secondary file are read. These numbers are chosen so that no block will be only partially in memory.

Before processing a block pair, a check is made to see that both blocks are in memory. If either is not, at most 128 blocks will be read from the appropriate file. This number of blocks is the same for both files so that less data is in fact read from the file of 32 by 32 blocks. Of course, if there are fewer than 128 blocks remaining in a file, the number of blocks remaining will be read.

For each block, the main part of the block is converted to 32-bit floating point and stored in the array RAW. This main part of the block is 66 image lines of 64 pixels each for the 64 by 64 blocks and 34 image lines of 32 pixels each for the 32 by 32 blocks and thus includes the extra top and bottom edges but not the extra left and right edges. The extra left and right edges, converted to 32-bit floating point, are stored in EDGE. Since only one channel is used for each pixel, only one 32-bit floating point number is generated so that two pixel values may be stored in each PE in each memory row. Thus, one memory row holds exactly two image lines from a 64 by 64 block and four image lines from a 32 by 32 block. Including the extra left and right edge elements would destroy this convenient division of image lines into memory rows.

Once a block has been read into RAW and EDGE, the gradient is computed. The computation is similar for the 64 by 64 blocks and the 32 by 32 blocks, but some of the details are different.

The resulting gradient images are stored in G64 for the 64 by 64 blocks and in G32 for the 32 by 32 blocks.

The vertical portion of the gradient for the 64 by 64 blocks is computed by subtracting successive memory rows and squaring since each memory row contains data for two image lines. For the horizontal portion of the gradient, the top image line is no longer needed so each memory row is routed by 32, the length of an image line, and the left most 32 words of the next memory row, which are the next image line and are now routed to the rightmost part of the memory row, are brought up as the rightmost part of the memory row. Of course, this procedure also eliminates the extra bottom image line. The horizontal portion of the gradient is then computed for both pixels in each PE by squaring after subtracting corresponding parts (inner and outer) of the next PE to the right gotten with a route of 63. Care is taken to exclude left and right edges during this computation. Finally, the left and right edge elements are routed from EDGE to the proper PEs to compute the horizontal portion of the gradient at the left and right edges of the blocks.

For the 32 by 32 blocks, there are four image lines per memory row. Therefore, for the vertical portion of the gradient, the elements desired are 32 PEs apart. These are lined up by routing and the vertical portion of the gradient is computed. For the horizontal portion of the gradient, the extra first and last image lines occupy only 16 PEs so routing takes place accordingly with one image line always from the next memory row down being brought up to compute the horizontal portion of the gradient, exclusive of the left and right edges of the image lines, in parallel. The edge elements are taken from EDGE and routed to the proper PEs to complete computation of the horizontal portion of the gradient.

CORRELATION

The correlation is done using the gradient image of the blocks as stored in G64 and G32. Certain elements of the correlation formula depend only on the 32 by 32 secondary block and thus remain the same for all 32 by 32 subblocks of the 64 by 64 primary block. These are computed and stored in XBAR and X2BAR. Next, the 32 by 32 subblocks of the 64 by 64 primary image block must be accessed and aligned with the 32 by 32 secondary block so that computation of the remaining elements of the correlation formula may be done in parallel.

The procedure is to access the first 16 words of each image line and use four of these 16-word pieces to make one memory row of YL. YL then contains enough information, and is correctly aligned with respect to the 32 by 32 block, to compute the components of the correlation formula dependent on the 64 by 64 block for one column of the correlation surface. For succeeding columns, the memory rows constituting G64 are moved left by one pixel position and YL is rebuilt. Note that since 32-bit mode is used, this movement really involves taking the outer part of each PE, placing it in the inner part, and then taking as the new outer part the inner part of the next PE to the right (next higher numbered PE). The components of the correlation formula dependent on the subblock of the 64 by 64 block are computed and stored in YBAR, Y2BAR, and XYBAR with the value associated with each subblock being stored in only one 32-bit word.

Finally, with all these partial results, the complete correlation surface is generated with 128 different correlation values at a time being computed in parallel. The sign bit of each result is reset to obtain the absolute value. The correlation values are stored in CORR. Then, a value from 0 to 1088 is assigned as the number of each correlation and stored in YBAR which is no longer needed since the complete correlations have been computed. This number will remain associated with the correlation and is used to remember the original position of the value in the correlation surface.

When computing the values in CORR, care is taken that unused elements in the last memory row of CORR are set to zero since the total number of correlation elements is 1089, a number not evenly divisible by 128. Since the absolute value of the correlation function is used, all generated values will be non-negative and the zero values will not be candidates for the largest correlation. In searching for the highest correlation value, the search is first done in parallel in each PE. Then, by routing and comparing, the highest value of all is found. At each step, the associated correlation number is saved. The correlation number is used to compute the initial values of the horizontal and vertical shifts. Doing an integer divide of the number by 33 yields the horizontal shift from the left as the result and the vertical shift from the top as the remainder since the correlation surface was formed by columns but stored across memory rows. However, since there is no integer division on the ILLIAC-IV, a rather tedious routine must be performed to get those values. The values obtained are adjusted relative to center by subtracting them from 16. Finally, the interpolation on the shifts is performed. The correlation numbers previously saved are used to obtain the addresses of the neighboring correlations and the values are obtained, one at a time, through the CU. The two interpolations, for horizontal and vertical, are done in parallel.

OUTPUT

The correlation value and interpolated shifts for each block pair are stored in OUTBUF. The correlation values are each stored twice so as to occupy an entire ILLIAC-IV 32-bit word, making position indexing easier for storage of the results. The second ILLIAC-IV word for a block pair contains the interpolated shifts. All output values are saved in memory and written to the output file after correlation has been performed on all block pairs.

FILES

WINDOW is the input file of 64 by 64 blocks.

WINDOX is the input file of 32 by 32 blocks.

COEFFS is the output file of correlation results.

DATA AREAS

COMPWORD is a single word used as the word to store completion results for all I/O operations.

RAW holds the data for one 64 by 64 or one 32 by 32 block after it has been converted to 32-bit floating point but before the gradient has been computed. RAW contains the extra upper and lower image lines needed for gradient computation, but not the extra left and right edge pixels so as to have each memory row contain an integral number of image lines for the two sizes of blocks.

EDGE contains the extra left and right edge elements needed to compute the horizontal portion of the gradient.

G64,G32 hold the gradient images of one 64 by 64 and one 32 by 32 block respectively.

XBAR,X2BAR contain terms of the correlation formula which depend only on the 32 by 32 block and are thus computed once only for each block pair. XBAR is also used as a temporary during the interpolation.

YBAR,Y2BAR,XYBAR contain, for each of the 1089 subblocks of the 64 by 64 block, terms of the correlation formula which depend on the subblock. YBAR is also used as a temporary during the interpolation.

YL contains a subblock of the 64 by 64 block which is 32 pixels across and 64 pixels down so that it contains all subblocks for a column of the correlation surface.

CORR contains the correlation results for all 32 by 32 subblocks of the 64 by 64 block as correlated with the 32 by 32 block.

HEAD64,SPAGE64 are the areas used to read the data for the 64 by 64 blocks.

HEAD32,SPAGE32 are the areas used to read the data for the 32 by 32 blocks.

OUTHEAD,OUTBUF are the areas used to write the output parameter file of correlation values and shifts.

PEN contains the PE numbers, numbered from 0 to 63.

EDITOR MULTITEMPORAL SYSTEM
PROGRAM DESCRIPTION (EDITCR)

EDITCR

EDITCR is the program used to analyze the output from the ILLIAC-IV correlation job to determine if the correlation succeeded and, if so, to create the final parameter file for use in creating the multitemporal data set on the IBM 370. EDITCR is called using the EDITOR command sequence

```
! REGISTRATION AND DIGITIZATION FUNCTIONS
2! SCENE TO SCENE REGISTRATION
3! EXAMINE CORRELATION OUTPUT
```

The main source file for EDITCR is EDITCR.F4, written in FORTRAN. Only the routines in that file will be described. However, EDITCR also requires LSQ.REL and SIMEQ.F4 to handle computation of the least squares polynomials, XREAD.MAC and RDWR.MAC to handle I/O, and STRING.MAC to handle some string manipulation functions and some I/O.

EDITCR.F4 is made up of a lengthy main program and some subroutines. Both the main program and the various subroutines will be described.

MAIN PROGRAM

The main program first asks for the input file. The file is read using READH. The header of the file is checked to be sure the file is really an output file from an ILLIAC-IV correlation job. If so, the coordinates of the primary and secondary blocks are read from the header. The primary block coordinates are stored in IRWIN for rows and ICWIN for columns. Similarly, the secondary block coordinates are stored in JRWIN and JCWIN. In each case, only the coordinates of the upper left corner of each block are stored. The correlation values and shifts are read in and converted from ILLIAC-IV 32-bit form. The correlation values are stored in VAL, the horizontal shifts are stored in CPOS, and the vertcial shifts are stored in RPOS. If the second of the two correlation values is negative, the block is marked as deleted in ICR. Note that the second correlation value could only be negative if EDITCR had previously created the file. The ILLIAC-IV program GCORR always returns the two correlation values as the same value and always as non-negative.

The user is then asked to enter a command. The action for each command, except 'Q' for quit, is to jump somewhere, perform the tasks required by the command, and jump back to wait for another command. Most commands are handled right in the main program and the implementation of those commands will be now described. The implementation of the others will be described under the appropriate subroutines.

The deletion and undeletion of control points by specifying a range of control points is handled by the 'D' and 'U' commands. For these commands, GETRAN is called to get the range of control points to be acted on. Then the variable KUD is set to zero to indicate delete and one to indicate undelete and the value of ICR is set to the value of KUD for all control points in the range specified. POLY is called to recompute the least squares polynomials.

The deletion of control points with correlation value less than a value specified by the user is handled by the 'DC' command. For this command, the test value is obtained from the user and checked for validity. Then, for each undeleted control point, if the correlation value is less than the user-specified value, the control point is marked as deleted. DISBLK is called to show the user what happened.

The deletion of control points with absolute value of at least one of the two shifts greater than some value is handled by the 'DS' command. For this command, the test value of the shift is obtained from the user and checked for validity. Then, for each undeleted control point, the absolute value of the shifts from RPOS and CPOS are checked against the user-specified value. If either is greater, the control point is marked as deleted. DISBLK is called to show what happened.

Copying of control points from the input buffer to the output buffer is done using the 'CP' command. For this command, GETRAN is called to get the range of control points to copy. After checking that the copy would not put more than the maximum number of control points (340) in the output buffer, the copy takes place. The block coordinates are copied to IDBUF which will be used to construct the header of the output file. The correlation values and shifts are converted to ILLIAC-IV 32-bit form and stored in DBUF. If a control point has been marked as deleted, the second correlation value is set to -1.0 as a flag to indicate deletion.

EDITOR MULTITEMPORAL SYSTEM
PROGRAM DESCRIPTION (EDITCR)

Control points from the output buffer are written to an output file using the 'W' command. This command assumes everything to be written is in IDBUF and DBUF and writes it out using WRITEH.

Deletion of control points with the absolute value of either residual greater than a user-specified value is done using the 'DE' command. For this command, POLY is first called to recompute the least squares polynomials and the residuals. The test value is then gotten from the user and checked for validity. For each undeleted control point, the absolute values of the residuals taken from ERRORX and ERRORY are checked against the user-supplied value. If either is greater, the control point is marked as deleted.

Listing of control points in sorted order is done by the 'SC', 'SE', and 'SS' commands. The sorts are done on the absolute value of the correlation value, residual, or shift respectively. For the residuals and shifts, the maximum of the two values is used for the sort. The procedure is to find the undeleted control point not yet printed with the largest value of the key, print it, and mark it as printed in ISORT. After each group of five control points has been printed, the user is asked if the printout should continue.

The automatic deletion of control points based on residual values is handled by the 'AUTO' command. The test value for maximum residuals is obtained from the user and checked for validity. Each step of this command involves first calling POLY to compute the least squares polynomials and residuals from the undeleted control points. Then, the control point is found with the largest absolute value residual in either the horizontal or vertical direction, using the residual values from ERRORX and ERRORY. If this value is larger than the value specified by the user, the control point is marked as deleted and the procedure continues. Otherwise, the automatic deletion is stopped and summary information is printed.

DISBLK

Subroutine DISBLK displays the number of control points deleted by some operation, assuming this number is stored in NBLOX. Optionally, POLY is called to recompute the least squares polynomials and residuals and display them.

DISPCO

Subroutine DISPCO displays the correlation values, shifts, residuals, and block coordinates for each control point.

SPACEP

Subroutine SPACEP prints ten lines with one dot each to space the output from DISPCO up on the terminal.

GETRAN

Subroutine GETRAN queries the user for a range of control point numbers so that some operation may be performed on all the control points in the range. The low number in the range is returned in IBOT and the high number in the range is returned in ITOP. Checks are performed to ensure that the values are in the range of the number of control points currently in memory. Optionally, GETRAN will accept carriage return only (seen as both values set to zero) as equivalent to the range specifying all control points.

POLY

Subroutine POLY is used to compute, evaluate, and, optionally, display the least squares polynomials. For each undeleted control point, the row and column coordinates of the center of the primary image block, with the shifts subtracted, are put into PAY and PAX respectively. Similarly, for each undeleted control point, the center row and column of the secondary block are put into PBY and PBX respectively. The center of the primary block is taken as the coordinates of the upper left corner with 32 added to the row and column. The center of the secondary block is taken as the coordinates of the upper left corner with 16 added to the row and column. LSQ is alled to compute the coefficients of the polynomials and the associated scaling factors. LSQEVAL is called to evaluate the polynomials at each of the primary image points in PAX and PAY and compute points in the secondary image. The residuals, stored in ERRORX and ERRORY, are the difference between the values in PBX and PBY and the computed values.

OUTPLY

Subroutine OUTPLY is used to print the coeeficients of the least squares polynomials on the terminal. In addition it prints the scaling factors which are used in evaluating the least squares polynomials. These values are found in the common block LSQOUT, and are generated within subroutine POLY.

OUTPARM

Subroutine OUTPARM outputs a parameter file which will be used as input to the program OVER3. First the user is asked for an output file. Then the scaling factors and the polynomial cooeficients are written to the file. Finally the second half of the control points list is searched for the first undeleted control point. The secondary window coordinates of this control point are evaluated using the least squares polynomials and the primary and secondary row and column values are written out to the file. These values are used as check values by OVER3 to catch keypunching errors in the OVER3 input. If there are no undeleted block pairs in the second half of the list of control points an appropriate message is printed to the terminal. In this case, the primary row and column coordinates used are those corresponding to the control point number equal to half the number of control points.

DISPMP

EDITOR MULTITEMPORAL SYSTEM
PROGRAM DESCRIPTION (EDITCR)

This subroutine displays a grid showing which blocks have been deleted. The array I2ROW contains the row number for each block. I2ROW is created in the main program with the assumption that if the difference in the row coordinates of two consecutive blocks is greater than 20 then the blocks are in two different rows. A "." indicates that the block has been deleted, a "*" indicates that the block has not been deleted.

COMBINE

Subroutine COMBINE implements the 'CO' command and is used to combine the results from two output files created using the ILLIAC-IV block correlation program on different channels, and thus improve the resulting control point collection.

In order to use COMBINE the two files should contain the same number of block pairs and the same block coordinates. First the secondary file is read in. Note that in this context, the secondary file refers to an output file from the ILLIAC-IV block correlation program and not to a file of 32 by 32 blocks. Then for each secondary block pair the program tries to find a matching primary block pair, i.e. a block pair with the same row and column coordinates. The deletion flags and correlation values of the two block pairs are then checked and substitution of the secondary correlation value and shift values for the primary values is made where appropriate. A block pair is added to the primary file if the primary block pair has been deleted and the secondary block pair remains undeleted. A primary block pair is replaced by a secondary block pair if both remain undeleted and the secondary block pair has a higher correlation value than the primary. A message is printed for each block pair which is added or replaced. Also a message is printed telling the number of block pairs added, the number replaced and the total number of block pairs now in the merged set. Finally subroutine POLY is called to recompute the least squares polynomials.

WRITCP

Subroutine WRITCP writes an ASCII file of the undeleted block coordinates. The user is asked for an output file name. Block coordinates are then written out with an adjustment made to both primary and secondary block coordinates for the block center. In addition, the primary block coordinates are adjusted by the horizontal and vertical shift values.

DESCRIPTION OF MAJOR ARRAYS

IRWIN upper left row coordinate of primary blocks

ICWIN upper left column coordinate of primary blocks

JRWIN upper left row coordinate of secondary blocks

JCWIN upper left column coordinate of secondary blocks

CPOS the interpolated horizontal shift relative to the enter of the correlation surface, a positive shift indicates left of the center

RPOS	the interpolated vertical shift relative to the center of the correlation surface, a positive shift indicates above the center
VAL	correlation value of control points
ICR	flag indicating deletion of blocks, ICR(I)=0 indicates block I has been deleted
ERRORX	the difference between the secondary block column coordinate and the expected column coordinate given the least squares polynomial
ERRORY	the difference between the secondary block row coordinate and the expected row coordinate given the least squares polynomial
CBUF	input buffer (equivalenced to ICBUF)
DBUF	output buffer (equivalenced to IDBUF)

OVER3

OVER3 is a PL/I program which creates an eight-channel EDITOR tape from two four-channel tapes corresponding to a primary and secondary scene. By means of an input-defined bivariate cubic polynomial, each pixel inside a specified window of the primary scene is mapped to a point in the secondary scene. Pixels are then overlaid by the nearest-neighbor rule; i.e., a pixel in the primary scene is overlaid with the secondary-scene pixel which is closest to the point predicted by the overlay polynomial. The OVER3 source code resides in file RAD01.PLIB.SOURCE(OVER3) and object code resides in RAD01.PLIB.OBJECT(OVER3).

INPUT

The OVER3 input consists of two four-channel EDITOR tapes and 32 overlay parameters which appear in the SYSIN input stream. File SYSIN is declared with attributes LRECL=80, BLKSIZE=80, and RECFM=FB. The variables read from SYSIN are the following:

N, W, S, E	Primary scene coordinates of window for which eight-channel tape is to be created.
F1A, F1B, F2A, F2B	Scaling factors for overlay calculations.
Array C1	Coefficients for row polynomial.
Array C2	Coefficients for column polynomial.

PR,PC,SR,SC — Primary-scene row and column values and secondary-scene row and column values for check-point calculations.

OUTPUT

OVER3 output consists of an eight-channel EDITOR tape and line-printer output which includes the input-data echo, intermediate trace output, and the job log message "...I THINK OVERLAY WAS SUCCESSFUL" upon successful job completion.

MAIN PROGRAM

When a given row of overlaid pixels is being created, it is necessary for only one row of primary-scene pixels to be kept in memory. However, many rows of secondary-scene pixels may have to be in memory at the same time since a given row in the primary-scene can map into several different rows in the secondary scene. Consequently, the OVER3 main program maintains a queue for rows of secondary-scene pixels from which pixels are selected to overlay a single row of primary-scene pixels. The program variables which specify the size of this queue are the following:

MINMINX — Column number of west-most pixel in the queue.

MAXMAXX — Column number of east-most pixel in the queue.

MAXROWS — Number of rows in the queue.

EDITOR MULTITEMPORAL SYSTEM
PROGRAM DESCRIPTION (OVER3)

The processing steps which the OVER3 main program performs are the following:

1. Opens input tapes.
2. Reads SYSIN input data.
3. Computes MINMINX, MAXMAXX, and MAXROWS by performing overlay calculations for every tenth row and every tenth column in the overlay window.
4. Writes the output tape header.
5. Performs the following overlay cycle for each row of primary-scene pixels in the overlay window:
   a. Reads into memory a row of primary-scene pixels.
   b. Discards from the queue any unneeded rows.
   c. Reads in secondary-scene rows until the queue is full.
   d. Overlays pixels from the queue onto primary-scene pixels.
   e. Outputs a row of overlaid pixels.
6. Closes the output tape.

SUBROUTINES

The OVER3 main program calls the following subroutines:

CPTEST	Applies overlay polynomials to PR and PC. If results differ from SR and SC by more than five pixels the program terminates.
READHDR	Reads the input tape header.
WRITEHDR	Writes the output tape header.

REFERENCES

1. Anuta, P.E., Bauer, M., "An Analysis of Temporal Data for Crop Species Classification and Urban Change Detection", LARS Information Note 110873, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana, 1974.

2. Svedlow, M., McGillem, C.D., Anuta, P.E., "Analytical and Experimental Design and Analysis of an Optimal Processor for Image Registration", LARS Information Note 090776, Laboratory for Applications of Remote Sensing, West Lafayette, Indiana, 1976.