Date: January 28, 1977

Project Title: Design of a Low Cost Earth Resources System

Project No: A-1938

Project Director: Mr. N. L. Faust

Sponsor: NASA, George C. Marshall Space Flight Center, Alabama 35812

Agreement Period: From 1/13/77 Until 1/12/78 (Contr. Period)

Type Agreement: Contract No. NAS8-32397

Amount: $24,827 NASA + $1,000 GIT Contribution

Reports Required: Monthly Progress Reports; Final Report

Sponsor Contact Person(s):

Contractual Matters

Contracting Officer Rep.
NASA - Marshall Space Flight Center
Alabama 35812

Technical Matters

ONR Resident Representative
325 Hinman Research Building
Campus


Assigned to: Electromagnetics Laboratory

(School/Laboratory)

Copies To:

Project Director
Division Chief (EES)
School/Laboratory Director
Dean/Director - EES
Accounting Office
Procurement Office
Security Coordinator (OCA)
Reports Coordinator (OCA)

Library, Technical Reports Section
Office of Computing Services
Director, Physical Plant
EES Information Office
Project File (OCA)
Project Code (GTRI)
Other

CA-3 (5/76)
Design of a Low Cost Earth Resources System

A-1938

Mr. N. L. Faust

NASA, George C. Marshall Space Flight Center, AL 35812

5/31/78

Final Invoice and Closing Documents

Final Fiscal Report

Final Report of Inventions

Govt. Property Inventory & Related Certificate

Classified Material Certificate

Subcontract Closeout

Electromagnetics Laboratory (School/Laboratory)

COPIES TO:

Library, Technical Reports Section
Office of Computing Services
Director, Physical Plant
EES Information Office
Project File (OCA)
Project Code (GTRI)
K. E. Newkirk
DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

1 April 1977

Prepared for

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

by

Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332
I. INTRODUCTION

State and local government agencies, and private industry in Georgia and throughout the Nation are becoming increasingly aware of the advantages of digital processing of LANDSAT data. In the last several years the technical interest in the digital processing techniques of LANDSAT data has evolved from the "gee whiz" stage to the practical adaptation stage. Many agencies at least at the State level have the technical capability and the professional interest needed to operate an earth resources digital analysis system with the technical assistance of universities or private industry. To date, however, very little progress has been made in the transfer of the digital processing capability from NASA to user agencies. This failure is in part due to the complexity of the techniques used to analyze LANDSAT data and also in part due to the prohibitive costs associated with designing an Earth Resources Digital Analysis System (ERDAS). The Georgia Tech Engineering Experiment Station (EES) has designed and made operational a practical ERDAS system at Georgia Tech. The experience gained in the design of such a system is a valuable asset in the design of more efficient and less expensive versions of an ERDAS system that may become a necessary part of a regional, state or local LANDSAT applications program. Judicious choice of the most cost effective equipment and the most usable software will be necessary to justify the operational application of LANDSAT information.

The State of Georgia is on the threshold of approving the statewide, digital processing of LANDSAT data on the Georgia Tech ERDAS system. Many state agencies including the Department of Natural Resources, Department of Transportation, and the Office of Planning and Budget are beginning to realize that digital LANDSAT data is the only available data source that will meet both accuracy and timing considerations in the accomplishment of federal and state regional natural resource related programs.

A genuine interest is now being shown by most LANDSAT users including the LANDSAT information into an efficient data base manipulation system such as the IMGRID system developed at Harvard. This system is operationally efficient and is specifically designed for the layman. A keyword structure is used to access program options which interact with the data base. This type of analysis capability will enhance the value of LANDSAT digital data by combining it with other natural resource, economic, and political sources of data. In fact, any type of data that can be geographically referenced may be inserted into the data base and used later
Monthly Progress Report No. 3
For the Period March 1, 1977 through March 31, 1977

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

by
Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332
I. INTRODUCTION

EES plans to document requirements for the assembly of a low cost Earth Resources Digital Analysis System (ERDAS). The tasks under this project call for:

1) A review of existing software and hardware capabilities in the southeast.
2) A description of various hardware and software configurations
3) An estimate of continuing support requirements for an ERDAS system
4) The assembly of a FORTRAN software library for digital processing of LANDSAT data in an ERDAS system, and
5) The implementation of a limited version of a geographic data base manipulation program (IMGRID).

II. TECHNICAL PROGRESS SUMMARY

During this period EES continued investigations into the implementation of IMGRID on the NOVA II minicomputer system. Also during this period EES was able to determine a cost figure for the ERDAS system by comparing the amount of time a program needed on the NOVA to the cost of a duplicate program on the Georgia Tech CYBER 74 using identical data and peripheral devices. The cost figure arrived at was approximately ten dollars per hour for the NOVA excluding manpower costs. Other test programs and data sets will be used to verify the preliminary cost data.

Further development was also done on software development and documentation for the ERDAS system.

A subcontract was worked out with METRICS, Inc. for the part of the contract associated with evaluating the present digital analysis capability in the Southeast for processing LANDSAT data.

III. PROBLEMS

There were no new problems encountered during this period.
IV. WORK PLANNED FOR NEXT PERIOD

During the next period, EES plans to finalize the terms of the subcontract with METRICS, Inc. and ask for authorization from MSFC. Further work will be done on software documentation and IMGRID implementation. Further computer runs will be made to verify the preliminary cost figures for the ERDAS system.

V. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Services</td>
<td>$23,327</td>
<td>$6,397</td>
<td>$16,930</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>354</td>
<td>846</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>$24,827</td>
<td>$6,751</td>
<td>$18,076</td>
</tr>
</tbody>
</table>


Monthly Progress Report No. 4
For the Period April 1, 1977 through April 30, 1977

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

by

Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332
I. INTRODUCTION

EES plans to document requirements for the assembly of a low cost Earth Resources Digital Analysis System (ERDAS). The tasks under this project call for:

1) A review of existing software and hardware capabilities in the southeast.
2) A description of various hardware and software configurations
3) An estimate of continuing support requirements for an ERDAS system
4) The assembly of a FORTRAN software library for digital processing of LANDSAT data in an ERDAS system, and
5) The implementation of a limited version of a geographic data base manipulation program (IMGRID).

II. TECHNICAL PROGRESS SUMMARY

During this period tasks have been assigned to Metrics, Inc. for their portion of the project. EES has also spent time on the development of packing and unpacking software which will allow IMGRID to be implemented on a minicomputer. Software documentation also is progressing. Earlier cost estimates as to the operational cost of ERDAS seem to be comparing favorably with current costs. A hardware failure during this period, however, gave us limited time on the ERDAS system.

III. PROBLEMS

Two memory board failures occurred on the two 16K NOVA boards. These were repaired, but down time was approximately one week.

IV. WORK PLANNED FOR THE NEXT PERIOD

EES plans to document more software during the next month and to exercise IMGRID with packing and unpacking options on the CDC in preparation for the transfer to the NOVA. Metrics, Inc. should begin a survey in the southeast for digital processing capability for LANDSAT data.
V. **Budget**

<table>
<thead>
<tr>
<th></th>
<th>Budget</th>
<th>Expended</th>
<th>Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Services</td>
<td>$23,327</td>
<td>$6,483</td>
<td>$16,844</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>422</td>
<td>778</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>416</td>
<td>(116)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>$24,827</td>
<td>$7,321</td>
<td>$17,506</td>
</tr>
</tbody>
</table>
Monthly Progress Report No. 5
For the Period May 1, 1977 through May 31, 1977

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

by

Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332
I. TECHNICAL PROGRESS SUMMARY

During this period EES continued development of a minicomputer version of the IMGRID data base management system. Separate overlays will be necessary to allow the full range of options for the programs. A limited data base will be used for this version of IMGRID because of core constraints. Packing of 2 data words to a 16 bit word will be implemented to double the size of the available data base at the expense of cpu time for packing and unpacking.

Documentation of EES earth resources ERDAS programs is proceeding with internal program comments providing most of the information.

The subcontract to METRICS, INC. is now underway. They are now determining what earth resources digital analysis systems are currently available in the public domain.

II. PROBLEMS

There were no significant problems during this period.

III. WORK PLANNED FOR NEXT PERIOD

EES plans to restructure IMGRID in subroutines to allow overlay capability during June. Packing and unpacking logic for IMGRID will be implemented on the ERDAS system. Further documentation and program development will occur during the next period.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$8,073</td>
<td>$15,254</td>
</tr>
<tr>
<td>M &amp; S</td>
<td>1,200</td>
<td>491</td>
<td>709</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>337</td>
<td>(37)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$8,901</td>
<td>$15,926</td>
</tr>
</tbody>
</table>
DIGITIZING OF LANDSAT DATA

by

N. L. FAUST

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

by

Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia  30332
I. TECHNICAL PROGRESS SUMMARY

During this period work was continued on the documentation of the ERDAS system software. Much of this documentation will be comment statements inside the computer programs themselves. The IMGRID system is almost ready for transfer to the NOVA minicomputer. The original program has been rewritten in a set of subroutines that may be used as overlays in the NOVA system. All of the primary functions of the new version are being verified by comparison to the old version of IMGRID. Metrics, Inc. is set to begin on their part of the project.

II. PROBLEMS

No new problems were encountered this period.

III. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$9,447</td>
<td>$13,880</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>550</td>
<td>650</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>337</td>
<td>(37)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$10,334</td>
<td>$14,493</td>
</tr>
</tbody>
</table>
DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

by

Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332
I. TECHNICAL PROGRESS SUMMARY

During this period operation of the rewritten IMGRID program was verified on the CYBER and was transferred to a NOVA disk for implementation. This version of IMGRID is amenable to overlaying on the NOVA system. Work was also continued on documentation of earth resources software on the NOVA. METRICS, INC. made preparations to initiate their survey of existing hardware and software within the southeastern United States. Several hardware configurations are being considered for prototype low-cost systems. Support requirements for the ERDAS system are being compiled.

II. PROBLEMS

No technical problems were encountered.

III. WORK PLANNED FOR NEXT PERIOD

EES plans to work on implementation of IMGRID on the ERDAS system. Further efforts will be directed at the software documentation. METRICS, INC. will begin their hardware-software survey in earnest.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$12,337</td>
<td>$10,990</td>
</tr>
<tr>
<td>Material &amp; Supplies</td>
<td>1,200</td>
<td>661</td>
<td>539</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>423</td>
<td>(123)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$13,421</td>
<td>$11,406</td>
</tr>
</tbody>
</table>
Monthly Progress Report No. 8
For the Period August 1, 1977 through August 31, 1977

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

by
Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332
I. TECHNICAL PROGRESS SUMMARY

During this period a final estimate for cost of NOVA CPU time was determined, and continuing support requirements for the ERDAS system are being compiled. Flecs indented listings of all earth resources programs on the ERDAS system were produced. Efforts continued on the design of an ERDAS system using more powerful minicomputers. Initial work was performed on implementation of the IMGRID system on the NOVA.

II. PROBLEMS

No new technical problems occurred during this period.

III. WORK PLANNED FOR NEXT PERIOD

Further work will be done on implementation of IMGRID. Software documentation, including flowcharts, will continue. The hardware-software survey should have preliminary results by the end of next period.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$13,852</td>
<td>$9,475</td>
</tr>
<tr>
<td>Material &amp; Supplies</td>
<td>1,200</td>
<td>2,812</td>
<td>(1,612)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$17,091</td>
<td>$7,736</td>
</tr>
</tbody>
</table>

Monthly Progress Report No. 9
For the Period September 1, 1977 through September 30, 1977

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Prepared by
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332
I. TECHNICAL PROGRESS SUMMARY

During this period effort was put into testing the IMGRID algorithm on the NOVA. Sizing of the program will definitely be a problem. The overlaying of the program is not as easy as we anticipated. A sample data set of data was also transferred to the NOVA during this period. Documentation is proceeding at a low rate. Metrics is gathering material for hardware and software components for a low cost system.

II. PROBLEMS

No technical problems were encountered.

III. WORK PLANNED FOR NEXT PERIOD

EES will continue its attempts to overlay IMGRID on the NOVA. Work is continuing both at EES and Metrics on gathering cost and system parameters for candidate systems.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$14,797</td>
<td>$ 8,530</td>
</tr>
<tr>
<td>Material &amp; Supplies</td>
<td>1,200</td>
<td>2,840</td>
<td>(1,640)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$18,064</td>
<td>$ 6,763</td>
</tr>
</tbody>
</table>
DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332
I. TECHNICAL PROGRESS SUMMARY

During this period efforts at overlaying IMGRID on the NOVA were continued, but it is not yet in an operational state. Documentation continues and EES has gathered a substantial amount of information on various computer systems. Metrics is gathering information on peripherals for inclusion into an analysis hardware system.

II. PROBLEMS

We are having trouble with the overlay structure of the EES version and some modifications are being made.

III. WORK PLANNED FOR NEXT PERIOD

EES will continue its overlay attempts on IMGRID. Metrics & EES will try to compile all information on systems and decide what other information is needed.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$14,817</td>
<td>$8,510</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>4,162</td>
<td>(2,962)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$19,406</td>
<td>$5,421</td>
</tr>
</tbody>
</table>
Monthly Progress Report No. 11

For the Period November 1, 1977 through November 30, 1977

DIGITIZING OF Landsat DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332
I. TECHNICAL PROGRESS SUMMARY

During this period a meeting was held with Bill Spann (Metrics) to discuss the material he had gathered on components of a digital analysis system. Also during this period the overlay structure of IMGRID was finalized. Testing of each module of the program will be performed in the coming months.

II. PROBLEMS

The overlay problem has been solved.

III. WORK PLANNED FOR NEXT PERIOD

During the next period the input/output subsystem of IMGRID will be tested and verified. Work will be done on trying to display data base elements in color on the Earth Resources Digital Analysis System (ERDAS).

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$14,817</td>
<td>$ 8,510</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>6,752</td>
<td>(5,552)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$21,996</td>
<td>$ 2,831</td>
</tr>
</tbody>
</table>
Monthly Progress Report No. 12

For the Period December 1, 1977 through December 31, 1977

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Prepared by
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332
I. TECHNICAL PROGRESS SUMMARY

During this period the input/output subsystem of IMGRID was tested successfully. Maps of the test data base are now being put out on the EES printer/plotter. Further work was done on defining concepts for digital analysis systems.

II. PROBLEMS

No technical problems were encountered during this period.

III. WORK PLANNED FOR NEXT PERIOD

Next period EES intends to use the ERDAS color display for display of the data variables in color. The spatial search algorithm will be tested in this period since it is the algorithm which gives IMGRID its flexibility in spatial analysis. Work will continue on development of candidate digital systems and software documentation of EES computer programs.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$14,817</td>
<td>$8,510</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>7,431</td>
<td>(6,231)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$22,675</td>
<td>$2,152</td>
</tr>
</tbody>
</table>
Monthly Progress Report No. 13

For the Period January 1, 1978 through January 31, 1978

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Prepared by
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332

May 1, 1978
I. TECHNICAL PROGRESS

Flow charting of NOVA analysis programs is nearing completion. The final report will include listings, verbal description, and flow charts of major earth resources and data base software. The IMGRID analysis system is now implemented on the NOVA and tests are being run to verify each module. Each program module has been implemented as an overlay on the NOVA to conserve space and allow reasonable array sizes for data elements. Several preliminary system concepts were developed during this period.

II. PROBLEMS

More time is being required for the costing analysis of the systems than expected. It is anticipated that a revised final report date will be requested.

III. WORK PLANNED FOR NEXT PERIOD

Flow charting of programs will be completed during the next period. Work is being started on optimizing and making conversational the IMGRID program. It is envisioned that a totally new program with a different file structure will result from this effort. Costing of the already developed system concepts will begin during the next period.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$14,817</td>
<td>$8,510</td>
</tr>
<tr>
<td>Material &amp; Supplies</td>
<td>1,200</td>
<td>6,423</td>
<td>(5,223)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$21,667</td>
<td>$3,160</td>
</tr>
</tbody>
</table>
Monthly Progress Report No. 14

For the Period February 1, 1978 through February 28, 1978

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332

May 1, 1978
I. TECHNICAL PROGRESS

During this period work was done on the development of new system concepts and the costing of these systems. Flow charting of the major earth resources programs is complete. A new version of IMGRID, NIMGRID is being developed at EES for geographic data base management. This program is a modification of IMGRID to provide for conversational entry, visual display on the ERDAS system, and a raster analysis capability which provides the ability of searching from a given condition in the data base on a line-by-line basis. Data files are stored, line at a time, in a random access mode. Thus, the size of a data base that may be treated is dependent only on disk space, not array size in the program.

II. PROBLEMS

More time is needed for the costing of potential analysis systems and the development of NIMGRID. A no cost extension is being requested until May, 1978.

III. WORK PLANNED FOR NEXT PERIOD

Some program listings will be generated to go into the report during the next period. New system concepts will be devised and evaluated on a cost and flexibility basis. Further work on NIMGRID will proceed next month.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$14,817</td>
<td>$8,510</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>7,197</td>
<td>(5,997)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$22,441</td>
<td>$2,386</td>
</tr>
</tbody>
</table>
Monthly Progress Report No. 15
For the Period March 1, 1978 through March 31, 1978

DIGITIZING OF LANDSAT DATA

by

N. L. Faust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Prepared by
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332

May 1, 1978
I. TECHNICAL PROGRESS

   During this period the Georgia Department of Natural Resources (DNR) was involved in the testing of the minicomputer version of IMGRID and the design of NIMGRID. They are extremely interested in the versatility of both programs and the speed at which they operate on the NOVA. DNR is in the process of developing a hand coded geographic data base for North Fulton County, Georgia, an area just north of Atlanta. They wish to transfer the data to the ERDAS system and use the analysis and color graphics virtues of ERDAS in the furthering of geographic data base efforts in many state agencies. To this end, several presentations were made to state of Georgia groups at EES. Further work was done on NIMGRID in structuring the data file elements and testing of the analysis and file handling subroutines.

II. PROBLEMS

   None

III. WORK PLANNED FOR NEXT PERIOD

   During the next period a meeting with Sandford Downs - NASA/MSFC is planned to go over the structure of the final report. EES will transfer parts of the hand encoded North Fulton data base to the ERDAS system. Work will continue on the development of NIMGRID.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$14,817</td>
<td>$8,510</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>7,882</td>
<td>(6,682)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$23,126</td>
<td>$1,701</td>
</tr>
</tbody>
</table>
Monthly Progress Report No. 16

For the Period April 1, 1978 through April 30, 1978

DIGITIZING OF LANDSAT DATA

by

N. L. Fust

Contract No. NAS8-32397
(A-1938)

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Prepared by
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332

May 22, 1978
I. TECHNICAL PROGRESS

During this period EES interacted daily with Georgia DNR in the transfer of the North Fulton data base to the ERDAS. The file structure of IMGRID would not allow a data base as large as the North Fulton area so a decision had to be made as to whether the data base would be broken up or NIMGRID used as the analysis tool. A joint decision was made to speed up the development of NIMGRID, and to use it for analysis of the North Fulton data base. Work is continuing on the final report, and several new systems were designed and evaluated. Five potential systems from very low to moderate cost ($25,000 - $160,000) are suggested and will be included in the final report.

II. PROBLEMS

None

III. WORK PLANNED FOR NEXT PERIOD

More of the data variables will be transferred from DNR to EES. Color graphic displays of data elements will be generated and NIMGRID analysis capability will be further tested. The final report draft and hopefully final copy will be produced during this period.

IV. BUDGET

<table>
<thead>
<tr>
<th></th>
<th>BUDGET</th>
<th>EXPENDED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Effort</td>
<td>$23,327</td>
<td>$14,817</td>
<td>$8,510</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>1,200</td>
<td>8,632</td>
<td>(7,432)</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>427</td>
<td>(127)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$24,827</td>
<td>$23,876</td>
<td>$ 951</td>
</tr>
</tbody>
</table>
FINAL REPORT

DESIGN OF A LOW COST EARTH RESOURCES SYSTEM

by

N. L. Faust
M. D. Furman
Engineering Experiment Station

and

G. W. Spann
Metrics, Inc.

Contract No. NAS8-32397
GT/Project No. A-1938

Prepared for
National Aeronautics & Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

Prepared by
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332

August 31, 1978
FINAL REPORT

DESIGN OF A LOW COST EARTH RESOURCES SYSTEM

by

N. L. Faust
M. D. Furman
Engineering Experiment Station

and

G. W. Spann
Metrics, Inc.

Contract No. NAS8-32397
GT/Project No. A-1938

Prepared for
National Aeronautics & Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

Prepared by
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332

August 31, 1978
# TABLE OF CONTENTS

I. INTRODUCTION........................................................................................................... 1

II. CAPABILITIES FOR AND USE OF DIGITAL LANDSAT DATA IN THE SOUTHWEST........................................................................................................ 4

   Alabama......................................................................................................................... 4
   Georgia............................................................................................................................ 4
   Kentucky.......................................................................................................................... 9
   Missouri........................................................................................................................... 11
   North Carolina............................................................................................................... 11
   South Carolina............................................................................................................... 12
   Tennessee......................................................................................................................... 12
   Summary.......................................................................................................................... 13

III. DESIGN OF LOW COST EARTH RESOURCES DATA PROCESSING SYSTEM..................................................................................................................... 14

IV. THE IMGRID GEOGRAPHIC ANALYSIS PROGRAM...................................................... 22

V. APPENDIX A
   DESCRIPTION OF MAJOR SOFTWARE RELATED TO IMAGE PROCESSING AND EARTH RESOURCES AT EES............................................................. 30

   RECRAW....................................................................................................................... 31
   RECL................................................................................................................................. 31
   3DPL0T............................................................................................................................ 31
   MAX18............................................................................................................................. 32
   2DFFT.............................................................................................................................. 32
   FILTER............................................................................................................................. 33
   IMGRID........................................................................................................................... 33
   TOPO.............................................................................................................................. 33
   SCORECARD.................................................................................................................... 34
   THERMAL....................................................................................................................... 34
TABLE OF CONTENTS
CONTINUED

CHAN24.......................... 34
M6OCL.......................... 35
ASTEP.......................... 35
SEARCH.......................... 36
COORD.......................... 36
CLUSTER........................ 37
CLUST.......................... 37
SUPERG........................ 38
TRAIN.......................... 38

VI. APPENDIX B
PROGRAM LISTING.................. 39
ADDSIGM........................ 40
ALARM (SUBROUTINE)................. 42
ANGDIS (SUBROUTINE).............. 43
ANGLE (SUBROUTINE).............. 44
AUTO811Z........................ 45
CHAN24.......................... 48
CLASIFY........................ 51
CLEANUP........................ 56
CLUSTA (SUBROUTINE)............. 57
COMBINE........................ 61
COPY5.......................... 63
COUNTY.......................... 64
CRDEM4.......................... 65
CYTAPE.......................... 66
<table>
<thead>
<tr>
<th>Module</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISDEM4</td>
<td>67</td>
</tr>
<tr>
<td>DRAWER</td>
<td>68</td>
</tr>
<tr>
<td>DTAPE</td>
<td>69</td>
</tr>
<tr>
<td>EBCDIC (SUBROUTINE)</td>
<td>70</td>
</tr>
<tr>
<td>EDIST (SUBROUTINE)</td>
<td>71</td>
</tr>
<tr>
<td>EIGEN (SUBROUTINE)</td>
<td>72</td>
</tr>
<tr>
<td>ELIP</td>
<td>74</td>
</tr>
<tr>
<td>FACANL (SUBROUTINE)</td>
<td>75</td>
</tr>
<tr>
<td>FILTER 1</td>
<td>76</td>
</tr>
<tr>
<td>GETPOLY</td>
<td>77</td>
</tr>
<tr>
<td>GSIG (SUBROUTINE)</td>
<td>79</td>
</tr>
<tr>
<td>HISTO (SUBROUTINE)</td>
<td>80</td>
</tr>
<tr>
<td>HISTPR</td>
<td>81</td>
</tr>
<tr>
<td>INITCL</td>
<td>83</td>
</tr>
<tr>
<td>INPUTSIG</td>
<td>84</td>
</tr>
<tr>
<td>LINBOX (SUBROUTINE)</td>
<td>85</td>
</tr>
<tr>
<td>LINCLASS</td>
<td>86</td>
</tr>
<tr>
<td>LINEAR (SUBROUTINE)</td>
<td>88</td>
</tr>
<tr>
<td>LISTALL</td>
<td>89</td>
</tr>
<tr>
<td>LISTSIG</td>
<td>90</td>
</tr>
<tr>
<td>MATIN (SUBROUTINE)</td>
<td>91</td>
</tr>
<tr>
<td>MATPRT</td>
<td>93</td>
</tr>
<tr>
<td>MESS</td>
<td>94</td>
</tr>
<tr>
<td>MODIFY (SUBROUTINE)</td>
<td>95</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------</td>
<td>------</td>
</tr>
<tr>
<td>PACK (SUBROUTINE)</td>
<td>96</td>
</tr>
<tr>
<td>POLY2 (SUBROUTINE)</td>
<td>97</td>
</tr>
<tr>
<td>POLYCL</td>
<td>101</td>
</tr>
<tr>
<td>PUTIN</td>
<td>105</td>
</tr>
<tr>
<td>RAINBOW</td>
<td>106</td>
</tr>
<tr>
<td>RATIO</td>
<td>107</td>
</tr>
<tr>
<td>RDATA</td>
<td>108</td>
</tr>
<tr>
<td>RDIMG (SUBROUTINE)</td>
<td>109</td>
</tr>
<tr>
<td>RETIMG (SUBROUTINE)</td>
<td>110</td>
</tr>
<tr>
<td>SAVIMG (SUBROUTINE)</td>
<td>111</td>
</tr>
<tr>
<td>SCORECARD</td>
<td>112</td>
</tr>
<tr>
<td>SEQCOV (SUBROUTINE)</td>
<td>114</td>
</tr>
<tr>
<td>SEQST (SUBROUTINE)</td>
<td>115</td>
</tr>
<tr>
<td>STRIPLESS</td>
<td>116</td>
</tr>
<tr>
<td>STRIPNF</td>
<td>117</td>
</tr>
<tr>
<td>SUMCAT</td>
<td>121</td>
</tr>
<tr>
<td>SYMINV (SUBROUTINE)</td>
<td>122</td>
</tr>
<tr>
<td>THRDST (SUBROUTINE)</td>
<td>123</td>
</tr>
<tr>
<td>TOPO2</td>
<td>124</td>
</tr>
<tr>
<td>TRAIN3</td>
<td>125</td>
</tr>
<tr>
<td>UNPAC1 (SUBROUTINE)</td>
<td>130</td>
</tr>
<tr>
<td>UNPAC4 (SUBROUTINE)</td>
<td>131</td>
</tr>
<tr>
<td>UPPLT (SUBROUTINE)</td>
<td>132</td>
</tr>
<tr>
<td>ZCOUNTY</td>
<td>133</td>
</tr>
</tbody>
</table>
VII. APPENDIX C
MAIN PROGRAM FLOWCHARTS ............................................. 134

ADDSIGM ................................................................. 135
CLUSTER ................................................................. 136
LINCLASS ................................................................. 138
M60CL ................................................................. 140
STRIPLESS .............................................................. 142
STRIPNF ................................................................. 143
TRAIN3 ................................................................. 145
I. INTRODUCTION

The Georgia Institute of Technology Engineering Experiment Station (EES) has been funded since 1973 by NASA Marshall Space Flight Center (MSFC) to assist the State of Georgia in utilizing Landsat digital analysis. In support of the Georgia state and local agencies and the usefulness of Landsat applications, the purpose of this project was to facilitate a transfer of technology, especially as related to low-cost Landsat data analysis systems. The major tasks accomplished during this project are discussed below.

The initial task involved a comprehensive survey of capabilities for digital processing of Landsat data in the Southern United States. Included was a review of the software and hardware currently being used by the southeastern states. Also, information was obtained on the commercial services used by those desiring to employ a Landsat data processing service.

As a second task, EES developed a set of minimum requirements (hardware and software) for a stand-alone Earth Resources Data Analysis System (ERDAS). The requirements developed in this phase represent a system that can be constructed at a minimum cost to the prospective user and can also satisfy the user's basic needs. The hardware for this system includes a minicomputer, line printer, disk drive, and tape drive while the software includes basic Landsat analysis techniques -- level slicing, maximum likelihood pattern recognition, clustering, and software scaling of the data. The primary products from such a system are computer printouts and statistics.

Since the user groups supporting an ERDAS system may vary greatly in size and available resources, a comprehensive plan was developed for expansion of the basic ERDAS system. This plan details a set of options, including approximate costs, and discusses the advantages of special equipment as well as the way that each equipment type complements the ERDAS concept.

One of the prime concerns in setting up an operational system is the hardware and software support needed to sustain a satisfactory operating schedule. Therefore, as the third task for this project, EES, through experience with its own ERDAS system, developed minimum re-
quirements for continuing support for such a facility (software and hardware). A system such as ERDAS is versatile and is likely to be used for purposes other than earth resources analysis. It therefore will be in use most of the time and will necessarily require the availability of a skilled electrical technician. Software support may be obtained "in-house" or by contract. Some equipment manufacturers have extensive service contracts which prove cost effective.

As the fourth task of the project, EES has assembled a software library for the digital processing of Landsat and other multispectral scanner data using, whenever possible, existing techniques from NASA, the University System, and other public agencies. Software techniques for maximum likelihood classification, linear classification, clustering, level slicing, registration and rectification, and table look-up classification are available for implementation into an ERDAS system. Each module is designed on a user keyword structure so that no detailed knowledge of programming is needed. Documentation is available on programs written or changed exclusively at EES with references provided for other programs. Even though the basic software was developed on a Data General minicomputer at Georgia Tech, the software library was designed entirely in Fortran IV and the routines are transferable to other 16 bit minicomputers.

The final task of the project was the implementation of a limited version of the IMGRID Geographic Analysis program on the ERDAS system. By providing this system to state and local users, the wide range of applications for rectified Landsat data becomes far more evident than if the system were geared to produce only land cover maps. IMGRID produces a dynamic modeling tool for site planning, erosion control, environmental impact, and many other uses.

Report Organization

The remainder of this report discusses each of the five tasks in detail. Section II discusses the results of the comprehensive survey of capabilities for digital processing of Landsat data in the South-eastern United States. Section III presents various options for low
cost earth resources processing systems and the minimum requirements for support of such facilities. Section IV discusses the IMGRID geographic analysis program. Appendix A contains descriptions of some of the major software available at EES, Appendix B contains a listing of the available Fortran software, and Appendix C presents flowcharts of some of the main analysis programs.
II. CAPABILITIES FOR AND USE OF DIGITAL LANDSAT DATA IN THE SOUTHEAST

Numerous individuals and/or organizations who are involved in the use of remote sensing data in the southeastern states were contacted in order to obtain information on their states' capabilities for digital processing of Landsat data. Included is a review of the software and hardware which are currently being used by these states as well as information on the commercial services used by those who wish to employ a Landsat data processing service.

The states involved are: Alabama, Georgia, Kentucky, Missouri, North Carolina, South Carolina and Tennessee. A brief summary of each state's capabilities also includes information on state funded projects utilizing Landsat data digitally processed by companies or organizations outside the state.

Alabama

NASA/Marshall Space Flight Center (MSFC) has the only current capabilities in Alabama for digital processing of Landsat data. Their hardware system consists of an IBM 360/75 with two megabytes of memory and several discs and tapes and a PDP11/45 with two 250-megabyte discs, three tape drives, and various displays and terminals. Approximately 375 computer software routines are available including all algorithms developed by any of the NASA centers. However, very little analysis of Landsat tapes is actually done with this system at present.

Georgia

In order to assist the State of Georgia agencies with their desire to incorporate digital Landsat data into their planning activities on an operational basis, the Georgia Tech Engineering Experiment Station (EES) approved the design and acquisition of the Earth Resources Data
Figure 1. Georgia Tech ERDAS System
Analysis System (ERDAS). EES was responsible for selecting and integrating both the hardware and software components of the system.

**Hardware.** ERDAS was designed and constructed to allow true interactive digital processing of all types of remote sensing data. The system consists of a set of four modules: (1) minicomputer subsystem, (2) input medium, (3) hardcopy output medium, and (4) display subsystem.

The minicomputer subsystem consists of a NOVA-2/10 minicomputer with 64K bytes of core memory and a dual Diablo disk system with 5.0 megabytes of storage for programs or data.

The input medium for the ERDAS system is a set of two nine track dual density (phase encoded/NRZI selectable) magnetic tape drives and controller — both drives with a capacity for 10-1/2 inch reel tape.

One hardcopy output device is a twenty inch electrostatic dot matrix printer/plotter. Scaled maps of Earth Resources data can be made using this device. A CROMALIN(R) photographic process may then be used to generate a color coded output hardcopy product. Another output method currently in use consists of storing images on a magnetic tape and sending these tapes to be made into images by the use of a digital film writer. This method is currently very inexpensive.

The display subsystem consists of a high quality color video image analysis system that is interfaced to the minicomputer for complete user interaction in the choice of training samples for earth resources classification.

**Software.** Initially, EES implemented a basic Landsat digital analysis program called ASTEP (Algorithm Simulation Test and Evaluation Program) which was obtained from NASA/JSC and has been extensively modified by EES personnel. The ASTEP system was designed as a modular program whereby various classification algorithms may be tested against one another using a standard input/output system.
Software for the rectification of Landsat data to map coordinates using a least squares fit of Landsat data to control points was obtained from NASA/MSFC and transferred to the Georgia Tech computer. Other software, such as various spatial clustering algorithms, was studied but has not yet been transferred. Software for a table lookup formulation for Landsat classification (ELLTAB) was obtained from NASA/ERL along with software for rectification, destriping, and polygon location of Landsat data. A fast combination table lookup and maximum likelihood classifier from NASA/ERL has been implemented which significantly decreases the length of time needed for a scene classification. A fast clustering algorithm from NASA/ERL has also been implemented. In addition, many software algorithms for image manipulation, spatial filtering, rectification, training field selection, and high speed classification have been developed at EES. EES software exists for image analysis on UNIVAC and CDC large computers and Data General minicomputers.

Projects. EES has been deeply involved in the formulation, planning, and implementation of a Georgia Natural Resources Inventory since its conception in 1972. During 1972 and 1973, EES presented various State of Georgia agencies with the background information needed to make an initial assessment of the usefulness of digital Landsat information. A trial project was initiated between the Georgia Department of Natural Resources (DNR) and EES in 1973 to test the capability for using digitally processed Landsat data to determine land use in the Atlanta area.

EES has been funded since 1973 by NASA/MSFC to assist the State of Georgia in utilizing Landsat digital analysis for various resource problems within the state. In this multi-year effort, several related tasks have been performed in conjunction with numerous local and state agencies within Georgia, including the Department of Natural Resources, the Department of Transportation (DOT), and the Office of Planning and Budget (OPB).
In 1975 Georgia Tech EES was designated as the technical interface with NASA/ERL for the transfer of NASA software to the State of Georgia. Under this technology transfer project EES purchased the necessary digital processing equipment required for operational processing of Landsat data. EES then coordinated a project with the Georgia DNR office of Planning and Research for classifying and mapping land cover for the State of Georgia. Products of the effort assured further acceptance of digital processing of Landsat data as an operational tool for environmental analysis.

Probably the best indicator of the success of the technology transfer efforts of this project is the number of agencies which have committed funds and/or personnel time to a project to map the entire State of Georgia using Landsat data. This project is concerned with mapping land cover using Landsat data processed with ERDAS and, where appropriate, inferring land use.

The agencies which have committed funds to the mapping project include:

- Georgia Department of Natural Resources
- Environmental Protection Division
- Game and Fish Division
- Office of Planning and Research
- Georgia Forestry Commission
- Georgia Office of Planning and Budget
- Bureau of Community Affairs
- U.S. Department of Agriculture
- Soil Conservation Service
- Forest Service
- United States Army Corps of Engineers
- Fort Benning
- Savannah Engineer District
- North Georgia Area Planning and Development Commission
- Coosa Valley Area Planning and Development Commission

Other organizations which are interested but as yet have supplied no funds include:

- Georgia Department of Natural Resources
- Earth and Water Division
- Georgia Department of Transportation
- Five other area planning and development commissions.
EES is currently assisting the Georgia DNR in a geographic database demonstration project. DNR manually obtained 30 different data variables on a 10 acre cell basis for North Fulton County, Georgia. EES is assisting with transfer of the data to the ERDAS system and analysis of the data using IMGRID and NIMGRID.

In addition to the activities at Georgia Tech EES, the Atlanta Regional Commission (ARC) was involved in a project to test the feasibility of using automatic processing of Landsat data to detect land use changes in the seven county planning area of the ARC, and thereby update the USGS/LUDA land use map of the Atlanta Region.

The technique chosen to accomplish the land use change detection was the ratioing of two different dates of Landsat data. This was accomplished at the EROS Data Center utilizing the Image 100 System. The results of the project indicated that ratioing Landsat data was a feasible technique for ARC to use in updating the USGS land use maps of the area. Accuracy evaluations showed that 91% of the change theme was accurate to within about 3 pixels (accuracy sufficient for ARC's purposes). Additional manual analysis was required to identify accurately the types of land use changes.

Kentucky

Kentucky is currently developing the capability for in-house processing of Landsat digital tapes. A few state agencies and universities have small interactive graphics systems which are capable of digital processing, but these are being discarded in favor of a central state computer system. The central system includes an IBM 370/168 MP with 9 megabytes of main storage and various discs. The Bureau of Computer Services has been established as the central hardware agency. Their personnel have recently visited NASA/ERL to obtain compatible software for the system.
A representative from Eastern Kentucky University recently attended a workshop at GSFC on a joint Appalachian Regional Commission–NASA project to apply Landsat to the study of geological lineaments using GE's Image 100 System. An ORSA package was ordered from GSFC which provides an offline printout of similar pixels vs. significant changes, but this is not adequate for final classifications.

Five professors from Murray State University attended a short course at ERL and had plans to return the end of October for hands-on experience with digital tapes for strip mining studies in Western Kentucky.

Previous projects in Kentucky include a survey in 1973 by Earthsat of water impoundments larger than two acres. The results were not beneficial because the imagery was taken after a flood and strip mines appeared as impoundments. However, this led to a Landsat-2 follow-on study in 1976 for which ERIM provided the digital processing. The objective of the project was to look at the operations of inspection and enforcement to detect significant violations of surface mining laws.

Several projects have been contracted with Bendix Corporation for Area Development (AD) Districts in conjunction with EPA 208 land use planning programs. Color-coded maps and overlays were provided in 1976 for the Kentuckiana AD (seven Kentucky counties, two Indiana counties) at 1:48,000 scale. Area tabulations were also provided for each county. Processing for the Big Sandy AD (Prestonsburg, Kentucky) is in progress and maps are scheduled for delivery this fall. A contract has also been signed for the Green River AD (Owensboro, Kentucky) to be delivered in December, 1977. This project will provide information to meet HUD 701 requirements for land use planning.

Two counties of Kentucky were included in a digital processing project by Bendix for the Ohio-Kentucky-Indiana Council of Government (OKI COG). Color-coded maps at 1:62,500 scale and computer tabulations for each of 229 drainage basins were provided. Also, five counties centered around Fayette County (Lexington, Kentucky) known as the
Central Blue Grass Region (not an AD district) were mapped for the Army Corps of Engineers.

The LARS program at Purdue was used to produce a land cover map of Henderson County in Western Kentucky (part of Southeastern Indiana COG).

**Missouri**

According to the final report of a project on Earth Observation Data Management Systems in December, 1976, "few agencies (in the five state Midwestern region of Illinois, Iowa, Minnesota, Missouri, and Wisconsin) now have the staff or computer capabilities to handle digital satellite data."¹ At the present time, Missouri still has no capabilities for digital processing of Landsat data. However, the University of Missouri at Rolla recently received a grant from the National Science Foundation and has ordered a Comtal Interactive Image Analysis System which will be used for such processing. Delivery occurred in early 1978. Software will be obtained from NASA/ERL.

Several Missouri agencies have funded projects for digital processing in the past. A demonstration study of land use in the Ozarks Planning Region of Southern Missouri was conducted by NASA/ERL in 1975.² The Soil Conservation Service in Missouri has digital tapes processed by LARS/Purdue for water analysis and study of soil patterns. Also, the Missouri Geological Survey had the University of Kansas in Lawrence process some digital tapes for an area around Kansas City, Missouri.

**North Carolina**

Currently North Carolina has no facilities for processing digital Landsat data although considerable interest exists for establishing a centralized state system. Several projects have been completed by Bendix Corporation for EPA 208 planning regions in North Carolina. Color-coded land use overlays at 1:96,000 scale were produced using the Multispectral Data Analysis System (M-DAS) at Bendix in 1975 for


Planning Region J (also known as Triangle J -- five counties in central North Carolina -- Raleigh, Durham, Chapel Hill areas). An additional product of the analysis process was the generation of statistical data by 50x50 meter grid cells in data sets corresponding to 54 7-1/2 minute USGS quadrangle maps. Overall classification accuracy of the land use categories was judged to be around 90% and the cost of processing the Landsat scene and generating the products was approximately $4.00 per square mile.

A similar analysis was done in 1976 by Bendix for Planning Region D in the Northwestern part of the state. Also, analysis of the Dan River Sub-Basin (Roanoke River Basin) was completed this year for the Corps of Engineers. NASA had some involvement in the most recent project.

South Carolina

South Carolina has no capabilities at present for digital processing of Landsat data. A proposal has been submitted to NASA/ERL for projects which may involve digital processing. Previous activities include a contract with General Electric Company for Image 100 processing of Landsat data to produce color-coded land use maps and area calculations for three Council of Government (COG) regions comprising approximately 25% of the state. Landsat tapes were processed by NASA to produce a map of the Congaree Swamp area for the Wildlife and Marine Resources Department. The Land Reserve Conservation Commission and the Bureau of Mines visited the EROS Data Center for processing of digital tapes for a study of mining areas in South Carolina.

Tennessee

The only current capabilities in Tennessee for digital processing of Landsat tapes are at the University of Tennessee, Knoxville. The facilities are presently being used for image processing projects other than earth resources, although previously they have been used for Landsat analysis. Oak Ridge National Laboratory has accomplished strip mining surveys using some Landsat data.
Bendix Corporation completed a digital processing project on water resources and strip mining in the New River Drainage Basin in North-central Tennessee for the Soil Conservation Service in early 1976. They provided a color-coded map of the Basin at 1:62,500 scale, fifteen color-coded 7-1/2 minute quads, computer tabulations, and rescanned-resampled tapes for the area.

Summary

Of all the Southeastern states, Georgia is the most advanced in the use of digital Landsat data. The University of Tennessee at Knoxville has the capability for digital processing of Landsat data but no such projects are currently underway. All other states have relied on NASA or commercial facilities. The most extensive use of digital Landsat data among these states has been in fulfillment of EPA 208 and HUD 701 planning requirements.

The results of this survey indicate that users of remote sensing data in the Southeastern U.S. are increasingly turning to digital processing techniques. All the states surveyed have had some involvement in projects using digitally processed data. Even those states which do not yet have in-house capabilities for digital processing are extremely interested in and are planning to develop such capabilities.
III. DESIGN OF LOW COST EARTH RESOURCES DATA PROCESSING SYSTEMS

In the design of an earth resources data processing system there are many factors to be considered. In some cases potential users should buy turn-key systems that are currently on the market; in other cases, users might consider the design and implementation of their own systems by buying components and assembling the systems using their own technical expertise. This section deals with alternative systems that might be considered by users having the technical expertise for assembly of such a system within their agencies or support groups.

With the appropriate technical personnel available, a significant cost savings often may be realized by user design and implementation of systems. At least one part- or full-time computer hardware technician, one applications software analyst, and several applications programmers are desirable for all phases of system design. These requirements may be lessened, however, if sufficient support is available from the various equipment manufacturers.

Computer software for digital processing of earth resources and other geographically based data is currently becoming available for minicomputers at minimal or no cost. Since a significant amount of this computer software has been developed under government contracts, it is in the public domain and readily available. Thus the costs involved in the acquisition of the systems listed here are primarily the actual hardware costs. If they are needed, personnel training costs and systems interface costs are extra.

Figure 2 indicates five alternative configurations for low cost earth resources data processing systems. At the low end is a nominal system consisting of a minicomputer, floppy disk, magnetic tape unit, color terminal, and line printer. The estimated cost range for this system (depending on the exact components selected) is $22,000 to $45,000.
At the upper end of the range (still, however, at a price significantly less than many systems on the market) are systems costing an estimated $165,000 to $220,000. This configuration, with substantially increased capability over the lower cost system, consists of a minicomputer with an array processor, a 96 megabyte disk, dual magnetic tape drives, a digitizer, a color display, a line printer, and a film recorder.

In general, as the systems increase in cost, the processing sophistication is improved and the speed with which a data set can be analyzed increases rapidly. Thus, for users requiring only a low volume of processed data, a system in the low or middle cost range might be suitable. For users desiring a faster processing speed and an increased throughput, a more expensive system might be in order.

The breakdown of the total costs for each system is shown in Table I. For each system, the high and the low cost estimate for each component is given. Thus, by selecting particular components with a greater or lesser capability, systems could be configured that cost anywhere within the range of the cost extremes given.

To facilitate the estimation of costs for system configurations other than those listed in Figure 2, the costs of the individual components are given in Table II. Using these data, the approximate costs of many more low cost system configurations could be derived. Typical vendors of such components are given in Table III.

**Estimated Cost for ERDAS Support**

A minimum cost estimate for the support of the Georgia Tech ERDAS system over a one year period totaled $2,500. This estimate includes approximately 400 hours of a resident part-time technician along with all electrostatic line printer supplies and any additional maintenance charges incurred when repairs exceeded in-house capabilities.

This estimate is approximately 3.3% of the total cost of the ERDAS system. As maintenance contracts usually run about 10% of system costs per year, a significant savings was realized using in-house methods of maintaining the equipment.
16 BIT CPU

COLOR TERMINAL

TERMINAL

MAG TAPE UNIT

9 TRACK

FLOPPY DISK

16 BIT CPU

LINE PRINTER

SYSTEM A: $22 - 45 K

SYSTEM B: $34 - 54 K

DUAL MAG TAPE UNITS

9 TRACK

CARTRIDGE DISK

2.5 MEGA BYTES

COLOR TERMINAL

LINE PRINTER

Figure 2
Figure 2 (Continued)

SYSTEM C: $ 53 - 74 K

SYSTEM D: $ 69 - 98 K
16 BIT CPU
ARRAY PROCESSOR
DIGITIZER
COLOR DISPLAY
3 IMAGES
256 X 256 X 8 BITS
DUAL MAG TAPE UNITS
9 TRACK
96 MEGABYTE DISK
FILM RECORDER
LINE PRINTER

SYSTEM E: $165 - 220 K

Figure 2 (Continued)
<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>COMPONENT</th>
<th>LOW - HIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TAPE DRIVE</td>
<td>$5 - 10 K</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>$8 - 18 K</td>
</tr>
<tr>
<td></td>
<td>DISK</td>
<td>$2 - 4 K</td>
</tr>
<tr>
<td></td>
<td>LINE PRINTER</td>
<td>$5 - 8 K</td>
</tr>
<tr>
<td></td>
<td>COLOR TERMINAL</td>
<td>$2 - 5 K</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>$22 - 45 K</td>
</tr>
<tr>
<td>B</td>
<td>TAPE DRIVE</td>
<td>$12 - 14 K</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>$8 - 18 K</td>
</tr>
<tr>
<td></td>
<td>DISK</td>
<td>$7 - 9 K</td>
</tr>
<tr>
<td></td>
<td>LINE PRINTER</td>
<td>$5 - 8 K</td>
</tr>
<tr>
<td></td>
<td>COLOR TERMINAL</td>
<td>$2 - 5 K</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>$34 - 54 K</td>
</tr>
<tr>
<td>C</td>
<td>TAPE DRIVE</td>
<td>$12 - 14 K</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>$8 - 18 K</td>
</tr>
<tr>
<td></td>
<td>DISK</td>
<td>$9 - 11 K</td>
</tr>
<tr>
<td></td>
<td>ELECTROSTATIC PRINTER/ PLOTTER</td>
<td>$10 - 20 K</td>
</tr>
<tr>
<td></td>
<td>COLOR DISPLAY</td>
<td>$19 - 23 K</td>
</tr>
<tr>
<td></td>
<td>(256 X 256)</td>
<td>$53 - 74 K</td>
</tr>
<tr>
<td>D</td>
<td>TAPE DRIVE</td>
<td>$12 - 14 K</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>$8 - 18 K</td>
</tr>
<tr>
<td></td>
<td>DISK</td>
<td>$9 - 11 K</td>
</tr>
<tr>
<td></td>
<td>ELECTROSTATIC PRINTER/ PLOTTER</td>
<td>$10 - 20 K</td>
</tr>
<tr>
<td></td>
<td>COLOR DISPLAY</td>
<td>$30 - 35 K</td>
</tr>
<tr>
<td></td>
<td>(3 images 256 X 256)</td>
<td>$69 - 98 K</td>
</tr>
<tr>
<td>E</td>
<td>TAPE DRIVE</td>
<td>$12 - 14 K</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>$8 - 18 K</td>
</tr>
<tr>
<td></td>
<td>DISK</td>
<td>$20 - 35 K</td>
</tr>
<tr>
<td></td>
<td>FILM WRITER</td>
<td>$40 - 50 K</td>
</tr>
<tr>
<td></td>
<td>ARRAY PROCESSOR</td>
<td>$40 - 45 K</td>
</tr>
<tr>
<td></td>
<td>LINE PRINTER</td>
<td>$5 - 8 K</td>
</tr>
<tr>
<td></td>
<td>DIGITIZER</td>
<td>$5 - 15 K</td>
</tr>
<tr>
<td></td>
<td>COLOR DISPLAY</td>
<td>$30 - 35 K</td>
</tr>
<tr>
<td></td>
<td>(3 images 256 X 256)</td>
<td>$165 - 220 K</td>
</tr>
</tbody>
</table>
TABLE II. INDIVIDUAL COMPONENT COST ESTIMATE

<table>
<thead>
<tr>
<th></th>
<th>Component Description</th>
<th>Cost Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 Mag Tape Drive &amp; Controller</td>
<td>$5 - 10 K</td>
</tr>
<tr>
<td>2</td>
<td>2 Mag Tape Drives &amp; Controller</td>
<td>12 - 14 K</td>
</tr>
<tr>
<td>3</td>
<td>16 Bit CPU with 32 K Memory</td>
<td>8 - 18 K</td>
</tr>
<tr>
<td>4</td>
<td>Line Printer</td>
<td>5 - 8 K</td>
</tr>
<tr>
<td>5</td>
<td>Floppy Disk</td>
<td>2 - 4 K</td>
</tr>
<tr>
<td>6</td>
<td>Dual 2.5 Megabyte Drive &amp; Controller</td>
<td>9 - 11 K</td>
</tr>
<tr>
<td>7</td>
<td>10.0 Megabyte Cartridge Disk</td>
<td>9 - 11 K</td>
</tr>
<tr>
<td>8</td>
<td>Electrostatic Printer/Plotter</td>
<td>10 - 20 K</td>
</tr>
<tr>
<td>9</td>
<td>Array Processor</td>
<td>40 - 45 K</td>
</tr>
<tr>
<td>10</td>
<td>96 Megabyte Disk</td>
<td>25 - 35 K</td>
</tr>
<tr>
<td>11</td>
<td>Color Terminal</td>
<td>2 - 5 K</td>
</tr>
<tr>
<td>12</td>
<td>Color Display Image 256 x 256</td>
<td>19 - 23 K</td>
</tr>
<tr>
<td>13</td>
<td>Color Display 3 Image 256 x 256</td>
<td>28 - 35 K</td>
</tr>
<tr>
<td>14</td>
<td>Film Writer</td>
<td>40 - 50 K</td>
</tr>
</tbody>
</table>
TABLE III. TYPICAL VENDORS

**COMPUTERS**

- Data General Corporation
- Digital Equipment Corporation
- Hewlett Packard Corporation

**MAGNETIC TAPE DRIVES**

- Digi-Data Corporation
- Pertec Computer Corporation
- Kennedy, C. J. Company

**DISKS**

- Control Data Corporation
- Data General Corporation
- Digital Equipment Corporation

**COLOR DISPLAY SYSTEMS**

- Aydin Corporation
- Comtal Systems Corporation
- ITT Grinnell Corporation
- Ramtek

**ELECTROSTATIC PRINTERS/PLOTTERS**

- Versatec Incorporated
- Varian Data Machines
- Gould Incorporated

**DIGITAL FILM RECORDERS**

- Dicomed Corporation
- Optronics International Incorporated

**DIGITIZERS**

- Summagraphics
- Bendix Corporation
- Talos System Incorporated
- Aristo Graphics Corporation

**LINE PRINTERS**

- General Electric Company
- Okidata Corporation
- IBM
- Varian Data Machines
IV. THE IMGRID GEOGRAPHIC ANALYSIS PROGRAM

IMGRID, an information manipulation system for grid cell data structures, is a package of computer programs designed for the analysis of natural resource and land planning data which is qualitative in its thematic content and varies over geographic space. Originally developed by David Sinton of the Harvard University Department of Landscape Architecture, the IMGRID system has been adapted for use on mini-computers at Georgia Tech.

The IMGRID system has been designed for people having no previous experience with computers. The basic operations are controlled with simple keyword commands which may be used with a basic knowledge of planning principles but without any knowledge of programming. Thus, a link is provided for easy access and manipulation of digital data bases. Some typical project applications of the IMGRID system are:

- River basin planning
- Siting of facilities such as airports or sanitary landfills
- Environmental impact statement review
- Visual Analysis
- Project review by regional planning agencies

KEYWORD STRUCTURE

The basic structure for an IMGRID keyword command involves three processes:

1. Retrieve one or more data elements from the data file.
2. Transform or manipulate the values for each grid cell in the data elements retrieved.
3. Store the new data element created in the data file.

On the Georgia Tech ERDAS, keyword commands are entered on a CRT terminal, executed by the NOVA II mini-computer, and output either in color on the Comtal video display, or in black & white on a dot matrix printer. The data files are usually stored on tape then read into a disk file. The IMGRID program itself also resides on a disk.
The keyword commands fall basically into 9 groups:

1.) **Data Entry & Management**

   Two keywords can be used to enter new information into the data base:
   - **STORE** - operates on an element (data variable) by basis. When a new variable, such as a soil type or slope category is added to the data base for all cells, STORE can be used.
   - **UPDATE** - operates on a cell by cell basis. If a single cell changes characteristics for a particular variable, such as a land use change from agricultural to residential, UPDATE is used along with the row and column location, plus the new value of the cell.

   Three keywords which are used for data management are:
   - **RELOC** - permits a data element to be moved to a new location within the data file, such as relocating the results of an analysis as a new data element in another location.
   - **RENAME** - allows the name of a data element to be changed without affecting the contents of the data.
   - **LIST** - allows a user to list the names associated with the contents of part or all of the data base.

2.) **Delimiter Keywords**

   - **MODEL** - used as a first keyword in a sequence of keywords defining an analysis. The primary function of MODEL is to assign a title to subsequent analyses. It also includes the function of the CLEAR keyword.
   - **CLEAR** - clears the results of operations performed by previous keywords.
   - **END** - identifies the completion of IMGRID input.
3.) **Display Keywords**

- **SYMBOL** - allows a user to enter a specific set of character symbols to be used when making a map display.
- **MAP** - makes a graphic display of the contents of a data element in the file. This can be a data variable such as a MAP of a slope, or of the results of an analysis such as vulnerability to soil erosion.
- **TEXT** - permits user to insert textural descriptions of the procedures being undertaken.

4.) **Spatial Analyses**

- **SEARCH** - generates a set of values which identify the proximity of each cell in the study area to a specified condition, such as roads, rivers, or airports. The analysis determines how far every other location is from the preselected data items.
- **ASEARCH** - operates on a cell by cell basis examining for a prespecified radius, the conditions around every cell, or a defined subset of cells in the study area, such as how many cells of wetlands are there within a five cell radius of a landfill.

5.) **Rescaling or Restructuring of Data Values**

- **RECODE** - assigns new values to an old set of values for a data element, such as recoding an old set of values for element "land market value" to reflect land use or tax policy changes. Recode assigns values in the range of 0-9.
- **XRECODE** - an extended recode which assigns values in the range of 0-19. The RECODE keywords assign the last value specified in the case where multiple elements are being combined with different value scales.
OVERLAY - assigns the highest value, rather than the last value, when multiple elements are being combined. The system will over-ride the values assigned on the first element if the new values generated by the rescaling are greater than the previously existing values.

6.) Logical Combinations of Elements

- MATRIX - results in a series of values which identify each of the possible combinations of features, such as the highly erodible soils on steep slopes.

7.) Reject Conditions

- REJECT - permits the user to identify a group of cells which must be ignored ("dropped out") in all analysis and display of the data, such as rejecting unstable soils in an analysis for siting a large industrial complex.

8.) Mathematical Manipulation of the Data

- MULTIPLY - allows user to multiply the values in one data element by the values in a second data element.
- INDEX - generates a weighted index of several data elements and also provides for addition and subtraction.
- NORMAL - takes data values over large ranges and normalizes them to the range of 0-99.
- REDUCE - generates values in the range of 0-19.

9.) User Written Fortran Subroutines

- USERSUB - allows user to write a Fortran subroutine to perform any set of computations that is desired and call that subroutine through the USERSUB keyword.

Figure 3 shows an example of IMGRID output where elements of slope, depth to water table, soils, vegetation, proximity to roads, and travel time were weighted, overlaid, recoded and mapped to produce an attractiveness model for industrial parks. The darkest cells are best for industrial sites using the criteria specified.
Figure 3
Figure 3 (Continued)
Because of the large size of IMGRID and the limited address space of a 16 bit computer, a method was devised to partition the IMGRID program into independent subroutines. Each subroutine corresponds to a keyword in the above discussion. By making each major subroutine an overlay, IMGRID was implemented on the NOVA 2 minicomputer. A core part of the IMGRID program resides in the computer at all times, and when a particular keyword is selected, only programs that provide that function are loaded into the main memory of the minicomputer from disk storage. While some delay occurs in the program execution due to the overlay procedure, the time required is negligible compared to the execution time of the analysis modules.

The structure of the minicomputer IMGRID program allows two data variables in a format of up to sixty elements by sixty elements to be analyzed at one time. In the Atlanta, Georgia area, this array size is sufficient to represent data for a USGS seven and one half minute quadrangle in approximately ten acre elements. For a large area, therefore, data for a data set of the whole area would have to be partitioned into segments dependent on the selected cell size.

All keyword functions with the exception of the search algorithms could easily be performed on such a segmented data set. Any application for which a search from some criterium is specified, however, would encounter problems when needing to search beyond the boundaries of one data segment. In addition, the normal mode of operation for IMGRID is execution in a batch runstream in which a set of input data is required in special formats. While this method is sometimes desirable when using cards as input to a large computer, an alternate interaction method should be considered when dealing with minicomputers where the user has direct contact with the computer itself and he may often be the only user using the system at one time.

In an effort to alleviate some of the difficulties given above a new minicomputer version of IMGRID, called NIMGRID, has been developed at EES. NIMGRID reflects to a large extent the program philosophy of the original IMGRID program regarding types of functions performed on
the data base, but a new philosophy of data storage and retrieval and program interaction was implemented in early 1978. The data storage philosophy envisions each data variable as being represented by a raster data set which is only constrained by the disk space available to the user of a particular system. The data are processed on a line by line basis, and even for the search algorithm, only two lines of the data set are needed in the computer at one time. While this method involves many more input/output operations than the IMGRID method, very little degradation in performance has been noted in comparing NIMGRID and IMGRID.

The prime advantage of this storage and retrieval philosophy is that the size of the data set that may be processed by a user is virtually unlimited. Also, searches may be made over large areas without problems associated with crossing data segment boundaries. Another virtue of the NIMGRID system is that the program has been made interactive. Prompting of the user occurs, giving him all possible choices and asking him to select the desired command. No knowledge of FORTRAN and very little knowledge about the particular minicomputer is required of the user. Although NIMGRID is still under development, applications personnel from the Georgia Department of Natural Resources are currently using the program at EES with ease.
APPENDIX A

DESCRIPTION OF MAJOR SOFTWARE
RELATED TO IMAGE PROCESSING AND
EARTH RESOURCES AT EES
PROGRAM NAME: RECRAW

LANGUAGE: FORTRAN IV

COMPUTER: SEL 32/55, CDC Cyber 74

SPECIAL PERIPHERALS: 2 Tape Drives
Disk Files

PURPOSE: RECRAW uses a first order transformation matrix from COORD to resample raw Landsat data by a selectable nearest neighbor or bilinear interpolation to format the data into a standard coordinate system.

PROGRAM NAME: RECL

LANGUAGE: FORTRAN IV

COMPUTER: Data General NOVA 2, CDC Cyber 74, SEL 32/55

SPECIAL PERIPHERALS: 2 Tape Drives
Disk Files

PURPOSE: RECL uses a first order transformation matrix computed by COORD to resample Landsat classified data using nearest neighbor and format the data into a Latitude-Longitude or UTM coordinate system.

PROGRAM NAME: 3DLPLOT

LANGUAGE: FORTRAN IV

COMPUTER: Data General NOVA 2, CDC Cyber 74

SPECIAL PERIPHERALS: Calcomp plotter or printer/plotter
Disk Files

PURPOSE: 3DLPLOT provides a perspective view of three dimensional data ($z = f(x,y)$). Viewing angle and scaling may be specified by the user. This technique is especially useful in analysis of topographic related data.
PROGRAM NAME: MAX18 (NASA/ERL)

LANGUAGE: FORTRAN IV

COMPUTER: SEL 32/55, CDC Cyber 74

SPECIAL PERIPHERALS: Two Tape Drives
Disk Files

PURPOSE: MAX18 was developed by Ronnie Pearson of NASA/ERL (Slidell, Louisiana) as a fast classifier of Landsat MSS data. The program is a combined table look-up-maximum likelihood type of classifier which uses the best points of each technique. Instead of creating a look-up table defining the boundaries of statistical distribution of signatures, this technique iterates quickly through the data building a table of where in channel space the majority of the data in a Landsat scene lie. Then, these vectors are classified using a maximum likelihood scheme. A second iteration is made through the data for classification. Each pixel of MSS data is checked to see if the data vector associated with that pixel is in the already classified data table. If so, the classification is derived by simply indexing into the classified table. If the data vector has not already been classified a maximum likelihood decision rule is used. There is a distinct trade off between amount of storage and speed of classification. On the SEL machine a Landsat scene may be classified into 60 classes in 1.5 - 2 hours.

PROGRAM NAME: 2DFFT

LANGUAGE: FORTRAN IV

COMPUTER: Data General NOVA 2, CDC Cyber 74

SPECIAL PERIPHERALS: Magnetic Tape Drives
Disk Files

PURPOSE: 2DFFT performs a two dimensional Fast Fourier Transform on image data. Options in the program include:

a) Forward transform
b) Inverse transform
c) Image filtering by specification of one of many optional filters.
PROGRAM NAME: FILTER
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Image display system and 1 tape drive or 2 magnetic tape drives

PURPOSE: FILTER is a program whose concept was taken from a technique used at USGS, Flagstaff, Az. for high frequency enhancement of image data. The basic method involves taking a windowing approach to create a low pass filtered image and subtracting that image from the original image. The resultant is essentially a high pass filtered image. By adding the high pass filtered image to the original image the high frequency enhancement is achieved.

PROGRAM NAME: IMGRID (Harvard U.)
LANGUAGE: FORTRAN IV
COMPUTER: CDC Cyber 74, NOVA 2 (Data General)
SPECIAL PERIPHERALS: Disk Files

PURPOSE: IMGRID is a general purpose geographic data base manipulation program developed by the Harvard University Graduate School of Landscape Design. This program provides for manipulation of multiple data variables related to the same geographic area and in a gridded format. Spatial searching, statistics generation, and modeling via multivariable weighting parameters are key to its analysis capability. Visual and environmental impact analysis are two uses of such a system.

PROGRAM NAME: TOPO
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Tape Drive
Image Display

PURPOSE: After breaking down the NCIC topographic tapes into NOVA 4096 record blocks with the CDC Cyber 74, TOPO unpacks and displays the data on the imaging system. Various scales may be represented by selection of parameters in the program.
PROGRAM NAME: SCORECARD
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2

SPECIAL PERIPHERALS: Tape Drive

PURPOSE: SCORECARD performs a maximum likelihood classification on specific polygons within a data set. This is used to evaluate accuracy of classification by comparing the classified data for test fields to known ground truth.

PROGRAM NAME: THERMAL
LANGUAGE: FORTRAN
COMPUTER: Data General NOVA 2

SPECIAL PERIPHERALS: Tape Drive
Image Display

PURPOSE: THERMAL is a program designed to unpack and display digital thermal data from a NASA owned thermal scanner – RS18.

PROGRAM NAME: CHAN24
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2

SPECIAL PERIPHERALS: Image Display
Tape Drives

PURPOSE: CHAN24 unpacks data from the Bendix 24 channel aircraft scanner and reformats the data such that single channels may be accessed and written out to either tape or an image display.
PROGRAM NAME: M6OCL
LANGUAGE: FORTRAN IV
COMPUTER(S): Data General NOVA 2
SPECIAL PERIPHERALS: 2 Tape Drives
Disk

PURPOSE: M6OCL uses a maximum likelihood decision rule to classify Landsat data into one of up to sixty classes for which means and covariances are available on a disk file. Thresholds (probability of correct classification) are output for each record of data. This program is a record by record classifier. One record is read from tape, classified, and then written to an output tape before the next record is processed.

PROGRAM NAME: ASTEP
LANGUAGE: FORTRAN IV
COMPUTER: CDC Cyber 74, UNIVAC 1108
SPECIAL PERIPHERALS: Tape Drives
Disk Files

PURPOSE: ASTEP is a general purpose earth resources analysis program developed by TRW Systems for NASA Johnson Space Center. The acronym ASTEP stands for Algorithm Simulation Test and Evaluation Program. The program is of modular construction with standardized input-output such that they are essentially transparent to the user. Different classification, clustering, statistics generating, or feature selection algorithms may be tested against one another with a minimum of programming change to the whole system. In addition to its usefulness as an algorithm test bed, ASTEP has been used effectively as an operational, interactive classification system for Landsat data.
PROGRAM NAME: SEARCH (NASA/ERL - Ronnie Pearson)

LANGUAGE: FORTRAN IV

COMPUTER: SEL 32/55

SPECICAL PERIPHERALS: 1 Tape Drive
 Disk Files

PURPOSE/DESCRIPTION: SEARCH is a program developed by Ronnie Pearson of NASA/ERL for unsupervised development of signatures for use in a maximum likelihood classification scheme. A 3 x 3 or 6 x 6 pixel moving window is used in a single iteration through the raw Landsat data to form candidate signatures. A maximum number of acceptable signatures is specified and a divergence criterion is used for merging, splitting, and selection of signatures. This program normally takes approximately one hour for development of signatures for one Landsat scene. Auxiliary programs are available for intuitively assigning color values for each class for use on a color display based on a two dimensional plot of the signature means for channels 2 and 4 of Landsat MSS data.

PROGRAM NAME: COORD

LANGUAGE: FORTRAN IV

COMPUTER(S): CDC Cyber 74

SPECIAL PERIPHERALS: Disk Files

PURPOSE: COORD accepts pairs of Latitude-Longitude or UTM coordinates and Landsat pixel coordinates for Ground Control Points (GCP) and computes a least squares fit of the transformation matrix needed to map Landsat data into a standard coordinate system.
PROGRAM NAME: CLUSTER
LANGUAGE: FORTRAN IV
COMPUTER(S): Data General NOVA 2, CDC Cyber 74
SPECIAL PERIPHERALS: 2 Tape Drives

PURPOSE: CLUSTER is a sequential clustering algorithm which creates an unsupervised classification of Landsat or aircraft multispectral scanner (MSS) data using a Euclidean distance measure as a decision criterion. This system decides how many "different" types of land cover there are in a MSS scene. This system is dependent on user input parameters which specify the criteria for number of clusters, merging, creation of new clusters, and exclusion of clusters. This technique is often used to define training fields for supervised classification.

PROGRAM NAME: CLUST (NASA/ERL - Ronnie Pearson)
LANGUAGE: FORTRAN IV
COMPUTER: SEL 32/55
SPECIAL PERIPHERALS: 1 Tape Drive
Disk Files

PURPOSE/DESCRIPTION: CLUST is structurally similar to the SEARCH program for unsupervised development of class signatures. CLUST, however, uses a Euclidean distance measure in its sort and merge control logic for clusters. CLUST is often more useful than SEARCH in very broken terrain where fields of 40 acres are not common.
PROGRAM NAME: SUPERG (NASA/ERL - Marcellus Graham)

LANGUAGE: FORTRAN IV

COMPUTER: SEL 32/55

SPECIAL PERIPHERALS: 2 Tape Drives
                      Disk Files

PURPOSE/DESCRIPTION: SUPERG is a rectification program for Landsat MSS data which includes 2nd order mirror corrections into a least squares determination of a covariance matrix for conversion of Landsat pixels into a UTM coordinate system. The data are resampled along scan lines to satisfy scaling considerations and the output file contains new pixels which are directly related to the UTM system. Between 10 and 30 Ground Control Points are suggested for complete determination of the transformation. Rotation of the data to true North is not accomplished by this program.

PROGRAM NAME: TRAIN

LANGUAGE: FORTRAN IV

COMPUTER: Data General NOVA II

SPECIAL PERIPHERALS: COMTAL Interactive Color Video Display (3 images)
                      Magnetic Tape Drive
                      Disk Files

PURPOSE: TRAIN is an interactive training field selection and statistics generation program for Landsat digital analysis. In conjunction with a video display system with a cursor or joystick, a subset of a Landsat image may be selected by drawing an arbitrarily shaped polygon (up to 100 vertices) around an area on the display screen. The program keeps track of position on the input Goddard format CCT and calculates the normal statistics (mean and covariance) of the selected training fields. Histograms of the multivariate distributions are displayed on the display screen and the mean and variance, polygon vertices, and histograms may be saved on a disk file. The program acts in a question-answer mode which requires no knowledge of computer languages by the user.

A parallelopiped classifier is also implemented in this program which will classify a 256 x 256 element scene for one class in near real time.
ADD SIGM

This routine adds specified signatures to form new mean and covariance sequence:
ADD SIGM INPUT SIG FIL NEWSIG

CREATED AT GEORGIA TECH EES
PROGRAMMER: NICKOLAS L. FAUST

DIMENSION COV1(4,4), COV2(4,4), NUM1(12), NUM2(12), IORDER(60)
DIMENSION ID(6), AMEAN(4,60), BCOV(4,4,60), I1(30), I2(30), I3(30)
DIMENSION NP(60), NAM1(60), NAM2(60), NAM3(60), NAM4(60)
DIMENSION VM1(12), VM2(12), VM(4), ISW(2)
COMMON/NP/NP
ND=4
IP=12

COMARG AND OPEN STATEMENTS

CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, 11, ISW, IERR)
CALL COMARG(1, 11, ISW, IERR)
CALL COMARG(1, 12, ISW, IERR)
CALL COMARG(1, 13, ISW, IERR)
CALL OPEN(2, I1, 0, IE)
CALL FOPEN(4, I3, "B")

TYPE " INPUT NEWSIG NAME "
READ(11,200) N1, N2, N3, N4
READ(2) NSIG
WRITE(IP) NSIG

GET SIGNATURES LISTED IN INPUT FILE

CALL GSIG(AMEAN, BCOV, NSIG, 2, 3, I2, IORDER, NAM1, NAM2, NAM3, NAM4)
NUM1(1)=NP(1)
DO (J=1,4)
  VM1(J)=AMEAN(J,1)
  DO (L=1,4)
    COV1(J,L)=BCOV(J,L,1)
  ..FIN
..FIN
NS1=NSIG-1
LOOP TO COMBINE NSIC SIGNATURES

DO (K=1,NS1)
  : KP1=K+1
  : DO (J=1,4)
  :   : VM2(J)=ANEAN(J,KP1)
  :   : DO (L=1,4)
  :   :     : COV2(J,L)=BCOV(J,L,KP1)
  :   : ..FIN
  : ..FIN
  : NUM2(I)=NP(KP1)

CALL ADDSIG(COV1,COV2,VM1,VM2,ND,NUM1,NUM2)

DO (I=1,4)
  : VH(I)=VM1(I)
..FIN
WRITE(IP,201)N1,N2,N3,N4
WRITE(IP)NUM1(I),VM,COV1
DO (L=1,6) ID(L)=0
WRITE BINARY(4) ID
WRITE BINARY(4) N1,N2,N3,N4
WRITE BINARY(4) NUM1(I),VM,COV1
200 FORMAT(4A2)
201 FORMAT(2X,4A2)
STOP
END
C***********************************************************************
C
C 	 ALARM (SUBROUTINE)
C
C
THIS SUBROUTINE ALARMS 1 CLASS ON THE COMTAL C 
CORRESPONDING TO THE LAST SIGNATURE, C 
IT ALARMS TO GRAPHICS OVERLAY #10V
C
C***********************************************************************
C
C 	 CREATED AT GEORGIA TECH EES
C
C 	 PROGRAMMERS: NICKOLAS L. FAUST C 
ROBERT A. MADDOX
C
C***********************************************************************

SUBROUTINE ALRM2(IOV)
DIMENSION IMIN(4), IMAX(4), IX2(0:255, 3)
COMMON/TRANS/IMAG(0:255), IMX(0:511, 4), IMAG(0:511)
COMMON/HISTY/ICOUNT(4, 100)
EQUIVALENCE (IX2, IX2)
DO (I=0, IMIN) IMAG(I)=0
DO (I=0:255) CALL GWR(IOV, I, IMAG, 16)
DO (I=1, 4)
: DO (J=1, 100)
: : IF (ICOUNT(I, J) .GT. 0) GO TO 30
: ...FIN
30 CONTINUE
: ININC(I)=J
: DO (JJ=1, 100)
: : N=101-JJ
: : IF (ICOUNT(I, N) .GT. 0) GO TO 50
: ...FIN
50 CONTINUE
: IMAX(I)=N
...FIN
IMIN= IMIN(1)
IMIN2= IMIN(2)
IMIN4= IMIN(4)
IMAX= IMAX(1)
IMAX2= IMAX(2)
IMAX4= IMAX(4)
DO (K=0, 255)
: CALL IMLD(0, K, IMAG(0), 128)
: CALL IMLD(1, K, IMAG(128), 128)
: CALL IMLD(2, K, IMAG(256), 128)
: CALL UPAC8(IMAG, IX2(0, 1), 384)
: DO (L=0, 255)
: : IMAG1(L)=0
: : IF (IX2(L, 1) .LT. IMIN1) GOTO 70
: : IF (IX2(L, 1) .GT. IMAX1) GOTO 70
: : IF (IX2(L, 2) .LT. IMIN2) GOTO 70
: : IF (IX2(L, 2) .GT. IMAX2) GOTO 70
: : IF (IX2(L, 3) .LT. IMIN4) GOTO 70
: : IF (IX2(L, 3) .GT. IMAX4) GOTO 70
: : IMAG1(L)=1
70 CONTINUE
...FIN
: CALL PACB(IMAG1, IMAG, 16)
: CALL GWR(IOV, K, IMAG, 16)
...FIN
RETURN
END
ANGDIS (SUBROUTINE)

PURPOSE:
COMPUTES THE DISTANCES AND ANGLES BETWEEN A SET OF
VECTORS OF ARBITRARY DIMENSION

DESCRIPTION OF PARAMETERS

INPUT:
CALLING SEQUENCE
VM - NVM VECTORS OF DIMENSION ND, STORED BY COLUMNS
NVM - NUMBER OF VECTORS
ND - DIMENSION OF EACH VECTOR
IDISF - .EQ.1 COMPUTE EUCLIDEAN DISTANCE
         .NE.1 COMPUTE L1 DISTANCE

OUTPUT
CALLING SEQUENCE
R - AN NVM BY NVM MATRIX WITH I TH - J TH ELEMENT
  CORRESPONDING TO I TH - J TH VECTORS AND EQUALS ANGLE
  IF ABOVE DIAGONAL AND EQUALS DISTANCE IF BELOW
  DIAGONAL

CREATED AT NASA/JSC (ASTEP)

SUBROUTINE ANGDIS(VM,NVM,ND,IDISF,R)
DIMENSION VM(ND,NVM),R(NVM,NVM)
N = NVM - 1
DO 20 J = 1,N
R(J,J) = 0.
II = J + 1
DO 10 I = II,NVM
IF(IDISF.EQ.1) GO TO 6
D=0.
DO 4 K=1,ND
4 D=D+ABS(VM(K,J)-VM(K,I))
GO TO 8
6 CALL EDIST(VM(1,J),VM(1,I),ND,D)
8 CONTINUE
CALL ANGLE(VM(1,J),VM(1,I),ND,A)
R(J,J) = D
R(J,I) = A
10 CONTINUE
20 CONTINUE
R(NVM,NVM) = 0.
RETURN
END

43
C*********************************************************************
C             ANGLE (SUBROUTINE)
C*********************************************************************
C PURPOSE: 
COMPUTES THE ANGLE BETWEEN TWO VECTORS OF ARBITRARY 
DIMENSION 
C DESCRIPTION OF PARAMETERS 
C INPUT 
CALLING SEQUENCE 
V1 - 1ST VECTOR 
V2 - 2ND VECTOR 
ND - DIMENSION OF V1 AND V2 
C OUTPUT 
CALLING SEQUENCE 
A - ANGLE BETWEEN V1 AND V2 IN DEGREES 
C*********************************************************************

SUBROUTINE ANGLE ( V1, V2, ND, A )
DIMENSION V1(ND), V2(ND)
D1 = 0.
D2 = 0.
A = 0.
DO 10 I = 1, ND
D1 = D1 + V1(I)**2
D2 = D2 + V2(I)**2
10 A = A + V1(I)*V2(I)
IF ( D1.EQ.0.0 .OR. D2.EQ.0.0 ) GO TO 20
A = A/(SQRT(D1)*SQRT(D2))
A = 57.29578*ACOS( A )
15 RETURN
20 A = 90.0
GO TO 15
END

C*********************************************************************
C CREATED AT NASA/JSC (ASTEP) 
C*********************************************************************
AUT0811Z

DESIGNED TO OUTPUT CLASSIFIED TAPE IN PATTERN FORMAT
TO DISPLAY ON THE VERSATEC

SEQUENCE: AUT0811Z MTU: F INPUTFILE

CREATED AT GEORGIA TECH EES
PROGRAMMER: MICHAEL D. FURMAN

INTEGER IMAG(180), ITAPE(10), IOUT(360), IORDER(25,15), ISW(2)
INTEGER NC(25), IWORK(4100), IMAG3(180), INEM0:255, IMAG2(360)
INTEGER IWORK1(360), IWORK2(360), IWORK3(360), IWORK4(360)
INTEGER IWORK5(360), IWORK6(360), IWORK7(360), IWORK8(360)
INTEGER IWORK9(360), IWORK10(360), IWORK11(360), IWORK12(360)
INTEGER IMES(15,40), IFLD(10)
EQUIVALENCE (IWORK(1), IWORK1(1))
EQUIVALENCE (IWORK(359), IWORK2(1))
EQUIVALENCE (IWORK(717), IWORK3(1))
EQUIVALENCE (IWORK(1075), IWORK4(1))
EQUIVALENCE (IWORK(1433), IWORK5(1))
EQUIVALENCE (IWORK(1791), IWORK6(1))
EQUIVALENCE (IWORK(2149), IWORK7(1))
EQUIVALENCE (IWORK(2507), IWORK8(1))
EQUIVALENCE (IWORK(2865), IWORK9(1))
EQUIVALENCE (IWORK(3223), IWORK10(1))
EQUIVALENCE (IWORK(3581), IWORK11(1))
EQUIVALENCE (IWORK(3939), IWORK12(1))
COMMON/DUM/ ID(165), IWORK

DATA ID/
10,0,0,0,0,0,0,0,0,0,0,0,0
20,176K,102K,102K,102K,102K,102K,102K,102K,176K,0,
30,0,16K,16K,16K,16K,16K,16K,16K,16K,16K,7,3,
5176K,74K,30K,0,291K,303K,201K,0,30K,74K,176K,
6303K,303K,303K,303K,0,0,0,303K,303K,303K,303K,
730K,30K,30K,30K,30K,30K,30K,30K,30K,30K,30K,
830K,30K,30K,30K,30K,30K,30K,30K,30K,30K,30K,
93,7,16K,16K,14K,30K,30K,30K,30K,30K,30K,
1377K,303K,303K,303K,303K,303K,303K,303K,303K,303K,201K,
CALL OPEN(1,"COM. CM",1,IERR)
CALL COMARG(1,ITAPE,ISW,IERR)
CALL COMARG(1,ITAPE,ISW,IERR)
CALL COMARG(1,IFLD,ISW,IERR)
CALL FOPEN(3,IFLD,"B")
I0=3
CALL MTOPD(2, ITAPE, 0, IE)
DO (I1=1, 25)
  DO (I2=1, 15) IORDER(I1, I2)=0
  ...FIN
TYPE "DOES INPUT TAPE HAVE THRESHOLD? (2=Yes, 1=No)"
READ(IO) ITH
TYPE "LINE"
READ(IO) ILINE
TYPE "MAP PORTION (1 - 12) " , IANS2
READ(IO) IANS2
TYPE "LENGTH OF DATA"
READ(IO) IST
IEND=IEL+359
IF (IEND.GT.4100) IEND=4100
TYPE "PRODUCE OVERALL MAP? (1=Yes)"
READ(IO) IANS
TYPE "NUMBER OF GENERALIZED CLASSES"
READ(IO) IGEN
DO (I=1, IGEN)
  TYPE "INPUT CLASS DESCRIPTION"
  READ(IO, 101) (IMES(I, K), K=1, 20)
  TYPE "NUMBER OF SUBCLASSES"
  READ(IO) NC(I)
  NC1=NC(I)
  TYPE "INPUT SUBCLASSES"
  READ(IO) (IORDER(I, K), K=1, NC1)
  IF (I.NE.IGEN) TYPE "NEXT GROUP"
  ...FIN
50 DO (I=1, 253) IMAG2(I)=0
DO (I=1, 180) IMAG3(I)=-1
DO (I=0, 255) IMEM(I)=0
WRITE(12)
L2=154
DO (I=1, ILINE) CALL MTDIO(2, 0, IWORK2, IS, IE, IC)
IF (IANS.EQ.1)
  DO (I=1, 180) IOUT(I)=0
  DO (M5=1, IGEN)
    : L3=NC(M5)
    : DO (J2=1, L3) IMEM(IORDER(M5, J2))=L2
    : WRITE(12, 100) (IMES(M5, K), K=1, 20) (IORDER(M5, I), I=1, L3)
    : DO (J=1, 11)
    : : IOUT(15)=ID(L2+J)
    : : CALL MTX(IOUT, 180)
    : : ...FIN
    : L2=L2-11
  : ...FIN
WRITE(12)
: ...FIN

46
DO (M1=1, IGEN)
  IF (IANS.NE.1)
    L3=NC(M1)
    WRITE(12,100)(INES(N1,K),K=1,20)(IORDER(M1,I),I=1,L3)
  DO (I=1,180) IOUT(I)=0
  DO (J=0,255) IMEM(I)=0
  DO (J=1,11)
    L2=154
    DO (J2=1,L3)
      IMEM(IORDER(N1,J2))=L2
      IOUT(J2*4+9)=ID(L2+J)
      L2=L2-11
    ..FIN
    CALL MTX(IOUT,180)
    WRITE(12)
  ..FIN
  DO (M2=1,IST)
    DO (I=1,ITH)
      CALL MTDIO(2,0,IWORK,IS,IE,IC)
      IF (IC.LT.5) GOTO 55
    ..FIN
    ..i
    ..i
    .. FIN
    55 DO (I=1,11) CALL MTX(IMAG3,180)
    DO (I=1,700)
      DO (J=1,360) IOUT(J)=ID(1)
      DO (N10=1,1100) N11=N10
      CALL PAC8(IOUT,IMAG,360)
      CALL MTX(IOUT,180)
    ..FIN
    CALL MTDIO(2,10000K,IWORK,IS,IE,IC)
    DO (I=1,ILINE) CALL MTDIO(2,0,IWORK2,IS,IE,IC)
    IF (IANS.EQ.1)
      IANS=0
    GOTO 50
..FIN
DO (I=1,10) WRITE(12)
STOP FINISHED
100 FORMAT(1X,/,1X,20A1,/," CLASSES ",2014)
101 FORMAT(20A1)
END
**CHAN24**

**UNPACKS AND REFORMATS BENDIX 24-CHANNEL DATA FOR DISPLAY**

SEQUENCE CHAN24 MTU:F MTU:F

**C**********************************************************************
C
C
CIIAN24
C
CUNPACKS AND REFORMATS BENDIX 24-CHANNEL DATA FOR DISPLAY
C
CSEQUENCE CIIAN24 MTU:F MTU:F
C
C**********************************************************************
C
CCREATED AT GEORGIA TECH EES
C
CProgrammer: Michael D. Furman
C
C**********************************************************:
INTEGER IDATA(4100), IMAG(512), IMAC2(256), ITAPE1(5), ITAPE2(5)
INTEGER IWORK(256), IHEAD(30), IX(3), IY(3), ISW(2)
COMMON IX, IY
EQUIVALENCE (IX(1), IX1), (IX(2), IX2), (IY(1), IY1), (IY(3), IY3)
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, ITAPE1, ISW, IERR)
CALL COMARG(1, ITAPE2, ISW, IERR)
CALL COMARG(1, ITAPE1, ISW, IERR)
CALL COMARG(1, ITAPE2, ISW, IERR)
DO (I=1, 30) IHEAD(I)=0

10 FORMAT(1X, I2, "/", I2, "/", I2)
11 FORMAT(1X, "INPUT TAPE NO.", Z)
12 FORMAT(1X, "INPUT TAPE NUMBER? XXXX")
13 FORMAT(1X, "BLOCK SIZE= ", I3)
14 FORMAT(10A2)

C

**TYPE "DATE M. D. Y"**
ACCEPT IHEAD(4), IHEAD(5)
**TYPE "INPUT TAPE NUMBER? XXXX"**
ACCEPT IHEAD(1)
**TYPE "COMMENTS ON RUN -- 20 CHARACTERS"**
READ (11, 14) (IHEAD(I), I=10, 20)
CALL COLORSUB
**TYPE "FAST SCAN? (2= YES, 1=N0) "**
ACCEPT IFSCAN
**TYPE "DATA BLOCKS TO SKIP?"**
ACCEPT ISKB
**TYPE "INPUT BLOW-UP FACTOR"**
ACCEPT IBLUP
IBLUPM1=IBLUP-1
IELEM=1
IF (IFSCAN.EQ.1)
: TYPE "START WITH ELEMENT?"
: ACCEPT IELEM
: ..FIN
CALL SCALESUB
TYPE "INPUT CHANNEL NUMBER(1-24)"
ACCEPT ICHAN
ISKB=(ICHAN+2)/3+(ISKB*9)+2
INUM=5+ICHAN-(((ICHAN+2)/3+1)*3)
L=INUM*393+2+IELEM
LAST=L+256+((IFSCAN-1)*92)
IF (IELEM.GT.94) LAST=351+(INUM*393)
CALL MTOPD(3,ITAPE1,0,IER)
CALL MTOPD(4,ITAPE2,0,IER)
DO (I=1,ISKB)
CALL MTDIO(3,0,IDATA,IS,IER,ICNT)
M2=512/IBLUP
M3=0
DO (M1=1,M2)
: DO (N=1,IFSCAN)
: : CALL MTDIO(3,30010K,IDATA,IS,IE,IC)
: : CALL MTDIO(3,0,IDATA,IS,IE,IC)
: : IF (IC.LT.10) GOTO 150
: : ..FIN
: J=1
: DO (K=L,LAST,IFSCAN)
: : IMG(J)=IDATA(K)
: : J=J+1
: ..FIN
: BLOWUP-AREA
: DO (I=1,IBLUP)
: : CALL IMWRITE(0,M3,IMAG2,256)
: : M3=M3+1
: ..FIN
..FIN
150 DO (L1=1,3)
: PAUSE -- POSITION CURSOR AND HIT RETURN
: CALL RTARG(IX(L1),IY(L1))
: ..FIN
IX1=IX1/2
IX2=IX2/2
DO (1=1,3,2)
: CALL IMREAD(0,IY(1),IWORK,256)
: DO (12=IX1,IX2) IWORK(12)=255
: CALL IMWRITE(0,IY(1),IWORK,256)
: ..FIN
DO (1=1,IY1,IY3)
: CALL IMREAD(0,1,IWORK,256)
: IWORK(IX1)=255
: IWORK(IX2)=255
: CALL IMWRITE(0,1,IWORK,256)
: ..FIN
TYPE "PROPER RECTANGLE? (1=YES; 2=NO)"
ACCEPT IANS
IF (IANS.EQ.2) COTO 150
CALL MTD10(3, 10000K, IDATA, IS, IE, ICNT)
IX1=IX1*IFSCAN+IELEM-1
IX2=IX2*IFSCAN+IELEM-1
IY1=IY1*IFSCAN
IY3=IY3*IFSCAN
IPAS=9*IY1
IDIF=(IY3-IY1)
IDIF2=IX2-IX1
WRITE-HEADER
CALL MTD10(3, 30001K+IPAS, IDATA, IS, IE, ICNT)
DO (I=1, IDIF)
  CALL MTD10(3, 30001K, IDATA, IS, IE, ICNT)
  DO (K5=1,8)
    CALL MTD10(3, 0, IDATA, IS, IE, ICNT)
    DO (L6=3, 789, 393)
      J=1
      DO (L5=IX1, IX2)
        IWORK(J)=IDATA(L5+L6)
        J=J+1
      END
    CALL MTD10(4, 50000K+IY1, IWORK, IS, IE, ICNT)
   END
  CALL MTD10(4, 50036K, IHEAD, IS, IE, ICNT)
  END
END
TO WRITE-HEADER
  WRITE(12, 11)
    WRITE(12) IHEAD(1)
    WRITE(12, 10) IHEAD(3), IHEAD(4), IHEAD(5)
    WRITE(12, 12) (IHEAD(I), I=10,20)
    WRITE(12, 13) IDIF2
    CALL MTD10(4, 50036K, IHEAD, IS, IE, ICNT)
  END
TO BLOWUP-AREA
  DO (K7=0,255)
    K9=256-K7
    KDB=K9/IUBLUP
    IMAG(K9)=IMAG(KDB)
    END
  CALL PABC(0, IMAG, IMAG2, 512)
  END
STOP FIELD ERROR -- CHAN24 MTU:F MTU:F
END
**CLASSIFY**

This routine classifies polygons with a MAXLIK classifier. A table look-up feature is included.

**SEQUENCE:** CLASIFY INPUT DATAP MAXTAP SIGFIL

**CREATED AT GEORGIA TECH EES**

**PROGRAMMERS:** NICKOLAS L. FAUST
ROBERT A. MADDOX

**DIMENSION** AMEAN(4,60), BCOV(4,4,60), IVX(101), IVY(101), DET(60)
1, LA(133), COUNT(0:61), IBUF(1640), CLCON(61)
2, NAM1(60), NAM2(60), NAM3(60), NAM4(60), B1(4,4), B2(4,4)
DIMENSION JBUF(4,810), FIELD(17), ISWS(2), IORD(0:61)
DIMENSION DUM(61), DUM1(4,4,60), DUM2(4,4,60), ITABL(5,300)
CONM  JBUF
EQUIVALENCE (JBUF, DUM), (JBUF, DUM1), (JBUF, DUM2), (JBUF(1,20), DET),
(1,JBUF,B1), (JBUF(1,10), B2)

**CONARG CALLS AND OPEN STATEMENTS**

CALL OPEN(1, "COM.CM", 1, IE)
CALL CONARG(1, FIELD, ISWS, IE) ;GET PROGRAM NAME
CALL CONARG(1, FIELD, ISWS, IE) ;GET INPUT FILE NAME
CALL OPEN(2, FIELD, 0, IE) ;OPEN INPUT FILE
CALL CONARG(1, FIELD, ISWS, IE) ;GET DATA TAPE NAME
CALL NTOPD(3, FIELD, 0, IE) ;OPEN TAPE FILE
CALL CONARG(1, FIELD, ISWS, IE) ;GET OUTPUT TAPE NAME
CALL NTOPD(5, FIELD, 0, IE) ;OPEN TAPE OUTPUT FILE
CALL CONARG(1, FIELD, ISWS, IE) ;GET SIGNATURE FILE NAME

**IP=12**
**ND=4**
**BNV=-1.E+12**
**ITPTR=0** ;BIG NEGATIVE NUMBER
**INITIAL TABLE POINTER**
**READ(2,200) NSIG**

**GET SIGNATURES FOR CLASSIFICATION**

CALL GSIG(AMEAN, BCOV, NSIG, FIELD, NAM1, NAM2, NAM3, NAM4, CLCON)
DO (1=1, NSIG)
  : DO (JH=1,4)
    : : DO (JI=1,4)
      : : : : FIN
      : : FIN
    : CALL SYMINV(B2, B1, DETP, 4)
  : DO (JH=1,4)
    : : DO (JI=1,4)
      : : : : FIN
    : : FIN
  : DIVIDE DIAGONAL ENTRIES OF COVARIANCE MATRIX BY TWO
    : DO (J=1,4) BCOV(J,J,L) = BCOV(J,J,L)/2.
  : GET NATURAL LOG OF DETERMINANT
    : DET(L)=ALOG(DETP)
  : FIN

51
IYMIN = 10000
IXMIN = 10000
IYMAX = 0
WRITE (IP, 202)
READ (2, 200) NC
DO (I = 1, NC)
  WRITE (IP, 205) I
  READ (2, 206) IVX(I), IVY(I)
  IF (IVY(I) .LT. IYMIN) IYMIN = IVY(I)
  IF (IVX(I) .LT. IXMIN) IXMIN = IVX(I)
  IF (IVY(I) .GT. IYMAX) IYMAX = IVY(I)
END DO
IVX(NC+1) = IVX(1)
IVY(NC+1) = IVY(1)
NV = NC
IDEL = IYMAX - IYMIN
ISKIP = IYMIN + 1

SET UP THE APRIORI INFORMATION EITHER FROM,
A. THE INPUT FILE IF 'IAPR' EQUALS 1 IN INPUT FILE
OR B. SET APRIORI FOR ALL CLASSES EQUAL TO 1.0

READ (2, 201) NTE    ; NUMBER OF TABLE ENTRIES
READ (2, 200) ITH
IF (ITH .NE. 0)
  READ (2, 310) THRS
  THRS = ALOG (THRS)
END IF
READ (2, 200) IAPR
IF (IAPR .EQ. 0)
  DO (I = 1, NSIG)
    CLCON(I) = 1.
  END DO
END IF
WRITE (IP, 410)
DO (I = 1, NSIG)
  WRITE (IP, 420) I, NAM1(I), NAM2(I), NAM3(I), NAM4(I), CLCON(I)
END DO
WRITE (IP, 400)

COMPUTE CLASS CONSTANTS
PILOG = 2. * ALOG (2. * 3.141593)
DO (I = 1, NSIG)
  CLCON(I) = ALOG (CLCON(I)) - 0.5 * DET(I) - PILOG
END DO
CLCON(NSIG + 1) = BNV

SORT CLASS CONSTANTS FROM LARGEST TO SMALLEST
AND ALSO REORDER STATISTICS
DO (J = 1, NSIG)
  BG = -1. E + 10
  DO (I = 1, NSIG)
    IF (CLCON(I) .GT. BG)
      BG = CLCON(I)
      IND = I
    END IF
  END DO
  IORD(J) = IND
  DUM(J) = CLCON(IND)
  CLCON(IND) = BNV
END DO
IORD(61) = 61
IORD(0) = 0

52
REORDER CLASS CONSTANT ARRAY

DO (J=1,NSIG) CLCON(J)=DUM(J)
CLCON(NSIG+1)=BNV

REORDER COVARIANCE MATRICES

DO (I=1,NSIG)
  DO (J=1,ND)
    DO (K=1,ND) DUM1(J,K,I) = BCOV(J,K,IORD(I))
    ...FIN
  ...FIN
DO (I=1,NSIG)
  DO (J=1,ND)
    DO (K=1,ND) BCOV(J,K,I)=DUM1(J,K,I)
  ...FIN
...FIN

REORDER MEAN VECTORS

DO (I=1,NSIG)
  DO (J=1,ND) DUM2(J,I)=AMEAN(J,IORD(I))
...FIN
DO (I=1,NSIG)
  DO (J=1,ND) AMEAN(J,I)=DUM2(J,I)
  ...FIN
...FIN

SKIP RECORDS

CALL MTDIO(3,30000K+ISKIP,IBUF,IS,IEE)
DO (LL=1,812) IBUF(LL)=0
IBUF(1)=IDEL
IBUF(2)=IX1IIN
CALL MTDIO(5,50000K+810,IBUF,IS,IER,NW)
ISW=2
JQ=810
TOT=0
TOTIHITS=0
DO (I=1,61) COUNT(I)=0
DO (I=1,5) DO (J=1,NTE) ITABL(I,J)=0
...FIN
CALL FGTIME(IHR,IMIN,ISEC)
TYPE "LINE PROCESSING BEGAN @", IHR, IMIN, ISEC
DO (J=1,IDEL)
  HITS=0
  CALL MTDIO(3,0,IBUF,IS,IE,NW)
  CALL POLY2(ISKIP+J-1,ISW,IVX,IVY,NV,LA)
  JD=0
  DO (KO=1,JQ)
    DO (KP=1,4) JBUF(KP,KO)=0
    ...FIN
  ...FIN
  LA1=LA(1)
  DO (K1=1,LA1)
    LSUB=2*K1
    L2=LSUB+1
    JS=LA(LSUB)
    JF=LA(L2)
    JD=JD+JF-JS+1
    IFC(JF,CT,JQ) JF=JQ
    DO (K2=1,4)
      LL=((JS-1)/2)*4+K2
      LAST=LL+(JF-JS+1)*2
      J1=1
      DO (II=LL,LAST,4)
        JBUF(K2,J1)=ISIIFT(IBUF(II),-8)
        JBUF(K2,J1+1)=IAND(IBUF(II),377K)
        J1=J1+2
      ...FIN
  ...FIN
...FIN
CLASSIFY THE POINTS BY MAXIMUM LIKELIHOOD ALGORITHM

DO (JPT=1,JQ)
   IF (JBUF(1,JPT).EQ.0) GOTO 1000
   JB1=JBUF(1,JPT)
   JB2=JBUF(2,JPT)
   JB3=JBUF(3,JPT)
   JB4=JBUF(4,JPT)
   DO 1010 IC=1,NTE
      IF (JB3.NE.ITABL(3,IC)) GOTO 1010
      IF (JB2.NE.ITABL(2,IC)) GOTO 1010
      IF (JB4.NE.ITABL(4,IC)) GOTO 1010
      IF (JB1.NE.ITABL(1,IC)) GOTO 1010
      INDEX=ITABL(5,IC)
      HITS=HITS+1
      GOTO 1050
  1010 CONTINUE
   N=1
   PROB=BNV
  1040 VD1=JB1-AMEAN(1,M)
   VD2=JB2-AMEAN(2,M)
   VD3=JB3-AMEAN(3,M)
   VD4=JB4-AMEAN(4,M)
   CLTHR=CLCON(M) - (VD1*VD1*BCOV(1,1,M) + VD2*VD2*BCOV(2,2,M) + VD3*VD3*BCOV(3,3,M) + VD4*VD4*BCOV(4,4,M))
   IF (CLTHR.LT.PROB) GOTO 1020
   PROB=CLTHR
   INDEX=M
  1020 M=M+1
   IF (PROB.LT.CLCON(M)) GOTO 1040
   IF (ITH.EQ.0) GOTO 1030
  1030 ITPTR=ITPTR+1
   IF (ITPTR.GT.NTE) ITPTR=1
   ITABL(1,ITPTR)=JB1
   ITABL(2,ITPTR)=JB2
   ITABL(3,ITPTR)=JB3
   ITABL(4,ITPTR)=JB4
   ITABL(5,ITPTR)=INDEX
   GOTO 1050
  1000 INDEX=0
  1050 IBUF(JPT)=IORD(INDEX)
..FIN

TOT=TOT+JD
TOTHITS=TOTHITS+HITS
DO (K=1,JQ)
   IBK=IBUF(K)
   COUNT(IBK)=COUNT(IBK)+1
..FIN
CALL FGTIME(IHR,IMIN,ISEC)
WRITE(10,207)J,HITS
TYPE "TIME = ",IHR,IMIN,ISEC
CALL MTDIO(5,50000K+810,IBUF,IS,IER)
..FIN

54
WRITE (IP, 400)
WRITE (IP, 430)
DO (K=1, NSIG)
  PER = (COUNT(K) * 100) / TOT
  IKK = IORD(K)
  WRITE (IP, 440) K, NAM1(K), NAM2(K), NAM3(K), NAM4(K), IKK
  CLCON(IKK), COUNT(K), PER
END
WRITE (IP, 450) COUNT(61)
WRITE (IP) TOTHITS, TOT
WRITE (IP, 400)
DO (KL=1, 810) IBUF(KL) = 0
DO (KL=1, 200) CALL MTDIO(5, 50000K+810, IBUF, IS, IER)
CALL MTDIO(5, 60000K, IBUF, IS, IER)
CALL MTDIO(5, 60000K, IBUF, IS, IER)
TYPE "TOTAL HITS, TOTAL POINTS ", TOTHITS, TOT

200 FORMAT(12)
201 FORMAT(13)
202 FORMAT(2X, "INPUT # OF CORNERS")
205 FORMAT(2X, "INPUT CORNER # ", I3, "J, I")
206 FORMAT(2I4)
207 FORMAT(2X, "LINE #", I5, " PROCESSED THERE WERE ", F8.0, " HITS")
310 FORMAT(F10.6)
400 FORMAT(///)
410 FORMAT(2X, "NUMBER", 2X, "NAME", 6X, "APRIORI")
420 FORMAT(4X, I2, 2X, 4A2, 2X, F10.6)
430 FORMAT(2X, "NUMBER NAME RANKING CONSTANT # PIXELS PERCENT")
440 FORMAT(4X, I2, 2X, 4A2, 3X, I2, 2X, F10.6, 2X, F8.1, 3X, F5.1)
450 FORMAT(///, 4X, "TOTAL POINTS NOT CLASSIFIED = ", F10.0, ///)
STOP
END
CLEANUP

USED TO CLEAN UP SIGNATURE, VERTEX, OR HIST FILES

SEQUENCE: CLEANUP NAMES FILEIN FILEOUT

CREATED AT GEORGIA TECH EES

PROGRAMMER: NICKOLAS L. FAUST

DIMENSION IFIL(20), JFIL(20), KFIL(20), SM(4), COV(4,4), I(S)
DIMENSION ICOUNT(4, 100), IVX(101), IVY(101), ISW(2)
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARC(1, IFIL, ISW, IERR)
CALL COMARC(1, IFIL, ISW, IERR)
CALL COMARC(1, JFIL, ISW, IERR)
CALL COMARC(1, JFIL, ISW, IERR)
CALL FOPEN(2, IFIL, "B")
CALL FOPEN(4, JFIL, "B")
WRITE(10, 401)
READ(11) ICON
CALL FOPEN(3, JFIL, "B")
READ BINARY(3) I
WRITE BINARY(4) I
CALL FCLOSE(3)
FOREVER:
CALL FOPEN(3, JFIL, "B")
READ BINARY(3) I
ISW=0
READ(2, 100, END=121) NA1, NA2, NA3, NA4
REPEAT WHILE (ISW.EQ.0):
   READ BINARY(3, END=120) NS1, NS2, NS3, NS4
   IF (ICON.EQ.1) READ BINARY(3) NP, SM, COV
   IF (ICON.EQ.2)
     READ BINARY(3) K, IYMIN, IYMAX
     DO (IV=1, K) READ BINARY(3) IVV, IVX(IV), IVY(IV)
   FIN
   IF (ICON.EQ.3) READ BINARY(3) ICOUNT
   IF (ICON.EQ.1) WRITE BINARY(4) NS1, NS2, NS3, NS4
   IF (ICON.EQ.2)
     WRITE BINARY(4) K, IYMIN, IYMAX
     DO (IV=1, K) WRITE BINARY(4) IV, IVX(IV), IVY(IV)
   FIN
   IF (ICON.EQ.3) WRITE BINARY(4) ICOUNT
   ISW=1
   CALL FCLOSE(3)
   FIN
   FIN
FIN
STOP
100 FORMAT(4A2)
401 FORMAT(2x, "INPUT SWITCH, 1 - SIG, 2 - VER, 3 - HIS")
120 STOP - SIGNATURE NOT FOUND
121 STOP - NORMAL EXIT
END
SUBROUTINE CLUSTA(V, VM, ND, NV, NVG, NVMMAX, C, S, RP, R, NPC, NPT, PLIST, NEC, NET, NMIN, NMC, NMT, IPASS, IMG, TDIS, JPTP, RMINM, RMINV, VMP, VAR)

COMMON/INOUT/NOUT, NIN
INTEGER PLIST
INTEGER TDIS, V
DIMENSION V(ND, NV), VM(ND, NVMMAX), NVG(NVMMAX), PLIST(NVMMAX),
* VS(24), IMG(NV), TDIS(NV)
COMMON/DIST/IDIST
DIMENSION RMINM(NVMMAX), RMINV(NVMMAX)
DIMENSION VMP(ND, NVMMAX), VAR(ND, NVMMAX)
1111 FORMAT(' ND ', I5)
JPT=0
NS=0
TESTS ON - NEXT POINT, SMALL CLUSTER ELIMINATION, MERGING TIME, AND PLIST UPDATE TIME

12 CONTINUE
   IF(IPASS.EQ.2) GO TO 16
   NEC=NEC+NS
   IF(NEC.GE.NET) GO TO 70
14 NMC=NMC+NS
   IF(NMC.GE.NMT) GO TO 80
16 NPC=NPC+NS
   IF(NPC.GE.NPT) GO TO 90
18 JPT=JPT+1
   IF(JPT.GT.(NV-1)) GO TO 100
   JSC=JPT
   JPT=JPT+1

STRIP GENERATION

19 I=1
20 T=ABS(V(I,JPT) - V(I,JSC))
   IF(T.GT.S) GO TO 22
      J=J+1
   IF(J.LE.ND) GO TO 20
      JPT=JPT+1
   IF(JPT.LE.NV) GO TO 19
22 JPT=JPT-1

COMPUTE MEAN OF STRIP

NS=JPT-JSC+1
   DO 26 I=1,ND
26 VS(I)=0.
   DO 30 J=JSC,JPT
      DO 28 I=1,ND
28 VS(I)=VS(I)+V(I,J)
   CONTINUE
   T=FLOAT(NS)
   DO 32 I=1,ND
32 VS(I)=VS(I)/T

PRIORITY SEARCH FOR NEAREST CLUSTER

34 RMIN=1.E+10
   DO 40 J=1,NVM
      L=PLIST(J)
      D=0.
      IF(IDIST.EQ.2) GO TO 3601
      DO 36 J=1,ND
36 D=D+ABS(VM(J,L)-VS(J))
      GO TO 3602
3601 CONTINUE
   DO 3603 J=1,ND
3603 D=D+(VM(J,L)-VS(J))**2
   D=SQR(D)
3602 CONTINUE
1112 FORMAT( )
   IF(D.GT.RMIN) GO TO 40
      RMIN=D
      J1=L
   IF(RMIN.LT.RP) GO TO 42
40 CONTINUE
TEST DISTANCE TO CLUSTER FOR ASSIGNMENT OF STRIP

42 IF(RMIN.GT.R) GO TO 50

FIRST PASS - ASSIGN TO J1

IF(IPASS.EQ.2) GO TO 200
CALL MODIFY(VM(1,J1),VS,NVG(J1),NS,ND)
GO TO 12

FIRST PASS - NEW GROUP, POSSIBLE ELIMINATION OF SMALLEST

50 IF(IPASS.EQ.2) GO TO 210
NVM=NVM+1
J1=NVM
IF(NVM.LE.NVMMAX) GO TO 60
NVM=NVMMAX
NVGP=10000
DO 52 I=2,NVM
IF(NVG(I).GT.NVGP) GO TO 52
IMIN=I
NVGP=NVG(I)
52 CONTINUE

SMALL CLUSTER TESTS AND POSSIBLE ELIMINATIONS

70 NEC=0
NPC=NPT
I=1
72 I=I+1
74 IF(I.GT.NVM) GO TO 14
IF(NVG(I).GT.NMIN) GO TO 72
JPTT=JPTP+JPT
IF(IP.NE.0) WRITE(NOUT,900) I,NVG(I),JPTT,NVM
NVG(1)=NVG(1)+NVG(I)
call pack(VM,ND,NVM,1)
call pack(NVG,1,NVM,1)
NVH=NVH-1
GO TO 74

TESTS FOR MERGING AND POSSIBLE MERGING

80 NPC=NPT
NMC=0
C
C 82  I2=NVM-1
   IF(I2.LE.0) GO TO 16
   RMIN=1.E+10
   DO 88 I=1,I2
      J=I+1
      IF(IDIST.EQ.2) GO TO 8400
      DO 84 L=1,ND
         D=D+ABS(VM(L,I)-VM(L,J))
      84 J=J+1
      84 CONTINUE
      1007 FORMAT(   )
      IF(D.GT.RMIN) GO TO 86
      RMIN=D
     J1=I
     J2=J
  86 CONTINUE
  88 CONTINUE
C
C THRESHOLD TEST
C
C 82  IF(RMIN.GT.C) GO TO 16
C
C CLUSTER MERGING, J2 INTO J1
C
JPTT=JPT+JPT
IF(IP.NE.0) WRITE(NOUT,902) J2,NVG(J2),J1,NVG(J1),NVM,JPTT
902 FORMAT(7H MERGER,6H J2 = ,I2,7H NJ2 = ,I4,
   *6H J1 = ,I2,7H NJ1 = ,I4,7H NVM = ,I2,7H JPT = ,I4)
CALL MODIFY(VM(1,J1),VM(1,J2),NVG(J1),NVG(J2),ND)
CALL PACK(VM,ND,NVM,J2)
CALL PACK(NVG,1,NVM,J2)
NVM=NVM-1 GO TO 82
C
C PLIST UPDATE
C
90 NPC=0
CALL UPPLT(PLIST,NVG,NVM)
GO TO 18
C
C POSSIBLE SPECIAL CASE FOR LAST POINT
C
100 IF(JPT.NE.NV) RETURN
   DO 102 I=1,ND
      VS(I)=V(I,NV)
      NS=1
      JSC=JPT
  102 IMG(J)=J1
GO TO 34
C
C SECOND PASS - ASSGN TO J1 AND UPDATE STATISTICS
C
200 CALL THRDST(RMIN(J1),RMINV(J1),NVG(J1),RMIN,NS)
   DO 202 J=JSC,JPT
      CALL SEQST(VMP(1,J1),VAR(1,J1),NVG(J1),ND,V(1,J))
      TDIS(J)=RMIN
  202 IMG(J)=J1
GO TO 12
C
C SECOND PASS - ASSGN TO UNASSIGNED
C
210 DO 212 J=JSC,JPT
      TDIS(J)=100000
  212 IMG(J)=1
      NVG(1)=NVG(1)+NS
GO TO 12
END
**COMBINE**

**THIS PROGRAM COMBINES SIGNATURES**

**WITH A GIVEN SPECIFIER FROM TWO FILES**

**SEQUENCE:** COMBINE FILE1 FILE2 FILE3

**CREATED AT GEORGIA TECH EES**

**PROGRAMMER:** NICKOLAS L. FAUST

**DIMENSION** A(4,20), B(4,4,20), N1(20), N21(20), N31(20), N41(20)

**DIMENSION** IF1(20), IF2(20), IF3(20), C(4,20), D(4,4,20)

**DIMENSION** E(4,30), F(4,4,30), N12(20), N22(20), N32(20), N42(20)

**DIMENSION** N1(30), N2(30), N3(30), N4(30)

**DIMENSION** SM(4), COV(4,4), ISW(2)

**CALL OPEN(1, "COM.CH", 1, IERR)**

**CALL COMARC(1, IF1, ISW, IERR)**

**CALL COMARC(1, IF1, ISW, IERR)**

**CALL COMARC(1, IF2, ISW, IERR)**

**CALL COMARC(1, IF3, ISW, IERR)**

NSIG=20

WRITE(10,103)

ND=4

IP=10

READ(11) ISW

CALL FOPEN(2, IF1, "B")

CALL FOPEN(3, IF2, "B")

CALL FOPEN(4, IF3, "B")

READ BINARY(2) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3

IF(ISW.NE.1)READ BINARY(3) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3

WRITE(10,101)

READ(11,100) NA1, NA2

WRITE(10,102)

NUN2=2

READ(11,104) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3

NSIG=5

CALL GSIG1(A, B, NA1, NA2, NUM1, NSIG, NUN1, N11, N21, N31, N41)

WRITE(12,105)

NUN2=3

NSIG=20

IF(ISW.EQ.2)

: CALL GSIG1(C, D, NA1, NA2, NUM2, NSIG, NUN2, N12, N22, N32, N42)

: WRITE(12,106)

: ...FIN
DO (K=1, NUM1)
  N1(K)=N11(K)
  N2(K)=N21(K)
  N3(K)=N31(K)
  N4(K)=N41(K)
  DO (K1=1, ND)
    E(K1,K)=A(K1,K)
  DO (K2=1, ND)
    F(K1,K2,K)=B(K1,K2,K)
  ...FIN
  ...FIN
IF(ISW.EQ.1)
  DO (M=1, NUM2)
    ML=NUM1+M
    N1(ML)=N12(M)
    N2(ML)=N22(M)
    N3(ML)=N32(M)
    N4(ML)=N42(M)
    DO (M1=1, ND)
      E(M1,ML)=C(M1,M)
    DO (M2=1, ND)
      F(M1,M2,ML)=D(M1,M2,M)
    ...FIN
    ...FIN
  ...FIN
  IF(ISW.EQ.1) ML=NUM1
NP=100
WRITE BINARY(4) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
DO (I=1, 11L)
  WRITE BINARY(4) N1(I), N2(I), N3(I), N4(I)
  DO (I1=1, ND)
    SM(I1)=E(I1, I)
  DO (12=1, ND)
    COV(I1, 12)=F(I1, 12, I)
  ...FIN
  ...FIN
  WRITE BINARY(4) NP, SM, COV
  WRITE(IP, 104) N1(I), N2(I), N3(I), N4(I)
  WRITE(IP) NP, SM, COV
...FIN
100 FORMAT(2A2)
101 FORMAT(2X, "INPUT NA1, NA2")
102 FORMAT(2X, "INPUT TAPE, DATE IN 216 FORMAT")
103 FORMAT(2X, "INPUT # OF FILES")
104 FORMAT(6A2)
105 FORMAT(/2X, "ABOVE ARE SIGNATURES FROM FILE1")
106 FORMAT(/2X, "ABOVE ARE SIGNATURES FROM FILE2")
STOP
END
C************************************************************************
C *
C COPY5
C
THIS ROUTINE ADDS 5 SIGNATURE VERTICES, OR HIST FILES AND PUTS THEM IN A COPYFILE
C
SEQUENCE: COPY5 OUT IN1 IN2 IN3 IN4 IN5
C
C************************************************************************
C *
C CREATED AT GEORGIA TECH EES
C *
C PROGRAMMER: NICKOLAS L. FAUST
C
C************************************************************************

DIMENSION IOUT(20), IN1(20), IN2(20), IN3(20), IN4(20), IN5(20)
DIMENSION ICOUNT(4,100), I(6), SM(4), COV(4,4), IVX(101)
DIMENSION IVY(101), ISW(2)

CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, IOUT, ISW, IERR)
CALL COMARG(1, IOUT, ISW, IERR)
CALL COMARG(1, IN1, ISW, IERR)
CALL COMARG(1, IN2, ISW, IERR)
CALL COMARG(1, IN3, ISW, IERR)
CALL COMARG(1, IN4, ISW, IERR)
CALL COMARG(1, IN5, ISW, IERR)

IPR=10
WRITE(IPR, 401)
READ(1) ICON
WRITE(IPR, 402)
READ(1) M
L=N+2
CALL FOPEN(2, IOUT, "B")
CALL FOPEN(3, IN1, "B")
CALL FOPEN(4, IN2, "B")
CALL FOPEN(5, IN3, "B")
CALL FOPEN(6, IN4, "B")
CALL FOPEN(7, IN5, "B")
DO (K=3, L) READ BINARY(K) I
WRITE BINARY(2) I
 TYPE " L = " ,L
DO (KL=3, L)
: FOREVER
: : READ BINARY(KL, END=120) NS1, NS2, NS3, NS4
: : WRITE(IPR, 403) NS1, NS2, NS3, NS4
: : WRITE BINARY(2) NS1, NS2, NS3, NS4
: : IF (ICON.EQ.1) : READ BINARY(KL) NP, SM, COV
: : TYPE " NP = " ,NP
: : WRITE BINARY(2) NP, SM, COV
: : ..FIN
: : IF (ICON.EQ.2) : READ BINARY(KL) K, IYMIN, IYMAX
: : DO (K1=1, K) READ BINARY(KL) K1, IVX(K1), IVY(K1)
: : WRITE BINARY(2) K, IYMIN, IYMAX
: : DO (K1=1, K) WRITE BINARY(2) K1, IVX(K1), IVY(K1)
: : ..FIN
: : IF (ICON.EQ.3) : READ BINARY(KL) ICOUNT
: : WRITE BINARY(2) ICOUNT
: : ..FIN
: ..FIN
120 : CONTINUE
: ..FIN
401 FORMAT(2X, " INPUT SWITCH - 1 - SIG , 2 - VER , 3 - HIS")
402 FORMAT(2X, " INPUT # OF FILES ")
403 FORMAT(2X, 4A2)
STOP
END
INTEGER IC(2,256), IXY(2,2)
ACCEPT "# OF POINTS, STRING OF POINTS ",NP,( (IC(I,J),I=1,2),J=1,NP)
DO (I=1,2)
 : DO (J=I,2) IXY(I,J)=IC(I,1)
 : : .FIN
DO (J=1,NP)
 : DO (K=1,2)
 : : IF ( (IC(K,J).LT.IXY(K,1)) .AND. (IC(K,J).GE.0) ) IXY(K,1)=IC(K,
 : : : : IF (IC(K,J).GT.IXY(K,2)) IXY(K,2)=IC(K,J)
 : : : .FIN
 : ..FIN
FAX=(IXY(1,2)-IXY(1,1))/511.0
FAY=(IXY(2,2)-IXY(2,1))/511.0
WHEN (FAX.GT.FAY) FACT=FAX
ELSE FACT=FAY
DO (I=1,NP)
 : DO (J=1,2) IC(J,I)=(IC(J,1)-IXY(J,1))/FACT
 : : .FIN
NP1=NP-1
DO (I=1,NP1) CALL VECTOR(0,IC(1,I),IC(2,I),IC(1,I+1),IC(2,I+1),0,200)
CALL VECTOR(0,IC(1,1),IC(2,1),IC(1,1),IC(2,1),0,200)
STOP
END
**CRDEM4**

CREATE DEMO FROM DISPLAY IMAGE INCLUDING FUNCTION MEMORY, COLOR MEMORY, AND GRAPHICS

SEQUENCE: CRDEM4 MTU:F

CREATED AT GEORGIA TECH EES

PROGRAMER: FRED L. THOMPSON

```
INTEGER ISW(2), INPUT(512), IARRAY(64), IBRAY(0:255), IFLD(10)
CALL OPEN (1, "CON.CM", 1, IE)
CALL COMARG (1, IFLD, ISW, IE)
CALL COMARG (1, IFLD, ISW, IE)
CALL MTOPO (3, IFLD, 0, IE)
DO (I=1,512) INPUT(I)=O
CALL RCM (0, INPUT(3))
ACCEPT "512 OR 256", ISZ
ACCEPT "TYPE 1 TO RECORD GRAPHICS", IANS
TYPE "TYPE A THIRTY CHARACTER DESCRIPTION"
TYPE ".........................."
READ (11,100) (INPUT(I), I=68,98)
100 FORMAT (30A1)
INPUT(1)=ISZ
INPUT(2)=IANS
CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
DO (I=0,2)
: CALL RFUM (I, INPUT)
: CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
:..FIN
IEND=ISZ-1
ISZ1=ISZ/2
DO (IY=0,IEND)
: CALL IMRD (0, IY, INPUT, ISZ1)
: CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
: IF (ISZ.EQ.256)
: : DO (I=1,2)
: : : CALL IMRD (I, IY, INPUT, ISZ1)
: : : CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
: : :..FIN
: : IF (IANS.EQ.1)
: : : DO (I=0,3)
: : : : CALL GRD (I, IY, INPUT, 16)
: : : : CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
: : : :..FIN
: :..FIN
..FIN
DO (I=1,2) CALL MTDIO (3, 60000K, INPUT, IST, IE, ICNT)
STOP
END
```
C**********************************************************************
C
C
CYTAPE
C
MAKES FILE INTO A CYBER 74 COMPATABLE FORMAT
C
SEQUENCE: CYTAPE FILE FILE
C
C*********************************************************************
C
C
CREATED AT GEORGIA TECH EES
C
PROGRAMMER: MICHAEL D. FURMAN
C
C**********************************************************************

DIMENSION S(30), IFILE(10), NS(4), IFILE2(10), ITAP(3), IDAT(3), ISW(2)
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, IFILE, ISW, IERR)
CALL COMARG(1, IFILE2, ISW, IERR)
CALL FOPEN(2, IFILE, "B")
CALL FOPEN(3, IFILE2, "B")
READ BINARY(2, END=205) ITAP, IDAT
FOREVER
: DO (I=1,30) S(I)=0.
: READ BINARY(2, END=205) NS
: READ BINARY(2, END=205) NP,(S(I), I=1,20)
: WRITE BINARY(3) NS
: WRITE(3)
: WRITE(3, 100) NP,(S(I), I=1,20)
: FIN
STOP
205 STOP FINISHED
100 FORMAT(1X,14,5(F12.8,/))
END
**DISDEM4**

**Displays Demo Tapes to Comtal Color Display**

**With Graphics Color Memory and Function Memory**

**Sequence:** DISDEM4 MTU:F

**Created at Georgia Tech EES**

**Programmer:** FRED L. THOMPSON

```
INTEGER ISW(2), INPUT(512), IFLD(10)
CALL OPEN (1, "COM.CM", 1, IE)
CALL COMARG (1, IFLD, ISW, IE)
CALL COMARG (1, IFLD, ISW, IE)
CALL NTOPD (3, IFLD, 0, IE)
FOREVER
  CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
  ISZ=INPUT(1)
  IANS=INPUT(2)
  WRITE (10, 100) (INPUT(I), I=68, 98)
100  FORMAT (1X, 30A1)
  IF (ISZ.EQ.512) PAUSE SET BIC FOR 512 HIT RETURN
  IF (ISZ.EQ.256) PAUSE SET BIC FOR 256 HIT RETURN
  CALL WCM (0, INPUT(3))
  DO (I=0, 2)
    CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
    CALL WFUM (I, INPUT)
    ..FIN
  IEND=ISZ-1
  ISZ1=ISZ/2
  DO (IY=0, IEND)
    CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
    CALL IMWR (0, IY, INPUT, ISZ1)
    IF (ISZ.EQ.256)
      DO (*1, 1, 2)
        CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
        CALL IMWR (I, IY, INPUT, ISZ1)
        ..FIN
      *1, IF (IANS.EQ.1)
      DO (I=0, 3)
        CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
        CALL GWR (1, IY, INPUT, 16)
        ..FIN
      ..FIN
  ..FIN
  CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
  PAUSE HIT RETURN FOR NEXT FILE CTRL A TO END
..FIN
STOP
END
```
C***********************************************************************
C                         DRAWER
C                        USED WITH ELIP TO DRAW 2-D ELLIPSES OF SIGNATURES
C                        ON THE VERSATEC
C***********************************************************************
C                        CREATED AT GEORGIA TECH EES
C                        PROGRAMMER: G. DAVID GENTRY
C***********************************************************************

SUBROUTINE DRAWER(SN, E, COV, AX1, AX2, R, ID)

DIMENSION R(2,2),SM(2),E(2,4),COV(2,2)

IF(ID.GE.1)GO TO 9
CALL MODE(7,8,8,9999.)
CALL MODE(9,180.,10.,0.)
CALL AXES(7.0, "CHANNEL", 7.0, "CHANNEL")
CALL NOTE(-.5,.5,AX2,1000)
CALL NOTE(.5,.5,AX1,1000)
9 10=10
IF(ID.EQ.0)IDC=43
IF(ID.EQ.1)IDC=31
IF(ID.EQ.2)IDC=42
THETA=ATAN2(R(2,1),R(1,1))
IF(COV(1,2) .LT. 0.)THETA=ATAN2(R(2,2),R(1,2))
IF(COV(I,2) .LT. 0.)WRITE(10,12)
12 FORMAT(//" COV(X,Y) IS NEGATIVE")

WRITE(10,11)THETA
11 FORMAT(//" THE ROTATION ANGLE IS",F6.2," DEGREES")
DO (I=1800.,2600.,10)
  J=((I-1800.)/10)+1
  X1=FLOAT(J)/10.
  X=X1-SM(1)
  IF(COV(1,2) .LT. 0.)GO TO 6
  A=E(2,1)*SIN(THETA)**2+E(1,1)*COS(THETA)**2
  B=2.*X*(E(2,1)*SIN(THETA))*(COS(THETA))-E(1,1)*SIN(THETA)
  C=X**2*(E(1,1)*COS(THETA)**2+E(2,1)*SIN(THETA)**2)-3.*E(1,1)*E(2,1)
1: CONTINUE
5 CONTINUE
6 CONTINUE
10 CONTINUE
END
C************************************************************************
C                  DTAPE
C************************************************************************
GENERAL TAPE DUMPING ROUTINE FOR READING AND DEBUGGING FOREIGN TAPES

SEQUENCE: DTAPE MTU: F

CREATED AT GEORGIA TECH EES
PROGRAMMER: MICHAEL D. FURMAN

C************************************************************************

INTEGER IDATA(4110), IOUT(8220), IFLD(10), ISW(2)
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARC(1, IFLD, ISW, IERR)
CALL COMARC(1, IFLD, ISW, IERR)

DO (1=1,4110) IDATA(I)=0
DO (1=1,8220) IOUT(I)=0
CALL MTDIO(3,0,IDATA,IS,IE,IC)
TYPE "RECORDS PER BLOCK=",IC
IF (IP.EQ.0)
CALL UPACS(IOUT,IC)
IF (IQ.EQ.2) CALL EBCDIC(IOUT,IC)
IC=IC*2
WRITE(12,102)(IOUT(I),I=1,IC)
ELSE
IF ((IN.EQ.1).AND.(IP.EQ.1)) WRITE(12,100)(IDATA(I),I=1,IC)
IF ((IP.EQ.0).AND.(IN.EQ.0)) WRITE(12,103)(IDATA(I),I=1,IC)
IF ((IP.EQ.0).AND.(IN.EQ.1)) WRITE(12,101)(IOUT(I),I=1,IC)
WRITE(12,104)(IOUT(I),I=1,IC)
ELSE
WRITE(12)
END STOP
SUBROUTINE EBCDIC (INPUT, IOUT, IPR)
INTEGER INPUT( IPR), IOUT( IPR)
COMMON/DUM1/ ICNTAB(263)
DATA ICNTAB/1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
    1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
    1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
    1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
    1H?, 1H? / 
C
DO (I=1, IPR)
    KI=INPUT(I)+1
    WHEN (KI.LE.256)
        IOUT(I)=ICNTAB(KI)
    ELSE IOUT(I)=ICNTAB(1)
..FIN
END

CREATED AT GEORGIA TECH EES
PROGRAMMER: MICHAEL D. FURMAN
SUBROUTINE EDIST (V1, V2, ND, D)
DIMENSION V1(ND), V2(ND)
D = 0.
DO 10 I = 1, ND
10 D = D + (V1(I) - V2(I))**2
D = SQRT(D)
RETURN
END

C************************************************************************
C
EDIST (SUBROUTINE)
C
PURPOSE:
COMPUTES THE DISTANCE BETWEEN TWO VECTORS
C
DESCRIPTION OF PARAMETERS
C
CALLING SEQUENCE
INPUT
V1, V2 - TWO VECTORS
ND - DIMENSION OF THE VECTORS
C
OUTPUT
CALLING SEQUENCE
D - CONTAINS DISTANCE BETWEEN V1 AND V2
C
************************************************************************

CREATED AT NASA/JSC (ASTEP)

SUBROUTINE EDIST (V1, V2, ND, D)
DIMENSION V1(ND), V2(ND)
D = 0.
DO 10 I = 1, ND
10 D = D + (V1(I) - V2(I))**2
D = SQRT(D)
RETURN
END
EIGEN (SUBROUTINE)

COMPUTES EIGENVALUES AND EIGENVECTORS OF REAL SYMMETRICAL MATRIX

CREATED AT NASA/JSC (ASTEP)

SUBROUTINE EIGEN(AA,N,MV,A,E,R)
DIMENSION AA(N,N),R(N,N),E(N),A(N,N)

C***************************************************

SUBROUTINE EIGEN(AA,N,MV,A,E,R)
DIMENSION AA(N,N),R(N,N),E(N),A(N,N)

202 FORMAT("SUBROUTINE EIGEN ERROR RETURN-FINAL NORM HAS NOT BEEN REACHED AFTER 100 ITERATIONS")
DO (I=1,N)
: : DO (J=1,N)
: : : A(I,J)=AA(I,J)
: : : FIN
: : FIN
RANGE=1.E-6
IF(MV-1)10,25,10
10
DO (I=1,N)
: : DO (J=1,N)
: : : IF(I .EQ. J) GO TO 15
: : : R(I,J)=0.
: : : GO TO 20
15
: : CONTINUE
: : FIN
: : FIN
25
ANORM=0.
DO (I=1,N)
: : DO (J=1,N)
: : : IF(I .EQ. J) GO TO 35
: : : ANORM=ANORM+A(I,J)*A(I,J)
35
: : CONTINUE
: : FIN
: : FIN
IF(ANORM)165,165,40
40
ANORM=SQRT(ANORM)
ANRM2=ANORM*RANGE
ICNT=0
IND=0
THR=ANORM
45
THR=THR/FLOAT(N)
ICNT=ICNT+1
IF(ICNT .EQ. 100) GO TO 200
50
MQ=2
55
MP=1
X=.5*(A(MP,MP)-A(MQ,MQ))
62
IF(ABS(A(MP,MQ))-THR)138,65,65
65
IND=1
X=.5*(A(MP,MP)-A(MQ,MQ))
Y=-A(MP,MQ)/SQRT(A(MP,MQ)*A(MP,MQ)+X*X)
IF(X) 70, 75, 75

70 Y = -Y

75 SINX = Y / SQRT(2. * (1. + SQRT(1. - Y * Y))
SINX2 = SINX * SINX
COSX = SQRT(1. - SINX2)
COSX2 = COSX * COSX
SINCS = SINX * COSX
XX = A(MP, MP)
YY = A(MQ, MQ)
ZZ = A(MP, MQ)
DO (I = 1, N)
   X = A(I, MP) * COSX - A(I, MQ) * SINX
   A(I, MQ) = A(I, MP) * SINX + A(I, MQ) * COSX
   A(I, MP) = X
   IF(MV - 1) 120, 125, 120
120 X = R(I, MP) * COSX - R(I, MQ) * SINX
   R(I, MQ) = R(I, MP) * SINX + R(I, MQ) * COSX
   R(I, MP) = X
125 CONTINUE
..FIN
X = 2. * ZZ * SINCS
Y = (XX * COSX2) + (YY * SINX2) - X
X = (XX * SINX2) + (YY * COSX2) + X
A(MP, MP) = Y
A(MQ, MQ) = X
A(MP, MQ) = (XX - YY) * SINCS + ZZ * (COSX2 - SINX2)
A(MQ, MP) = 0.
DO (I = 1, N)
   A(MP, I) = A(I, MP)
   A(MQ, I) = A(I, MQ)
..FIN
138 IF(MP .NE. (MQ - 1)) GO TO 140
140 IF(MQ .NE. N) GO TO 145
145 IF(IND .NE. 1) GO TO 160
IND = 0
GO TO 50
160 IF(THR - ANRMX) 165, 165, 45
165 CONTINUE
DO (I = 1, N)
   DO (J = 1, N)
      IF(I .NE. J) GO TO 190
      E(I) = A(I, J)
190   CONTINUE
..FIN
GO TO 210
200 WRITE(10, 202)
210 RETURN
END
C******************************************************************************
C
C
C
C ELIP
C
C DRAWS 2-D ELLIPSES OF COVARIANCE MATRICIES
C
C SEQUENCE: ELIP REQFIL SIGFILC
C
C******************************************************************************
C
C CREATED AT GEORGIA TECH EES
C
C PROGRAMMER: G. DAVID GENTRY
C
C******************************************************************************

DIMENSION REQ(30), SIGFIL(30), E(2, 4), R(2, 2), D(2, 2), IORDER(20)
DIMENSION COVIN(4, 4, 20), SMIN(4, 20), N1(20), N2(20), N3(20), N4(20)
DIMENSION EXVE(2, 2), SM1(2), ISW(2)
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, REQ, ISW, IERR)
CALL COMARG(1, REQ, ISW, IERR)
CALL COMARG(1, SIGFIL, ISW, IERR)
CALL OPEN(2, REQ, "B")
I0=10
ACCEPT "INPUT NUMBER OF ELIPSES PER PLOT ", NUNS
IN=11
ND=2
NSIG=20
CALL MODE(1, 1, 1, -1, -1)
MSIG=NSIG/NUMS
DO(JV=1, MSIG)
1
NSIG=NUMS
2
KO=0
2
CALL GSIG(SNIN, COVIN, NSIG, 2, 3, SIGFIL, IORDER, N1, N2, N3, N4)
2
DO (JJ=1, 3)
2
DO (KK=K, 4)
2
AX1=JJ
2
AX2=KK
2
ID=0
2
DO (II=1, NSIG)
2
SM1(1)=255.-SMIN(JJ, II)
2
SM1(2)=255.-SMIN(KK, II)
2
COVIN(1, 1)=COVIN(JJ, JJ, II)
2
COVIN(1, 2)=COVIN(JJ, KK, II)
2
COVIN(2, 1)=COVIN(KK, JJ, II)
2
COVIN(2, 2)=COVIN(KK, KK, II)
2
CALL FACANL(COV1, SM1, ND, E, R, D)
2
CALL DRAWER(SH1, E, COV1, AX1, AX2, R, ID)
2
ID=ID+1
2
FIN
2
KO=KO+1
2
IF(KO.EQ.1)
2
CALL DRAW(0., 0., 1., 0000)
2
CALL DRAW(-.5, .5, 1, 1001)
2
FIN
2
IF(KO.EQ.2)
2
KO=0
2
CALL DRAW(0., 0., 1., 0000)
2
CALL DRAW(0., -9, 5, 1, 1001)
2
CALL DRAW(0., 0., 1., 9000)
2
FIN
FIN
CALL DRAW(0, 0, 0, 9999)
STOP
END
**FACANL (SUBROUTINE)**

**USED WITH ELIP TO DO FACTOR ANALYSIS OF COV. MATRIX**

**CREATED BY NASA JSC**

---

```fortran
SUBROUTINE FACANL(COV, SM, ND, E, R, D)
DIMENSION COV(ND, ND), SM(ND), E(ND, 4), R(ND, ND), D(ND, ND)
MV=0
CALL EIGEN(COV, ND, MV, D, E, R)
ND1=ND-1
IF(ND1 .EQ. 0) RETURN
DO (J=1, ND1)
   EMAX=0.
   DO (I=J, ND)
      IF(E(I, 1) .LT. EMAX) GO TO 50
      EMAX=E(I, 1)
      IS=I
   CONTINUE
   E(J, 1)=E(IS, 1)
   DO (I=1, ND)
      DUM=R(I, J)
      R(I, J)=R(I, IS)
      R(I, IS)=DUM
   CONTINUE
   FIN
   DUM=1./DUM
   EMAX=0.
   DO (I=1, ND)
      E(I, 2)=DUM*E(I, 1)
      E(I, 3)=EMAX+E(I, 2)
      EMAX=E(I, 3)
   CONTINUE
   DUM=0.
   DO (I=1, ND)
      DUM=DUM+SM(I)**2
   CONTINUE
   DUM=SQR(T(DUM))
   DO (J=1, ND)
      E(J, 4)=0.
      DO (I=1, ND)
         E(J, 4)=E(J, 4)+SM(I)*R(I, J)
      CONTINUE
      E(J, 4)=57.29578*ACOS(E(J, 4))
   CONTINUE
END
```
INTEGER IN(512,3), IMAG1(256), IMAG2(512), IMAG3(256)
INTEGER IN2(512,3), IN3(512,3)
ACCEPT "512 OR 256 MODE ", IMODE
IMOM1=IMODE-1
IMOM2=IMODE-2
IANS=0
IF (IMODE.EQ.256)
  ACCEPT "ALL 3 IMAGES AT ONCE? (1=YES) ", IANS
  IF (IANS.NE.1) ACCEPT "IMAGE NUMBER(0-2) ", IM2
  ..FIN
DO (IM=0,2)
  IF (IANS.NE.1).AND.(IMODE.NE.512)) IM=IM2
  M3=2
  M4=3
  DO (I=0,1)
    CALL IMREAD(IM,I,IMAG1,256)
    CALL UPACB(IMAG1, IN(I,(I+1)), 256)
    ..FIN
  DO (N5=2,IMOM2)
    M5M1=M5-1
    CALL IMREAD(IM,M5,IMAG1,256)
    CALL UPACB(IMAG1, IN(1, M4), 256)
    DO (I=1,MODE) IMAG2(I)=0
    DO (J1=1,3)
      DO (K=2, IMOM1) IMAG2(K)=IMAG2(K)+IN(K-1,J1)+IN(K,J1)+IN(K+
      ..FIN
      DO (I=1,IMODE)
        IMAG2(I)=IN(1, M3)*2-(( IMAG2(I)/9.0)+0.5)
        IF (IMAG2(I).LT.0) IMAG2(I)=0
        IF (IMAG2(I).GT.255) IMAG2(I)=255
        ..FIN
      CALL PACB(IMAG2, IMAG3, 512)
      CALL IMWRITE(IM, M5M1, IMAG3, 256)
      WHEN (M4.EQ.3) M4=1
      ELSE M4=M4+1
      WHEN (M3.EQ.3) M3=1
      ELSE M3=M3+1
      ..FIN
    IF (IANS.NE.1).OR.(IMODE.EQ.512)) STOP
  ..FIN
CALL BACK
END
GETPOLY

This program sums areas of all maximum likelihood class within a specified polygon. Polygon data is then dumped to tape.

Sequence: GETPOLY INPUT MTU:F MTU:F

Created at Georgia Tech EES

Programmer: Nickolas L. Faust

DIMENSION COUNT(60), JBUF(3300), IVX(201), IVY(201)
DIMENSION LA(233), KBUF(3300)
DIMENSION IFIL(20), ITAP(20), JTAP(20), ISW(2)
IN=11
IO=10
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, IFIL, ISW, IERR)
CALL COMARG(1, ITAP, ISW, IERR)
CALL COMARG(1, JTAP, ISW, IERR)
CALL FOPEN(IIN, IFIL, "B")
CALL MTOPD(3, ITAP, 0, IE)
CALL MTOPD(4, JTAP, 0, IE)
WRITE(I10, 400)
DO(1=1, 133) LA(I)=0
400 FORMAT(2X, "INPUT 1 FOR OUTPUT OF POLYGON TO TAPE")
DO(1=1, 133) COUNT(I)=0.0
READ(IIN) IOUT
WRITE(IIO) IOUT
ICT=10000
WRITE(I10, 401)
401 FORMAT(2X, "INPUT 1 TO USE A THRESHOLD")
READ(IIN) IZ
WRITE(IIO) IZ
IF(IZ.EQ.i)
: WRITE(I10, 402)
: FORMAT(2X, "INPUT THRESHOLD ")
: READ(IIN) ICT
: WRITE(IIO) ICT
: FIN
ACCEPT " START LINE OF CLASSIFIED DATA ? ", ISTART
IYMIN=10000
IXMIN=10000
IYMAX=0
READ(IIN) NV
WRITE(IIO) NV
201 FORMAT(212)
DO(1=1, NV)
: READ(IIN, 206) IVX(I), IVY(I)
: IVY(I)=IVY(I)-ISTART+1
: IF(IVY(I).LT.0) IVY(I)=0
: IF(IVX(I).LT.0) IVX(I)=0
: WRITE(IIO) IVX(I), IVY(I)
: IF(IVY(I).LT.IYMIN) IYMIN=IVY(I)
: IF(IVX(I).LT.IXMIN) IXMIN=IVX(I)
: IF(IVY(I).GT.IYMAX) IYMAX=IVY(I)
: IF(IVX(I).GT.IXMAX) IXMAX=IVX(I)
: FIN

77
IVX(NV+1) = IVX(1)
IVY(NV+1) = IVY(1)
IDEL = IYMAX - IYMIN + 1
WRITE(IIO) IDEL
IF(IOUT.EQ.1) THEN
  JBUF(1) = IDEL
  CALL MTDIO(4,50000K+3300,JBUF,IS,IE)
END IF
ISKIP = (IYMIN-1)*2+1
IF(ISKIP.GT.0)
206 FORMAT(2I4)
   DO (K=1,ISKIP)
      CALL MTDIO(3,0,JBUF,IS,IE)
      WRITE(IIO,28) JS,JF,JV
      JS1 = JS-1
      IF(JS1.NE.0)
         DO (K2=JV,JS1) JBUF(142)=60
         END DO
      END IF
      JV=JF+1
      DO (I=J1,3300) JBUF(I)=60
         DO (I=J1,3300) KBUF(I)=10000
      END DO
   END DO
28 FORMAT(2X,"JS,JF,JV",3I10)
   JS1 = JS-1
   IF(JS1.NE.0)
      DO (K2=JV,JS1) JBUF(K2)=60
      END DO
   IF(JS1.GT.60) LS1=60
   COUNT(LS1) = COUNT(LS1) + 1
   JV=JS1+1
   DO (J1=JS1,JF1)
   END DO
   JF1 = JF + 1
   DO (J1=JF1,3300) JBUF(J1)=60
      DO (J1=JF1,3300) KBUF(J1)=10000
   END DO
   IF(IOUT.EQ.1) THEN
      CALL MTDIO(4,50000K+3300,JBUF,IS,IE)
      CALL MTDIO(4,50000K+3300,KBUF,IS,IE)
      WRITE(10,211) J
      ACRE = COUNT(K)*1.05
      WRITE(10,100) K,COUNT(K),ACRE
   END IF
211 FORMAT(2X,"LINE # ",I5," PROCESSED")
   DO (K=1,60)
      ACRE = COUNT(K)*1.05
      WRITE(10,100) K,COUNT(K),ACRE
   END DO
END
SUBROUTINE GSIG(A, B, NSIG, ITEMSIG, N1, N2, N3, N4, APR)
DIMENSION A(4, 60), B(4, 4, 60), SM(4), COV(4, 4), ITEMSIG(34)
DIMENSION N1(60), N2(60), N3(60), N4(60), APR(60)
IP=12
NUN1=2
NUN2=4
DO (I=1, NSIG)
    READ(NUN1, 100) NA1, NA2, NA3, NA4, APRI
    N1(I)=NA1
    N2(I)=NA2
    N3(I)=NA3
    N4(I)=NA4
    APRI(I)=APRI
    CALL FOPEN(NUN2, ITEMSIG, "B")
    READ BINARY(NUN2) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
    ISET=0
    WHILE(ISET.EQ.0)
        READ BINARY(NUN2, END=120) NS1, NS2, NS3, NS4
        READ BINARY(NUN2) NP, SM, COV
        IF(NS1.EQ.NA1.AND.NS2.EQ.NA2)
            IP(NS3.EQ.NA3.AND.NS4.EQ.NA4)
            DO (K=1, 4)
                A(K, I)=SM(K)
                DO (L=1, 4)
                    B(K, L, I)=COV(K, L)
                END DO
            END DO
            ISET=1
        CALL FCLOSE(NUN2)
        END DO
    END WHILE
RETURN
100 FORMAT(4A2, F10.8)
120 STOP FILE NOT FOUND
END
SUBROUTINE HIST(NUM)
DIMENSION ICOUNT(4,100), MAX(4), MIN(4)
COMMON /DTRANS/ IMODL(256), IX(512,4), IDUM(512)
COMMON /HISTO/ ICOUNT

ND=4
DO ( IJ=1,100)
  DO ( J=1,ND)
    ICOUNT(J1, IJ)=0
  END
  END

DO ( K=1, NUM)
  J=IX(K, I)
  IF(J, EQ, 99) = 99
  IF(J, GT, 99) = 99
  ICOUNT(1, J) = ICOUNT(1, J) + 1
  END

IF(KCH=1, ND)
  MIN(KCH)=1000
  MAX(KCH)=0
  DO ( KK=1, 60)
    MIN(KCH)=MIN(KCH, KK)
    MAX(KCH)=MAX(KCH, KK)
    END
  IF(MAX(KCH, EQ, 0) = MAX(KCH)
  DO ( JJ=1, 60)
    ICOUNT(KCH, JJ) = (ICOUNT(KCH, JJ)*20)/MAX(KCH)
  END

DO ( L=1, 20)
  LI=256-L
  DO ( K=1, 512)
    IDUM(K) = 0
    DO ( KCH=1, 4)
      ISUB=(KCH-1)*60+NJ
      IF(ICOUNT(KCH, NJ)) GT, LI) IDUM(ISUB) = 140
    END
  IDUM(1) = 140
  IDUM(50) = 140
  IDUM(120) = 140
  IDUM(180) = 140
  CALL CPACK1(IDUM, IMODL, 0)
  WHEN(L, NE, 1) CALL IMWRITE(0, LI, IMODL, 256)
  ELSE
    DO ( L2=1, 255)
      IMODL(L2) = 106214K
    END
    CALL IMWRITE(0, LI, IMODL, 256)
  END
ACCEPT "HISTO ENDED, ENTER 1 TO CONTINUE : ", ISTOP
RETURN
END
HISTPR

THIS PROGRAM PRINTS HISTOGRAMS AND STATISTICS FOR ALL SIGNATURES IN A FILE

SEQUENCE: HISTPR HISTFILE SIGFILE

CREATED AT GEORGIA TECH EES

PROGRAMMER: NICKOLAS L. FAUST

DIMENSION ICOUNT(4,100), IFIL(20), IMAX(4), ICOL(123)
DIMENSION JFIL(20), COV(4,4), SM(4)
INTEGER Z, ISW(2)
COMMON /LBLANK/ IBL
DATA IBL /1H/
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, IFIL, ISW, IERR)
CALL COMARG(1, JFIL, ISW, IERR)
CALL COMARG(1, IFIL, ISW, IERR)
ACCEPT " WHICH OUTPUT DEVICE DO YOU WANT (10 OR 12) ? ", Z
CALL FOPEN(2, IFIL, "B") CALL FOPEN(3, JFIL, "B")
READ BINARY(2) IT1, IT2, IT3, ID1, ID2, ID3
READ BINARY(3) IT1, IT2, IT3, ID1, ID2, ID3
WRITE(Z, 102) IT1, IT2, IT3
WRITE(Z, 101) ID1, ID2, ID3
FOREVER
: READ BINARY(2, END= 120) NA1, NA2, NA3, NA4
: WRITE(Z, 102) NA1, NA2, NA3, NA4
: READ BINARY(2) ICOUNT
: READ BINARY(3) NA1, NA2, NA3, NA4
: READ BINARY(3) NP, SM, COV
: WRITE(Z, 102) NA1, NA2, NA3, NA4
: WRITE(Z) NP, SM, COV
: DO (I=1,4) IMAX(I)=0
: DO (I=1,4)
: DO (J=1,100)
: IF (ICOUNT(I,J).GT.IMAX(I)) IMAX(I)=ICOUNT(I,J)
: FIN
: IF (IMAX(1).EQ.0) IMAX(1)=1
: DO (JJ=1,60)
: ICOUNT(1, JJ)=(ICOUNT(JJ)*10)/IMAX(1)
: FIN
: ICOL(2)=1H
: ICOL(63)=1H
: WRITE(Z, 104) ICOL
: WRITE(2, 105)
: WRITE(2, 105)
: WRITE(2, 105)
: WRITE(2, 105) :FIN
: ICOL(KO)=1H*
: IF (ICOUNT(KO, K3).GE.K2) ICOL(K1)=1H*
: IF (ICOUNT(K12, K3).GE.K2) ICOL(K10)=1H*
: FIN
: ICOL(2)=1H
: ICOL(63)=1H
: WRITE(Z, 104) ICOL
: DO (KO=1,123) ICOL(KO)=1H=
: WRITE(Z, 104) ICOL
: WRITE(Z, 105)
: WRITE(Z, 105)
: WRITE(Z, 105)
: WRITE(Z, 105)
: FIN
100  FORMAT(2X, "TAPE IDENTIFIER = ", 312)
101  FORMAT(2X, "SIGNATURE TAKEN ON = " , I2, "/", I2, "/", I2)
102  FORMAT(1X, 4A2)
104  FORMAT(123A1)
106  FORMAT(4A2)
105  FORMAT(//////
120  STOP
     END
SUBROUTINE INITCL(VM,NVG,NVM,KNO)
DIMENSION VM(240),NVG(20)
COMMON/INOUT/IOUT,INP
1040 WRITE(IOUT,1050)
1050 FORMAT(3811 CHOOSE VALUES FOR INITIALIZATION FROM)
WRITE(IOUT,1055)
1055 FORMAT(15H ZERO OLD NEW)
READ(INP,1030)IAN
1030 FORMAT(A6)
IF(IAN.EQ.4HZERO) GO TO 1100
IF(IAN.EQ.3HOLD) GO TO 1200
IF(IAN.EQ.3HNEW) GO TO 1150
WRITE(IOUT,1060)IAN
1060 FORMAT(1H ,A6,23H IS NOT A VALID CHOICE.) GO TO 1040
1100 CONTINUE
DO 51 I=1,240
51 VM(I)=0.0
DO 52 I=1,20
52 NVG(I)=0
NVM=1
GO TO 1200
1150 CONTINUE
1170 WRITE(IOUT,1160)
1160 FORMAT(21H SININIT VM,NVG,NVM)
WRITE(IOUT,1022)NVM,(NVG(I),I=1,NVM)
CALL MATPRT(VM,KNO,KNO,NVM,5HMEANS)
WRITE(IOUT,1020)
1020 FORMAT(32H TYPE YES IF INPUTS ARE CORRECT.)
READ(INP,1030)IAN
IF(IAN.NE.3HYES) GO TO 1170
1200 CONTINUE
RETURN
END
INPUT SIG
ALLOWS MANUAL INPUT OF Signature FILE

CREATED AT GEORGIA TECH EES
PROGRAMMER: NICKOLAS L. FAUST

DIMENSION COV(4,4), XM(4)
CALL OPEN(3, "INSIG", 0, IE)
CALL FOPEN(4, "HSIG", "B")
READ(3) NSIG
ID1 = 01
ID2 = 22
ID3 = 77
IT1 = 99
IT2 = 99
IT3 = 99
WRITE BINARY(4) IT1, IT2, IT3, ID1, ID2, ID3
DO (L = 1, NSIG)
  WRITE(10, 101)
  READ(3, 100) NA1, NA2, NA3, NA4
  WRITE(10, 102)
  READ(3) NP
  WRITE(10, 103)
  READ(3) XM
  DO (K = 1, 4)
    WRITE(10, 104) K
    READ(3) (COV(K, J), J = 1, 4)
  END DO
  WRITE BINARY(4) NA1, NA2, NA3, NA4
  WRITE BINARY(4) NP, XM, COV
END DO

100 FORMAT(4A2)
101 FORMAT(2X, "INPUT NA1, NA2, NA3, NA4")
102 FORMAT(2X, "INPUT NUMBER OF POINTS")
103 FORMAT(2X, "INPUT (XM(I), I=1,4")
104 FORMAT(2X, "INPUT ROW # " , I1, " OF COV - (COV(K, J), J=1,4")
STOP
END
C****************************************************************************************************************************
C
C L I NBOX (SUBROUTINE)
C
C USED TO CALCULATE EXTREMES FOR A LINEAR CLASSIFICATION
C
C****************************************************************************************************************************
C
C CREATED AT GEORGIA TECH EES
C
C PROGRAMER: NICKOLAS L. FAUST
C
C****************************************************************************************************************************

SUBROUTINE LINBOX(A_MEAN, B_COV, NSIG, ND)
DIMENSION A_MEAN(4,10), B_COV(4,4,10)
INTEGER DELLIN
COMMON/BOX/DELLIN(4,10,2)
DO (K=1,NSIG)
 : DO (I=1,ND)
 : DIF=(SQRT(B_COV(I,I,K))/2.)*3.
 : DELLIN(I,K,1)=A_MEAN(I,K)-DIF
 : DELLIN(I,K,2)=A_MEAN(I,K)+DIF
 : ..FIN
 : ..FIN
RETURN
END
LINCLASS

THIS ROUTINE CLASSIFIES POLYGONS WITH A LINEAR CLASSIFIER AND CREATES THRESHOLD ARRAY

SEQUENCE: LINCLASS INPUT DATAP SIGFIL LINTAP

CREATED AT GEORGIA TECH EES

PROGRAMMER: NICKOLAS FAUST

FILE STATEMENTS

CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, I1, ISW, IERR)
CALL COMARG(1, I2, ISTNT, IERR)
CALL COMARG(1, I3, ISW, IERR)
CALL COMARG(1, I4, ISW, IERR)

CALL OPEN(2, I1, 0, IE)
CALL MTOPD(5, I4, 0, IE)

CALL MTOPD(3, I2, 0, IE)
ND=4
NUN1=2
NUN2=4
READ(2, 200) NSIG
WRITE(IP) NSIG

CALL GSIC(AMEAN, BCOV, NSIG, NUN1, NUN2, I3, IORDER, NAM1, NAM2, NAM3, NAM4)

IYMIN=1000
CALL LINBOX(AMEAN, BCOV, NSIG, ND)
IXMIN=1000
IYMAX=0
WRITE(IP, 202)
READ(2, 200) NC
DO (I=1, NC)
    WRITE(IP, 205) I
    READ(2, 206) IVX(I), IVY(I)
    WRITE(IP, 206) IVX(I), IVY(I)
    IF(IVX(I) .LT. IYMIN) IYMIN=IVX(I)
    IF(IVY(I) .LT. IXMIN) IXMIN=IVY(I)
    IF(IVX(I) .GT. IYMAX) IYMAX=IVX(I)
    IF(IVY(I) .GT. IXMAX) IXMAX=IVY(I)
..FIN
IVX(NC+1) = IVX(1)
IVY(NC+1) = IVY(1)
NV=NC
IDEL=IYMAX-IYMIN
ISKIP=IYMIN+1
CALL MTD10(3, 30000K+ISKIP, IBUF, IS, IEE)
DO (LL=1, 812)
    IDUM(LL) = 0
..FIN
IDUM(1) = IDEL
IDUM(2) = IXMIN
WRITE(1P) IDUM(1), IDUM(2)
CALL MTD10(5, 50000K+812, IDUM, IS, IER, NW)
WRITE(1P) NW
DO (J=1, IDEL)
    CALL MTD10(3, 0, IBUF, IS, IE, NW)
    ISW = 2
    CALL POLY2(ISKIP+J-1, ISW, IVX, IVY, NV, LA)
    JS = LA(2)
    JF = LA(3)
    CALL UNPAC4(IBUF, JBUF, JS, JF, 810, 1650)
    JQ = JF-JS+1
NN = NSIG
CALL LINEAR(JBUF, NN, IC, ND, NN, AMEAN, BCOV, JQ, IDUM, NVG, THRESH)
WRITE(10, 207) J
JMOVE = JS-1
DO (LL=1, 810)
    LV = 813-LL
    IDUM(LV) = IDUM(LV-2)
    THRESH(LV) = THRESH(LV-2)
..FIN
IDUM(1) = JMOVE
IDUM(2) = JQ
CALL MTD10(5, 50000K+812, IDUM, IS, IER)
CALL MTD10(5, 50000K+812, THRESH, IS, IER)
TOT = TOT+JQ
DO (K=1, JQ)
    LSUB = IDUM(K+2)
    COUNT(LSUB) = COUNT(LSUB) + 1
..FIN
..FIN
DO (K=1, NSIG)
    PER = (COUNT(K)*100)/TOT
    WRITE(12, 102) K, COUNT(K), PER, NAM1(K), NAM2(K), NAM3(K), NAM4(K)
..FIN
CALL MTD10(5, 60000K, IDUM, IS, IER)
CALL MTD10(5, 60000K, IDUM, IS, IER)
FORMAT(2X, "CLASS ", I4, " NUMBER ", F7.0, " PERCENT ", F4.0, 2X, 4A2)
102 FORMAT(2X, "INPUT # OF CORNERS")
103 FORMAT(2X, 4A2)
200 FORMAT(12)
202 FORMAT(2X, "INPUT # OF CORNERS")
205 FORMAT(2X, "INPUT CORNER ", I3, "J, I")
206 FORMAT(214)
207 FORMAT(2X, "LINE #", I5, " PROCESSED")
STOP
END
C**********************************************************
C
C       LINEAR (SUBROUTINE)
C
C       CLASSIFIES LANDSAT DATA WITH LINEAR DISTANCE
C
C**********************************************************
C
C       CREATED AT GEORGIA TECH EES
C
C       PROGRAMMER:  NICKOLAS L. FAUST
C
C**********************************************************

SUBROUTINE LINEAR(V, NVM, ICOUNT, ND, NSIC, AMEAN, BCOV, NL, IDUN, NVC, ITRESH)
    DIMENSION AMEAN(ND, NVM), BCOV(ND, ND, NVM)
    INTEGER DELLIN
    DIMENSION VC(ND, NL)
    DIMENSION IDUM(812), THRESH(812), NVC(10)
    INTEGER V
    COMMON/BOX/DELLIN(4, 10, 2)
    ICOUNT=1
    DO (LL=1, NL) IDUM(LL)=9
    DO (LL=1, NL) THRESH(LL)=0.
    JPT=0
    FOREVER
      JPT=JPT+1
      IF(JPT.GT.NL) RETURN
      IMIN=1
      DIST=1.E+6
      DO (K=1, NVM)
        KSUM=0
        SUM=0.0
        DO (L=1, ND)
            SUM=SUM+ABS(AMEAN(L, K)-V(L, JPT))
          KSUM=KSUM+1
          FIN
          IF(KSUM.EQ.4)
            IF(SUM.LT.DIST)
              DIST=SUM
              IMIN=K
              FIN
          FIN
      FIN
      THRESH(JPT)=DIST
      IDUN(JPT)=IMIN
      NVC(IMIN)=NVC(IMIN)+1
    FIN
100 FORMAT(2X, "ENTERED LINEAR ", I6, " TIMES")
101 FORMAT(2X, "EXITED LINEAR")
RETURN
END
LISTALL

PRINTS ALL SIGNATURES, VERTECIES, OR HISTOGRAMS
FOR A SPECIFIC FILE

SEQUENCE: LISTALL FILE

CREATED AT GEORGIA TECH EES

PROGRAMMER: NICKOLAS L. FAUST

DIMENSION IFIL(20), SM(4), COV(4,4), ICOUNT(4,100)
DIMENSION IVX(101), IVY(101)
DIMENSION ISWS(2), FIELD(17)
CALL OPEN(1, "CON.CMV", 1, IE)
CALL COMARG(1, FIELD, ISWS, IE)
CALL COMARG(1, FIELD, ISWS, IE)
CALL FOPEN(2, FIELD, "B")
WRITE(10, 555)
READ(11) IOUT

FORMAT(2X, " INPUT OUTPUT DEVICE ≠ (10, 12")
WRITE(10, 401)
READ(11) ICON
WRITE(10, 403)
READ(11) ITOT
READ BINARY(2) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
WRITE(IOUT, 100) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
FOREVER :
READ BINARY(2, END=129) NS1, NS2, NS3, NS4
IF( ICON.EQ.1) READ BINARY(2) NP, SM, COV
IF( ICON.EQ.2) :
READ BINARY(2) K, IYMIN, IYMAX
DO (K1=1, K) READ BINARY(2) K2, IVX(K1), IVY(K1)
FIN
IF( ICON.EQ.3) READ BINARY(2) ICOUNT
WRITE(IOUT, 101) NS1, NS2, NS3, NS4
IF( ICON.EQ.1) :
IF( ITOT.EQ.1) WRITE(IOUT) NP, SM, COV
FIN
IF( ICON.EQ.2) :
IF( ITOT.EQ.1) WRITE(IOUT) K, IYMIN, IYMAX
DO (K1=1, K) WRITE(IOUT) K1, IVX(K1), IVY(K1)
FIN
FIN
IF( ICON.EQ.3) :
IF( ITOT.EQ.1) WRITE(IOUT) ICOUNT
FIN
FIN
FORMAT(1X, 6I2)
FORMAT(1X, 6I2)
FORMAT(2X, " INPUT SWITCH, 1 -SIG, 2 -VER, 3 -HIS")
FORMAT(4I10)
FORMAT(2X, " TOTAL LIST - 1, OR NAME LIST - 0 ?")
STOP
END
LISTSIG

PRINTS ALL SIGNATURES FOR A GIVEN FILE

SEQUENCE: LISTSIG FILE

DIMENSION IFIL(20), SM(4), COV(4,4), ISW(2)
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, IFIL, ISW, IERR)
CALL COMARG(1, IFIL, ISW, IERR)
CALL FOPEN(2, IFIL, "B")
READ BINARY(2) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
WRITE(12, 100) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
FOREVER
: READ BINARY(2, END=120) NS1, NS2, NS3, NS4
: READ BINARY(2) NP, SM, COV
: WRITE(12, 101) NS1, NS2, NS3, NS4
: WRITE(12) NP, SM, COV
:..FIN
100 FORMAT(1X, 6I2)
101 FORMAT(1X, 4A2)
120 STOP
END
SUBROUTINE MATIN(A, N, B, /M, KEY, DETERM)
DIMENSION A(N, N), B(N), IPIVOT(20), INDEX(20, 2)
DOUBLE PRECISION PIVOT(20), T, SWAP, DETERM, AMAX, ZERO, A, B

C *****************************************************************
C ** MATIN (SUBROUTINE) **
C ** MATRIX INVERSION PROGRAM **
C *****************************************************************
C ** CREATED AT NASA/JSC (ASTEP) **
C ***************************************************************** 

SUBROUTINE MATIN(A, N, B, /M, KEY, DETERM)
DIMENSION A(N, N), B(N), IPIVOT(20), INDEX(20, 2)
DOUBLE PRECISION PIVOT(20), T, SWAP, DETERM, AMAX, ZERO, A, B

**inizializzazione**
5 KEY = N
10 DETERM = 1.D+0
15 DO 20 J = 1, N
20 IPIVOT(J) = 0
30 DO 550 I = 1, N

**Cerca l'elemento di maestra**
40 AMAX = 0.D+0
45 DO 105 J = 1, N
50 IF (IPIVOT(J) - 1) 60, 105, 60
60 DO 100 K = 1, N
70 IF (IPIVOT(K) - 1) 80, 100, 740
80 IF (DABS(AMAX) - DABS(A(J, K))) 85, 100, 100
85 IROW = J
90 ICOLUMN = K
95 AMAX = A(J, K)
100 CONTINUE
105 CONTINUE

ZERO = 1.D-16
107 IF(DABS(AMAX) - ZERO) 745, 745, 110
110 IPIVOT(ICOLUMN) = IPIVOT(ICOLUMN) + 1

**Interchange rows to put pivot element on diagonal**
130 IF(IROW - ICOLUMN) 140, 260, 140
140 DETERM = -DETERM
150 DO 200 L = 1, N
160 SWAP = A(IROW, L)
170 A(IROW, L) = A(ICOLUMN, L)
180 A(ICOLUMN, L) = SWAP
200 CONTINUE
205 IF(M) 260, 260, 210
210 SWAP = B(IROW)
230 B(IROW) = B(ICOLUMN)
250 B(ICOLUMN) = SWAP
260 INDEX(1, 1) = IROW
270 INDEX(1, 2) = ICOLUMN
310 PIVOT(1) = A(ICOLUMN, ICOLUMN)
DIVIDE PIVOT ROW BY PIVOT ELEMENT

330 A(ICOLUMN,ICOLUMN) = .1D+1
340 DO 350 L=1,N
350 A(ICOLUMN,L) = A(ICOLUMN,L) / PIVOT(I)
355 IF(M) 380,380,360
360 B(ICOLUMN) = B(ICOLUMN) / PIVOT(I)

REDUCE NON-PIVOT ROWS

380 DO 550 L1=1,N
390 IF(L1-ICOLUMN) 400,550,400
400 T=A(L1,ICOLUMN)
420 A(L1,ICOLUMN) = .0D+0
430 DO 450 L=1,N
450 A(L1,L) = A(L1,L) - A(ICOLUMN,L) * T
455 IF(M) 550,550,460
460 B(L1) = B(L1) - B(ICOLUMN) * T
550 CONTINUE

INTERCHANGE COLUMNS

600 DO 710 I=1,N
610 L=N+1-I
620 IF (INDEX(L,1) - INDEX(L,2)) 630,710,630
630 JROW = INDEX(L,1)
640 JCOLUMN = INDEX(L,2)
650 DO 705 K=1,N
660 SWAP = A(K,JROW)
670 A(K,JROW) = A(K,JCOLUMN)
700 A(K,JCOLUMN) = SWAP
705 CONTINUE
710 CONTINUE
DO 800 I=1,N
J=N+1-I
800 DETERM = DETERM * PIVOT(J)
740 RETURN
745 DETERM = 0.D+0
746 KEY = I - 1
750 RETURN
END
SUBROUTINE MATPRT(MAT, NR, RD, NC, NAM)
INTEGER RD, P, Q
REAL MAT(RD, NC)
1000 FORMAT(1H, 12, 6F10.3)
1010 FORMAT(1H, 6I10)
KOUT=12
WRITE(KOUT,1020)NAM, NR, NC
1020 FORMAT(1H, 15X, A2, 16, 3H BY, 13)
DO 100 I=1, NR
DO 100 J=1, NC
IF(MAT(I, J))150, 100, 150
100 CONTINUE
WRITE(KOUT,1030)
1030 FORMAT(1H, 15X, 12H ALL ZEROES.)
GO TO 30
150 CONTINUE
P=0
Q=-5
10 P=P+6
Q=Q+6
IF(NC.LT.P)P=NC
WRITE(KOUT,1010)(J, J=Q, P)
WRITE(KOUT,1000)
DO 20 I=1, NR
WRITE(KOUT,1000)I, (MAT(I, J), J=Q, P)
20 CONTINUE
IF(NC.GT.P)GO TO 10
30 CONTINUE
WRITE(KOUT,1000)
RETURN
END
MESS
UNPACKS AND DISPLAYS MULTISPECTRAL 12 CHANNEL
DAEDALEUS SCANNER DATA AND DISPOSES REFORMATED DATA
TO TAPE
SEQUENCE:  MESS MTU:F MTU:F

CREATED AT GEORGIA TECH EES
PROGRAMMER: MICHAEL D. FURMAN

INTEGER INPUT(700), IOUT(850), ITAPE1(10), ITAPE2(10), IMAG2(1700)
INTEGER IMAG3(1100), ISW(2)
CALL OPEN(1,"COM.CM",1,IERR)
CALL COMARG(1, ITAPE1, ISW, IERR)
CALL COMARG(1, ITAPE2, ISW, IERR)
CALL MTOPD(2, ITAPE1, 0, IE)
CALL MTOPD(3, ITAPE2, 0, IE)
ACCEPT "START WITH WHAT ELEMENT ", IEL FOREVER
CALL MTDIO(2, 0, INPUT, IS, IE, IC)
L=1
DO (J1=1,565,3)
M0=IAND(I$HFT(INPUT(J1),2),177400K)
M1=IOR(M0, IAND(I$HFT(INPUT(J1),6),300K))
IOUT(L)=I0R(M1, I$HFT(INPUT(J1+1),-10))
M2=IOR(I$HFT(INPUT(J1+1),10), IAND(I$HFT(INPUT(J1+2),-6),1400K)
IOUT(L+1)=IOR(M2, IAND(I$HFT(INPUT(J1+2),-2),377K))
L=L+2
.FIN
CALL UPAC8(IOUT, IMAG2, 850)
DO (I=1,40)
IF ((IMAG2(I).EQ.255).AND.(IMAG2(I+1).LE.10)) IPAS1=I+4
.FIN
IPAS2=0
DO (I=IPAS1,512)
IF ((IMAG2(I).EQ.255).AND.(IMAG2(I+1).LE.10)) IPAS2=I+4
.FIN
L30=1
WHEN (IPAS2.GT.0)
DO (I=IPAS2,756)
IMAG3(L30)=IMAG2(I)
L30=L30+1
.FIN
ELSE
L31=1
DO (I=IPAS1,756)
IMAG3(L31)=IMAG2(I)
L31=L31+1
.FIN
.FIN
CALL PAC8(IMAG3(IEL),IOUT,512)
CALL RIMWRITE(0,0,IOUT,256)
CALL MTDIO(3,50000K+850,IMAG3, IS, IE, IC)
.FIN
STOP
END
C************************************************************************
C    *  
C    MODIFY (SUBROUTINE)    
C    *  
C    PURPOSE    
C    Computes weighted average of two mean vectors    
C    *  
C    DESCRIPTION OF PARAMETERS    
C    *  
C    INPUT    
C    CALLING SEQUENCE    
C    V1 - 1ST VECTOR MEAN    
C    V2 - 2ND VECTOR MEAN    
C    N1 - NUMBER OF VECTORS USED TO COMPUTE V1    
C    N2 - NUMBER OF VECTORS USED TO COMPUTE V2    
C    ND - DIMENSION OF V1 AND V2    
C    *  
C    OUTPUT    
C    CALLING SEQUENCE    
C    V1 - WEIGHTED AVERAGE OF INPUT V1 AND V2    
C    N1 - NUMBER OF VECTORS USED TO COMPUTE OUTPUT V1    
C    *  
C************************************************************************
C    *  
C    CREATED AT NASA/JSC (ASTEP)    
C    *  
C    SUBROUTINE MODIFY ( V1, V2, N1, N2, ND )    
DIMENSION V1(ND), V2(ND)    
REAL N1    
XN1 = (N1)    
XN2 = FLOAT(N2)    
XN1N2 = 1.0/(N1+FLOAT(N2))    
DO 10 I=1,ND    
10 V1(I) = ( XN1*V1(I) + XN2*V2(I) )*XN1N2    
N1 = N1 + FLOAT(N2)    
RETURN    
END
**PACK (SUBROUTINE)**

**PURPOSE**
Packs a storage array to eliminate a vacated slot, moves all vectors with index greater than index of vacated slot down one position in the array.

**DESCRIPTION OF PARAMETERS**

**Input Calling Sequence**
- V - data array
- ND - dimension of each vector in V
- NV - number of vectors in V
- IND - index in V of vacated slot

**Output Calling Sequence**
- V - packed array

**CREATED AT NASA/JSC (ASTEP)**

```fortran
SUBROUTINE PACK(V, ND, NV, IND)
DIMENSION V(ND,NV)
IF(IND.EQ.NV) RETURN
I1 = IND+1
DO 20 I = I1, NV
   I2 = I - 1
   DO 10 J = 1, ND
   10 V(J, I2) = V(J, I)
20 CONTINUE
RETURN
END
```
SUBROUTINE POLY2(JY, ISW, IX, IY, N, LA)

VERTEX -(IX(I), IY(I))=(ELEMENT, SCAN) FOR ITH VERTEX. DIMENSION IX(101), IY(101), D(100), F(101), LA(133), S(101)

IF(ISW.EQ.1)GO TO 1

SLOPE STORE BLOCK X/Y

DO 20 I=1, N

IF(IY(I).EQ.IY(I+1))GO TO 20

D(I)=FLOAT(IX(I+1)-IX(I))/FLOAT(IY(I+1)-IY(I))

CONTINUE

Y=FLOAT(JY)

N=0

NOV=0

DO 40 I=1, N

IF(IY(I).LE.JY.LT.IY(I+1))SCAN=JY

IF(IY(I).EQ.JY)GO TO 30

IF(IY(I).LT.JY)GO TO 22

IF(IY(I+1).LT.JY)GO TO 28

GO TO 40

22

IF(IY(I+1).GT.JY)GO TO 28

GO TO 40

M=M+1

F(M)=D(I)*(Y-FLOAT(IY(I)))+FLOAT(IX(I))

GO TO 40

28

M=M+1

NOV=NOV+1

LA(NOV)=IX(I)

40

CONTINUE

K=N

BOUNDARY SORT - INCREASING ORDER.

DO 50 J=1, K

S(J)=F(1)

IND=1

DO 48 L=1, M

IF(F(L).GE.S(J))GO TO 48

S(J)=F(L)

IND=L

48 CONTINUE

F(IND)=10000

50 CONTINUE

FOR (NOV.LT.2) INTERVAL S(J), S(J+1) MEMBERSHIP CAN BE DETERMINED IF(NOV.EQ.0)GO TO 500

IF(NOV.EQ.1)GO TO 490

DATA SHIFT FOR CONSISTENCE AFTER MEMBERSHIP IS DETERMINED

DO 52 K=1, M

F(K)=S(K)

NOI=N-1

N=0
INTERVAL F(K), F(K+1) MEMBERSHIP SECTION FOR (NOV. GE. 2).

DO 200 K=1,N01
A IS THE POINT CHECKED TO DETERMINE INTERVAL F(K), F(K+1) MEMBERS
A=(F(K)+F(K+1))/2.

L=0
J=0
L1=0
J1=0
DO 122 I=1,N
LOCATION OF INTERVALS SUCH THAT IX(I).LE.A.LT.IX(I+1)

CX=FLOAT(IX(I))
DX=FLOAT(IX(I+1))
IF(CX.EQ.DX)GO TO 122
IF(CX.EQ.A)GO TO 128
IF(CX.LT.A)GO TO 124
IF(DX.LT.A)GO TO 128
GO TO 122

124 IF(A.GE.DX)GO TO 122
128 CY=FLOAT(IY(I))
DX=FLOAT(IY(I+1))
DM=(DY-CY)/(DX-CN)
BOUNDARY POINTS (A,F(A)) FOR GIVEN INTERVAL IX(I).LE.A.LT.IX(I+1):
FX=DM*(A-CX)+CY

IN=1 INDICATES F(A)=JY
IF(FX.EQ.Y)IN=1
IF(FX.LT.Y)GO TO 130
J INDICATES F(A).GT.JY.

J=J+1
IF(FX.NE.CY)GO TO 122
IF(A.NE.CX)GO TO 122

130 IF(FX.NE.CY)GO TO 122
IF(A.NE.CX)GO TO 122

JV INDICATES F(A)=IY(I).GT.JY, FOR SOME I.

JV=JV+1
GO TO 122

L INDICATES F(A).LT.JY.

130 L=L+1
IF(FX.NE.CY)GO TO 122
IF(A.NE.CX)GO TO 122

LV INDICATES F(A)=IY(I).LT.JY. FOR SOME I.

LV=LV+1
122 CONTINUE
INTERVAL $F(K), F(K+1)$ MEMBERSHIP DETERMINATION.

IF (IN .EQ. 1) GO TO 150
IF (JV .EQ. 0) GO TO 140
J = L
IF (LV .EQ. 0) GO TO 140

IF NO DETERMINATION CAN BE MADE, ADD ANOTHER INCREMENT AND TRY AGAIN

A = A + .01
GO TO 54

L = 2 * (J / 2)

IF (J .NE. L) THE INTERVAL $F(K), F(K+1)$ IS IN POLYGON.

IF (J .NE. L) GO TO 150

CHECK FOR VERTEX AT $(F(K), JY)$

DO 146 J = 1, NOV
   CX = FLOAT(LA(J))
   IF (F(K) .NE. CX) GO TO 146

VERTEX $(F(K), JY)$ INCLUDED.

M = M + 1
S(N) = F(K)
M = M + 1
S(N) = F(K)
CONTINUE

IF (K .NE. NOI) GO TO 200
FX = F(K+1)
DO 148 J = 1, NOV
   CX = FLOAT(LA(J))
   IF (FX .NE. CX) GO TO 148
   M = M + 1
   S(M) = FX
   M = M + 1
   S(M) = FX
CONTINUE

GO TO 200

INTERVAL $F(K), F(K+1)$ INCLUDED

M = M + 1
S(M) = F(K)
M = M + 1
S(M) = F(K+1)
CONTINUE

GO TO 500
ALL INTERVALS FORTHCOMING ARE BOUNDARY TO BOUNDARY ON JY

CHECK FOR INTERVAL THAT CONTAINS ONLY A VERTEX

\[ L = 2 \times (\frac{M}{2}) \]

IF \( L \), EQ. M) GO TO 500

VAV = FLOAT(LA(1))

M = M + 1

K = K - 1

IF(S(K), EQ. VAV) GOTO 494

S(K+1) = S(K)

GO TO 492

STOP INSERT FOR VERTEX LOOP

494 S(K+1) = VAV

IF(M, EQ. 0) NO POINTS ON SCAN JY ARE INCUBED

IF(M, EQ. 0) GO TO 555

I = 0

LOOP START AND STOP ROUND OFF TO MIN AND MAX INTEGER VALUES INCL

501 I = I + 1

LA(I) = S(I) + 0.999999

I = I + 1

IF(LA(I), LT. M) GO TO 501

I = -1

INTERVAL(START.GT.STOP) COMPRESS

504 I = I + 2

IF(LA(I), LE. LA(I+1)) GO TO 510

M = M - 2

IF(M, EQ. 0) GO TO 555

IF(I, GT. M) GO TO 511

DO 506 K = I, M

506 LA(K) = LA(K+2)

GO TO 513

LOOPSTOP ON INTERVAL(J).EQ. LOOP START ON INTERVAL(J+1) COMPRESS

510 I = 0

DO 516 K = I, M

516 LA(K) = LA(K+2)

GO TO 513

LA(1) = NUMBER OF LOOPS RETURNED

IF(M, EQ. 0) GO TO 560

DO 557 I = 1, M

J = M - I + 1

LA(J+1) = LA(J)

560 LA(1) = M/2

RETURN

END
C************************************************************************
C
C POLYCL
C
C PERFORMS CLUSTERING OPERATION
C
C SEQUENCE: POLYCL INPUT OUTPUT
C
C************************************************************************
C
CREATED AT NASA/JSC (ASTEP)
C
************************************************************************

COMMON/UNTNUM/IMGUNT,DATUNT
COMMON/BLANK/KNUM
INTEGER CAR, BUF, ISW(2)
REAL NVG
COMMON/COL/JS, JF
COMMON/BOUND/YMIN, XMIN, YMAX, XMAX
INTEGER YHIN, YMAX, XMIN, XMAX
DIMENSION JPX(101), JPY(101), LA(133)
COMMON/NVEC/NVM
COMMON/BUFFER/BUF(3300)
COMMON/A1/1OUT(612)
INTEGER DATUNT
COMMON/INOUT/NOUT, NIN
COMMON/CHARAC/CAR
COMMON/DIST/DIST
INTEGER PLIST
DIMENSION VM(240), AD(20,20)
DIMENSION NVG(20), PLIST(20), CAR(20), IACOP(4)
DIMENSION RN(20), RV(20)
DIMENSION VMP(60), VAR(60)
DIMENSION ACRE(20), II(30), I3(30)

C = 3
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARGC(1, 11, ISW, IERR)
CALL COMARGC(1, 11, ISW, IERR)
CALL COMARGC(1, 12, ISW, IERR)
CALL MTOPD(4, 11, 0, IER)
CALL MTOPD(3, 12, 0, IER)
RP=3
S=1.
R1=20.
R2=20.
NVMAX=15
NPT=100
NET=500
NMT=100
IP=0
IACOP(1)=2HME
IACOP(2)=2HSI
IACOP(3)=2HAN
IACOP(4)=2HQU
ND=4
NRT=1
IMGUNT=3
DATUNT=4
NOSCAL=1
NBUSZ=3260
IQ=NBUSZ/(ND+2)
IBUF2=NBUSZ-IQ+1
IBUF1=IBUF2-IQ
NIN=11
NOUT=10
CAR(1) = 1HA
CAR(2) = 1H.
CAR(3) = 1H:
CAR(4) = 1H-
CAR(5) = 1H,
CAR(6) = 1H/
CAR(7) = 1H+
CAR(8) = 1H0
CAR(9) = 1H*
CAR(10) = 1H#
CAR(11) = 1H$
CAR(12) = 1H%
CAR(13) = 1HB
CAR(14) = 1H@
CAR(15) = 1H*

USER INPUTS

1 CONTINUE
WRITE(NOUT, 1008)
1008 FORMAT(2X, 'INPUT VALUES FOR C, RP, R1, R2 AND NVMAX')
READ(NIN)C, RP, R1, R2, NVMAX
WRITE(NOUT) C, RP, R1, R2, NVMAX
WRITE(NOUT, 102)
102 FORMAT(22H TYPE YES IF INPUTS OK)
READ(NIN, 104) IAN
104 FORMAT(A6)
IF(IAN.NE.3HYES) GO TO 1
IF(NVMAX.GT.20) NVMAX= 20

FIRST PASS INITIALIZATION

CALL INITCL(VM, NVG, NVM, ND)
WRITE(10, 199)
199 FORMAT(2X, 'INPUT 1 FOR POLY, 0 - NORMAL')
READ(11) IPOLY
IF(IPOLY.NE.1) GO TO 201
WRITE(10, 202)
202 FORMAT(2X, 'INPUT # OF CORNERS')
READ(11) NV
WRITE(10, 205) I
DO 204 I=1, NV
205 FORMAT(2X, 'INPUT CORNER # ', 13, ' J, I')
204 READ(11) JPX(I), JPY(I)
JPX(NV+1) = JPX(1)
JPY(NV+1) = JPY(1)
JMIN= 10000
IMIN= 10000
DO 405 K1=1, NV
405 IF(JPX(K1).LT.JMIN) JMIN= JPX(K1)
IF(JPY(K1).LT.IMIN) IMIN= JPY(K1)
405 NRT= IMAX- IMIN+1
DO 406 K2=1, 812
406 IOUT(K2) = 0
IOUT(1) = NRT
IOUT(2) = JMIN
LINIT= IMIN
LIN= IMIN+1
CALL MTDIO(3, 50000K+812, IOUT, IS1, IER)
CALL MTDIO(4, 30000K+LIN, IOUT, IS1, IER)
INRT= 0
NEC= 0
NMC= 0
NPC= 0
IPASS= 1
JPTP= 0
RM(1) = 0.
RV(1) = 0.
DO 8 I=1, ND
8 VMP(I) = 0.
DO 10 I=1, NV
10 PLIST(I) = I

102
FIRST PASS PROCESSING

20 INRT = INRT + 1
IF (INRT.GT.NRT) GO TO 50
IL = LINIT + INRT - 1
ISW = IL + 1
CALL POLY2 (IL, ISW, JPX, JPY, NV, LA)
JS = LA (2)
JF = LA (3)
IF (JS.EQ.0) JS = 1
WRITE (NOUT) JS, JF, INRT, "1"
CALL UNPAC1
NX = JF - JS + 1
CALL CLUSTA (BUF (1), VM, ND, NX, NV, NVMAX, NVG, C, S, RP, R1, NPC, *NPT, PLIST, NEC, NET, NMIN, NMC, NNT, IPASS, BUF (IBUF1), BUF (IBUF2), *
*IP, JPTP, RM, RV, VMP, VAR)
JPTP = JPTP + NX
GO TO 20

SPECIAL ELIMINATION AND MERGER TESTS

50 NX = 0
JPTP = 0
NEC = NET
NMC = NNT
CALL CLUSTA (BUF (1), VM, ND, NX, NV, NVMAX, NVG, C, S, RP, R1, NPC, NPT, *
PLIST, NEC, NET, NMIN, NMC, NNT, IPASS, BUF (IBUF1), *
*Buf (IBUF2), IP, JPTP, RM, RV, VMP, VAR)

SECOND PASS Initialization

INRT = 0
NBACK = LIN + NRT
CALL MTD10 (4, 40000K + NBACK, BUF, IS, IE)
WRITE (10, 222) NBACK

222 FORMAT (2X, " BACKSPACED ", I10, " LINES")
NEC = 0
IF (LIN.LE.0) GO TO 444
CALL MTD10 (4, 30000K + LIN, BUF, IS, IE)
CONTINUE

SECOND PASS PROCESSING

DO 60 I = 1, NVM
PLIST (I) = 1
60 NVG (I) = 0

62 INRT = INRT + 1
IF (INRT.GT.NRT) GO TO 70
IL = LINIT + INRT - 1
ISW = IL + 1
CALL POLY2 (IL, ISW, JPX, JPY, NV, LA)
JS = LA (2)
JF = LA (3)
JQ = JF - JS + 1
DO 402 LL = 1, 812
402 IOUT (LL) = 0
IOUT (1) = JS
IOUT (2) = JQ
WRITE (NOUT) JS, JF, INRT, "2"
IF (JS.EQ.0) JS = 1
CALL UNPAC1
NX = JF - JS + 1
CALL CLUSTA (BUF (1), VM, ND, NX, NV, NVMAX, NVG, C, S, RP, R2, NPC, NPT, *
*PLIST, NEC, NET, NMIN, NMC, NNT, IPASS, BUF (IBUF1), *
*Buf (IBUF2), IP, JPTP, RM, RV, VMP, VAR)
JPTP = JPTP + NX
DO 403 LL = 1, 810
403 L2 = IBUF2 + LL - 1
IF (L2.GT.NBUFSZ) GO TO 404
IOUT (LL + 2) = BUF (L2)
CONTINUE
CALL MTD10 (3, 50000K + 812, IOUT, IS1, IER)
GO TO 62
PRINT RESULT SUMMARY

70 WRITE(NOUT, 106)
106 FORMAT(' CLUSTER SYMBOL SIZE R MEAN R SIGMA ACRES ') NSAVE=0.0
DO 72 I=1, NVM
RV(I)=SQRT(RV(I))
CC=1.0541515
IF(NOSCAL.EQ.0) CC=1.53046
ACRE(I)=NVG(I)*CC
IF(NVG(I).LT.NSAVE) GO TO 2000
NSAVE=NVG(I)
KNUM=I
2000 CONTINUE
72 WRITE(NOUT, 108) I, CAR(I), NVG(I), RM(I), RV(I), ACRE(I)
108 FORMAT(' ', 6X, A1, F10.0, 2X, 2F7.2, F10.2)
I=ND*NVM
DO 74 J=1, I
74 VM(J)=VMP(J)
CALL MTDIO(3, 60000K, IOUT, IS1, IER)
CALL MTDIO(3, 60000K, IOUT, IS1, IER)
C USER OPTION SELECTION
C
76 WRITE(NOUT, 110)
110 FORMAT(' CHOOSE OPTION FROM ') WRITE(NOUT, 112) IACOP
112 FORMAT(' ', 4A8)
READ(NIN, 104) IAN
IF(IAN.EQ.5HBEANS) GO TO 80
IF(IAN.EQ.6HSIGMAS) GO TO 85
IF(IAN.EQ.6HANGDIS) GO TO 90
IF(IAN.EQ.4HQUIT) STOP
WRITE(NOUT, 114) IAN
114 FORMAT(' ', A6, 22H IS NOT A VALID CHOICE) GO TO 76
C MEANS DISPLAY
C
80 CALL MATPRT(VM, ND, ND, NVM, 5HMEANS) GO TO 76
C SIGMAS DISPLAY
C
85 I=ND*NVM
DO 86 J=1, I
86 VAR(J)=SQRT(VAR(J))
CALL MATPRT(VAR, ND, ND, NVM, 6HSIGMAS) GO TO 76
C ANGDIS DISPLAY
C
90 IDISF=2
CALL ANGDIS(VM, NVM, ND, IDISF, AD)
CALL MATPRT(AD, NVM, NVM, NVM, 6HANGDIS) GO TO 76
END
C************************************************************************
*                    PUTIN                                              *
* INPUT SIGNATURES                                            *
* CREATED AT GEORGIA TECH EES                              *
* PROGRAMMER: NICKOLAS L. FAUST                              *
************************************************************************

DIMENSION COV(4,4), XM(4)
CALL OPEN(3, "INSG", 0, IE)
CALL FOPEN(4, "HSIG", "B")
READ(3) NSIG
ID1=01
ID2=22
ID3=77
IT1=99
IT2=99
IT3=99
WRITE BINARY(4) IT1, IT2, IT3, ID1, ID2, ID3
DO (L=1, NSIG)
   WRITE(10, 101)
   READ(3, 100) NA1, NA2, NA3, NA4
   WRITE(10, 102)
   READ(3) NP
   WRITE(10, 103)
   READ(3) XM
   DO (K=1, 4)
      WRITE(10, 104) K
      READ(3) (COV(K, J), J=1, 4)
   FIN
   WRITE BINARY(4) NA1, NA2, NA3, NA4
   WRITE BINARY(4) NP, XM, COV
FIN
100 FORMAT(4A2)
101 FORMAT(2X, "INPUT NA1, NA2, NA3, NA4")
102 FORMAT(2X, "INPUT NUMBER OF POINTS")
103 FORMAT(2X, "INPUT (XM(I), I=1, 4)")
104 FORMAT(2X, "INPUT ROW # ", I1, " OF COV - (COV(K, J), J=1, 4) ")
STOP
END
INTEGER IORAN(94)
CALL FOPEN (5, "RYGBV.")
ACCEPT "TYPE 1 TO SAVE PRESENT PSEUDOCOLOR ", IANS1
IF (IANS1.EQ.1)
  REWIND 5
  DO (J=1,1000) READ BINARY (5, END=10) IORAN
10 : DO (I=1,94) IORAN(I)=0
   CALL RCM (0, IORAN)
   TYPE "TYPE A 30 CHARACTER DISCREPTION"
   TYPE "................................."
   READ (11,20) (IORAN(I), I=65,94)
20 : FORMAT (30A1) WRITE BINARY (5) IORAN
   TYPE "PSEUDOCOLOR MEMORY NUMBER ", J
   ..FIN
   ACCEPT "TYPE 1 TO REPLACE PRESENT PSEUDOCOLOR ", IANS2
   IF (IANS2.EQ.1)
     ACCEPT "TYPE 1 IF YOU KNOW THE PSEUDOCOLOR MEMORY NUMBER ", IANS3
     IF (IANS3.NE.1)
       PAUSE PUT UP PSEUDOCOLOR TEST PATTERN HIT RETURN
30 : REWIND 5
   : DO (J=0,1000)
   : : READ BINARY (5, END=30) IORAN
   : : WRITE (10,40) (IORAN(I), I=65,94)
40 : FORMAT (1X,30A1)
   : : CALL WCM (0, IORAN)
   : : ACCEPT "TYPE 1 FOR NEXT PATTERN ", IANS4
   : : IF (IANS4.NE.1) STOP NORMAL EXIT
   : ..FIN
   IF (IANS3.EQ.1)
     REWIND 5
     : ACCEPT "PSEUDOCOLOR MEMORY NUMBER ", INUM
     : DO (J=1,INUM READ BINARY (5) IORAN
     : WRITE (10,60) (IORAN(I), I=65,94)
60 : FORMAT (1X,30A1)
     : CALL WCM (0, IORAN)
     ..FIN
   ..FIN
STOP NORMAL EXIT
END
THIS PROGRAM DIVIDES ONE ERTS CHANNEL INTO THE OTHER
THREE TO LEAVE THREE CHANNELS OF DATA. IT ALSO WILL TAKE
TWO CHANNELS AND BY DIVIDING ONE INTO THE OTHER TWO, CREATE
A THIRD CHANNEL, ALL TO BE DISPLAYED.

SEQUENCE: RAT10 MTU:F

CREATED AT GEORGIA TECH EES
PROGRAMMER: MICHAEL D. FURMAN

INTEGER INPUT(1700), IMP(3400), IMAG(0:255), IMAG2(0:255), ITAPE(10)
INTEGER ISW(2)
CALL OPEN(1,"COM.C11",1,IERR)
CALL COMARG(1,ITAPE,ISW,IERR)
CALL COMARG(1,ITAPE,ISW,IERR)
CALL NTOPD(2,ITAPE,0,IE)
ACCEPT "TWO (2) OR FOUR (4) CHANNEL RATIO ",IANS1
WHEN (IANS1.EQ.2) ACCEPT "INPUT NUM. AND DENOM. CHANNELS ",INUM,IDEN
ELSE ACCEPT "INPUT DENOMINATOR CHANNEL ",IDEN
ACCEPT "SCALE FACTOR ",ISCAL
ACCEPT "INPUT LINE AND ELEMENT ",LINE,N
IF ((N/2)*2).EQ.N) N=N-1
ISKP=1+LINE
IEL=((N-4)-(6/(1+N-((N/2)*2)))-1
IEND=IE+19
IDEN2=IDEN*2-1
DO (I=1,ISKP) CALL MTDIO(2,0,INPUT,IS,IE,IC)
DO (K2=1,255)
: CALL MTDIO(2,0,INPUT,IS,IE,IC)
: CALL UPACB(INPUT,IMP,1700)
: M2=0
: M1=0
: DO (K=1,7,2)
: L=0
: IF (K.NE.IDEN2)
: : DO (J1=1EL,IEND,8)
: : : IMAG(L)=(IMP(J1+K)/(IMP(IDEN2+J1)+1.0))*ISCAL
: : : IMAG(L+1)=(IMP(J1+K+1)/(IMP(IDEN2+J1+1)+1.0))*ISCAL
: : : L=L+2
: : : FIN
: : DO (I=0,255)
: : : IF (IMAG(I).GT.255)
: : : : IMAG(I)=255
: : : FIN
: : FIN
: CALL PACB(IMAG,IMAG2,256)
: CALL RIMWRITE(M1,M2,IMAG2,256)
: WHEN (M2.EQ.0) M2=255
: ELSE M2=0
: M1=M1+1
: FIN
FIN
STOP
END
SUBROUTINE RDATA( I TOP, I BOT, NV, IX, IY, IXD, NP)
D I M E N S I O N LA(133), IX(101), IY(101), JBUF(100,4), IXD(512,4)
C O M M O N / BUFFER / IBUF(1700)
C O M M O N / I N D E V / IPOS
IC=0
ISUM1=ITOP-IPOS-1
222 FORMAT(2X, "SKIPPED ", I5, " RECORDS")
IF(ISUM1.GT.0)CALL MTD10(2,30000K+ISUM1,IBUF,IS,IER)
IF(ISUM1.LT.0)CALL MTD10(2,40000K-ISUM1,IBUF,IS,IER)
IDEL=IBOT-ITOP+1
DO ( IL=1, IDEL)
  CALL MTD10(2,0,IBUF,IS,IE)
  ISW=ITOP+1
  ILIN=ITOP+IL-1
  CALL POLY2(ILIN,ISW,IX,IY,NV,LA)
  NN=100
  NSEG=1
  MM=1650
  DO ( K=1,NSEG)
    ISUB1=2*K
    JS=LA(ISUB1)
    JF=LA(ISUB1+1)
    JD=JF-JS+1
    CALL UNPAC3(IBUF,JBUF,JS,JF,NN,MM)
    DO ( K=1,JD)
      IC=IC+1
      DO ( KD=1,4)
        IXD(IC,KD)=JBUF(K,KD)
      END
    DO
  END
NP=1G
IPOS=IBOT
WRITE(10,222) IDEL
TYPE " TOTAL # PTS = ", NP
RETURN
END
**RDIMG (SUBROUTINE)**

**READ AND DISPLAY IMAGE TO SCREEN**

**CREATED AT GEORGIA TECH EES**

**PROGRAMMER: ROBERT A. MADDOX**

**SUBROUTINE RDIMG(IL, IEL, NCH, IBLUP)**

**INTEGER IMAG(260), IMAG2(0:512), NCH(0:2)**

**COMMON BUFFER/INPUT(1700),/INDEV/IP0S**

ISUM1=IL-IP0S-1
IELP=((IEL+254)/2)*4
IEL2=((IEL-1)/2)*4
IF (IBLUP.LE.0)
   TYPE "ILLEGAL FACTOR", IBLUP
RETURN
FIN
IF (ISUM1.GT.0) CALL MTDIO(2,30000K+ISUM1,INPUT,IS,IE,IC)
IF (ISUM1.LT.0) CALL MTDIO(2,40000K-ISUM1,INPUT,IS,IE,IC)
BLOW-UP-IMAGE: DO (I=1,IBLUP)
   ICC=ICC+1
   IF (ICC.GT.768) GO TO 100
   CALL RIMWR(I5,N5,IMAG,256)
   F.IN
   CONTINUE
FIN
100 CONTINUE
IP0S=IL+M7
RETURN
TO BLOW-UP-IMAGE
CALL UPACB(IMAG,IMAG2,256)
DO (K7=0,255)
   K9=256-K7
   KDB=K9/IBLUP
   IMAG2(K9)=IMAG2(KDB)
   F.IN
   CALL PACB( IMAG2, IMAG, 512)
   F.IN
END
SUBROUTINE RETIMG(IFLAG,NUNIT,IMG,THR,NP)
DIMENSION IMG(112)
COMMON/COJJS,JF
INTEGER THR
IF(IFLAG.NE.1) GO TO 10
REWIND NUNIT
10 READ BINARY(NUNIT)JS,NP,(IMG(I),I=1,NP)
RETURN
END
SAVIMG (SUBROUTINE)

PURPOSE
SAVES, ON TAPE, IMAGE AND THRESHOLD ARRAYS CORRESPONDING TO A DATA RECORD

DESCRIPTION OF PARAMETERS

INPUT
CALLING SEQUENCE
IFLAG - FIRST ENTRY OR REWIND AND WRITE HEADING RECORD
FLAG
NUNIT - UNIT NUMBER
IMG - IMAGE ARRAY TO BE SAVED
THR - THRESHOLD ARRAY CORRESPONDING TO IMG
NP - NUMBER OF POINTS IN EACH ARRAY

CREATED AT NASA/JSC (ASTEP)

SUBROUTINE SAVIMG(IFLAG,NUNIT,IMG,THR,NP)
DIMENSION IMG(NP), THR(NP)
INTEGER THR
DIMENSION BLK(166)
COMMON/COL/JS,JF
BLK(1)=1.
IF(IFLAG.NE.1) GO TO 10
REWIND NUNIT
WRITE BINARY(NUNIT)JS,NP,(IMG(K),K=1,NP)
WRITE(12)NP, IMG(1)
RETURN
END
SCORECARD

THIS ROUTINE CLASSIFIES TRAINING FIELDS WITH A MAXLIK CLASSIFIER

SEQUENCE: SCORECARD INPUT TAPNAM SIGFIL VERTEX

CREATED AT GEORGIA TECH EES

PROGRAMMER: NICHOLAS L. FAUST

FIELD AND OPEN STATEMENTS

CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, I1, ISW, IERR)
CALL COMARG(1, I2, ISW, IERR)
CALL COMARG(1, I3, ISW, IERR)
CALL COMARG(1, I4, ISW, IERR)

CALL OPEN(2, I1, 0, IE)
CALL FOPEN(7, "MAXFLD", "B")

OPEN TAPE FILE

CALL MTOPD(3, I2, 0, IE)
ND=4
NUN1=2
NUN2=4
READ(2,200) NSIG
WRITE(1F) NSIG

GET SIGNATURES FOR CLASSIFICATION

CALL GSIG(AMEAN, BCOV, NSIG, NUN1, NUN2, I3, IORDER, NAM1, NAM2, NAM3, NAM4)
CALL ICD(AMEAN, BCOV, ND, NSIG, DET, B, B)

ISIG=0
DO (KSIG=1, NSIG)
  :  ISIG= ISIG+1
  :  CALL FOPEN(5, 14, "B")
  :  READ BINARY(5) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
  :  K3= IORDER(KSIG)
  :  DO (K2=1, K3)
    :   READ BINARY(5) NA1, NA2, NA3, NA4
    :   READ BINARY(5) K, IYMIN, IYMAX
    :   DO (I=1, K)
      :     READ BINARY(5) K1, IVX(I), IVY(I)
      :     .FIN
      :     .FIN
      :     CALL FCLOSE(5)
      :     WRITE(IP) ITAP1, ITAP2, ITAP3
      :     WRITE(IP) IDAT1, IDAT2, IDAT3
      :     WRITE(IP, 103) NA1, NA2, NA3, NA4
      :     WRITE(IP) K, IYMIN, IYMAX
      :     DO (L3=1, K)
        :       WRITE(IP) K1, IVX(L3), IVY(L3)
        :       .FIN
        :     .FIN
        :     NV= K
        :     ISKIP= IYMIN+1
        :     IVX(K+1)= IVX(1)
        :     IVY(K+1)= IVY(1)
        :     IDEL= IYMAX- IYMIN+1
        :     DO (I1=1, 20)
          :       ICOUNT(I1)= 0
          :     .FIN
          :     JTOT= 0

        /*SKIP RECORDS
          CALL MTD10(3, 30000+ ISKIP, IBUF, IS, IEE)
          WRITE BINARY(7) IDEL, ISIG
          DO (J=1, IDEL)
            :   CALL MTD10(3, 0, IBUF, IS, IE, NW)
            :     ISW= 2
            :     CALL POLY2(ISKIP+J-1, ISW, IVX, IVY, NV, LA)
            :     JS= LA(2)
            :     JF= LA(3)
            :     CALL UNPAC3(IBUF, JBUF, JS, JF, 512, 1650)
            :     JQ= JF- JS+1
            :     WRITE(IP) J, JS, JF, JQ, IDEL, ISKIP
            :     DO (L=1, JQ)
              :       DO (L1=1, ND)
              :         JSUB=(L-1)*ND+L1
              :         IBUF(JSUB)= JBUF(L, L1)
              :       .FIN
              :     .FIN
          CALL MAXL1(IBUF, ND, JQ, AMean, BCOV, NSIG, DET, IEUM, NVG)

          JMOVE= JS-1
          WRITE BINARY(7) JMOVE, JQ, IDUM
          JTOT= JTOT+ JQ
          DO (K=1, JQ)
            :   LSUB= IDUM(K)
            :   ICOUNT(LSUB)= ICOUNT(LSUB)+1
            :     .FIN
            :   FIN
            :   DO (K=1, NSIG)
            :     JPER= (ICOUNT(K)*100)/JTOT
            :     WRITE(12, 102) K, ICOUNT(K), JPER, NAM1(K), NAM2(K), NAM3(K), NAM4(K)
            :     .FIN
            :   ISUM= IDEL+ ISKIP
            :   CALL MTD10(3, 40000+ ISUM, IBUF, IS, IER)
          .FIN
          102 FORMAT(2X, "CLASS ", 14, " NUMBER ", 16, " PERCENT ", 13, 2X, 4A2)
          163 FORMAT(2X,4A2)
          200 FORMAT(12)
          STOP
        END
SUBROUTINE SEQCOV(X, N, V, M, I, D)

INTEGER X
DIMENSION X(N)
DIMENSION V(N, N)
REAL M(N)
DIMENSION D(N, 2)

WHEN (I .LE. 0)
    DO (L=1, N)
        V(K, L) = 0.0
        I = 1
        RETURN
    END

ELSE
    FIP = FLOAT(I)
    I = I + 1
    FIP = FLOAT(I)
    FI = 1.0 / FIP
    DO (K=1, N)
        D(K, 1) = X(K) - M(K)
        D(K, 2) = D(K, 1) * FIP
        M(K) = M(K) + D(K, 2)
    END
    FIP = 1.0 - FIP
    DO (K=1, N)
        DO (L=1, N)
            V(K, L) = FIP * V(K, L) + D(K, 2) * D(L, 1)
        END
    END
RETURN
END
SEOST (SUBROUTINE)
PURPOSE
SEQUENTIALLY CALCULATED MEAN VECTOR AND VARIANCES

DESCRIPTION OF PARAMETERS

INPUT
CALLING SEQUENCE
XM - CURRENT MEAN
XV - CURRENT VARIANCES
N - NUMBER OF POINTS USED TO COMPUTE XM AND XV
ND - DIMENSION OF DATA VECTOR
X - NEW DATA VECTOR TO BE ADDED TO XM AND XV

OUTPUT
CALLING SEQUENCE
XM, XV, N - UPDATED VALUES

CREATED AT NASA/JSC (ASTEP)

SUBROUTINE SEOST(XM, XV, N, ND, X)

DIMENSION XM(ND), XV(ND), X(ND)
INTEGER N
IF(N.GT.0) GO TO 20
DO 10 I = 1, ND
XM(I) = X(I)
10 XV(I) = 0.
N = 1
RETURN
20 F1 = (1., -1.)/FLOAT(N)
N = N + 1
F2 = 1./FLOAT(N)
DO 30 I = 1, ND
T = X(I) - XM(I)
XM(I) = XM(I) + F2*T
30 XV(I) = F1*XV(I) + F2*(T**2)
RETURN
END
C**********************************************************************
C
C
C STRIPLESS
C
C PROGRAM TO DESTRIPE A DATA SET GIVEN THE CONVERSION TABLE FROM STRNF2
C
C SEQUENCE: STRIPLESS MTIN:F MTOUT:F TABLE
C
C**********************************************************************
C
C PROGRAMMERS: RONNIE PIERNER (NASA/ERL)
C NICKOLAS FAUST (GEORGIA TECH EES)
C
C**********************************************************************
DIMENSION IEM(4000), LT(4, 6, 128), IZ(2, 4, 412)
DIMENSION ITAP(15), JTAP(15), IFIL(15), ISW(2)
CALL OPEN( 1, "COM.CM", 1, IERR)
CALL COMARG( 1, ITAP, ISW, IERIL)
CALL COMARG( 1, ITAP, ISW, IEMI)
CALL COMARG( 1, JTAP, ISW, IER0)
CALL COMARG( 1, IFIL, ISW, IERO)
CALL FOPEN(3, IFIL, "E")
CALL MTOPD(2, ITAP, 0, IERR)
CALL MTOPD(4, JTAP, 0, IERR)
NOFT1=1
READ_BINARY(3) LT
DO (K=1,2)
  CALL MTDIO(2,0,IEM,IS,ILRR,NW)
  CALL MTDIO(4,50000K+NW, IEM, IS, IERR)
..FIN
N=0
NL=0
TYPE "HOW MANY LINES OF 6 TO SKIP?"
READ(11)KICK
TYPE "HOW MANY LINES TO PROCESS?"
READ(11)NLIN
KICK6=6*KICK
IF(KICK.CT.0)
  CALL MTDIO(2,30000K+KICK6, IEM, IS, IERR)
..FIN
FOREVER
  CALL MTDIO(2,0,IEM,IS,ILRR,NW)
  N=N+1
  NL=NL+1
  IF(NL .GE. NLIN)WRITE-LOF
  IF(NW .LE. 50)WRITE-EOF
  CALL PACKER(1, IEM, IZ)
  DO (I=1,4)
    DO (J=1,407)
      K=IZ(1, I, J)+1
      K1=IZ(2, I, J)+1
      IF(K .GT. 128)K=128
      IF(K1 .GT. 128)K1=128
      IZ(2, I, J)=LT(I, N, L1)
      IZ( 1, I, J)=LT( I , N,L) 
..FIN
    CALL PACKER(2, IEM, IZ)
  CALL MTDIO(4,50000K+1650, IEM, IS, IERR)
  IF(N .EQ. 6)N=0
..FIN
TO WRITE-EOF
  CALL MTDIO(4,60000K, IEM, IS, IERR)
  CALL MTDIO(4,60000K, IEM, IS, IERR)
  CALL MTDIO(4,10000K, IEM, IS, IERR)
  CALL MTDIO(2,10000K, IEM, IS, IERR)
STOP
..FIN
END
STRIPNF

TO ELIMINATE STRIPING IN LANDSAT DATA

SEQUENCE: STRNP2 TAPE TABFIL

CREATED AT NASA/ERL

PROGRAMMER: RON PIERSON

DIMENSION IER(620,8), LT(4,6,128), IUNPAK(2,4,3), DT(4,8,129)
DIMENSION IEM(1650), LIMS(4,6,2)
DIMENSION ITAP1(15), ITAB(15)
COMMON/DUM/IER, DT, LIMS, IUNPAK, ITAP1

EQUIVALENCE (IEM(1), IER(1,6))
EQUIVALENCE (LT(1,1,1), IER(1,1))
CALL FIELD(2, ITAP1, $701)
CALL FIELD(3, ITAB, $701)
CALL MTDPO(1, ITAP1, 0, IERR)
CALL FOPEN(3, ITAB, "B")

N0FTR=0
NL1=0
NOFT1=1
NOFT=1

WRITE (10, 2222)
2222 FORMAT (2X, "INPUT NPRNT NLINES")
READ (11) NPRNT, NLINES

CALL MTDIO(1, 0, IEM, IS, IERR)
CALL MTDIO(1, 0, IEM, IS, IERR)

SIX SCAN READ

10 N=1
IERROE=0
11 CALL MTDIO(1, 0, IEM, IS, IERR, NW)
IF (NW, LT, 1600) GO TO 400
IF (NL1, GT, NLINES) GO TO 400
DO (KV=1, 620) IER(KV, N) = IEM(KV)
N=N+1
NL1=NL1+1
IF (N, LT, 7) GO TO 11

SIX BY SIX PROCESS

N=1
NUM=0
DO 299 NE=1, 51
N=N+12
DO 280 J=1, 6
DO (L1=1, 3)
   DO (L2=1, 4)
      N1=N+(L1-1)*4+L2-1
      IWRD=IER(N1, J)
      IUNPAK(1, L2, L1)=ISHFT(IWRD, -8)
      IUNPAK(2, L2, L1)=IAND(IWRD, 377K)
   :   :   FIN
   :   : FIN

ELEMENT AVERAGE

DO 280 I=1, 4
DO 200 K1=1, 3
NUM=NUM+IUNPAK(1, I, K1)+IUNPAK(2, I, K1)
200 CONTINUE
DT(I, J, 1)=FLOAT(NUM) / 6.
NUM=0

280 CONTINUE
DO 284 I=1, 4
C DISCARD OF ABNORMAL DATA VALUES
SM=DT(I, I, 1)
BG=SM
DO 279 J=2,6
RAD=DT(I,J,1)
IF(RAD.GT.SM)GO TO 275
SM=RAD
GO TO 279
275 IF(RAD.LT.BG)GO TO 279
BG=RAD
279 CONTINUE
RAD=BG-SM
IF(RAD.GT.10.)GO TO 284
C DATA AGREGATION
AVG=0.
DO 282 J=1,6
282 AVG=DT(I,J,1)+AVG
MN=AVG/6.+.5
IF(MN.LT.0)GO TO 284
IF(MN.GT.127)GO TO 284
MN=MN+2
D=DT(I,7,MN)+6.
DO 283 J=1,6
DT(I,J,MN)=(DT(I,J,MN)*DT(I,7,MN)+6.*DT(I,J,1))/D
283 CONTINUE
C POINT TOTAL UPDATE
DT(I,7,MN)=D
284 CONTINUE
299 CONTINUE
GO TO 10
400 NOFTR=NOFTR+1
IF(NOFTR.LT.NOFT)GO TO 99
C DETERMINATION OF USEABLE DATA RANGES
DO 1099 I=1,4
J=2
1013 J=J+1
1017 J=J-1
1099 IER(I+4,6)=J
C MEAN ADJUST
DO 410 1=1,4
L1=IER(1,6) K1=IER(I+4,6)
DO 410 NE=L1,K1
SUM=0.
405 SUM=SUM+DT(I,J,NE)
AVG=SUM/6.
A=0.
L=1
K=3
DO 406 J=1,6
B=DT(I,J,NE)-AVG
IF(B.LT.0)B=-B
IF(B.LT.A)GO TO 406
A=B
L=J
406 CONTINUE
SUM=SUM+DT(I,L,NE)
AVG=SUM/5.
IF(L.EQ.K)K=1
A=0.
DO 407 J=1,6
407 IF(J.EQ.K)GO TO 407
B=DT(I,J,NE)-AVG
IF(B.LT.0.)B=-B
IF(B.LT.A)GO TO 407
A=B
K=J
407 CONTINUE
118
SMOOTHING OF SPARSE DATA OVER USEABLE DATA RANGES

DO 1199 1 = 1, 4
   J = IER(1, 6) + 1
   K = IER(1 + 4, 6) - 1
   DO 1199 L = J, K
   IF(DT(I, 7, L) .GT. 299.) GO TO 1199
   N = L
   IF(DT(I, 7, N) .LT. 299.) GO TO 1113
   M = L + 1
   IF(DT(I, 7, M) .LT. 299.) GO TO 1117
   RAD = DT(I, 7, L) / 300.
   SM = 1. - RAD
   IF(DT(I, 7, L) .LT. 10.) DT(I, 8, L) = FLOAT(L - 2)
   A = (DT(I, 8, L) - DT(I, 8, N)) / (DT(I, 8, M) - DT(I, 8, N))
   DO 1121 L1 = 1, 6
      BG = DT(I, L1, M) - DT(I, L1, N)
      DT(I, L1, L) = SM * BG + RAD * DT(I, L1, L)
   1121 CONTINUE
   CONTINUE
   IF(NPRNT.EQ.0) GO TO 500
   DO 499 NE = 2, 63
      MN = NE - 2
      WRITE(12, 401) NN
      DO 404 J = 1, 8
         A = DT(I, J, NE)
         B = DT(2, J, NE)
         C = DT(3, J, NE)
         D = DT(4, J, NE)
         WRITE(12, 402) J, A, B, C, D
      404 CONTINUE
      499 CONTINUE
   CONTINUE

MID-RANGE TABLE BUILDER

DO 498 L = 1, 4
   DO 498 NE = 1, 6
      L = IER(1, 6)
      K = IER(1 + 4, 6)
      SM = DT(I, NE, L)
      K = DT(I, NE, K)
      J = SM
      A = FLOAT(J)
      IF(SM.NE.A) J = J + 1
      LIMS(I, NE, 1) = J
      LIMS(I, NE, 2) = K + 2
      DO 498 M = J, K
      A = FLOAT(M)
      N = M + 1
      IF(DT(I, NE, L) .LT. A) GO TO 435
      L = L - 1
      GO TO 431
   431 L2 = L + 1
   IF(DT(I, NE, L2) .GT. A) GO TO 441
   441 L2 = L + 1
   L = L - 1
   GO TO 431
   435 A2 = DT(I, NE, L2) - DT(I, NE, L)
   B = A - DT(I, NE, L)
   B2 = DT(I, 8, L2) - DT(I, 8, L)
   LT(I, NE, N) = DT(I, 8, L) + B * B2 / A2 + .5
   498 L = L + 1
C BELOW-RANGE TABLE BUILDER
DO 415 I=1,4
DO 415 NE=1,6
K=LIMS(I,NE,1)
L=K
DO 415 J=1,K
M=LT(I,NE,L+1)-1
IF(M.LT.0)M=0
LT(I,NE,L)=M
415 L=L-1
C ABOVE-RANGE TABLE BUILDER
DO 425 I=1,4
DO 425 NE=1,6
K=LIMS(I,NE,2)
DO 425 L=K,128
M=LT(I,NE,L-1)+1
IF(M.GT.127)M=127
425 LT(I,NE,L)=M
C TABLE SMOOTHER
DO 525 I=1,4
DO 525 J=1,6
DO 525 K=2,126
M=K+1
L=LT(I,J,M)-1
IF(L.LE.LT(I,J,K))GO TO 525
N=K-1
IF(LT(I,J,N).LT.LT(I,J,K))GO TO 521
LT(I,J,K)=LT(I,J,K)+1
GO TO 525
521 L=K+2
IF(LT(I,J,L).GT.LT(I,J,M))GO TO 525
LT(I,J,M)=LT(I,J,M)-1
525 CONTINUE
DO 541 I=1,4
DO 541 J=1,6
541 LT(I,J,1)=0
IF(NPRNT.EQ.0)GO TO 550
DO 549 K=1,128
MN=K-1
WRITE(12,601)MN
601 FORMAT(6X,"CHNL 1 CHNL 2 CHNL 3 CHNL 4 
INPUT= ",I3)
DO 549 J=1,6
L=LT(I,J,K)
L2=LT(2,J,K)
M=LT(3,J,K)
N=LT(4,J,K)
WRITE(12,603)J,L,L2,M,N
603 FORMAT(2X," DET ",I1,3X,4(I3,6N))
549 CONTINUE
C DATA CORRECTION
C
550 CONTINUE
WRITE BINARY(3) LT
CALL FCLOSE(3)
CALL MTDIO(1,10000K,IEM,IS,IERR)
666 STOP
701 STOP STRNF2 TAPE TABFIL
END
**SUMCAT**

**THIS ROUTINE SUMS CLASSIFIED CATEGORIES INTO GENERAL CATEGORIES FOR OUTPUT**

**SEQUENCE:** SUMCAT IFIL

**CREATED AT GEORGIA TECH EES**

**PROGRAMMER:** NICKOLAS FAUST

---

```fortran
DIMENSION COUNT(60), IEQ(60,20), NUM(20), ICAR(10,20)
DIMENSION JFIL(34), ISUM(20), PER(60), KFIL(34), ISWS(2)

CALL OPEN(1, "COM.CL", 1, IE)
CALL COMARG(1, JFIL, ISWS, IE)
CALL COMARG(1, JFIL, ISWS, IE)
CALL OPEN(3, JFIL, "B")
CALL COMARG(1, KFIL, ISWS, IE)
CALL FOPEN(2, KFIL, "B")

READ(3) ISUM
DO (L=1, ISUM)
  READ (3,101) (ICAR(J,L),J=1,10)
101 : FORMAT (10A2)
  READ(3) NUM(L)
  NUM2=NUM(L)
  READ(3) (IEQ(J,L),J=1,NUM2)
FIN

READ(3) NWQ
DO (K=1, NWQ)
  STOT=0.
  DO (KC=1, 60) COUNT(KC)=0.
   READ(2,130) NA1
130 : FORMAT(14)
   WRITE(12,131)NA1
131 : FORMAT(2X, " WQMIJ # " , I5 /////)
   DO (KZ=1, 60)
     COUNT(KZ) , PER(KZ)
70 : WRITE(10) COUNT(KZ), PER(KZ)
   FIN
   DO (N=1, ISUM)
     NUM2=NUM(N)
     S2=0
     S1=0
     DO (L=1, NUM2)
       IQ=IEQ(L, N)
       S1=S1+COUNT(IQ)
       S2=S2+PER(IQ)
     FIN
     WRITE(12,102)N, (ICAR(J,N),J=1,10), S1, S2
102 : FORMAT(2X, 12, 2X, 10A2, 2X, F10.1, " ACRES", 2X, F6.3, " PERCENT"
   Fin
   WRITE(12,103)STOT
103 : FORMAT(6X, "TOTAL ", 10X, F12.1, " ACRES")
   WRITE(12,200)
200 : FORMAT( // // /// /)
FIN
STOP
END
```
SYMINV (SUBROUTINE)
INVERTS A SYMMETRIC MATRIX

CREATED AT NASA JSC

SUBROUTINE SYMI NV(A, AI, DET, N)
DIMENSION A(N,N), AI(N,N), QC(12), D(12), DI(12)
INTEGER R
DET=1.0
R=0
10 IF(R.GT.N)GO TO 19
QC(R)=1.0
I=0
20 I= I+1
IF(I.GT.R-1)GO TO 29
QC(I)=0.0
J=0
30 J= J+1
IF(J.GT.R-1)GO TO 20
QC(I)=QC(I)-AI(I,J)*A(J,R)
GO TO 30
29 CONTINUE
D(R)=A(R,R)
IF(D(R).LE.0.0)GO TO 52
K=0
40 K= K+1
IF(K.GT.R-1)GO TO 49
D(R)=D(R)+A(R,K)*QC(K)
GO TO 40
49 CONTINUE
DET=DET*D(R)
IF((D(R)/A(R,R)).LT.1.E-8)GO TO 52
51 D(R)=1.0/D(R)
GO TO 60
52 D(R)=0.0
WRITE(12,1010)R, D(R), DET
1010 FORMAT(9E10.5,6H DET = ,E10.5)
60 I=0
70 CONTINUE
I= I+1
IF(I.GT.R)GO TO 100
AI(R,I)=0.0
AI(I,R)=0.0
J=0
80 J= J+1
IF(J.GT.R)GO TO 70
QD=QC(I)*DI(R)
AI(I,J)=AI(I,J)+QD*QC(J)
GO TO 80
100 CONTINUE
GO TO 10
19 CONTINUE
RETURN
END
SUBROUTINE THRDST(X1,V1,N1,X2,N2)
REAL N1
IF(N1.GT.0.) GO TO 10
X1=X2
V1=0.
RETURN
10 X=(N1+FLOAT(N2))/N1
R1=(N1)/X
R2=FLOAT(N2)/X
X=R1*X1+R2*X2
V1=R1*(V1+X1**2)+112*(X2**2)-X**2
X1=X
RETURN
END
TOPO2

READS CYBER CONVERTED TOPO TAPES AND DISPLAYS THEM TO THE COMCAL.

SEQUENCE: TOPO2 NTU:F

CREATED AT GEORGIA TECH LEMS

PROGRAMMER: MICHAEL D. FURMAN

INTEGER IMP(1810), IWORK(512), IMAG(256), ITAPE(10), ISW(2)
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARCH(1, ITAPE, ISW, IERR)
CALL COMARCH(1, ITAPE, ISW, IERR)
ACCEPT "FAST SCAN? (1=YES, 0=NO) ", IFSCAN
WHEN (IFSCAN.EQ.1) IFSCAN=4 ELSE IFSCAN=1
ACCEPT "SKIP RECORDS? ", ISLP
ACCEPT "MAX ACCEPTED PEAK VALUE (AFTER SUBTRACTION) ", IPEAK
P2=IPEAK/255.0
ACCEPT "ASSIGNED PEAK SHADE VALUE? ", ISHade
ACCEPT "SUBTRACT WHAT VALUE FROM ALL DATA? ", ISUB
ACCEPT "BEGIN WITH ELEMENT? ", IEL
IEL=IEL+7
IEND=IEL+511
IF (IFSCAN.EQ.4) IEND=1800
IF (IEND.GT.1800) IEND=1800
CALL MTOPD(3, ITAPE, 0, IE)
LINE=ISKP
DO (I=1, ISKP) CALL NTDIO(3,0,IMP,IS,IE,IC)
DO (M5=1, 256)
: TYPE "LINE NUMBER ", LINE
: LINE=LINE+IFSCAN
: DO (I=1, IFSCAN) CALL MTDIO(3,0,IMP,IS,IE,IC)
: K=I
: DO (I=IEL, IEND, IFSCAN)
: IWORK(K)=IMP(I)
: K=K+1
: FIN
: K2=K-1
: SUBTRACT-CONSTANT-AND-CUT-OFF-PEAKS
: CALL PAC8(IWORK, IMAG, 512)
: CALL RINWRITE(0,0,IMAG,256)
: IF (M5.EQ.256) ACCEPT "TYPE A '1' TO CONTINUE ", IANS
: IF (IANS.EQ.1) M5=1
: FIN
STOP
TO SUBTRACT-CONSTANT-AND-CUT-OFF-PEAKS
: DO (M6=1, K2)
: WHEN (IWORK(M6).LT.ISUB) IWORK(M6)=0
: ELSE IWORK(M6)=IWORK(M6)-ISUB
: IF (IWORK(M6).GT.IPEAK) IWORK(M6)=IPEAK
: IWORK(M6)=IWORK(M6)/P2
: FIN
: FIN
END
THIS PROGRAM IS DESIGNED FOR TRAINING FIELD SELECTION AND STATISTICS GENERATION FOR THREE IMAGES

SEQUENCE: TRAIN3 MTU:F

DESIGNED AT GEORGIA TECH EES

PROGRAMMERS: NICKOLAS L. FAUST

ROBERT A. MADDOX

MICHAEL D. FURMAN

OPEN-FILES
INIT-VARIABLES FOREVER
: ASK-QUESTIONS : CONDITIONAL
: : (IFUN.EQ.1) READ-IMAGE
: : (IFUN.EQ.2) ALARM-SCREEN
: : (IFUN.EQ.3) MAGNIFY-SUBSET-OF-SCREEN
: : (IFUN.EQ.4) TAKE-TRAINING-SAMPLE
: : (IFUN.EQ.5) TAKE-GCP
: : (IFUN.EQ.6) DIGITIZE-BOUNDARY
: : (IFUN.EQ.7) CALL FSWAP("SCRIPTO.SV")
: : (IFUN.EQ.8) CALL FSWAP("INK.SV")
: : (IFUN.EQ.9) CALL FSWAP("WFMBOB.SV")
: : (IFUN.EQ.10) CALL FSWAP("CHCOLR.SV")
: : (IFUN.EQ.11) CALL FSWAP("FILTER2.SV")
: : (IFUN.EQ.12) CALL FSWAP("CLYDE5.SV")
: : (IFUN.EQ.98) BLANK-GRAPHICS
: : (IFUN.EQ.99) CLOSE-FILES (OTHERWISE)
: : TYPE "PLEASE ENTER ONE OF THE NUMBERS GIVEN"
: : ...FIN
: ...FIN
...

TO OPEN-FILES:
CALL OPEN(1, "COM.CM", 1, IFER)
CALL COMARG(1, FIELD, ISWS, IFER)
CALL COMARG(1, FIELD, ISWS, IFER)
 IF (FIELD(1).NE."MT")
 : TYPE "PARAMETER ERROR TRAIN3 MTU:F"
 : TYPE "TAPE NAME PROBABLY NOT ENTERED"
 : STOP PARAMETER ERROR IN TRAIN3 MTU:F ERROR HALT
 : ...FIN
 CALL NTOPOD(2, FIELD, 0, IFER)
 CALL FOPEN(3, "VERTEX", "B")
 CALL FOPEN(4, "HISTS", "B")
 CALL FOPEN(5, "SAVSIG", "B")
 TYPE "INPUT TAPE NUMBER IN 612 FORMAT"
 READ(11, 96) IT1, IT2, IT3, ID1, ID2, ID3

96
WRITE BINARY(3) IT1, IT2, IT3, ID1, ID2, ID3
WRITE BINARY(4) IT1, IT2, IT3, ID1, ID2, ID3
WRITE BINARY(5) IT1, IT2, IT3, ID1, ID2, ID3
CALL NTD10(2, 0, IBUF, IS, IFER)
CALL NTD10(2, 0, IBUF, IS, IFER)

...FIN
TO INIT-VARIABLES
  :  MODE=0
  :  MODE1=2
  :  MAG=1
  :  ICH=1
  :  ND=4
  :  ...FIN

TO ASK-QUESTIONS
  :  TYPE " "
  :  TYPE "THE FOLLOWING FUNCTIONS ARE AVAILABLE, SELECT BY NUMBER"
  :  TYPE " "
  :  TYPE  1 - READ IMAGE  9 - USE WFM"
  :  TYPE "  2 - ALARM 10 - USE CHCOLR"
  :  TYPE "  3 - MAGNIFY 11 - USE FILTER2"
  :  TYPE "  4 - TRAINING 12 - USE CLYDE5"
  :  TYPE "  5 - GCP"
  :  TYPE "  6 - DIGITIZE"
  :  TYPE "  7 - USE SCRIPTO 98 - BLANK GRAPHICS"
  :  TYPE "  8 - USE INK 99 - STOP"
  :  TYPE " "
  :  ACCEPT "FUNCTION NUMBER ? ", IFUN
  :  ...FIN

TO TAKE-TRAINING-SAMPLE
  :  K=1
  :  ICLOSE=1
  :  PAUSE  POSITION CURSOR FOR TRAINING FIELD THEN RETURN
  :  CALL GETXY(IXS, IYS)
  :  IXS=IAND(IXS,377K)
  :  IYS=IAND(IYS,377K)
  :  IVX(1)=IXS
  :  IVY(1)=IYS
  :  IXS=IXS
  :  IYS=IYS
  :  TYPE " CURSOR COORDS ", IXS, IYS
  :  TYPE " POSITION CURSOR AT NEXT VERTEX"
  :  ACCEPT " ENTER 1 TO READ POSITION, 2 TO CLOSE : ", ICLOSE
  :  REPEAT UNTIL (ICLOSE.NE.1)
  :  CALL GETXY(IX, IY)
  :  CALL VECTOR(1, IXS, IYS, IXS1, IYS1, 1, 1)
  :  K=K+1
  :  IVX(K)=IX
  :  IVY(K)=IY
  :  IXS=IX
  :  IYS=IY
  :  ACCEPT " ENTER 1 TO READ POSITION, 2 TO CLOSE : ", ICLOSE
  :  ...FIN
  :  CALL VECTOR(1, IXS, IYS, IXS1, IYS1, 1, 1)
  :  IYMAX=0
  :  IXMAX=0
  :  IXMIN=10000
  :  IYMIN=10000
  :  IVX(K+1)=IVX(1)
  :  IVY(K+1)=IVY(1)
  :  KP=K+1
  :  DO (IR=1, KP)
  :  :  IVX(IR)=IVX(IR)/MAG+ICOL
  :  :  IVY(IR)=IVY(IR)/MAG+IL
  :  :  IF(IVY(IR).GT.IYMAX)IYMAX=IVY(IR)
  :  :  IF(IVY(IR).LT.IYMIN)IYMIN=IVY(IR)
  :  :  IF(IVX(IR).GT.IXMAX)IXMAX=IVX(IR)
  :  :  IF(IVX(IR).LT.IXMIN)IXMIN=IVX(IR)
  :  :  ...FIN
ISTART=IYMIN
IEND=IYMAX
TYPE " BOX LIMITS", ISTART, IEND, IXMIN, IXMAX
CALL RDATA(I Hast  1, IEND, K, IVX, IVY, IXd, NP)
CALL HIST3(NP, 3) ; PUT HISTOGRAM ON GRAPHICS 3
ACCEPT " INPUT 1 FOR STATS : ", ISTAT
CONTINUE
IF(ISTAT.EQ.1)
  DO (J1=1,4)
    XM(J1)=0.0
  DO (J2=1,4)
    V(J1,J2)=0.0
  I=0
  DO (LK=1,NP)
    DO (LJ=1,4)
      MX(LJ)=IXD(LK,LJ)
    CALL SEQCov(XM,4,V,XM,I,D)
  CALL MATPRn(XM,ND,ND,ND,3HMEANS)
  CALL MATPRn(V,ND,ND,ND,3HCOV)
  TYPE " STATS CALCULATED FOR ":NP," POINTS"
  ACCEPT " DO U WISH TO SAVE THIS SIGNATURE?1=YES : ", ISIG
  IF(ISIG.EQ.1)
    TYPE " INPUT & CHARACTER NAME FOR SIGNATURE ":
    READ(11,107)NA1,NA2,NA3,NA4
    FORMAT(4A2)
    WRITE BINARY(5)NA1,NA2,NA3,NA4
    WRITE BINARY(5)NP,XM,V
    WRITE BINARY(3)NA1,NA2,NA3,NA4
    WRITE BINARY(3)K,IXMIN,IXMAX
    DO (KI=1,K)WRITE BINARY(3)K1,IVX(K1),IVY(K1)
    WRITE BINARY(4)NA1,NA2,NA3,NA4
    WRITE BINARY(4)ICOUNT
  CONTINUE
  ACCEPT " HAS SIGNATURE BEEN TREATED AS U WISH?1=YES : ", ISAT
  IF(ISAT. NE.1)
    ISTAT=1
  GO TO 25
  FIN
FIN/07
CLOSE-FILES:
CALL FCLOSE(3)
CALL FCLOSE(5)
CALL FCLOSE(4)
STOP TRAIN3 MTU:F NORMAL EXIT

TO CHECK-MAG-FACTOR:
  WHILE (MAG.NE.1.AND.MAG.NE.2.AND.MAG.NE.4.AND.MAG.NE.8)
    TYPE " INCORRECT MAG FACTOR"
    ACCEPT "ENTER MAG FACTOR (1,2,4, OR 8)", MAG
    ..FIN
..FIN
TO READ-IMAGE
  : ILT= IL
  : ICOLT= ICOL
  : TYPE "INPUT TOP LEFT COORDS FOR IMAGE (COL,LINE)"
  : ACCEPT " ENTER -1,-1 TO KEEP OLD COORDS ", ICOL, IL
  : IF (IL.EQ.-1)
  :   : IL=ILT
  :   : ICOL= ICOLT
  :   : ..FIN
  : ILS= IL
  : IES= ICOL
  : ACCEPT "MAGNIFICATION FACTOR (1,2,4, OR 8) ? ", MAG
  : CHECK-MAC-FACTOR
  : CALL RDIMG(IL,ICOL,1CH,MAG)
  : ..FIN

TO ALARM-SCREEN
  : ACCEPT "ALARM TO WHICH OVERLAY (0-3) ",I0V
  : CALL ALRM2(I0V)
  : ..FIN

TO MAGNIFY-SUBSET-OF-SCREEN
  : PAUSE POSITION CURSOR AT UPPER LEFT AND HIT RETURN
  : CALL GETXY(IX,IY)
  : CHECK-XY
  : ICOL= IX/MAG+IES
  : IL= IY/MAG+ILS
  : TYPE "SCREEN COORDS",IX,IY
  : TYPE " TAPE COORDS",ICOL,IL
  : ACCEPT "MAG FACTOR (1,2,4, OR 8) ? ",MAG
  : CHECK-MAG-FACTOR
  : CALL RDIMG(IL,ICOL,1CH,MAG)
  : ..FIN

TO TAKE-GCP
  : ACCEPT "INPUT 1 TO READ CURSOR POSITION ",IG
  : REPEAT UNTIL(IG.NE.1)
  :   : CALL GETXY(IX,IY)
  :   : CHECK-XY
  :   : TYPE "SCREEN COORDS",IX,IY
  :   : ICOLKT= IX/MAG+ICOL
  :   : ILKT= IY/MAG+IL
  :   : TYPE " TAPE COORDS",ICOLKT,ILKT
  :   : ACCEPT "INPUT 1 TO READ CURSOR POSITION ",IC
  :   : ..FIN
  : ..FIN
TO DIGITIZE-BOUNDARY
  ACCEPT "DRAW BOUNDARY ON WHICH OVERLAY (0-3) ? ", IOV
  PAUSE "POSITION CURSOR AT START AND HIT RETURN"
  CALL GETXY(IX, IY)
  CHECK-XY
  TYPE "SCREEN COORDS", IX, IY
  ICOLS = IX/MAG+ICOL
  ILS2 = IY/MAG+IL
  TYPE "TAPE COORDS", ICOLS, ILS2
  IX1 = IX
  IY1 = IY
  IXS = IX
  IYS = IY
  ACCEPT "ENTER: 1 TO READ POSITION, 2 TO CLOSE, 3 TO STOP ", ICLOSE
  REPEAT UNTIL (ICLOSE .NE. 1)
    CALL GETXY(IX, IY)
    CHECK-XY
    CALL VECTOR(IOV, IXS, IYS, IX1, IY1, 1, 1)
    TYPE "LAST VERTEX AT"
    TYPE "SCREEN COORDS", IX, IY
    ICOLS = IX/MAG+ICOL
    ILS2 = IY/MAG+IL
    TYPE "TAPE COORDS", ICOLS, ILS2
    IXS = IX
    IYS = IY
    ACCEPT "ENTER: 1 TO READ POSITION, 2 TO CLOSE, 3 TO STOP ", ICLOSE
    IF (ICLOSE .EQ. 2) CALL VECTOR(IOV, IXS, IYS, IX1, IY1, 1, 1)
..FIN

TO CHECK-XY
  IX = IAND(IX, 377K)
  IY = IAND(IY, 377K)
..FIN

TO BLANK-GRAPHICS
  ACCEPT "WHICH OVERLAY TO BLANK ? (0-3) ", IGRN
  DO (I=1,16) IDUM(I)=0
  DO (I=0,255) CALL GWR(IGRN, I, IDUM, 16)
..FIN

STOP
END
SUBROUTINE UNPAC1

INTEGER BUF
COMMON/BUFFER/BUF(3300)
COMMON/COL/JS,JF
CALL NTDIO(4,0,BUF,ISTAT,IERR,NW)
IF(NW.GT.1750)WRITE(10,1)
WRITE(10,2)NW
1 FORMAT(2X,"# OF WORDS READ = ",I5)
2 FORMAT(2X,"ERROR IN UNPAC1")
IM=1630
JC=0
DO 20 I=1,NW
NM=I+IM
20 BUFFER(NM)=BUF(I)
DO 3 J=JS,JF
I=J-1
L1=-16+(I+(I/2)*6)*8
JC=JC+1
DO 4 I=1,4
IL=L1+I*16
I2=IABS(IL-II*16)
II=IL/16
IF(I2.EQ.8)GO TO 10
II=ISHFT(BUF(II+IM),-8)
GO TO 5
10 CONTINUE
II=IAND(BUF(II+IM),377K)
5 CONTINUE
KK=I+(JC-1)*4
BUF(KK)=255-II
4 CONTINUE
3 CONTINUE
RETURN
END
C************************************************************************
C *
C UNPAC4 (SUBROUTINE)
C *
C UNPACKS PART OF LINE INSIDE POLYGON
C
C************************************************************************
CREATED AT GEORGIA TECH EES

SUBROUTINE UNPAC4(IBUF, JBUF, JL, JR, N, M)
DIMENSION JBUF(4, N), IBUF(M)
IF(JR.EQ.810)JR=810
DO (K=1,4)
   L=((JL-1)/2)*4+K
   LAST=L+(JR-JL+1)*2
   J1=1
   DO (I=L,LAST,4)
      JBUF(K,J1)=ISHFT(IBUF(I),-8)
      JBUF(K,J1+1)=IAND(IBUF(I),377K)
      J1=J1+2
   ..FIN
RETURN
END
SUBROUTINE UPPLT(PLIST,NVG,NVM)
REAL NVG,N
INTEGER PLIST,TLIST
DIMENSION PLIST(NVM),NVG(NVM),TLIST(20)
DO 10 I=1,20
10 TLIST(I)=0
DO 30 L=1,NVM
N=-1
DO 20 I=1,NVM
IF(TLIST(I).EQ.1) GO TO 20
IF(NVG(I).LE.N) GO TO 20
N=NVG(I)
J=1
20 CONTINUE
TLIST(J)=1
30 PLIST(L)=J
RETURN
END
ZCOUNTY

CALCULATES PIXEL COORDINATES FOR ANY COUNTY PARTIALLY OR WHOLLY CONTAINED IN A SCENE AND WRITES RESULTS TO DISK.

SEQUENCE: NCOUNTY TAPE FILE

CREATED AT GEORGIA TECH EES

PROGRAMMER: NICKOLAS L. FAUST

DIMENSION NCODE(50), ITEMP(2,300), FIELD(17), ISWS(2)
COMMON/DUM/IA(4100)
CALL OPEN(1, "COM.CM", 1, IERR)
CALL COMARG(1, FIELD, ISWS, IER)
CALL COMARG(1, FIELD, ISWS, IER)
CALL MTOPD(2, FIELD, O, IE)
CALL COMARG(1, FIELD, ISWS, IER)
CALL FOPEN(3, FIELD, O, IE)

ISET=1
IOU=3
NCOD=0
ACCEPT "A1,A2,A3,B1,B2,B3 ",A1,A2,A3,B1,B2,B3
ACCEPT "SKIP RECORDS? ",ISK
IF (ISK.GT.0) DO (I=1,ISK)
   CALL MTDIO(2,0,IA,IS,IE,IC)
   TYPE " DATA ", (IA(KK),KK=1,IC)
   NCOUNTY=IA(1)
   YI=IA(2)*10000.+IA(3)
   XI=IA(4)*10000.+IA(5)
   IF(XI.LE..001.AND.YI.LE..001)STOP
   ISET=1
   JSET=0
   IC=10
   UNTIL (IC.LT.5)
   CALL MTDIO(2,0,IA,IS,IE,IC)
   TYPE " DATA ", (IA(KK),KK=1,IC)
   NCOUNTY=IA(1)
   YI=IA(2)*10000.+IA(3)
   XI=IA(4)*10000.+IA(5)
   IF(XI.LE..001.AND.YI.LE..001)STOP
   IV=IV+1
   IF(NCOUNTY.NE.NCOD)
      ISET=1
      JSET=0
      IF(K.NE.0) NIV=IV-1
      DO (J=1,NIV)
         IF(ITEMP(1,J).GT.O.AND.ITEMP(1,J).LT.4000) JSET=1
         IF(ITEMP(2,J).GT.0,AND.ITEMP(2,J).LT.4000) KSET=1
         IF(JSET.EQ.1.AND.KSET.EQ.1) ISET=0
      FIN
      IF(ISET.EQ.0) FIN
      NIV=IV-1
      WRITE(IOU) NCOD
      DO (K1=1,NIV)
         WRITE(IOU,200) ITEMP(1,K1), ITEMP(2,K1)
      200 FORMAT(1X, " FOR COUNTY = ", I5)
      FIN
      IV=1
      NCOD=NCOUNTY
      K=1
      ISET=1
      FIN
      PY=A1+A2*XI+A3*YI
      PX=B1+B2*XI+B3*YI
      ITEMP(1,IV)=PX
      ITEMP(2,IV)=PY
   FIN
STOP
END
STATISTICALLY ADD SIGNATURES

INITIALIZATION

OPEN FILES

GET SPECIFIC SIGNATURES FROM SIGNATURE FILE

STATISTICALLY ADD SIGNATURES

OUTPUT NEW SIGNATURE

STOP

INPUT - 1) SIGNATURE NAME FILE
        2) SIGNATURE FILE
OUTPUT - NEW BINARY SIGNATURE FILE
READ
LINE OF
DATA

ASSIGN
DATA
to
CLUSTERS

UPDATE USING
SPECIAL
ELIMINATION AND
MERGER TESTS

LAST LINE?

yes
no

REWIND
TAPE

INPUT - LANDSAT RAW DATA TAPE
OUTPUT - CLUSTERED DATA SET
READ LINE OF DATA

ASSIGN PIXELS TO CLUSTER

OUTPUT LINE TO TAPE

LAST LINE?

yes

PRINT RESULTS

DISPLAY MEANS?

yes

PRINTOUT MEANS

no

DISPLAY SIGMA'S?

yes

PRINTOUT SIGMA'S

no

DISPLAY DISTANCE MATRIX?

yes

PRINTOUT DISTANCE MATRIX

no

STOP
LINCLASS

INITIALIZATION

READ NUMBER OF SIGNATURES AND NAMES

CALL GSIG GET SIGNATURES

GET MEAN AND CALCULATE STANDARD DEVIATION

READ POLYGON BOUNDARIES

CALCULATE MAXIMUM AND MINIMUM LINES IN POLYGON

SKIP TO POLYGON ON TAPE

INPUT - LANDSAT RAW DATA TAPE
OUTPUT - CLASSIFIED DATA TAPE

(next page)
LINCLASS (Continued)

1. **READ LINE OF DATA**
2. **PULL OUT DATA IN POLYGON**
3. **UNPACK DATA**
4. **CALL LINCAL CLASSIFY DATA IN POLYGON**
5. **UPDATE CLASS COUNTER**
6. **OUTPUT CLASS DATA TO TAPE**
7. **LAST LINE?**
   - yes: **OUTPUT CLASS SUMMARY PER SIGNATURE**
   - no: goto step 1

**STOP**
M6OCL

INITIALIZATION

READ NUMBER
OF SIGNATURES
AND NAMES

CALL GSIG
GET
SIGNATURES

INVERT
COVARIANCE
MATRICES

READ
POLYGON
BOUNDARIES

CALCULATE
MAXIMUM
AND
MINIMUM LINES
IN POLYGON

SKIP TO
POLYGON
ON
TAPE

INPUT - LANDSAT RAW DATA TAPE
OUTPUT - CLASSIFIED DATA TAPE
M60CL (Continued)

READ LINE OF DATA

PULL OUT DATA IN POLYGON

UNPACK DATA

CALL MAXLI CLASSIFY DATA IN POLYGON

UPDATE CLASS COUNTER

OUTPUT CLASS DATA TO TAPE

LAST LINE?

OUTPUT CLASS SUMMARY PER SIGNATURE

STOP
INPUT - DISK FILE
OUTPUT - DATA TAPE

STRIPLESS
INITIALIZATION

READ IN TABLE FROM DISK

READ IN DATA LINE FROM TAPE

CORRECT DATA LINE BY LINE AND OUTPUT TO TAPE

LAST LINE?

yes

OUTPUT 2 EOF'S

STOP

no
INITIALIZATION

READ 6 LINES FROM TAPE

SETUP 6 X 6 SLIDING WINDOW

AGGREGATE DATA FOR TABLE BUILDER

END OF TAPE?

LAST TAPE?

DETERMINE USABLE DATA RANGES

INPUT - LANDSAT RAW DATA TAPE
OUTPUT - DISK FILE AND PRINTER

(next page)
ADJUST MEAN

SMOOTH SPARSE DATA OVER USABLE DATA RANGES

MID RANGE TABLE BUILDER

BELOW RANGE TABLE BUILDER

ABOVE RANGE TABLE BUILDER

OUTPUT TABLE TO DISK

STOP
TRAIN3

INITIALIZE

INPUT OPTION

READ IMAGE?

ALARM DISPLAY?

MAGNIFY SUBSET OF DISPLAY?

TAKE TRAINING SAMPLE?

TAKE GROUND CONTROL POINT?

DIGITIZE BOUNDARY?

RETURN

INPUT - LANDSAT RAW DATA TAPE
OUTPUT - FILES FOR
a) SIGNATURES
b) VERTICES
c) HISTOGRAMS

...
WRITE TEXT TO IMAGE?

DRAW FROM CURSOR?

CHANGE FUNCTION MEMORY?

CHANGE COLOR MEMORY?

IMAGE HIGH PASS FILTER?

VERSATEC IMAGE DISPLAY?

BLANK A GRAPHICS OVERLAY?

RETURN

CLOSE FILES?
TRAIN3 (Continued)

A

INPUT COORDINATES

INPUT MAGNIFICATION FACTOR

CALL RDIMG (READ IMAGE)
READ AN IMAGE AND DISPLAY TO SCREEN

RETURN

B

INPUT OVERLAY NUMBER

CALL ALRM2 (ALARM)
ALARM THE IMAGE WITH LAST SIGNATURE

RETURN
CALL RDIMG
(READ IMAGE)
READ AN IMAGE
AND MAGNIFY FROM
CURSOR POSITION

RETURN

READ CURSOR POSITION

INPUT MAGNIFICATION FACTOR

CALL RDIMG
(READ IMAGE)
READ AN IMAGE
AND MAGNIFY FROM
CURSOR POSITION

RETURN

READ CURSOR POSITION

CALL VECTOR

CLOSE POLYGON?

CALL VECTOR

(next page)
TRAIN3 D (Continued)

CALL SEQCOV
COMPUTE MEAN AND COVARIANCE

CALL MATPRT
PRINT STATISTICS

TRAIN3 D (Continued)

CALL RDATA
READ DATA FROM TAPE FOR STATISTICS

CALL HIST3B
CALCULATES AND DISPLAYS HISTOGRAM

STATISTICS yes

CALL SEQCOV
COMPUTE MEAN AND COVARIANCE

CALL MATPRT
PRINT STATISTICS

SAVE?

READ NAME SAVE STATISTICS

RETURN yes

SIGNATURE TREATED PROPERLY no
TRAIN3 (Continued)

E

READ CURSOR POSITION

COMPUTE SCREEN AND TAPE COORDINATES

TYPE OUT COORDINATES

MORE POINTS?

RETURN

yes

no
TRAIN3 (Continued)

1. INPUT OVERLAY NUMBER
2. READ CURSOR POSITION
3. TYPE OUT SCREEN AND TAPE COORDINATES
4. CLOSE?
   - yes: CALL VECTOR TO CLOSE POLYGON
   - no: STOP?
5. STOP?
   - yes: RETURN
   - no: READ CURSOR POSITION
6. CALL VECTOR
7. TYPE OUT SCREEN AND TAPE COORDINATES
SWAP TO SCRIPTO FOR WRITING TEXT TO SCREEN

RETURN

SWAP TO INK TO DRAW FROM CURSOR

RETURN

SWAP TO WFM TO CHANGE FUNCTION MEMORY

RETURN
TRAIN3 (Continued)

J

SWAP TO CHCOLOR
TO CHANGE
COLOR MEMORY

RETURN

K

SWAP TO FILTER2
FOR
HIGH PASS FILTER

RETURN

L

SWAP TO CLYDE5
FOR IMAGE
TO VERSATEC

RETURN

153