General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
AgRISTARS

Supporting Research

"AS-BUILT" DESIGN SPECIFICATION FOR SEGMENT MAP (SGMAP) PROGRAM

M. A. Tompkins

Lockheed Engineering and Management Services Company, Inc.
1830 NASA Road 1, Houston, Texas 77058

Lyndon B. Johnson Space Center
Houston, Texas 77058
"AS-BUILT" DESIGN SPECIFICATION
FOR
SEGMENT MAP (SGMAP) PROGRAM

Job Order 71-308

Prepared By
M. A. Tompkins

Approved By
G. L. Clouette, Supervisor
Support Systems Software Section

R. A. McClane, Manager
Data Systems Department

R. Kent Lennington, Supervisor
Techniques Development Section

T. C. Minter, Manager
Development and Evaluation Department

Prepared By
Lockheed Engineering and Management Services Company, Inc.

For
Earth Observations Division
Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

June 1981

LEMSCO-15937
This document is the "As-Built" Design Specification for Segment Map (SGMAP) Program which is part of the CLASYF package. This program is designed to output symbolic maps or numerical dumps from Cluster/Classification files or ground truth/processed ground truth files which are in 'Universal' format.
CONTENTS

Section                                      Page
1. SCOPE                                      1-1
2. APPLICABLE DOCUMENTS                      2-1
3. SYSTEM DESCRIPTION
   3.1 SYSTEM FLOWCHART                      3-1
   3.2 HARDWARE DESCRIPTION                  3-5
   3.3 SOFTWARE DESCRIPTION                  3-5
   3.4 FILE DESCRIPTION                      3-6
   3.4.1 INPUT DATA FILE                     3-6
   3.4.2 USER DEFINED FILE (SYMBOL FILE NAME) (SYMBOL FILE TYPE) A  3-7
   3.4.3 USER DEFINED FILE (FILE NAME) CC A  3-8
   3.5 SOFTWARE DESCRIPTION                  3-10
   3.5.1 SGMAP PROGRAM                       3-10
   3.5.2 SOFTWARE COMPONENT NO. 1 (CRD2IN)   3-12
   3.5.3 SOFTWARE COMPONENT NO. 2 (NUMB)     3-14
   3.5.4 SOFTWARE COMPONENT NO. 3 (INTPAR)    3-16
   3.5.5 SOFTWARE COMPONENT NO. 4 (SPMAP)     3-18
   3.5.6 SOFTWARE COMPONENT NO. 5 (KUSKLA)    3-20
   3.5.7 SOFTWARE COMPONENT NO. 6 (GTMAP)     3-22
   3.5.8 SOFTWARE COMPONENT NO. 7 (CROPP)     3-24
4. OPERATION                                  4-1
   4.1 OPERATING INSTRUCTION                  4-1
   4.2 COMMANDS DESCRIPTION                   4-1
   4.2.1 START                                4-2
   4.2.2 SGMAP                                4-2
   4.2.3 END                                  4-2
   4.3 OPERATING EXAMPLE                      4-3
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendices</td>
<td></td>
</tr>
<tr>
<td>A. COMMON BLOCKS</td>
<td>A-1</td>
</tr>
<tr>
<td>B. PROGRAM LISTINGS</td>
<td>B-1</td>
</tr>
<tr>
<td>C. JOB CONTROL SOFTWARE</td>
<td>C-1</td>
</tr>
<tr>
<td>D. SAMPLE OUTPUT</td>
<td>D-1</td>
</tr>
</tbody>
</table>
### FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.1</td>
<td>System Flow for Ground/Processed Ground Truth Input Data File.</td>
<td>3-2</td>
</tr>
<tr>
<td>3.1.2</td>
<td>System Flow for Cluster/Classification Input Data File</td>
<td>3-3</td>
</tr>
<tr>
<td>3.1.3</td>
<td>SGMAP Hierarchy.</td>
<td>3-4</td>
</tr>
</tbody>
</table>
SGMAP PROGRAM

1.0 SCOPE

This document contains the description of the SGMAP Program. This program produces as output either a numeric dump or a symbolic map from an input Landsat Segment Cluster/Classification file or a digitized aircraft Ground Truth/Processed Ground Truth file.
2.0 APPLICABLE DOCUMENTS

The following documents form a part of this specification:

AD 63-2457-3308-01 Transferring Badhwar Software.

AD NAS 9-15200 Technical Memorandum Format Specifications for LACIE (Phase III) and Accuracy Assessment Computer Data Products.
3.0 SYSTEM DESCRIPTION

3.1 SYSTEM FLOWCHART

The system flowcharts for processing ground truth and cluster/classification data files are depicted in Figures 3.1.1 and 3.1.2, respectively.
Figure 3.1.1 System Flow for Ground/Processed Ground Truth Input Data File
Figure 3.1.2: System Flow for Cluster/Classification Input Data File
Figure 3.1.3 SGMAP Hierarchy
3.2 HARDWARE DESCRIPTION

The software for the SGMAP Program is operational on the IBM 3031 computer at Purdue.

3.3 SOFTWARE DESCRIPTION

This program produces numeric dumps or symbolic maps of two kinds of files: namely, (1) cluster or classification files and (2) ground truth files.

Cluster or classification files are single channel universal format image files produced by several different clustering and classification programs. They are "pixel level" files; i.e., they consist of 117 lines with 196 pixels per line. (See Section 3.4.1 for a complete file description).

Ground truth files are single channel universal format image files that are similar to the cluster or classification files except that they are at the sub-pixel level, i.e., they have six sub-pixels corresponding to each pixel on a cluster or classification file. They have 351 (3x117) lines with 392 (2x196) sub-pixels per line. (See Section 3.4.1 for a complete file description).

If a cluster or classification file is input, then the output numeric dump or symbolic map is necessarily at the pixel level also. However if the input is a ground truth file, then the output can be either sub-pixel level or pixel level. In the latter case the program combines the six sub-pixels that correspond to each pixel and determines the label for that pixel by majority rule. A numeric dump or symbolic map is then produced of this pixel data.

The valid range of sample values is 1-256.
3.4 FILE DESCRIPTION

3.4.1 INPUT DATA FILE

The two acceptable forms of input data are:

Cluster or Classification Files: These pixel level files are in universal format with one channel per physical record. There are 117 records, each having 360 8-bit bytes.

Ground Truth Files: These sub-pixel level files are in universal format with one channel per physical record. There are 351 records, each having 540 8-bit bytes. The contents of each byte have been biased with -128 and are stored in 8-bit two's-complement notation.
3.4.2 USER DEFINED FILE (SYMBOL FILE NAME) (SYMBOL FILE TYPE) A

For output maps, the symbol file defines a corresponding symbol for each of the 256 crop code values in the universal format input file. The user defines symbols for each crop code or crop code range; i.e., if only one crop code is mapped to a symbol, the beginning and ending of the range are the same. The usual procedure followed is that the user (1) defines all 256 codes to one symbol and (2) redefines any codes which merit unique symbols. The last entry in the file must be 0 0 0.

For output numeric dump the only entry is 0 0 N.

This file provides inputs in the given formats to the SGMAP program.

| Columns    | 1-5 | 6-10 | 15 |
|Format      | I5  | I5   | A1 |
| start      |     |      |    |
| end        |     |      |    |
| symbol     |     |      |    |
| Purpose    |     |      |    |
|            |     |      |    |
| 0 0 0      |     |      |    |
| Signifies the end of SYMBOL input for a MAP. |
| 0 0 N      |     |      |    |
| Signifies the end of SYMBOL input for a NUMERIC DUMP. |

The following is an example of a symbol file for an OUTPUT Numeric Dump.

0 0 N

The following is an example of a symbol file for an OUTPUT Map.

1 256 (blank)
1 1 *
40 40 C
127 127 B
50 50 W
0 0 0
This file is used to specify a number of parameters mainly related to output description. It is composed of card records which are input to the SGMAP routine.

The first six cards have a keyword beginning in column 1 followed by parameters in columns 11 through 72. Blanks are optional. The following description lists the keywords and describes the corresponding inputs.

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>ACCEPTABLE INPUTS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>GROUND TRUTH</td>
<td>This card defines the type of run. In the case where the output symbolic map is a classification or cluster map, the entry on this card is &quot;Class/Cluster&quot;. In the case where the output symbolic map is a ground truth map there are two possible entries; namely, (1) &quot;Ground Truth&quot; if pixel level output is desired and (2) &quot;Sub-Pixel&quot; if sub-pixel output is desired.</td>
</tr>
<tr>
<td></td>
<td>SUB-PIXEL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CLASS/CLUSTER</td>
<td></td>
</tr>
<tr>
<td>OUTPUT</td>
<td>MAP</td>
<td>This card specifies whether the output will be a numeric dump or a symbolic map. The corresponding entries are &quot;Numeric&quot; and &quot;Map&quot;.</td>
</tr>
<tr>
<td></td>
<td>NUMERIC</td>
<td></td>
</tr>
<tr>
<td>RECKSP</td>
<td>Positive number</td>
<td>This card specifies a number of records to skip following the header record. Normally no skipping is required. However, some input files do require it. If missing, the default is 0.</td>
</tr>
<tr>
<td>DATE</td>
<td>Free form</td>
<td>This card gives the date (in free-form) for report identification.</td>
</tr>
<tr>
<td>SEGMENT</td>
<td>Segment number</td>
<td>This card specifies the segment number for report identification.</td>
</tr>
<tr>
<td>AI</td>
<td>Analyst's name</td>
<td>This card specifies the analyst's name for report identification.</td>
</tr>
<tr>
<td>*END</td>
<td>Ignored</td>
<td>This card identifies the end of the user defined cards.</td>
</tr>
</tbody>
</table>

*Optional Input
Next there are one or more cards which specify the part of the image which is to be dumped or mapped. Each card specifies the PIXEL (never sub-pixel) coordinates of the upper left corner of the area to be dumped or mapped. The output always begins at this corner and produces a display that is 117 lines by 196 samples. If pixel level output has been specified, these lines and samples are pixels; if sub-pixel output has been specified, they are sub-pixels. When pixel level output is desired, the vertex 1,1 will cause all of the image to be output. However, when sub-pixel output is desired, six vertices are required to output the entire image. Of course, fewer vertices can be used if only part of the image is desired. The vertices (line, sample) are entered one to a card and may begin in any column 1 through 72. Leading, trailing and embedded blanks on these cards are ignored. The range of values permitted is 1 through 79 for lines and 1 through 99 for samples. A maximum of six vertex cards may be used.

A $END card is placed after the last vertex card.

All cards following the $END card are considered comments and are printed.

The following is an example of an user's CC file where the input data is in SUB-PIXEL structure. Maps giving complete coverage of all lines and samples are desired.

```
AI MARY ANN TOMPKINS
DATE DECEMBER 25, 1980
SEGMENT 1981
TYPE SUB-PIXEL
OUTPUT MAP
*END
  1,1
  40,1
  79,1
  1,99
  40,99
  79,99
$END
```
3.5 SOFTWARE DESCRIPTION

3.5.1 SGMAP PROGRAM

Purpose

SGMAP produces either a numeric dump or a symbolic map from an input Landsat Segment Cluster/Classification file or a digitized aircraft Ground Truth/Processed Ground Truth file.

Linkages

SGMAP calls CRD2IN, CPTIME, GTMAP, KUSKLA, and SPMAP.

Interface

Calling sequence:

N/A (A description for SGMAP EXEC which can be used to load and execute SGMAP is found in Section 4.0).

Calling sequence parameters:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function value:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Labeled COMMON parameters:

Full description of labeled COMMON blocks are contained in Appendix A.

<table>
<thead>
<tr>
<th>Label</th>
<th>Variable</th>
<th>Element Position</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/MAP/</td>
<td>IXLBUF(3060)</td>
<td>1</td>
<td>0</td>
<td>Pixel/subpixel input buffer.</td>
</tr>
<tr>
<td></td>
<td>IXLINE(98)</td>
<td>2</td>
<td>0</td>
<td>Output transformed data line.</td>
</tr>
<tr>
<td></td>
<td>IXLTRN(256)</td>
<td>3</td>
<td>0</td>
<td>Transformations for a sample value.</td>
</tr>
</tbody>
</table>

3-10
Blank COMMON parameters:

Full description of blank COMMON is contained in Appendix A.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Element Position</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Inputs

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>seq. data</td>
<td>Universal formatted ground truth, classification, or cluster file (See section 3.4.1).</td>
</tr>
<tr>
<td>13</td>
<td>seq. data</td>
<td>Symbolic mapping information (See section 3.4.2).</td>
</tr>
</tbody>
</table>

Outputs

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Report</td>
<td>Printout of the user defined control cards.</td>
</tr>
<tr>
<td>14</td>
<td>Report</td>
<td>Symbolic map or numeric dump of input data.</td>
</tr>
</tbody>
</table>

Storage requirement
Not applicable.

Description

SGMAP, according to user specification and type of input data file, produces symbolic maps or dumps in pixel or subpixel form (See Appendix C).

Flowchart
Not applicable.

Listing
See Appendix B for program listing.
3.5.2 SOFTWARE COMPONENT NO. 1 (CRD2IN)

Purpose
Subroutine CRD2IN reads, decodes, and writes to a printer file the user supplied card control file.

Linkages
CRD2IN is called by SGMAP.
CRD2IN calls INTPAR and NUMB.

Interface
Calling sequence:
CALL CRD2IN (MTYPE, KOUT, NRECSK, LFTCRS, IERR).

Calling sequence parameters:
<table>
<thead>
<tr>
<th>Argument</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTYPE</td>
<td>0</td>
<td>Identifies the type of conversion of input structure to output structure.</td>
</tr>
<tr>
<td>KOUT</td>
<td>0</td>
<td>Identifies the format of output.</td>
</tr>
<tr>
<td>NRECSK</td>
<td>0</td>
<td>Number of records to skip.</td>
</tr>
<tr>
<td>LFTCRS(6)</td>
<td>0</td>
<td>Identifies the upper left corner of output.</td>
</tr>
<tr>
<td>IERR</td>
<td>0</td>
<td>Input error indicator flag.</td>
</tr>
</tbody>
</table>

Function value:
N/A.

Labeled COMMON parameters:
Full description of labeled COMMON blocks are contained in Appendix A.
None.

Blank COMMON parameters:
Full description of blank COMMON is contained in Appendix A.
None.
**Inputs**

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>seq. data</td>
<td>Control card file (See Section 3.4.3).</td>
</tr>
</tbody>
</table>

**Outputs**

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Printer</td>
<td>File contains copy of report summary of user defined cards.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Used as a reread file.</td>
</tr>
</tbody>
</table>

**Storage requirement**
Not applicable.

**Description**

CRD2IN reads the key word (first four bytes) on the Control Card and then outputs the card. If the key word is not recognized, an error message is printed and execution continues. If the key word is recognized, the appropriate action regarding the card's parameter is taken. The routine determines if all required cards are present and if the parameters are acceptable. If either test fails, an error flag is set.

**Flowchart**
Not applicable.

**Listing**
See Appendix B for routine listing.
3.5.3 SOFTWARE COMPONENT NO. 2 (NUMB)

Purpose

NUMB decodes character information into numerical information.

Linkages

NUMB is called by CRD2IN.

Interface

Calling sequence:

CALL NUMB (KCHAR, NUMBER, ICOR).

Calling sequence parameters:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCHAR(80)</td>
<td>I</td>
<td>Card image.</td>
</tr>
<tr>
<td>NUMBER</td>
<td>I</td>
<td>Decoded number.</td>
</tr>
<tr>
<td>ICOL</td>
<td>I</td>
<td>Column to start decode procedure</td>
</tr>
</tbody>
</table>

Function value:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Labeled COMMON parameters:

Full description of labeled COMMON blocks are contained in Appendix A.

Label | Variable | Element Position | Input/Output | Description
------|----------|------------------|--------------|-----------------|
None   |          |                  |              |                 |

Blank COMMON parameters:

Full description of blank COMMON is contained in Appendix A.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Element Position</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Inputs
None.

Outputs
None.

Storage requirement
Not applicable.

Description
Decodes a number on a user supplied control card starting with the first non-blank character in the control card parameter field (ICOL) and ending with the first non-blank, non-numeric character.

Flowchart
Not applicable.

Listing
See Appendix B for routine listing.
3.5.4 SOFTWARE COMPONENT NO. 3 (INTPAR)

Purpose
Read and decode user defined control cards consisting of a pair of numbers.

Linkages
INTPAR is called by CRD2IN.

Interface
Calling sequence:

```
CALL INTPAR (LFTCRS, IERR).
```

Calling sequence parameters:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFTCRS(6)</td>
<td>0</td>
<td>Identifies upper left corner of output.</td>
</tr>
<tr>
<td>IERR</td>
<td>0</td>
<td>Identifies input error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IERR = 0 OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IERR = 1 Error</td>
</tr>
</tbody>
</table>

Function value:
N/A

Labeled COMMON parameters:
Full description of labeled COMMON blocks are contained in Appendix A.
None.

Blank COMMON parameters:
Full description of blank COMMON is contained in Appendix A.
None.

Inputs

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Seq. data</td>
<td>Control c-rd file (See Section 3.4.3).</td>
</tr>
</tbody>
</table>
**Outputs**

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Printer</td>
<td>File contains copy of report summary of user defined control cards.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Used as a reread file.</td>
</tr>
</tbody>
</table>

**Storage requirement**

Not applicable.

**Description**

INTPAR reads and decodes number pairs, input one to a card, separated by a comma.

**Flowchart**

Not applicable.

**Listing**

See Appendix B for routine listing.
3.5.5 SOFTWARE COMPONENT NO. 4 (SPMAP)

Purpose

SPMAP reads input data in pixel structure and transforms the data into output form.

Linkages

SPMAP is called by SGMAP.

Interface

Calling sequence:

CALL SPMAP (PIXPRO, MOFFS, LINE, LNS).

Calling sequence parameters:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIXPRO</td>
<td>0</td>
<td>Current number of pixels processed.</td>
</tr>
<tr>
<td>MOFFS</td>
<td>I</td>
<td>Output pixel offset.</td>
</tr>
<tr>
<td>LINE</td>
<td>I</td>
<td>Starting line of pass.</td>
</tr>
<tr>
<td>LNS</td>
<td>I</td>
<td>Current line.</td>
</tr>
</tbody>
</table>

Function value:

N/A.

Labeled COMMON parameters:

Full description of labeled COMMON blocks are contained in Appendix A.

<table>
<thead>
<tr>
<th>Label</th>
<th>Variable</th>
<th>Element Position</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>IXLBUF(3060)</td>
<td>1</td>
<td>0</td>
<td>Subpixel input buffer.</td>
</tr>
<tr>
<td></td>
<td>IXLTRN(256)</td>
<td>3</td>
<td>I</td>
<td>Transformations for a sample value.</td>
</tr>
<tr>
<td></td>
<td>IXLINE(98)</td>
<td>2</td>
<td>0</td>
<td>Transformed data line.</td>
</tr>
</tbody>
</table>
Blank COMMON parameters:

Full description of blank COMMON is contained in Appendix A.
None.

Inputs

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Seq. data</td>
<td>Universal formatted ground truth file (See Section 3.4.1).</td>
</tr>
</tbody>
</table>

Outputs

None.

Storage requirement

Not applicable.

Description

SPMAP reads one line of data into an internal buffer beginning with the first requested line. Each pixel is assigned an output symbol by using the pixel value to index into the pixel transformation buffer which contains a user defined symbol. This symbol is then stored into an output buffer for eventual output.

Flowchart

Not applicable.

Listing

See Appendix B for routine listing.
3.5.6 SOFTWARE COMPONENT NO. 5 (KUSKLA)

Purpose

KUSKLA reads input data in pixel structure and transforms the data into output form.

Linkages

KUSKLA is called by SGMAP.

Interface

Calling sequence:

CALL KUSKLA (MOFFS).

Calling sequence parameters:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOFFS</td>
<td>I</td>
<td>Output pixel offset.</td>
</tr>
</tbody>
</table>

Function value:

N/A.

Labeled COMMON parameters:

Full description of labeled COMMON blocks are contained in Appendix A.

<table>
<thead>
<tr>
<th>Label</th>
<th>Variable</th>
<th>Element Position</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/MAP/</td>
<td>IXLBUF(3060)</td>
<td>1</td>
<td>0</td>
<td>Pixel input buffer.</td>
</tr>
<tr>
<td></td>
<td>IXLINE(98)</td>
<td>2</td>
<td>0</td>
<td>Transformed data line.</td>
</tr>
<tr>
<td></td>
<td>IXLTRN(256)</td>
<td>3</td>
<td>I</td>
<td>Transformations for a sample value.</td>
</tr>
</tbody>
</table>

Blank COMMON parameters:

Full description of blank COMMON is contained in Appendix A.

None.
**Inputs**

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Seq. data</td>
<td>Universal formatted classification or cluster file (See Section 3.4.1).</td>
</tr>
</tbody>
</table>

**Outputs**

None.

**Storage requirement**

Not applicable.

**Description**

KUSKLA reads one line of data into an internal buffer, transforms each pixel into an output symbol, and stores the pixel into an output buffer for eventual output.

**Flowchart**

Not applicable.

**Listing**

See Appendix B for routine listing.
3.5.7 SOFTWARE COMPONENT NO. 6 (GTMAP)

Purpose

GTMAP reads input data in subpixel structure and converts the data to a pixel value.

Linkages

GTMAP is called by SGMAP.
GTMAP calls CROPP.

Interface

Calling sequence:

CALL GTMAP (PIXPRO, MCFSS).

Calling sequence parameters:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIXPRO</td>
<td>0</td>
<td>Current number of pixels processed.</td>
</tr>
<tr>
<td>MCFSS</td>
<td>I</td>
<td>Output pixel offset.</td>
</tr>
</tbody>
</table>

Function value:

None.

Labeled COMMON parameters:

Full description of labeled COMMON blocks are contained in Appendix A.

<table>
<thead>
<tr>
<th>Label</th>
<th>Variable</th>
<th>Element Position</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/MAP/</td>
<td>IXLBUF(3060)</td>
<td>1</td>
<td>0</td>
<td>Subpixel input buffer.</td>
</tr>
<tr>
<td></td>
<td>IXLTRN(256)</td>
<td>3</td>
<td>I</td>
<td>Transformations for a sample value.</td>
</tr>
<tr>
<td>/NSBIXL/</td>
<td>NSUBPX(6)</td>
<td>1</td>
<td>0</td>
<td>Subpixel that map to a pixel.</td>
</tr>
</tbody>
</table>

3-22
Blank COMMON parameters:

Full description of blank COMMON is contained in Appendix A.

N/A.

Inputs

<table>
<thead>
<tr>
<th>Unit</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Seq. data</td>
<td>Universal formatted ground truth file.</td>
</tr>
</tbody>
</table>

Outputs

None.

Storage requirement

Not applicable.

Description

GTMAP reads three lines of subpixel structure data into an internal buffer. The sub-pixels are divided into sub-pixel sets (2 sub-pixel across x 3 lines down) and CROPP is called to rule on a pixel value for the set. Each pixel is assigned an output symbol by using the pixel value to index into the pixel transformation buffer which contains a user defined symbol. This symbol is then stored into an output buffer for eventual output.

Flowchart

Not applicable.

Listing

See Appendix B for routine listing.
3.5.8 SOFTWARE COMPONENT NO. 7 (CROPP)

Purpose

CROPP accepts six subpixels as input and outputs a pixel value representative of the subpixels.

Linkages

CROPP is called by GRMAP.

Interface

Calling sequence:

```
CALL CROPP (KROP, IXLCNT).
```

Calling sequence parameters:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KROP</td>
<td>0</td>
<td>Majority crop code.</td>
</tr>
<tr>
<td>IXLCNT</td>
<td>0</td>
<td>Count of number subpixels that match the majority code.</td>
</tr>
</tbody>
</table>

Function value:

N/A.

Labeled COMMON parameters:

Full description of labeled COMMON blocks are contained in Appendix A.

<table>
<thead>
<tr>
<th>Label</th>
<th>Variable</th>
<th>Element Position</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/NSBIXL/</td>
<td>NSUBPX(6)</td>
<td>1</td>
<td>I</td>
<td>Subpixels that map to a pixel.</td>
</tr>
</tbody>
</table>

Blank COMMON parameters:

Full description of blank COMMON is contained in Appendix A.

None.
Inputs
None.

Outputs
None.

Storage requirement
Not applicable.

Description
The first sub-pixel value upon entry is considered to be the majority pixel (KROP) value. The NSUBPX array is traversed and the number of codes equal to KROP counted. This procedure continues until a count of each unique code is made. If at anytime a count becomes greater than the count of KROP this code becomes KROP's value.

Flowchart
Not applicable.

Listing
See Appendix B for routine listing.
4.0 OPERATION

4.1 OPERATING INSTRUCTIONS

SGMAP is operational on the IBM 3031 computer at LARS, West Lafayette, Indiana.

This program requires the use of a D disk which is assigned as a temporary disk. This disk will be assigned by the SGMAP EXEC; and the user, therefore, should not assign a disk to his machine using a MODE D.

Prior to invoking the SGMAP EXEC, the user must establish on his A disk a SYMBOL file as described in section 3.4.3 and a USER CONTROL file as described in section 3.4.4.

4.2 COMMANDS DESCRIPTION

To execute SGMAP, the user enters the following series of commands which invoke the JOB CONTROL SOFTWARE.

START
SGMAP.....
END

The following sections describe each of these commands in detail. Input fields are separated by blanks. If more than one word is required to describe an input field, the description is enclosed in pointed brackets <>. If an input is optional the field is enclosed in square brackets []. Do not include these explanatory characters <> [] when actually submitting input to the computer. To enter a command, the user types one input per defined input field and separates each field with a blank.
4.2.1 START

The START command spools the user's console file. The use of this command alone with the END command will provide a listing of all information appearing on the user's console file. (If running an interactive job, this is the terminal; if running a batch job, this is a system defined device). The START command is invoked by the user typing the following:

START

4.2.2 SGMAP

The SGMAP command executes the program. The SGMAP has the following as forms and is invoked by typing one of the following, according to the users requirement.

SGMAP <SYMBOL FILE> <SYMBOL FILE> <CC FILE> TAPE# FILE# [TAPE DENSITY]
< NAME > < TYPE > < NAME >

If the file is on disk -

SGMAP <SYMBOL FILE> <SYMBOL FILE> <CC FILE> <FILENAME> <FILETYPE> <FILEMODE>
< NAME > < TYPE > < NAME >

The output from the SGMAP program is spooled to the HOUSTON line printer. The output consists of a MAP and the USER IDENTIFICATION file.

4.2.3 END

This command closes the user's console file and causes a copy to be sent to the printer. This command has no effect if the START command was not previously issued. The END command is invoked by the user typing the following:

END
4.3 OPERATING EXAMPLE

For our example we will assume the following:

The user's symbol file is established as required on his A disk under the file description:

\[ 880 \text{ 79082 A} \]

The user's information file is established as required on his A disk under the filename of his choice and the required filetype (CC). (For our example we will use SYM880 CC A for our complete file description.)

The user has a 1600 BPI tape# 2645 which contains the file (17) he wishes to map.

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>EXPLANATION OR ACTION TAKEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Spools the console file.</td>
</tr>
<tr>
<td>SGMAP 880 79082 SYM880 2645 17</td>
<td>Executes the SGMAP PROGRAM. The user's tape is 1600 BPI; therefore, the user inputs tape#,file#, exercising the option of using the defined default of 1600 BPI.</td>
</tr>
<tr>
<td>END</td>
<td>Closes the user's console file and spools the file to the HOUSTON printer.</td>
</tr>
</tbody>
</table>
APPENDIX A
COMMON BLOCKS
COMMON BLOCKS

COMMON/MAP/IXLBUF(3060),IXLINE(98),IXLTRN(256)
IXLBUF  PIXEL/SUB-PIXEL INPUT BUFFER
IXLINE  OUTPUT TRANSFORMED DATA
IXLTRN  TRANSFORMATIONS FOR SAMPLE VALUE

COMMON/NSBIXL/NSUBPX(6)
NSUBPX  6 GROUND TRUTH SUB-PIXELS THAT ARE ASSOCIATED WITH
        A CORRESPONDING LACIE PIXEL
APPENDIX B
PROGRAM LISTINGS
SUBROUTINE CRZIN(MTYPE,KOUT,RECSK,LEFTS,IERH)

INTEGER MTYPE
TYPE OF OUTPUT REQUESTED

INTEGER KOUT
KIND OF OUTPUT

INTEGER RECSK
NUMBER OF RECS TO SKIP IN FILE

INTEGER LEFTS(4,7)
UPPER LEFT CORNER(S) OF OUTPUT

INTEGER IERH
ERROR FLAG
0 - NO ERROR
1 - ERROR(S)

HISTORY

MARY TOMPKINS        LEMSCO  10/30/80  ORIGINAL CODE

PURPOSE

HEAD AND ANALYZE CARDS DESCRIBING THE FOLLOWING:
FILE  - DOCUMENTATION LIST OF INPU FILES
TYPE  - INPUT: FIRST NON-BLANK CHARACTER AFTER COLUMN 10
ACCEPTABLE INPUTS:
G - GROUND TRUTH
O - OUTSIDE
S - SURPIX
C - CLASS/CLOSER
OUTPUT - INPUT: KIND OF OUTPUT
ACCEPTABLE INPUTS:
M - MAP
N - NUMERICAL DUMP
RECSK - INPUT: NUMBER OF RECORDS TO SKIP WITHIN FILE
DEFAULT 0
DATE  - DOCUMENTATION: CURRENT DATE
SEGMENT - DOCUMENTATION: SEGMENT NUMBER
ANALYST'S NAME
END   - SPECIFY END OF USER DEFINED RECORDS.

EXTERNAL REFERENCES

NUMB  - DECODE A NUMBER
INPAR  - DECODE INTEGER PAIR(S)

EXCEPTION

1. IF ANY OF THE FOLLOWING CARDS ARE MISSING OR DATA IS UNACCEPT-
ABLE ISSUE A DIAGNOSTIC MESSAGE AND SET IER = 1.

2. IF A CARD IS OF AN UNDEFINED TYPE ISSUE A DIAGNOSTIC MESSAGE
AND CONTINUE.

3. IF THERE ARE NO USER DEFINED UPPER LEFT CORNERS ISSUE A
DIAGNOSTIC MESSAGE AND SET IER = 3.

4. IF THERE IS NO RECSK CARD THEN RECSK DEFAULTS TO 0.

LOCAL DECLARATIONS

INTEGER CARD
CARD READ WORD AT A TIME

INTEGER CHAP
CARD HEAD CHARACTER AT A TIME

INTEGER KOUNT
COUNT OF NUMBERS RETURNED FROM FUNCTION NUMB

INTEGER NUMB
DECODED NUMBER

INTEGER INFC
ALLOWABLE CARD TYPES

INTEGER KANDR
CARD TYPE

"
LEVEL 7.3.0 (JUNE 70)  CRD2IN  OS/360 FORTRAN H EXTENDED  DATE 01.139/13.19.55  PAGE 4

***** FORTRAN CROSS REFERENCE LISTING *****

NAME    TAG    TYPE    ADD.   NAME    TAG    TYPE    ADD.   NAME    TAG    TYPE    ADD.   NAME    TAG    TYPE    ADD.
KHF
KHF SFA  *000174  ICOL SFA  *000174  TERR SFA  *000174  HARD SF  *000060
KHF SFA  *000326  KUNT S   *000326  NUMR F AF  *000326  INDEC  *000000
KHF SFA  *000126  UNK S    *000126  CFI2IZIN  *000010  SVCNDR  *000000
KHF SFA  *000174  INITPAR SF XF  *000060  ROMN2Y S  *000010  LFICRS SFA CR  *000000

SOURCE STATEMENT LABELS

LABEL ISN  ADDR   LABEL ISN  ADDR   LABEL ISN  ADDR   LABEL ISN  ADDR
10  13  000048   26  24  000410   36  24  000410   59  48  000554
70  51  000066   50  42  000554   53  48  000554   60  50  000608

COMPILER GENERATED LABELS

LABEL ISN  ADDR   LABEL ISN  ADDR   LABEL ISN  ADDR   LABEL ISN  ADDR
1000001  2  000010  1000002  13  000040  1000003  21  000040  1000004  29  000040
1000122  3  000010  1000123  31  000058  1000124  39  000058  1000125  47  000058
1000126  32  000020  1000127  44  000020  1000128  52  000020  1000129  60  000020
1000130  33  000038  1000131  41  000038  1000132  49  000038  1000133  57  000038
1000134  34  000048  1000135  46  000048  1000136  54  000048  1000137  62  000048
1000138  35  000056  1000139  43  000056  1000140  51  000056  1000141  59  000056
1000142  36  000064  1000143  44  000064  1000144  52  000064  1000145  59  000064
1000146  37  000072  1000147  45  000072  1000148  53  000072  1000149  59  000072

FORMAT STATEMENT LABELS

LABEL ISN  ADDR   LABEL ISN  ADDR   LABEL ISN  ADDR   LABEL ISN  ADDR
900  12  000024  1000  15  000044  1001  18  000044  1002  21  000044
1003  24  000057  1004  27  000057  1005  30  000057  1006  33  000057
1007  36  000064

*OPTIONS IN EFFECT=NAME(MAIN) OPTIMIZE(1) LINECOUNT(0) SIZE(MAX) AUTODATE(NONE)
*OPTICS IN =FFCT=SOURCE FRODIC NOLIST NODECK OBJECT MAP NOFORMAT NOGOSTHL XREF ALC NOANSF NOTERM IHM FLAG(1)
*STATISTICS SOURCE STATEMENTS = 75, PROGRAM SIZE = 1980, SUBPROGRAM NAME =CRD2IN
*STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPIIATION *****

2-KB BYTES OF CODE NOT USED
**FORTRAN CROSS REFERENCE LISTING**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>ADD</th>
<th>NAME</th>
<th>TYPE</th>
<th>ADD</th>
<th>NAME</th>
<th>TYPE</th>
<th>ADD</th>
</tr>
</thead>
<tbody>
<tr>
<td>KOUNT</td>
<td>SF</td>
<td>000000</td>
<td>IXLCNT</td>
<td>SF</td>
<td>000004</td>
<td>NEUDECK</td>
<td>SF</td>
<td>000000</td>
</tr>
</tbody>
</table>

**COMMON INFORMATION**

NAME OF COMMON BLOCK: NSBIRL

SOURCE STATEMENT LABELS

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ISN</th>
<th>ADDH</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>0000EF</td>
</tr>
</tbody>
</table>

COMPILER GENERATED LABELS

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ISN</th>
<th>ADDH</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>7</td>
<td>0000AC</td>
</tr>
</tbody>
</table>

- OPTIONS IN EFFECT: NAME(MAIN), OPTIMIZE(1), LINECOUNT(80), SIZE(MAX), AUTOBL(NONE)
- OPTIONS IN EFFECT: SOURCE EBCDIC, NOLIST, NUDECK, OBJECT MAP, NOFORMAT, NOGUSTMT, XREF, ALC, NUANS, NOTERM, IBM FLAG(1)
- STATISTICS: SOURCE STATEMENTS = 18, PROGRAM SIZE = 404, SUPPROGRAM NAME = CHUHP
- STATISTICS: NO DIAGNOSTICS GENERATED

**** END OF COMPILATION ****
OPTIONS IN EFFECT: NAME (MAIN) OPTICS (1) LINECOUNT (16) SIZE (MAX) AUTHOR (NONE)
SOURCE PL/CODING N/A LISEX EC OBJECT MAP FOR CSV N/A Bootstrap: NAME IMM FLG (1)

ISN 0007

---

SUBROUTINE GTMAP (PIXMAP, MOFS)

REAL PIxMAP
INTEGER MOFS

---

HISTORY

CAW AKERS
W AY TOWNS
LESCO

---

PURPOSE

THIS ROUTINE PROCESSES PIXEL MAPS FROM UNIVERSAL TAPES IN
SIM-PIXFL FORM (X) SIM-PIXELS X 301 LINES). THE CROP
CODES HAVE EFFECT ON THE SIM-PILXEL COUNT IN VALUE
ON SYM-CIC ACCORDING TO USFW SPECIFIED OPTIONS.

EXTERNAL REFERENCES

CROPP = EVALUATES SIM-PIXFL CROP CODES FOR PIXEL CODE

EXCEPTIONS

NONE.

LOCAL DECLARATIONS

INTEGER LEND
POINTER TO LINE END IN BUFFER
INTEGER ILAIJ
PIXEL # ADJUSTMENT FACTOR (ADVANCE FOR
EACH PAIR OF PIXELS)
INTEGER ULC
SIM-PIXEL CANDIDATE FOR PIXEL
INTEGER INTC
PIXEL COUNT OF SIM-PIXEL PER LINE
INTEGER KNP
CROP CODE
INTEGER KMPVL
CURRENT U LAST VALID CROP CODE
INTEGER NSHMAP
SIM-PIXELS TO ORCH PXEL
INTEGER KNPVL
CROP CODE SELECTED

PROCEDURE

---

LASSIL = IXLHUF
ISN 0004
CDTMAP/I XLHUF(1000), IXLHUF(1000), IXLHUF(1000)
ISN 0007
CDTMAP/NSHILXILRHUF(4)

READ 3 LINES INTO BUFFER

ISN 0005
L = 1640,540
ISN 0006
L = L + 53
ISN 0007
READ(1), ([400,400], IXLHUF(K), K = L:LEND)
ISN 0008
1000 PRINT(3,1000)
ISN 0009
CONTINUE

ISN 0010
ILAIJ = ILAIJ + MOFS
ISN 0011
CONTINUE

FOR EACH PIXEL IN LINE

ISN 0012
KMPVL = 1
SUBROUTINE INPAR(LETTERS, IFAM)

INTEGER LETTERS,LERROR,ICOL)
INTEGER IFAM
LEFT CORNER(S) OF OUTPUT
ERROR FLAG
0 - NO ERROR(S)
1 - ERROR(S)

HISTORY

MARY TOMPKINS  LEMSCU  10/27/80  ORIGINAL CODE

PURPOSE

READ AND INTERPRET CARDS CONSISTING OF A PAIR OF NUMBERS SEPARATED
BY COMMAS.

EXTERNAL REFERENCES

NONE.

EXCEPTIONS

IF A NON-NUMERIC CHARACTER OTHER THAN A BLANK WHICH IS
IGNORED OR A COMMA WHICH IS USED AS A SEPARATOR IS
FOUND A MESSAGE IS ISSUED AND IFAM = 1.

LOCAL DECLARATIONS

INTEGER KARD  CARD HEAD WORD AT A TIME
INTEGER KARM  CARD KID CHARACTER AT A TIME
INTEGER TYPE  TYPE OF OUTPUT REQUESTED
INTEGER KOUT  KIND OF OUTPUT
INTEGER UNIT  POWER OF 10 UNIT
INTEGER NUMER  DIGIT VALUE OF CHARACTER
INTEGER INTUP  CHARACTER DIGITS

DATA (BLANK/1/,COMMA/1/,IND*/SEND/)

ZERO FILL LEFT CORNER ARRAY

D) IS J = 1
D) VERY I = 1
D) J = 1
D) I = 1
D) SECONDARY CONTINUE
D) CONTINUE
D) READ CARD AND PLACE CARD IN REHEAD BUFFER
D) DO 200 INROW = 1,N
D) READ I(1),I = 1,20
D) FORMAT(2,284)
D) REMIND 3A
D) IF(30,1000,KARD(I)),I = 1,20
D) REMIND 3A
D) IF CARD IS NOT SEND ASSUME IT CONTAINS A PAIR OF NUMBERS
EXAMINE EACH CHARACTER. THE FIRST NUMBER STARTS WITH THE FIRST NON-HALAN NUMERIC CHARACTER AND ENDS WHEN A COMMA IS FOUND. THE SECOND NUMBER STARTS WITH THE FIRST NON-HALAN NUMERIC CHARACTER AND ENDS WHEN THE LAST NUMERIC CHARACTER IS FOUND.

IF (CHAR(1).EQ.,"L`) GO TO 100
IF (CHAR(1).NE.,"OMA") GO TO 20

CHAR IS A COMMA SECOND NUMBER FOLLOWS,
UNIT = 1
ICNL = 2
GO TO 100

CHECK FOR NON-NUMERIC CHARACTERS
IF (ICHAR(1).GE.,INT(1).AND.ICHAR(1).LE.,INT(10)) GO TO 40
WRITE (1,1002)
FORMAT (1,IX) ERROR ON CARD DEFINING UPPER LEFT CORNER OF *.
*OUTPUT*

IF (FMT = 1) GO TO 900

CHARACTER IS NUMERIC
DO 50 J = 1,10
   NUMBER (J) = I-J
GO TO 60

IF (ICHAR(1).EQ.,"O.MA") GO TO 60
CONTINUE

LITCN (INROW.INCOL) = LITCRS (INROW.INCOL)*UNIT + NUMBER
UNIT = 10
CONTINUE

END
SUBROUTINE KUSKLAL(MOFFS)

INTEGER MOFFS
OUTPUT OFFSET

HISTORY
-----
CARL KLIFES
MARY TOMPKINS

11/21/80

ORIGIN 1 CODE (PDP)

REDO (IBM)

PURPOSE
-----

REFERENCE TO A CODE CROP TRANSFORMATION SYMBOLS OR NUMERIC

VALUES OF PIXEL MAPS FROM UNIVERSAL TAPES IN PIXEL FORM

(196 X 177).

EXTERNAL REFERENCES
-----

NONE.

EXCEPTIONS
-----

NONE.

LOCAL DECLARATIONS
-----

INTEGER KALVY
INTERNAL TAPE CALIBRATION BYTE + OFFSETS

INTEGER XLNUM
PIXEL NUMBER

PROCEDURE
-----

LOGICAL ILBHUF
COMMON/XLBUF(3060)*XLINE(98),XALTNS(256)

READ ONE LINE OF SEGMENT

READ(I1,1001)(ILBHUF(I),I = 1,360)

1000 FORMAT(21/360A1)

GET PIXEL TRANSFORMATION FOR OUTPUT

KALVY = 72 + MOFFS

DO 100 I = 1,98

[XLNUM = KALVY + 1]

[XLINF(I) = XALTNS(I)*ILBHUF(I)*XLNUM]

[FI]ILBHUF(I),XLNUM, Eq,0),XLINE(I) = 0

100 CONTINUE
RETURN
END

*****FORT RAN CROSS REFERENCE LISTI N (*****

SYMBOL INTERNAL STATEMENT NUMBERS

MUFFS 0005 0006 0007 0008 0009 0010 0011

ILBHUF 0003 0004 0005 0010 0011

XLINF 0004 0010 0011

XALTNS 0004 0010

XLINE 0004 0010
FUNCTION NUMR(KHAR, NUMBER, ICOL)

INTEGER KHAR
INTEGER CARD (READ CHARACTER AT A TIME
INTEGER NUMBER
INTEGER ICOL
COLUMN IN CARD TO START DECODE

EXTERNAL REFERENCES

EXTERNAL REFERENCES

EXCEPTIONS

EXCEPTIONS

1. IF NO VALID NUMBER IS FOUND NUMR = 0.

LOCAL DECLARATIONS

LOCAL DECLARATIONS

PRICE FUNCTION

PRICE FUNCTION

LOOP UNTIL CHARACTER IS NON-NUMERICAL IGNORING BLANKS.

LOOP UNTIL CHARACTER IS NON-NUMERICAL IGNORING BLANKS.

SYMBOL LISTING:

SYMBOL LISTING:

**FORTRAN CROSS REFERENCE LISTING**

**FORTRAN CROSS REFERENCE LISTING**
SUBROUTINE SHARPPIXELMASK (IRLINE)

HEALPIXPRO  P I X E L S  P R O C E S S F U L
INTEGER MOFFS  O U T P U T  O F F S E T
INTEGER ILINE  S T A R T I N G  L I N E
INTEGER FIN  C O R R E C T I N G  L I N E

HISTORY
-------

CARL SKLERS, LFC  1/21/80  ORIGINAL CODE (POP)

NAME: SHARPPIXELMASK
AUTHORS: P. DE SOUZA

PURPOSE
-------

THIS ROUTINE PROCESSES SUM-PIXEL MAPS FROM UNIVERSAL TAPES
TO SUM-PIXEL FORM (2190, 14290, 3117 LINES). THE GRID

EXTERNAL REFERENCES
--------------------

NONE.

EXCEPTIONS
----------

NONE.

LOCAL DECLARATIONS
---------------------

INTEGER ILIN  L I N E  R E F E R R S  T O  (14290) / 1 7 7 7 2 0 9 7 ,
INTEGER PALYRT  C A L C U L A T I O N  I N F O R M A T I O N  O  F  O F F S E T
INTEGER ILINUM  C U R R E N T  P I X E L  N U M B E R
INTEGER KROM  C H R O P  C O D E
INTEGER KROMVL  C U R R E N T  O N  L A S T  V A L I D  C H R O P  C O D E

PROCEDURE
----------

ISN 0001  LOGICAL 1, ILHKU
CALL SYS/MP/ILHKU(3060), ILHLINE(948), ILHKR(256)

STARTING LINE IS NOT 1 SO FIND IT

ISN 0005  IFILS.NF.1, ILHLINE.EQ.11) GO TO 100
ISN 0007  ILYN = ILINE - 113
ISN 0009  IF (I Y N . N E . 1 I) GO TO 100
ISN 0010  IF (ILHKU(j,j) = 1, 460) 100  CONTINUE
ISN 0011  10 CONTINUE

ISN 0012  POSITIONS AT REQUESTED LINE

ISN 0014  IF (ILHKU(j,j) = 1, 460) 100  READ(11), ILHKU(1), I = 1, 460

SET PIXEL TRANSFORMATION FOR OUTPUT IF TAPE WAS WRITTEN FOR POP
WHICH USES THE EIGHTH BIT AS A SIGN BIT REQUIRING AN ADJUSTMENT
**FORTRAN CROSS REFERENCE LISTING**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>INTERNAL STATEMENT</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPMAP</td>
<td>SIZE OF PROGRAM</td>
<td>000257</td>
</tr>
<tr>
<td>NAME</td>
<td>TAG</td>
<td>TYPE</td>
</tr>
<tr>
<td>NAME</td>
<td>TAG</td>
<td>TYPE</td>
</tr>
<tr>
<td>NAME</td>
<td>TAG</td>
<td>TYPE</td>
</tr>
</tbody>
</table>

**COMMON INFORMATION**

NAME OF COMMON BLOCK: MAP

<table>
<thead>
<tr>
<th>VAR. NAME</th>
<th>TYPE</th>
<th>REL. ADDR.</th>
</tr>
</thead>
</table>

SOURCE STATEMENT LABELS

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ISN</th>
<th>ADDR</th>
</tr>
</thead>
</table>

COMPILER GENERATED LABELS

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ISN</th>
<th>ADDR</th>
</tr>
</thead>
</table>
APPENDIX C
JOB CONTROL SOFTWARE
FILE: SODAP.EXE  LANGUAGE: FORTRAN UNIVERSITY

CONTROL OFF
SODAP.EXE

HISTORY
----------
MARY TOMPSON 11/24/83 ORIGIHAL CODE

METHOD
--------
THIS EXE IS USED TO EXECUTE THE SODAP PROGRAM. THE ACCEPTABLE
ARGUMENTS TO THE EXE ARE:

ARGUMENT   DATA ON DISK   DATA ON TAPE
-----------   ------------   ------------
  A1        SYMBOL FILE NAME   SYMBOL FILE NAME
  A2        SYMBOL FILE TYPE   SYMBOL FILE TYPE
  A3        CC FILE NAME      CC FILE NAME
  A4        INPUT DATA FILENAME TAPE NUMBER
  A5        DATA FILE TYPE    FILE # ON TAPE
  A6        DATA FILE NAME    TAPE DE RITY (OPTIONAL)

BOTH CC FILE AND SYMBOL FILE ARE REQUIRED TO BE ON THE USERS A DISK.

EXCEPTIONS
-------
ERROR MESSAGES FOLLOWED BY PROGRAM TERMINATION ARE FOR
FOLLOWING REASONS:
1. NO TEMPORY DISK AVAILABLE.
2. WRONG PARAMETER COUNT.
3. NON-ALPHABETIC TAPE NAME.
4. UNSUCCESSFUL TAPE MOUNT.
5. UNEXPECTED INPUT.

PROCEDURE
---------
TAG IFY PRINTED INSTRUCTIONS.
SODAP PRINTS COMMAND LISTED TO RSCS
GLOBAL TITLE CMSLHN FORTMERICAN
CP QUERY VIRTUAL 192
LIF AREACOD NE 0 GFIDISK TEMP 2M CLEAR
CP QUERY VIRTUAL 192
LIF AREACOD NE 0 ATYPE NO TEMP DISK AVAILABLE.
LIF AREACOD NE 0 EXIT 1
  ASKP 3
  ATYPE SYMBOL FILE: A1 A2 A
  ATYPE CC FILE: A3 A4 A
  ATYPE INPUT DATA: A5 A6 A
  ATRACT = A000

  CHECK FOR ACCEPTABLE PARAMETER COUNT.
LIF A1NEOF 0 AQUITO -TAPE
LIF A1NEOF A AQUITO -CONT
ATYPE TOO MANY - TOO FEW INPUTS
EXIT 2

  CONT LIF A4 0 0R0 AQUITO -TAPE
LIF A4 0 AQUITO -DISC
ATYPE A4 NOT THE INPUT EXPECTED.
EXIT 3

  DATA IS ON TAPE TEST LIF TAPE 'NAME' IS NUMERIC
  -TAPE ATEST = AQUITO A4
  LIF ATEST 0 0R0 AQUITO 3
  ATYPE ILEGAL TAPE NUMBER A4

  DATA IS ON DISC - TEST LIF TAPE 'NAME' IS NUMERIC
  -DISC ATEST = AQUITO A4
  LIF ATEST 0 0R0 AQUITO 3
  ATYPE ILEGAL TAPE NUMBER A4
EXIT 4

IF A4 < NULL THEN SET TAPE4N = A4
ENDIF

IF TAPE HAS BEEN USED IN A PREVIOUS RUN A DISK FILE WILL
EXIST, DO NOT MOUNT AND TRANSFER DATA.

SAVE = COMCAT AS AS
STATE = DM DATA O
IF SERTCONF ME O SKIP A
A4 = AS
A5 = DATA
A6 = O
AGOTO -DISC

IF NO DISK FILE EXIST THEN MOUNT TAPE - TRANSFER TAPE FILE TO
DISK.

TAPE MOUNT A4 TAPE DO TAPE4N
IF SERTCONF ME O TYPE TAPE4N NOT MADE
IF SERTCONF ME O EXIT A
A1 = A5 = 1
TAPE MOUNT A4 TAPE
A2 = A5 = A6 AGOTO 1
TAPE END A4 (TAPE)
FILEDEF OSMOVE TAPE ( HLKSIZE 3060 NCFM U PERM DEN TAPE4N
FILEDEF OSMOVE DISK AS MDATA O (HLKFLC 3060 HLKSIZE 3060 NCFM U PERM
MOVFIL
DEF 141
A3 = AS
A4 = DATA
A5 = O

GENERATE FILEDEF:
UNIT 1 FILE DEFINITION
1 OUTPUT FROM CC FILE SYMBOL TABLE
2 INPUT DATA
3 INPUT CC FILE
4 INPUT SYMBOL TABLE
5 CAP OUTPUT
50 NEW ADD UNIT

-DISC -CONTINUING
FILEDEF FT15FO0 DISK AS A4 AS A6 (HLKFLC 3060 HLKSIZE 3060 NCFM U PERM
FILEDEF FT15FO0 DISK AS A3 AS A6 (HLKFLC 3060 HLKSIZE 3060 NCFM U PERM
FILEDEF FT15FO0 DISK OUT CC O (HLKFLC 131 HLKSIZE 131 PERM
FILEDEF FT15FO0 DISK OUT DATA O (HLKFLC 131 HLKSIZE 131 PERM
FILEDEF FT15FO0 DISK FILE FT15FO0 O (HLKFLC 131 HLKSIZE 131 PERM

BEGIN EXECUTION
LOAD SIMAP (CLEAN NO MAP) START
EXECUTION COMPLETE -- WRAP-UP
PRINT CC D IEC
PRINT OUT DATA D IEC
SPool PRINTP CLOSE
EXIT
END
<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td>B5</td>
<td>B6</td>
<td>B7</td>
<td>B8</td>
</tr>
<tr>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>C4</td>
<td>C5</td>
<td>C6</td>
<td>C7</td>
<td>C8</td>
</tr>
<tr>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>D7</td>
<td>D8</td>
</tr>
<tr>
<td>E1</td>
<td>E2</td>
<td>E3</td>
<td>E4</td>
<td>E5</td>
<td>E6</td>
<td>E7</td>
<td>E8</td>
</tr>
<tr>
<td>F1</td>
<td>F2</td>
<td>F3</td>
<td>F4</td>
<td>F5</td>
<td>F6</td>
<td>F7</td>
<td>F8</td>
</tr>
<tr>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>G5</td>
<td>G6</td>
<td>G7</td>
<td>G8</td>
</tr>
<tr>
<td>H1</td>
<td>H2</td>
<td>H3</td>
<td>H4</td>
<td>H5</td>
<td>H6</td>
<td>H7</td>
<td>H8</td>
</tr>
<tr>
<td>I1</td>
<td>I2</td>
<td>I3</td>
<td>I4</td>
<td>I5</td>
<td>I6</td>
<td>I7</td>
<td>I8</td>
</tr>
<tr>
<td>J1</td>
<td>J2</td>
<td>J3</td>
<td>J4</td>
<td>J5</td>
<td>J6</td>
<td>J7</td>
<td>J8</td>
</tr>
<tr>
<td>K1</td>
<td>K2</td>
<td>K3</td>
<td>K4</td>
<td>K5</td>
<td>K6</td>
<td>K7</td>
<td>K8</td>
</tr>
<tr>
<td>L1</td>
<td>L2</td>
<td>L3</td>
<td>L4</td>
<td>L5</td>
<td>L6</td>
<td>L7</td>
<td>L8</td>
</tr>
<tr>
<td>M1</td>
<td>M2</td>
<td>M3</td>
<td>M4</td>
<td>M5</td>
<td>M6</td>
<td>M7</td>
<td>M8</td>
</tr>
<tr>
<td>N1</td>
<td>N2</td>
<td>N3</td>
<td>N4</td>
<td>N5</td>
<td>N6</td>
<td>N7</td>
<td>N8</td>
</tr>
<tr>
<td>O1</td>
<td>O2</td>
<td>O3</td>
<td>O4</td>
<td>O5</td>
<td>O6</td>
<td>O7</td>
<td>O8</td>
</tr>
<tr>
<td>P1</td>
<td>P2</td>
<td>P3</td>
<td>P4</td>
<td>P5</td>
<td>P6</td>
<td>P7</td>
<td>P8</td>
</tr>
<tr>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
<td>Q4</td>
<td>Q5</td>
<td>Q6</td>
<td>Q7</td>
<td>Q8</td>
</tr>
<tr>
<td>R1</td>
<td>R2</td>
<td>R3</td>
<td>R4</td>
<td>R5</td>
<td>R6</td>
<td>R7</td>
<td>R8</td>
</tr>
<tr>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>S4</td>
<td>S5</td>
<td>S6</td>
<td>S7</td>
<td>S8</td>
</tr>
<tr>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
<td>T6</td>
<td>T7</td>
<td>T8</td>
</tr>
<tr>
<td>U1</td>
<td>U2</td>
<td>U3</td>
<td>U4</td>
<td>U5</td>
<td>U6</td>
<td>U7</td>
<td>U8</td>
</tr>
<tr>
<td>V1</td>
<td>V2</td>
<td>V3</td>
<td>V4</td>
<td>V5</td>
<td>V6</td>
<td>V7</td>
<td>V8</td>
</tr>
<tr>
<td>W1</td>
<td>W2</td>
<td>W3</td>
<td>W4</td>
<td>W5</td>
<td>W6</td>
<td>W7</td>
<td>W8</td>
</tr>
<tr>
<td>X1</td>
<td>X2</td>
<td>X3</td>
<td>X4</td>
<td>X5</td>
<td>X6</td>
<td>X7</td>
<td>X8</td>
</tr>
<tr>
<td>Y1</td>
<td>Y2</td>
<td>Y3</td>
<td>Y4</td>
<td>Y5</td>
<td>Y6</td>
<td>Y7</td>
<td>Y8</td>
</tr>
<tr>
<td>Z1</td>
<td>Z2</td>
<td>Z3</td>
<td>Z4</td>
<td>Z5</td>
<td>Z6</td>
<td>Z7</td>
<td>Z8</td>
</tr>
</tbody>
</table>