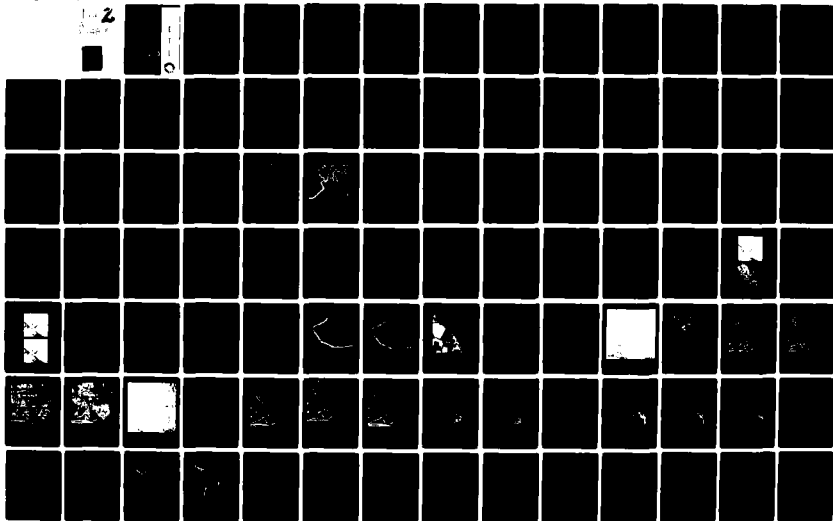


AD-A114 815

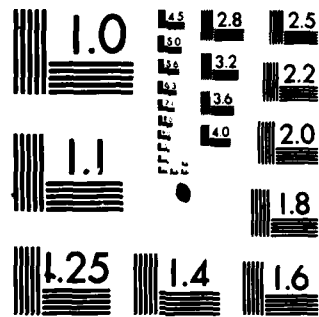
GENERAL ELECTRIC CO LANHAM MD SPACE SYSTEMS DIV F/8 9/2
INTERACTIVE DIGITAL IMAGE PROCESSING FOR TERRAIN DATA EXTRACTIO--ETC(U)
SEP 81 T F WESCOTT, W C DALLAM, C J PETERSON DAAK70-79-C-0183
ETL-0277 ML

UNCLASSIFIED

2



1148



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ETL-0277

2

Interactive digital image
processing for terrain
data extraction, phase 2

Thomas F. Wescott
William C. Dallam
Christopher J. Peterson

General Electric Company
Space Systems Division
4701 Forbes Blvd.
Lanham, MD 20706

AD A114815

SEPTEMBER 1981

DTIC
ELECTE
MAY 20 1982
S D E

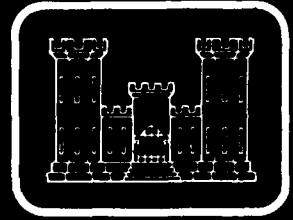
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Copy available to DTIC does not
permit fully legible reproduction

Prepared for 82 05 21 019

U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060

DTIC FILE COPY



E
T
L



Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official
Department of the Army position unless so designated by other
authorized documents.

The citation in this report of trade names of commercially available
products does not constitute official endorsement or approval of the
use of such products.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0277	2. GOVT ACCESSION NO. AD-A114 816	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) INTERACTIVE DIGITAL IMAGE PROCESSING FOR TERRAIN DATA EXTRACTION, PHASE 2	5. TYPE OF REPORT & PERIOD COVERED Final Report for Period Sept 1980 - Sept 1981	
	6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s) Thomas F. Wescott, William C. Dallam, and Christopher J. Peterson	8. CONTRACT OR GRANT NUMBER(s) DAAK70-79-C-0153	
9. PERFORMING ORGANIZATION NAME AND ADDRESS General Electric Company Space Systems Division 4701 Forbes Bldg Lanham, MD 20706	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62707A855C0006	
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, VA 22060	12. REPORT DATE Sept 1981	
	13. NUMBER OF PAGES	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Digital Image, Image Processing, Information Extraction, Interactive Analysis, Terrain Analysis, Water Resources Data Element Extraction		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Phase 2 of an experimental study has been performed to investigate the feasibility and use of man-machine interactive digital image processing techniques when applied to the extraction of terrain analysis data from aerial imagery. This phase of the study concentrated on applying techniques to extract water resources data elements and on the development and coding of software and algorithms. The developed software and algorithms, pertained to filtering of themes, and image scaling. Two digital processing		

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

techniques to "skeletonize" features, best mapped as lines, were established and tested with limited success. Relative to water resources data elements, success was achieved in the extraction of alignment of watercourses and shore line alignment of watercourses of water bodies. Partial success was achieved with the elements of delineation of wet areas and the identification of terminal points of watercourse segments. Image processing techniques for the extraction of areas subject to flooding and for the measurement of bank heights were not practical with the existing software.

The development/implementation of software, and the processing and analysis of imagery was conducted at the General Electric Digital Image Analysis Laboratory.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	23 P



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This document is generated under Contract DAAK70-79-C-0153 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060 by the General Electric Company, Space Division, Beltsville, Maryland. The Contracting Officer's Representative was Mr. Joseph H. Kitrosser.

TABLE OF CONTENTS

<u>Title</u>	<u>Page</u>
Preface	i
Illustrations	iv
I INTRODUCTION	1
A. Background	1
B. Objective	2
C. Scope	2
D. Water Resources Elements	3
II IMAGERY USED	4
A. Data Selection	4
B. Data Sets	4
1. Elizabeth City & North Carolina	4
2. Ripley, West Virginia Area	5
3. Clarion County, PA	5
4. Fort Belvoir/Woodbridge, VA Area imagery, USAETL, from Phase I Study.	6
5. Tennessee Valley Authority	6
III TECHNIQUES DEVELOPMENT	7
A. Image resolution and Scale Changing	7
1. Scale Changing via Pixel Replication	9
2. Fast Area Averaging.	11
3. Sub-pixel Interpolation for Image Enlargement	14
B. Feature Skeletonizing	18
1. Skeletonizing Features via Gradient Zero crossing	19
2. Skeletonizing Features via Golay Processing.	25
C. Theme Smoothing Method	31
D. Use of the Theme Filter.	40
1. Capabilities	40
Processing Example	41

TABLE OF CONTENTS (cont)

<u>Title</u>	<u>Page</u>
IV RESULTS	47
A. Extraction of Watercourse and Associated Elements .	47
1. Watercourses	47
2. Extracting Watercourses from Radar Imagery . .	52
3. Watercourse Extraction Using Combinations of Dissimilar Images	61
B. Extraction of Dry Gaps and Associated Elements . .	72
1. Dry Gaps.	72
2. Categorizing Watercourse by Drap Gap Width . .	73
3. Terminal Points of Watercourse Segments	78
4. Encoding Watercourse Delineation.	82
V CONCLUSIONS	85
VI RECOMMENDATIONS	87
APPENDIX A IMAGE RESOLUTION AND SCALE CHANGING PROGRAM . . .	A1
APPENDIX B THEME FILTERING PROGRAM	B1

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Legend</u>	<u>Page</u>
1	Image Scaling Program Flow Diagram	10
2	Scale Change by Replication	12
3	Fast Area Averaging	15
4	Four Point Context Dependent Interpolation	17
5	Typical Watercourse Theme (Purposely Left Rough to Provide a Difficult Test)	20
6	Watercourse Mask Where Connectivity Has Been Established by Slicing a Low Pass Filtered Image	21
7	Processing Sequence for Skeletonizing	23
8	Watercourse Skeletonizing by Gradient on Spatially Simple Stream	24
9	Unsuccessful Attempt at Watercourse Delineation By Zero-Crossing Line of Maximum Local Gradient. Note that discontinuities are introduced only when watercourse is oriented at 10 o'clock.	26
10	Golay Skeletonizing of Watercourse Theme	27
11	Golay Skeletonizing With Updated Images	29
12	Watercourse Delineation by Skeletonizing the Mask of Figure 6	30
13	Theme Filter	35
14	Theme Filter Data Structure	36
15	Theme Filter Operator Interface	38
16	Theme Filter Replacement Logic	39
17	Comparison of Fort Belvoir/Woodbridge Vegetation Themes (Top) with Manual Analysis Results (Bottom)	42
18	18A Theme Filter Results 18B Simplified Themes	44
19	Outlines of Vegetation and Water Themes	45

LIST OF ILLUSTRATIONS (CONT.)

<u>Figure</u>	<u>Legend</u>	<u>Page</u>
20	Outlines of Simplified Themes	46
21	Watercourse Extraction via Level Slicing the 1-Meter Resolution Panchromatic Photographic Image	49
22	Water Identification via Joint Use of the 1-Meter Resolution Panchromatic Photographic Image and the Derived Texture Image (smoothed)	50
23	Watercourse Extraction via Level Slicing the 4-Meter Resolution Panchromatic Photographic Image	51
24	Watercourse Identification via Joint Use of the 4-Meter Resolution Panchromatic Photographic Image and the Result of the Image Minus 7 x 7 Pixel Local Average	53
25	X-Band Horizontal Polarization Subscene	54
26	X-Band Vertical Polarization Subscene	54
27	L-Band Horizontal Polarization Subscene	54
28	L-Band Vertical Polarization Subscene	54
29	SAR Data and Subscene Locations	55
30	Water Theme Sliced from X-Band SAR w/Horizontal Polarization	56
31	Water Theme Derived from X-Band SAR w/Vertical Polarization	57
32	Water Theme Derived from X-Band SAR w/Horizontal Polarization	58
33	Water Theme Derived from L-Band SAR w/Vertical Polarization	59
34	X-Band Horizontal Polarization Subscene (Same as Figure 1)	60
35	X-Band Horizontal Polarization Data Smoothed by a 7x7 Pixel Moving Average Operator	60
36	Associated Microtexture Image	60
37	Microtexture Image Data Smoothed by a 7x7 Pixel Moving Average Operator	60

LIST OF ILLUSTRATIONS (CONT.)

<u>Figure</u>	<u>Legend</u>	<u>Page</u>
38	Water Theme Derived from w/Dimensional Partitioning of X-Band SAR w/Horizontal Polarization and Associated Microtexture Image Data	62
39	Water Theme Derived from 2-Dimensional Partitioning of X-Band SAR w/Horizontal Polarization Data and Textured Data Smoothed by a 7 x 7 Pixel Moving Average Operator . .	63
40	Theme Derived from the 7 x 7 Pixel Smoothed Version of the X-Band SAR Image w/Horizontal Polarization	64
41	Little River Water Course Theme Derived from Panchromatic Photography and SAR Data and From the Microtexture Version of the Photography	65
42	Smoothed Version of Figure 41	66
43	Edged Version of Figure 42	67
44	Chapel Creek Water Course Theme Derived from Panchromatic Photography and SAR Data	68
45	Chapel Creek Water Course Theme Derived from Panchromatic Photography and SAR Data and From the Smoothed Microtexture Version of the Photography	69
46	Smoothed Version of Figure 45	70
47	Edged Version of Figure 46	71
48	Manually Interpreted Water Course	74
49	Manually Interpreted Dry Gap	75
50	Skeletorized Water Course	76
51	Skeletorized Dry Gap	77
52	Categorizing a Dry Gap By Width	79
53	Dry Gap Categorized by Width and Displayed as Polygons . .	80
54	Combining Watercourse Center Line with Dry Gap Categories Performing a Logical Itersection Produces Water Course Segements Categorized by Width	81

LIST OF ILLUSTRATIONS (CONT.)

<u>Figure</u>	<u>Legend</u>	<u>Page</u>
55	Raster Operating Line Dotting Algorithm	83
56	Encoding Watercourse Delineation	84

I. INTRODUCTION

A. Background

During the period of September 1979 through September 1980, Phase 1 of the experimental study, "Interactive Digital Image Processing for Terrain Data Extraction", was performed. The 12-month study investigated the feasibility and utility of man-machine interactive digital processing techniques when applied to the extraction of terrain analysis data from aerial imagery. The study focused primarily on the extraction of vegetation data elements from digitized panchromatic photography, with a small amount of attention given to thermal infrared and side-looking radar imagery.

Of thirteen vegetation data elements listed in the USAETL Terrain Analysis Procedural Guide for Vegetation, eight were addressed in varying degrees of depth. Interactive digital techniques were developed for vegetation/land cover boundary extraction and for the extraction of several forest-related data elements. For the forest-related elements, three digital themes were developed from digitized panchromatic photography. The three digital themes and a developed automatic operation allowed estimates of percent canopy closure, stems per hectare; crown diameter; stem diameter; stems per hectare per diameter class; and stem spacing to be produced. Best results were achieved with a spatial resolution of approximately three feet in the input digital image.

In addition, techniques were developed enabling the extraction of vegetation/land cover boundaries. These techniques were most effective when the spatial resolution of the digital image is in the range of 8 to 20 feet. With a moderate amount of interactive analysis, processing can be employed to extend the boundary extraction rapidly and efficiently to relatively large areas. Procedures were also developed for distance, elevation, and area measurement from digital imagery. Further testing and development of new techniques were recommended as a result of Phase 1 work.

B. Objective

The objective of the investigation was to continue the study of man-machine digital image processing techniques to the extraction of terrain analysis data from aerial imagery.

C. Scope

The scope of the second phase part of the study involved the extraction of data identified in the Terrain Analysis Procedural Guide for Drainage and Water Resources. Within the scope was the development and coding of software and algorithms for filtering themes and image scaling. Input data for the second phase included panchromatic, thermal infrared and side looking synthetic aperture radar imagery.

D. Water Resources Elements

In this second phase, the water resources elements considered were:

1. Alignment of watercourses
2. Shore alignment of waterbodies
3. Delineation of wet areas
4. Identification of areas subject to flooding
5. Identification of terminal points of watercourse segments
6. Measurement of bank heights

II. IMAGERY USED

A. Data Selection

Data types used for Phase 2 consisted of two forms, i.e. digital and imagery. Imagery selected as transparencies were digitized into 512 x 512 pixel images (subscenes) with 8-bit gray level quantization. Certain data sets used in phase 1 were applied in Phase 2, specifically pertaining to Fort Belvoir/Woodridge, Va. area and Daedalus scanner images of Pennsylvania. Additional panchromatic photography, synthetic aperture radar (SAR), thermal infrared (TIR) and infrared (IR) data were acquired from the Tennessee Valley Authority (purchased) and General Electric Space Division (loaned) at Valley Forge. Selection of data used for analysis considered the following factors:

- Scene content (various watercourses and water bodies)
- Resolution (watercourses of 1-15 meters width)
- Absence of clouds, cloud shadows and haze
- Uniformity of scene feature brightness levels across the image
- Near-vertical view of the scene
- General quality of the transparency (i.e. absence of scratches, etc)

B. Data Sets

A list of all data sets used for the generation of selective 512 x 512 pixels images (subscenes) is shown in Table 1.

Table 1

1. Elizabeth City, North Carolina; imagery; USAETL

a. Knobbs Creek Area

- (1) 98° synthetic aperture radar #1 imagery
- (2) 52° synthetic aperture radar #9 imagery
- (3) Panchromatic #39 imagery
- (4) High resolution thermal flightlines 2, 3 and 4

Table 1 (Cont.)

- b. Little River Area
 - (1) 98° synthetic aperture radar #5 imagery
 - (2) 98° synthetic aperture radar #9 imagery
 - (3) 52° synthetic aperture radar #5 imagery
 - (4) 52° synthetic aperture radar #10 imagery
 - (5) 278° synthetic aperture radar #5 imagery
 - (6) High resolution thermal infrared #9 imagery
 - (7) Low resolution thermal infrared imagery
 - c. New Begun Creek Area
 - (1) 98° synthetic aperture radar #10 imagery
 - (2) 278° synthetic aperture radar #6 imagery
 - (3) Panchromatic #95 imagery
 - (4) Low resolution thermal infrared imagery
 - d. Chapel Creek Area
 - (1) 278° synthetic aperture radar #14 imagery
 - (2) Panchromatic #91 imagery
 - (3) Panchromatic #120 imagery
 - e. Symond Creek Area
 - (1) Panchromatic #47 imagery
 - (2) High resolution thermal infrared imagery
2. Ripley, West Virginia Area, digital, General Electric Company
- a. X-band horizontal polarization
 - b. X-band vertical polarization
 - c. L-band horizontal polarization
 - d. L-band vertical polarization
3. Clarion County, Pa, digital, from Phase 1 study
- a. 20,000 ft. altitude; green and thermal infrared channels
 - b. 10,000 ft. altitude; green and thermal infrared channels
 - c. 5,000 ft. altitude; green and thermal infrared channels

Table 1 (Cont.)

4. Fort Belvoir/Woodbridge, Va. Area, imagery, USAETL, from Phase 1 study

a. Panchromatic photographs

5. Tennessee Valley Authority

a. Holston River set

- Thermal infrared (TIR) low resolution (flight line)
- Five panchromatic photographs of flight line area

b. Night TIR (low air temperature)

- Six panchromatic photographs

c. Reservoir

- TIR
Five panchromatic photographs

III. TECHNIQUES DEVELOPMENT

A. Image Resolution and Scale Changing

As reported in the first phase of this project, spatial operators, which include the reported microtexture and crown template operators, are very useful tools for the interpretation of monochromatic imagery. The program of successfully categorizing monochrome imagery is in general a much more difficult program than with other types, such as multispectral, because of the relatively smaller amount of information available. Attempting to categorize an image by level slicing alone encounters serious problems, the most difficult being that signatures tend not to be unique enough to be used as the sole descriptor of the classes generally desired. Even seemingly simple classes like water bodies can be impossible to extract via level slicing. Clearly more information than gray level is required to perform categorization, and clearly such information is available, in the form of spatial patterns.

In particular, textural information is of considerable value in the analysis of high resolution images. Texture is an encompassing term with a multitude of types and definitions, most of which have limited use in remote sensing. The most basic and perhaps most widely applicable type of texture is simply surface roughness, which can be relatively easily determined by computing local image variance. Textural operations are scale dependent and for effective use, must be applied at a specific scale which is defined by the size of the microfeatures which generate the rough appearance of larger features being extracted. For example, a typical image might contain three features which we wish to separate, pasture, brush land, and forest. Each of the features is basically a random mixture of vegetation and shadow, and quite likely all three exhibit approximately the same integrated reflectance. Each of the features also exhibit similar textural structure, the appearance of random roughness, but they are distinctly different in textural scale.

To realize the full potential of scale dependent spatial techniques we must be able to efficiently vary the size of the spatial operators used (or at least the effective size), allowing the extraction of a

considerable range of spatial features. This scaling can be accomplished by either of two methods: (1) by scaling the operator itself, which is the historic method, or (2) by changing the scale and resolution of the image and applying a universal operator. The image scaling approach appears to have several significant advantages:

- Scaling of all but the simplest of spatial operators is not as straight forward as is reducing resolution and scaling an image. Any resolution up to that inherent in the original data can easily be simulated with very simple and rapidly executed algorithms. Allowing, for example, the highly effective and reasonably sized max-min microtexture operator to be used without modification, in the extraction of a wide variety of textural identifiable features.
- Matching spatial feature templates to objects of a particular size can be done much more precisely by changing image scale, as finer scaling increments can be used. Many effective spatial operators cover areas as small as 3 x 3 pixels, thus the finest scale adjustment would be a factor of 1/3.
- The reduction in the computation time required to apply an equivalent operation is significant when image scaling is chosen. Reduction in resolution means reductions in the volume of data processed. Reducing image size by a factor S reduces the number of pixels to be considered by S^2 , compared with increasing the operator size, which would increase most spatial computations by S^2 . The relative difference approaches S^4 .

During Phase I of this study, image scaling was accomplished by utilizing those techniques which were readily available at General Electric DIAL without software development. While the usefulness of image scaling to expand the utility of standard and simple spatial operators was firmly established, the actual methods were cumbersome and time consuming.

The ability to vary image scale and resolution rapidly enough for use in interactive analysis was quickly identified as being a missing and necessary tool.

Given the task of using spatial information as an input for the categorization of some image feature, a fairly lengthy procedure might be followed. First the image would be reduced in scale and resolution by some factor defined by the predominant spatial frequency of the object to be extracted. For example, the mean distance between shadows in forest canopy. A spatial operator is then applied to the reduced image generating a derived image where the spatial feature has been extracted or enhanced. This secondary image is then enlarged to the scale of the original for subsequent analysis. Since the derived image contains information of a nature independent from the spatial grey level information of the original, this operation essentially creates a multidimensional feature space which significantly increases an analyst's options when attempting to categorize a monochrome image. Repeating the operation at several scales generates an even more interesting feature space of higher dimensionality.

Based on the experience gained during the Phase 1 analysis of several images, a program was written to perform image scaling by any of three operator selected interpolation methods, nearest neighbor, area averaging, and a fairly unique subpixel interpolation method based on local scene context. Figure 1 is a flow chart of the program.

1. Scale Changing via Pixel Replication

Often simple replication of pixels is a satisfactory or even desirable method of changing the scale of an image. Whether or not it should be the chosen method is dependent on the specific image processing task being performed. For example, if one desires to quickly examine an 'overview' of a large image containing too much data for complete display on a CRT, interpolative resampling with its associated computation could be a waste of time. In other cases, such as resampling an image with a very small scale change, no noticable degradation will be suffered with this technique. Finally, there are many applications where original radiometric values must be preserved. In these cases replication or

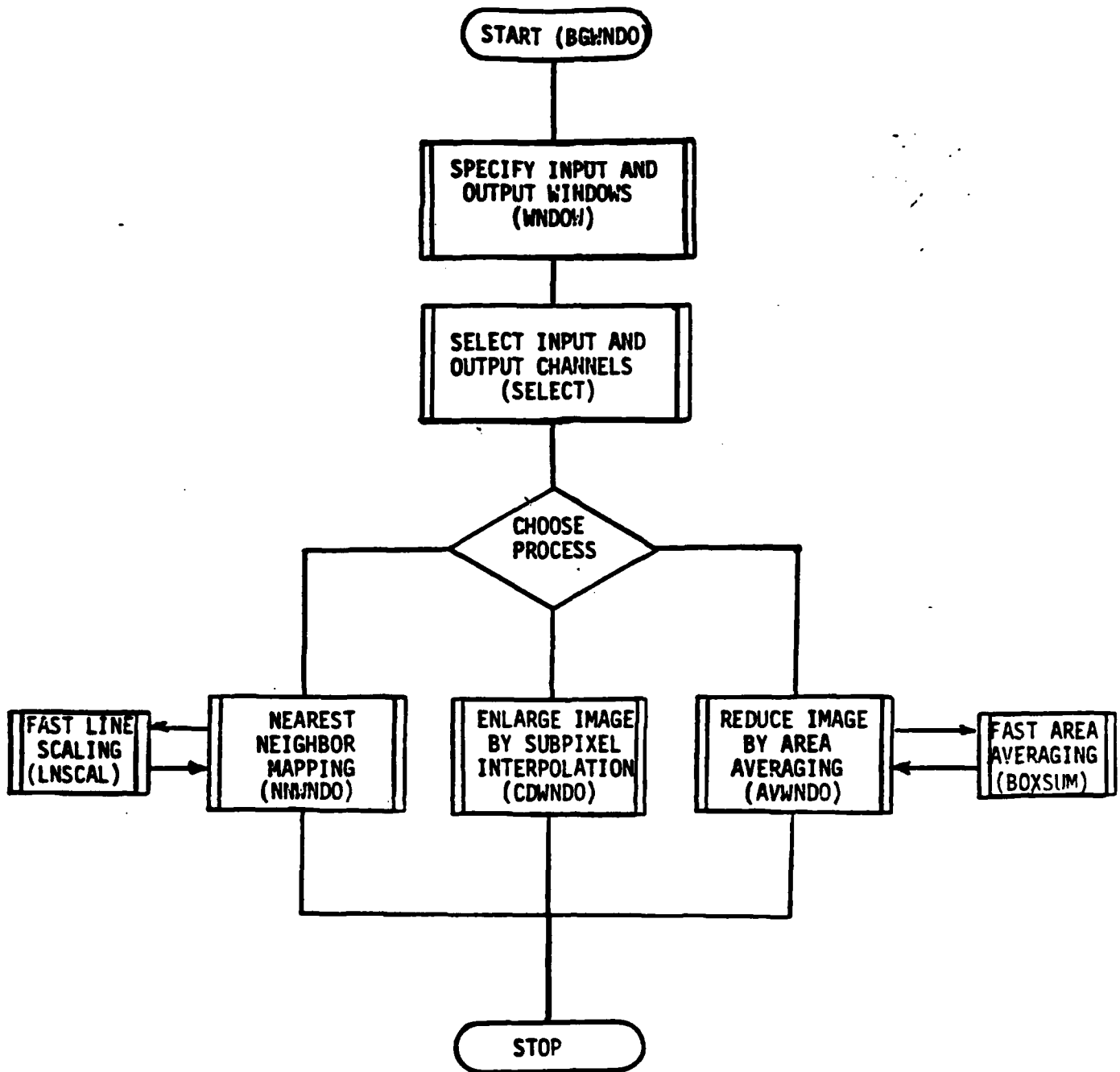


Figure 1. Image Scaling Program Flow Diagram

nearest neighbor resampling is the only simple method. Examples are, when sensor characteristics are being extracted, or when thematic or binary images, which are not gray scale in nature must be scaled.

Pixel replication sounds like a trivial problem, but the capability is so basic and heavily used in image processing, that a serious look at implementation and execution time is worthwhile. In several recent image processing systems the implementation is via hardware and operations are performed in near time. For other systems, a software solution is very reasonable, as rapid execution can be achieved.

Since no real pixel by pixel computation is required, either in determination of output pixel value, or in determination of output position, algorithms can be used which amount only to moving data in a high speed manner from a predetermined location to a predetermined destination. This is accomplished by setting up two address tables based on the replication or sampling required along lines and between lines. These tables, one with an element for each output line and one with an element for each output pixel, contain addresses in the input image. Each of these addresses is then the source location for each desired output pixel.

Using these tables, the actual execution is performed by indirect addressing. Data movement can be done by several methods including special purpose hardware and replacement statements from a high level programming language. In this case replacement was done in an assembler routine to minimize execution time. Omitting I/O limitations, the program takes approximately three seconds to construct a 512 x 512 pixel output image. Figure 2 shows the algorithm used.

2. Fast Area Averaging

This capability is fundamental for the extraction of scaled spatial information from high resolution images in the manner described in Phase 1. Starting with an image digitized and stored at the highest desired resolution area averaging is used to generate a secondary image of "super pixels" where digital numbers are related to the integrated

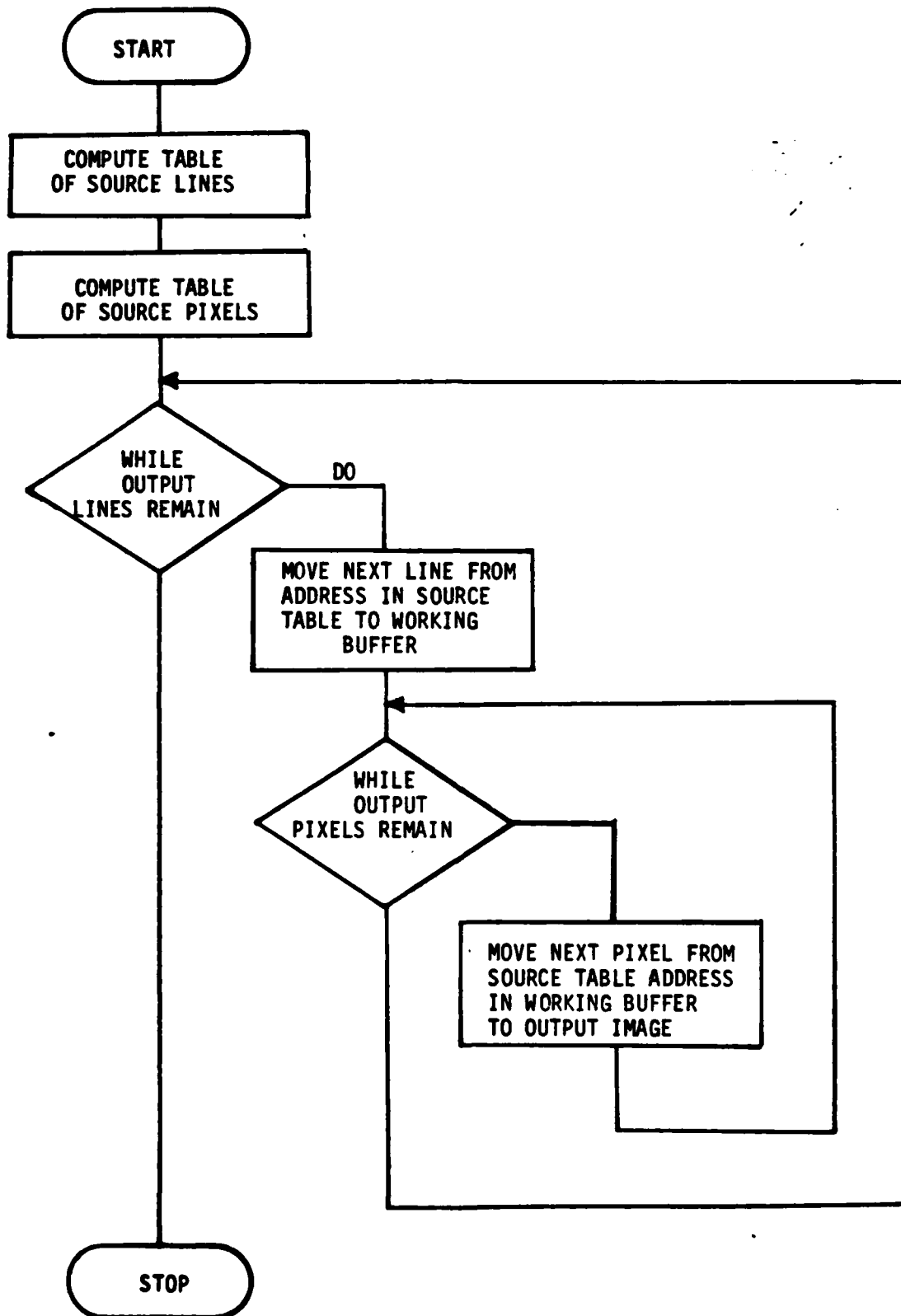


Figure 2. Scale Change by Replication

reflectance from a large number of original pixels. The spatial frequency content of the derived image is cut off by the span of the averaging area, simulating an image that might result if the same sensor and objects were used in a lower resolution (higher sensor altitude) configuration.

In use, as the averaging area increases, the observable texture of finely structured areas disappears from the derived image. Textural information can then be extracted by differencing the derived image with the original in a manner similar to a technique widely used for edge extraction. Area averaging is then applied to the derived difference image as a smoothing function which generates a grey level image where the digital number is proportional to the local high frequency content (texture) cut off by the original application of the filter.

The approach chosen to implement the filter was to use an adaption of the highly efficient box filtering technique. Box filtering computes a moving average over a rectangular area by using two linear moving averages, one down image columns and one along image rows. Execution is very efficient. If the box size is small in relation to the image size, the number of operations required to compute one output pixel approaches four, and is independent of box size. While box filtering computes more output pixels than are needed for scale changing, it is unlikely that any other algorithm can approach this efficiency. The scale changing program computes all of the column averages used in box filtering, but computes row averages only for the required output lines. Required output pixels are stripped from the row average at regular intervals. All along row operations are done in assembler subroutines to minimize execution time.

As with all spatial operations, some trickery must be used to avoid undesirable image border effects without reducing the size of the output image. Since, for our application, the derived images are often to be recombined with their full resolution parents, maintaining image size is very important. This need is particularly acute if large filters are used. Data must, in effect, be manufactured so that the moving averages

do not contain pixel values of zero at image edges. The method chosen as being the least objectionable, was to increase input image size by reflecting data values about image borders. Figure 3 shows the algorithm for fast area averaging.

3. Sub-pixel Interpolation for Image Enlargement

Selection of an interpolation algorithm for use in image enlargement is not quite as straight forward as many recent papers on the subject would lead us to believe. Truncated or otherwise modified versions of the "theoretically perfect" ($\frac{\sin x}{x}$) interpolation have been used to achieve good interpolation with a minimum of computation. However, some interesting artifact producing properties are made evident when we use these interpolators as an enlarging tool. In fact, making an enlargement is an excellent means of graphically showing many of the problems associated with various algorithms.

The often used and much discussed "cubic convolution," for example, which computed an interpolated value based on 16 surrounding pixels produces a curious effect on edges with certain orientations. When the point to be interpolated falls directly on one of the lines defined by the pixel grid of the original image, this algorithm performs admirably. In these cases, the algorithm degenerates to a one-dimensional operation. When the interpolation falls between the lines, two-dimensional interpolation is performed and considerable aliasing can occur. When the original image contains an edge feature oriented at 45° to the pixel grid, the aliasing is most apparent. The optimum ($\frac{\sin x}{x}$) interpolation would produce a blurred (from enlargement) but straight edge in the output image. The cubic convolution reproduces the 45° edge with a wobbly sine wave appearance, showing that individual interpolated values are as much a function of resampling position as they are of image content.

The other commonly discussed interpolator, bilinear interpolation, has been shown in numerous studies to perform just a little worse than cubic

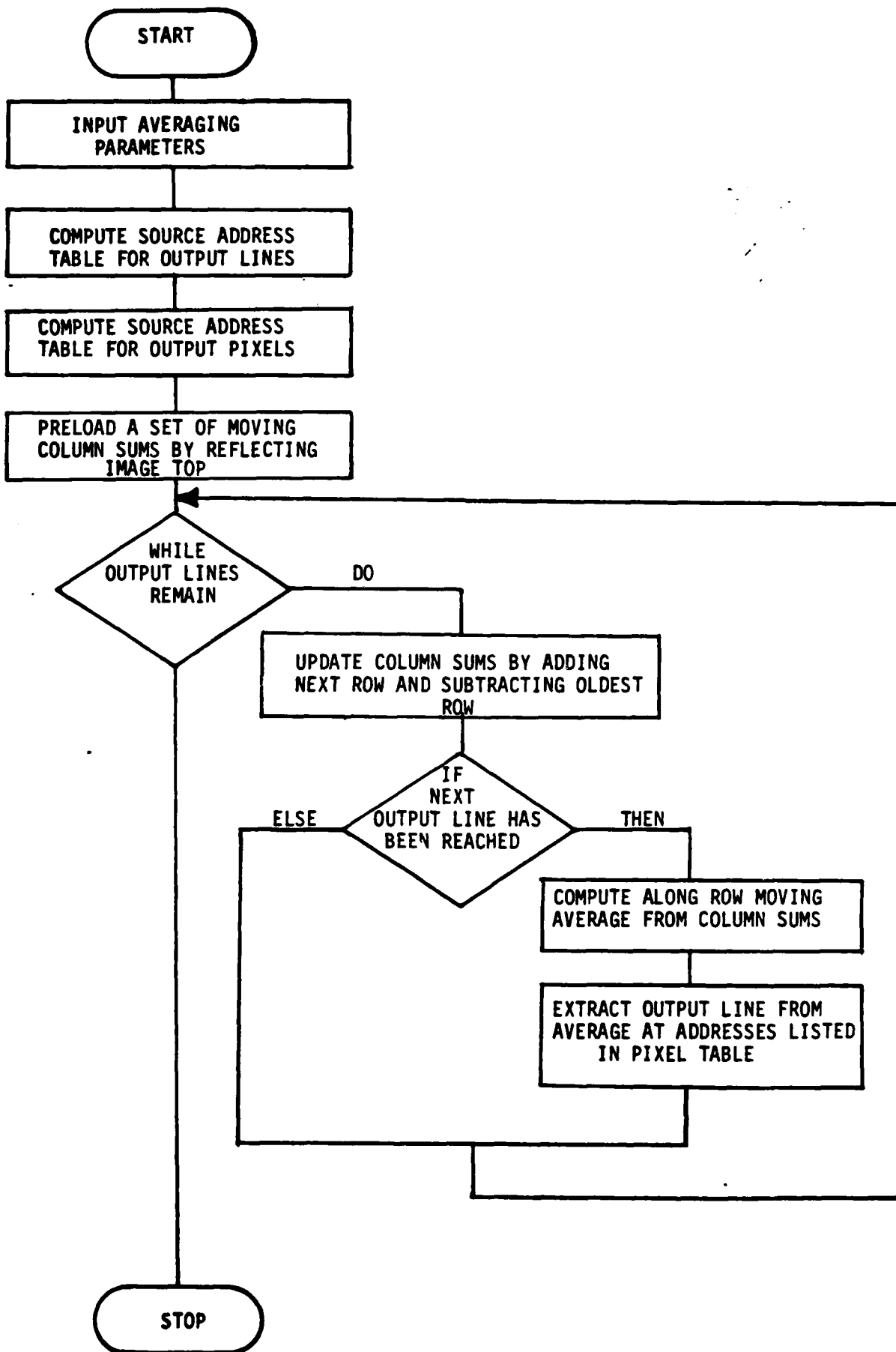
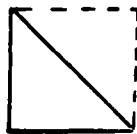
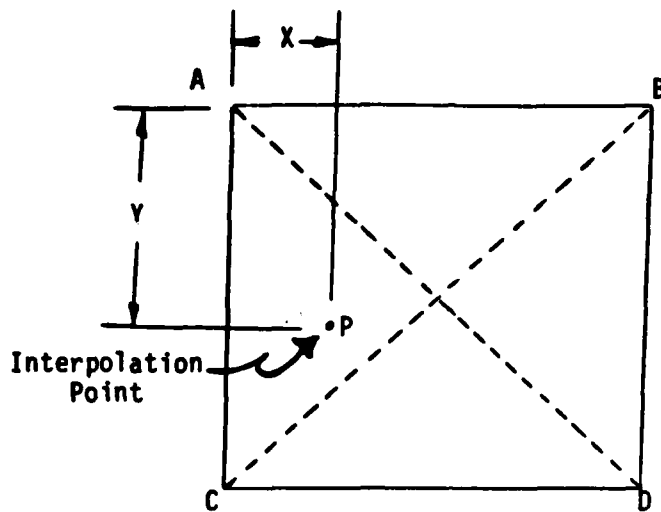


Figure 3. Fast Area Averaging

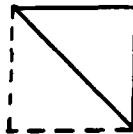
convolution in just about every respect. True to form, when subjected to a visual "enlargement test" it shows 45° artifacting a little more severe than cubic convolution does. Considering that it only requires four points, the algorithm does not perform all that poorly. Bilinear interpolation basically performs linear interpolation along two opposite sides of a quadrilateral defined by four pixels, and then performs a third interpolation between these two new points. It then, is based on the assumption that all four points lie in the same plane, a gross assumption which is incorrect whenever the four points are over an image edge.

The interpolation scheme chosen was a four point interpolation which breaks the quadrilateral defined by the four surrounding points into triangles. The most appropriate triangle for interpolation is chosen by the direction of the local image gradient and by the position of the desired point. The interpolation is then controlled by image content and handles 45° edge situations well. See Figure 4.

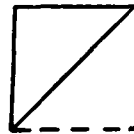
The context dependent interpolation has been coded and tested at GE DIAL and performs as was expected. The 45° artifacts are completely eliminated. The interpolator does have some remaining problems which could ultimately show up in factor analysis. Angular artifacts remain at $22\frac{1}{2}^{\circ}$ to the pixel grid and isolated points are reproduced as diamond shapes. Further use in the generation of factor overlays should determine the value of this method. It is likely that some combination of a linear interpolators and a context dependent interpolator would produce better results.



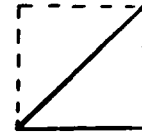
#1



#2



#3



#4

FOUR POINT CONTEXT DEPENDENT INTERPOLATION

STARTING WITH INTENSITY VALUES A,B,C,D SURROUNDING THE DESIRED POINT LOCATION, SELECT FROM THE FOUR DEFINED TRIANGLES, THE ONE INCLUDING BOTH THE POINT LOCATION AND THE DIAGONAL WITH THE SMALLEST INTENSITY GRADIENT. LINEAR INTERPOLATION IS DONE ON THE PLANE DETERMINED BY THE SELECTED TRIANGLE.

- CASE #1 IF (($|C-B| \geq |A-D|$) .AND. ($Y \geq X$))
 $P = A + X(D-C) + Y(C-A)$
- CASE #2 IF (($|C-B| \geq |A-D|$) .AND. ($Y < X$))
 $P = A + X(B-A) + Y(D-B)$
- CASE #3 IF (($|C-B| < |A-D|$) .AND. ($X \leq (1-Y)$))
 $P = A + X(B-A) + Y(C-A)$
- CASE #4 IF (($|C-B| < |A-D|$) .AND. ($X > (1-Y)$))
 $P = C - X(D-C) + (1-Y)(B-D)$

Figure 4

B. Feature Skeletonizing

At the end of Phase 1, another capability seen as very useful for the extraction of several terrain elements was the ability to transform those elements characterized by filament type objects into single pixel width lines. Two obvious examples of elements requiring the operation are the delineation of watercourses and roads. There are really two different levels to the program. The first case, where the feature is spatially connected i.e. no breaks, is relatively easy to handle by using a line stripping algorithm to reduce feature width to the point where only one strip of adjacent pixels remain, bisecting the original feature. In general, however, themes are not always well connected. As the desired feature nears the resolution limit of the imagery, or is hard to categorize for some other reason, the feature is usually mapped as a tenuous string of pixels spaced with many gaps. Any widely applicable skeletonizing algorithm should be able to bridge at least small gaps and then shrink the feature to produce the single pixel width map.

The problems are demonstrated vividly when one attempts to extract watercourse alignment. As described in the element extraction section of this report, several methods of extracting large watercourses are feasible and success was achieved with most of the images investigated. The most widely applicable methods rely on image intensity slicing in combination with the unique smoothness or lack of texture that water bodies exhibit to most sensors. Extracting the alignment of very narrow watercourses is a different problem as there is not enough resolution available to make use of texture or any other positively identifying features. While the watercourses can be roughly identified by density slicing, it is difficult to classify them as contiguous objects. Thin or intermittent watercourses are often revealed to the photo interpreter, not by the presence of clearly identifiable water, but only by subtle changes in the surrounding vegetation, or as a linear shadow with no water being imaged.

Prior to a skeletonizing operation, connectivity of the feature must be established if we are to work successfully with narrow watercourses and roads. Some encouraging results in this area were obtained by performing a very low pass filtering operation (box filtering) on unconnected binary theme maps. Filtering in this manner generates a slowly varying grey level output where losses in connectivity have been "bridged" by the slowly varying intensity ramps. The operation transforms strings of isolated pixels into a series of peaks and cols. Level slicing on the smoothed image results in a map much wider than the original feature, and one where connectivity has been established across gaps of width up to the size of the filter. The method has two undesirable effects. First the width of the feature is increased by up to twice the span of the filter size, making subsequent skeletonizing by line stripping methods a more lengthy process. Second, the filtering introduces a straightening of the feature centerline. The straightening can become pronounced and objectionable when filter size approaches the dimensions of feature characteristics such as stream meanders. Figure 5 shows a sample watercourse theme and Figure 6 is the slice where minor gaps have been bridged.

1. Skeletonizing Features via Gradient Zero Crossing

Several attempts were made to utilize linear spatial operators to locate feature centerlines. The results were less than satisfactory, and many problems remain to be addressed, but the use of differentiating operators on grey level images generated from smoothed themes, does present some interesting possibilities.

A binary watercourse theme is first changed by low pass filtering, into a continuous grey level image with a continuous first derivative. In this smoothed image, the watercourse center is characterized by a local intensity maximum which defines a ridge line curve where the local gradient taken perpendicular to the watercourse direction changes sign as it passes through zero. Zero crossing is a uniquely descriptive feature which is independent of the watercourse width, and also independent of the gray level magnitude corresponding to the watercourse. As long as the smoothing operator has established some non zero grey level

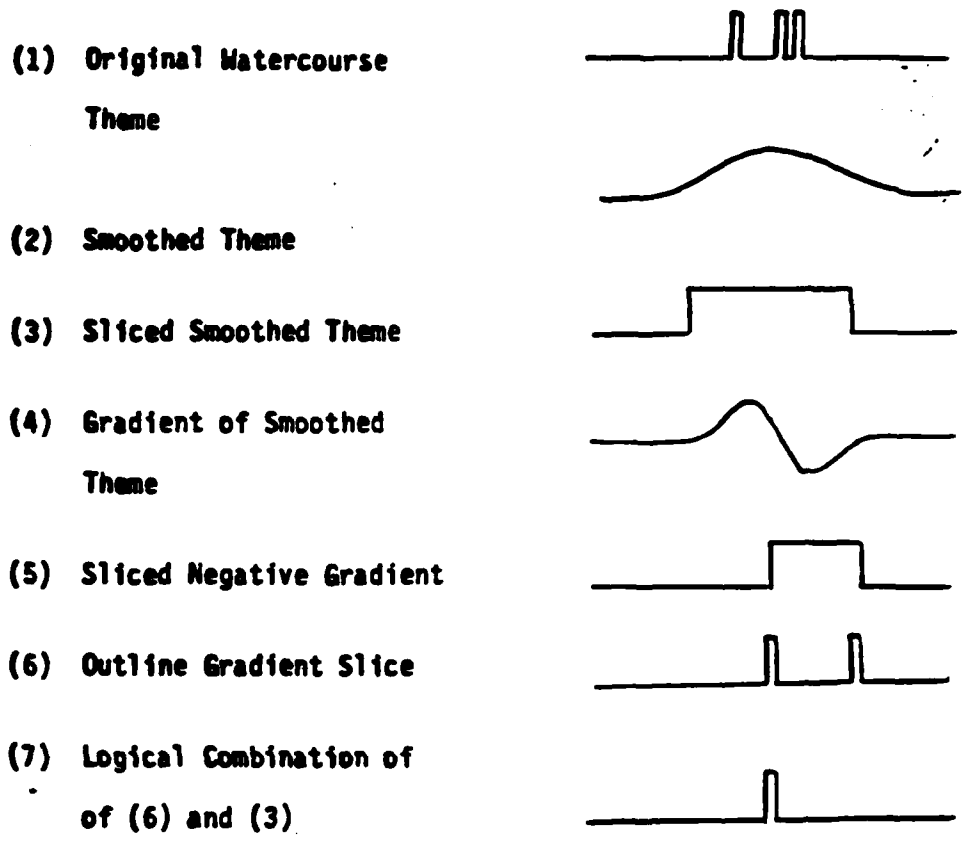
connection between watercourse pixels, a gradient zero crossing line will exist and can be located. Mapping the zero crossing can be achieved by level slicing all negative (or positive) pixel values and then outlining the area mapped by the slice. Figure 7 is a schematic of the processing sequence.

The zero crossing line has some very attractive features:

- As long as the filter size is greater than one half the watercourse width, gradient zero crossing will occur at the watercourse centerline.
- The zero crossing line tends to establish and maintain connectivity across small gaps in the watercourse.
- The line can be quickly and easily mapped into a single pixel width feature.

Despite the promise of this method, only limited success was achieved when applied to real images. The method was useful on spatially simple watercourses i.e. those without meanders, where the entire stream could be projected in some direction with a one to one mapping, allowing the use of a single gradient direction, as is shown in Figure 8. The method could be used on similar features with limited curvature. Whenever the range of curvature direction was greater than a semicircle, the method was unsuccessful. The problem reduces to one of preserving some of the properties of a direction sensing operator while making the operator work over the image in a directionally invariant manner. There is likely a solution which would make the method universally applicable, but during this project, time limitations brought an end to the investigation.

Most of the obvious adaptations of the gradient method were tried. Taking gradients from two directions and then combining the result into an image where grey scale was related to gradient magnitude failed. In general connectivity was not well preserved, and it was lost at every stream junction point. Likewise, treating orthogonal directions independently failed. This approach produced results which were not totally coincident and thus could not be recombined to form a single centerline.



(A single pixel width
Watercourse alignment map)

Reduction of an unconnected watercourse theme to a single pixel width watercourse alignment map by using low pass filtering and zero crossing of gradient.

Figure 7. Processing Sequence for Skeletonizing



Figure 8. Watercourse Skeletonizing by Gradient
On Spatially Simple Stream

The most interesting approach, which also failed, was to use a rotating gradient operator. With a 3 x 3 pixel moving operator four gradients arranged in a semicircle were extracted. Output of the operator was the signed magnitude of the gradient possessing the greatest absolute value. The idea was to always extract the gradient in a direction perpendicular to the watercourse so that the zero crossing line would be preserved, even when the gradient sign was reversed. Unfortunately the operator generated unacceptable discontinuities whenever the watercourse direction was oriented perpendicular to the line defining the semicircle. See Figure 9.

2. Skeletonizing Features via Golay Processing

Golay processing is a purely spatial logical operation which is applied to a binary theme map. For each pixel position, the operation examines pixels immediately adjacent to the central pixel, and adds or deletes the central pixel from the output based on the configuration of its adjacent neighbors. Golay transforms were meant to operate on images arranged in a hexagonal array where each pixel has six neighbors. There are several significant advantages to this arrangement when compared to the more commonly used rectangular array. These advantages are: fewer pixels are needed for spatial operators, between neighbors interpixel distances are constant, and a higher order of symmetry exists. Few interactive systems, however, work on hexagonal arrays, making implementation on these systems complex. At CE DIAL, the transforms have been applied in a compromising manner directly to a rectangular array, resulting in a capable program but also one which has some directional artifacts. Figure 10 shows the Golay patterns as implemented at DIAL.

Given a connected watercourse theme, Golay processing is a reasonable means of shrinking the watercourse to the single pixel width line needed to identify alignment. Skeletonizing is accomplished by deleting from the watercourse theme, all pixels with surrounds numbers three, four, and five as labeled in Figure 10. The operation basically removes the outside annular ring of pixels from every contiguous area and thus narrows the feature by two pixels per pass.

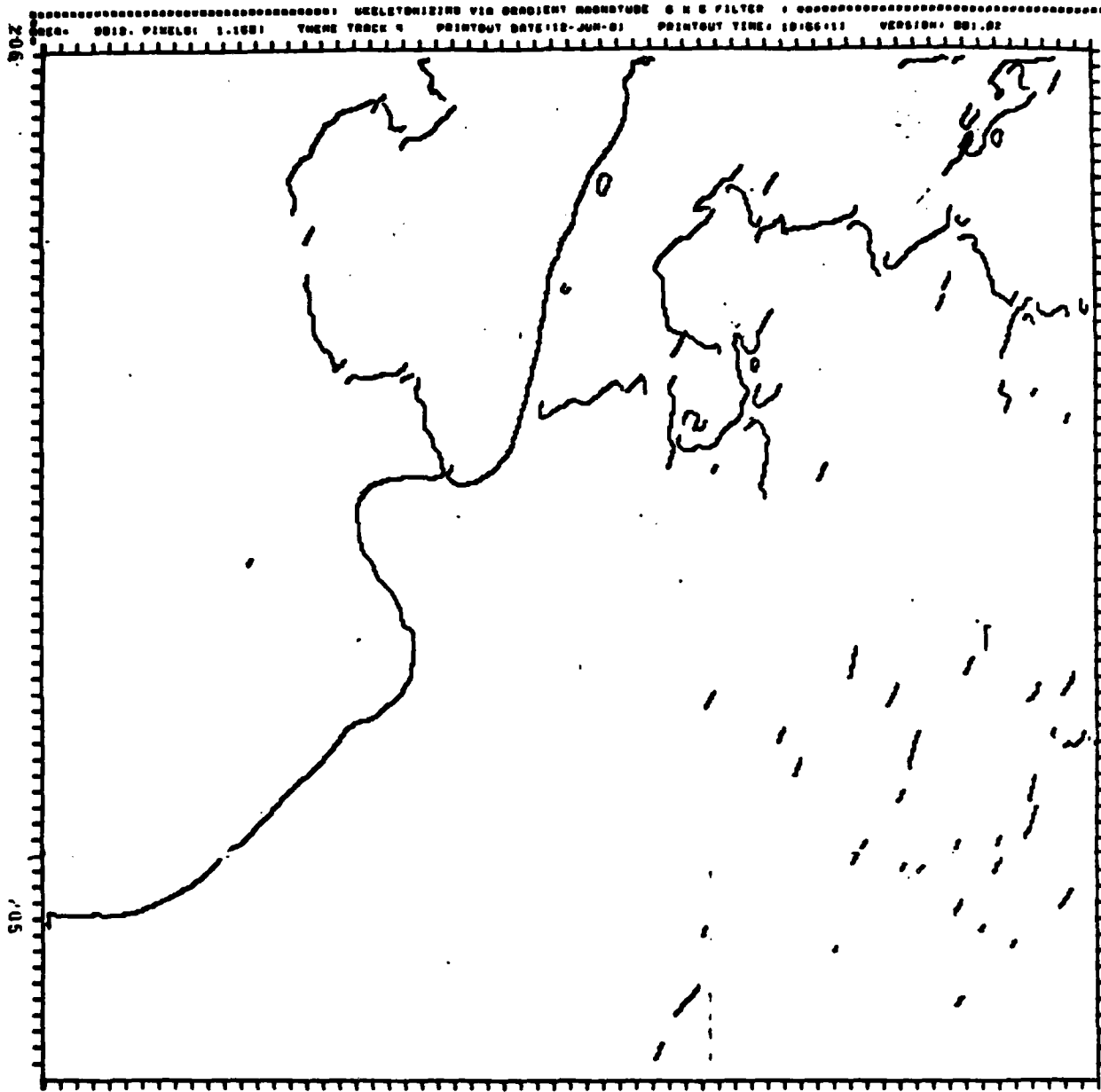
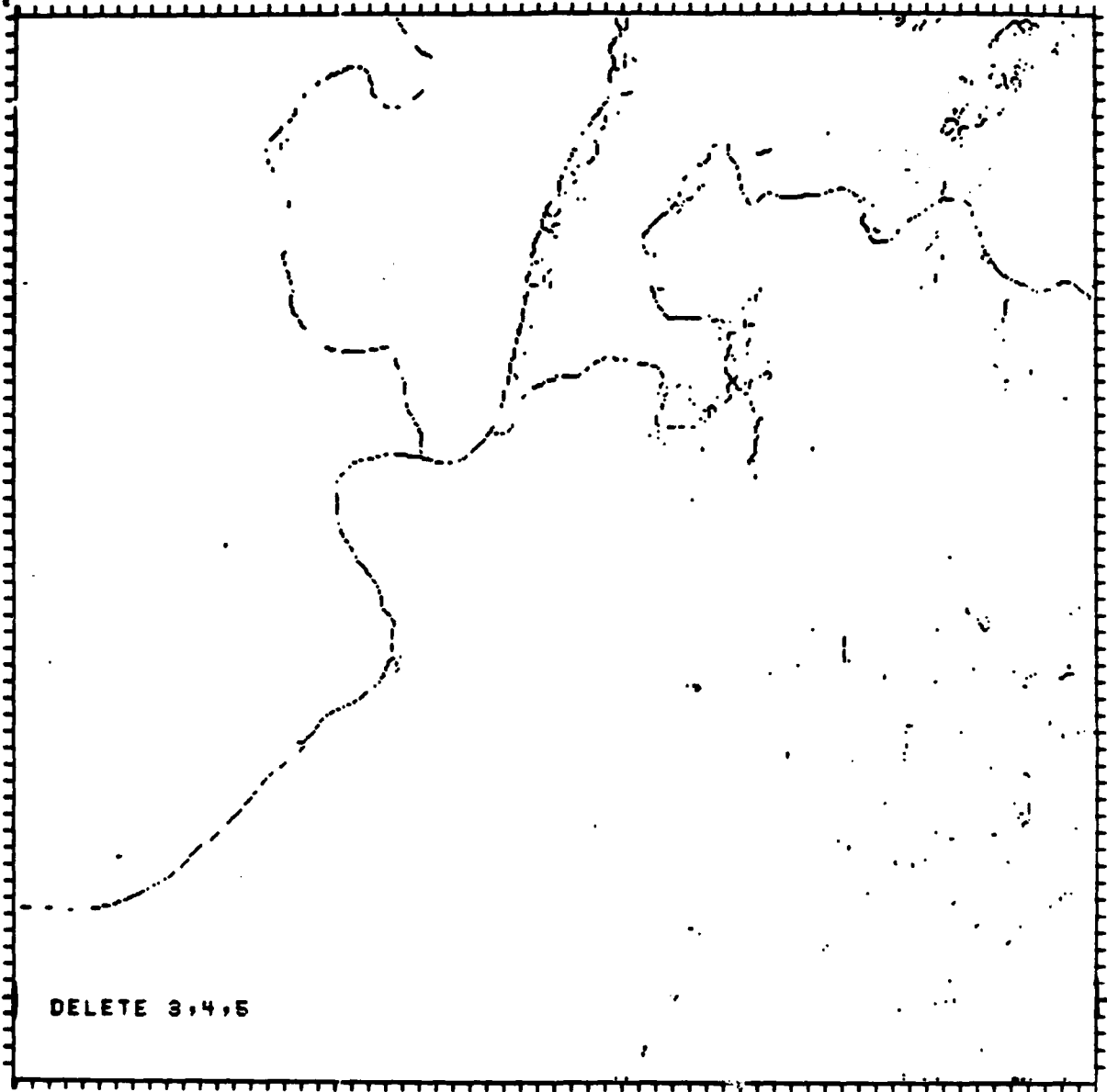


Figure 9. Unsuccessful Attempt At Watercourse Delineation
 By Zero Crossing Line Of Maximum Local Gradient.
 Note That Discontinuities Are Introduced Only When
 Watercourse Is Oriented At 10 o'clock.



DELETE 3,4,5

THE 14 GOLAY SURROUNDINGS ARE BY NUMBER:

00	10	11	11	11	11	11
0X0	0X0	0X0	0X1	0X1	0X1	1X1
00	00	00	00	01	11	11
01	02	03	04	05	06	07
10	11	11	11	10	00	11
0X1	0X0	0X0	0X1	0X1	1X1	0X0
10	01	10	10	00	00	11
08	09	010	011	012	013	014

Figure 10. Golay Skeletonizing of Watercourse Theme

A problem with the method results from the small neighborhood sampled by the Golay surround, this or any other rotationally invariant operator of 3 x 3 pixels or less, cannot sense the difference between the boundary of a large contiguous area, and a narrow linear feature two pixels wide. The result can be deletion of pixels from both sides of the narrow feature and thus loss of connectivity. Figure 10 shows the same watercourse skeletonized in this manner. There are ways to stop this action. For example, slight smoothing of the feature, a technique which has already been recommended to establish connectivity, can also be used to introduce curvature into long straight boundaries minimizing occurrence of the malfunction. Other techniques, such as slightly randomizing feature boundaries have also been used.

Alternatively, a modification of the process avoids the problem, but introduces other effects. The inability to recognize lines of two pixels width, is avoided if the process is changed so that image raster lines being processed are subjected to immediate updating. With updating, whenever an output pixel is changed, the surround of the subsequent pixel includes that change. The compromise with this approach is that the resulting single pixel width line is no longer centered on the feature. The line is instead shifted to the side of the feature which was last processed. Even with this undesirable effect, the technique might be of value for use on narrow watercourses where the induced offset is small. One very attractive feature of this modification is that two passes over the image are usually sufficient which can mean a significant reduction of execution time. Figure 11 shows a sample result.

Based on the approaches to skeletonizing investigated, the use of Golay pattern transforms of similar operations is viewed as the more promising means to achieve the delineation of narrow watercourses. The properties of rotational invariance, moderate execution time, and a versatile selection of operations make it useful. The main deficiency of the approach is that watercourse gaps of more than one pixel cannot be easily bridged, but this can be overcome by filtering. Figure 12

MODIFIED GOLAY SKELETONIZING

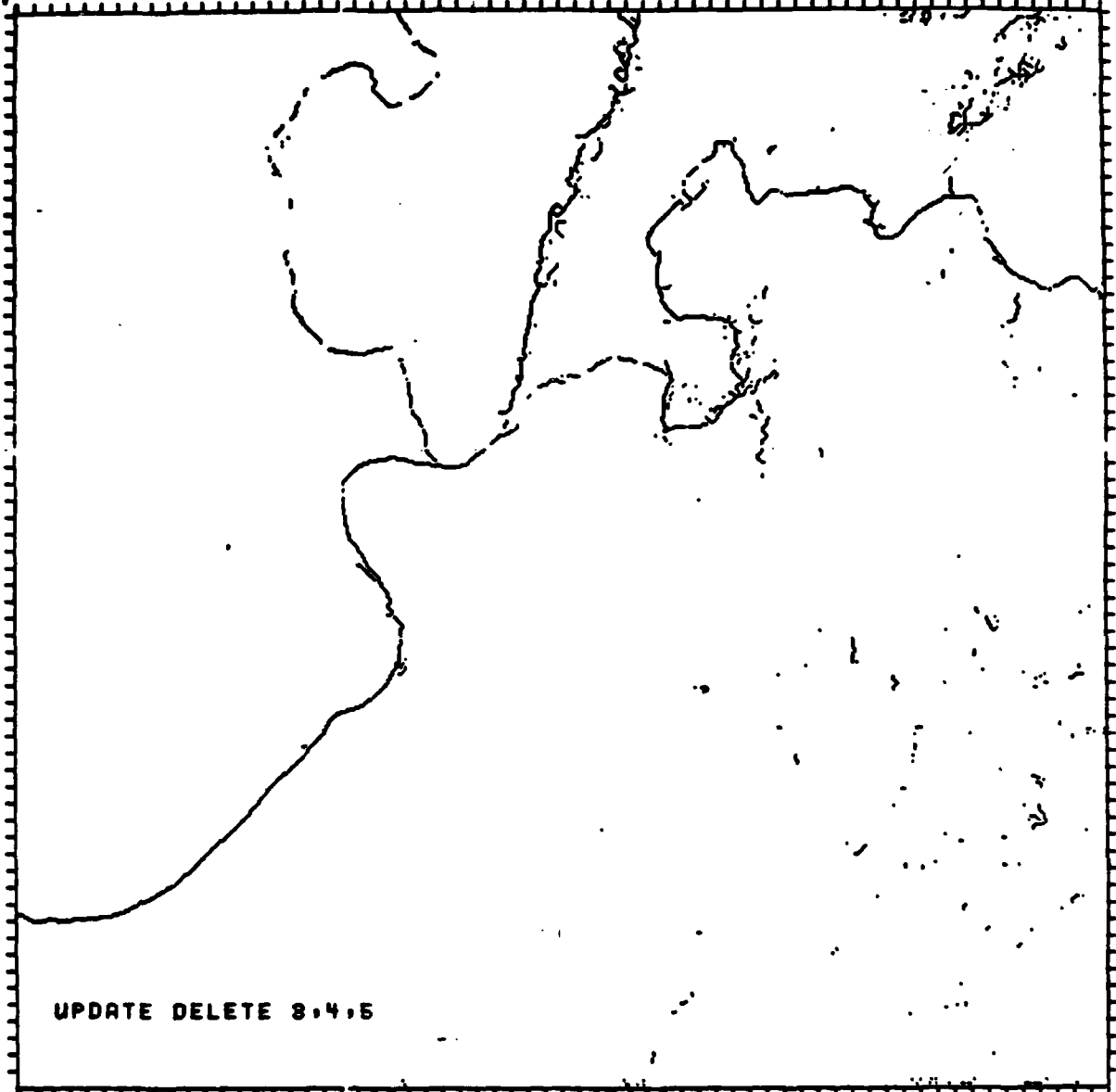


Figure 11. Golay Skeletonizing With Updated Image

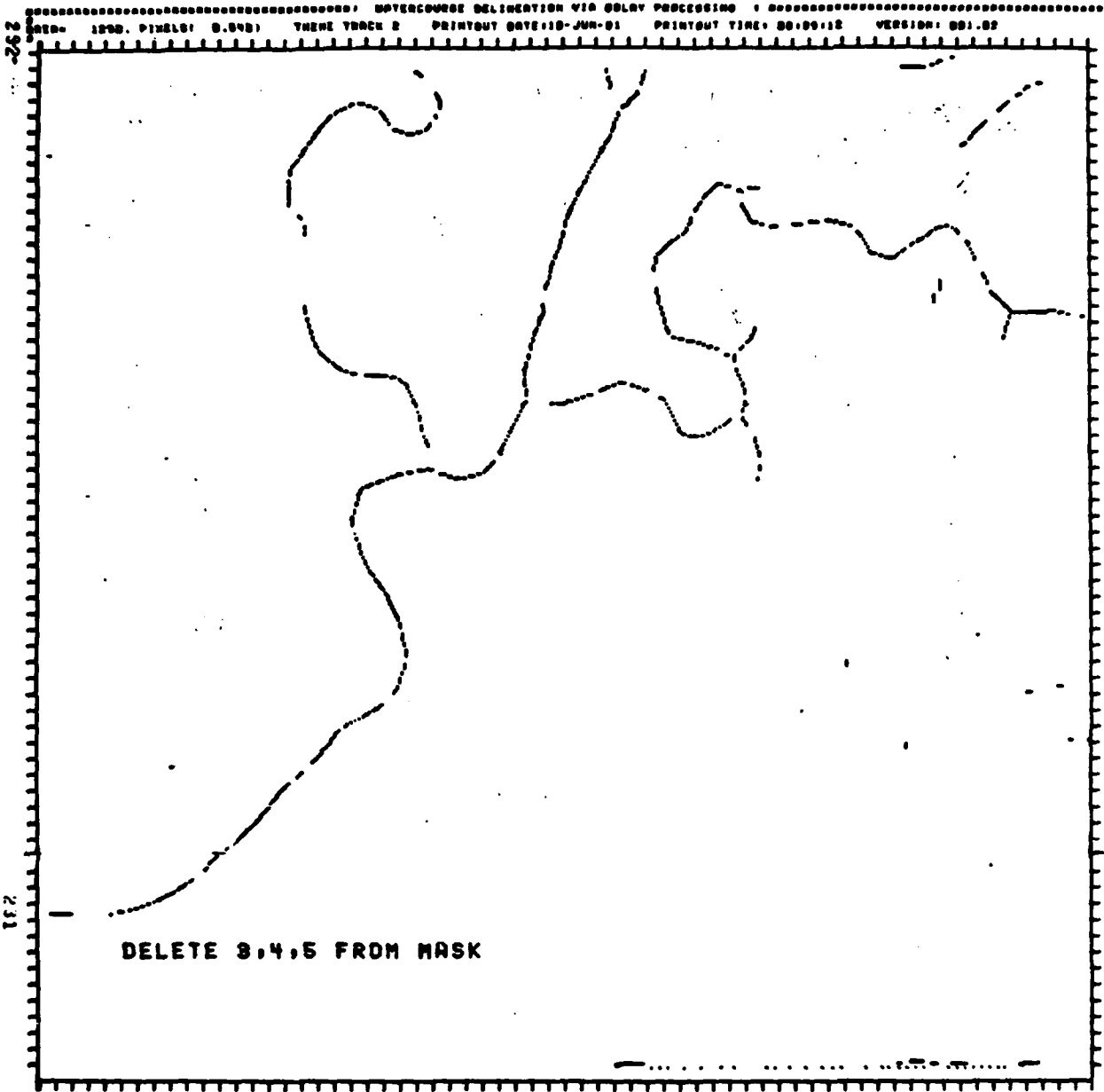


Figure 12. Watercourse Delineation by Skeletonizing the Mask In Figure 6.

represents Golay skeletonizing of the smoothed theme, and is probably the most successful skeletonization exercise. The gaps in this figure are largely a result of the way Golay transforms are currently implemented in GE DIAL with no attempt to compensate for use on a square pixel grid. These deficiencies could most likely be removed with an improved program.

C. Theme Smoothing Method

Categorization of images by computer assisted methods produces, except for the extraction of very easy and distinctly unique classes, results that are spatially complex. The classification may or may not be "noisy", depending on the suitability of the data for extraction of a given class, but will in general exhibit a salt and pepper appearance which might be interpreted as noise. (This effect comes from the fact that with current technology computers perform relatively poorly at spatial tasks.) Most of the commonly used classification schemes are totally aspatial, relying on intensity level in one or more spectral wavelength bands, while completely ignoring feature boundaries, shapes, and texture. Even those methods which aspire to use spatial information, for example, the textural operators described in this report, use spatial information in a very rudimentary manner when compared to the feature extraction capabilities of a human image interpreter.

The interpreter, in contrast, has a relatively poor ability to identify image intensity levels with any real accuracy, and relies very heavily on his spatial abilities, particularly when analysing monochrome imagery. As opposed to a computer classification, which takes place on a per pixel basis, the interpreter performs classification on a per feature basis. This leads to extraction of features by the location of their external boundaries and tends to force the classes into contiguous areas. The interpreter tends to ignore (or accommodate for) minor variances internal to the feature which might affect a computerized classification, and also performs considerable smoothing or straightening of feature outlines.

Deciding which of these two radically different approaches is more accurate is moot, and depends on ones definitions and objectives. What is certain is the fact that the interpreter generally produces a map that is far more suitable to the generation of factor overlays, and if computer assisted methods are to be used successfully, their results must be processed in a manner which changes them into a form which is more compatible with factor overlays. This requires spatial low pass filtering, and occasionally severe filtering. Factor overlays are very simple in spatial structure and contain only enough information to locate and describe a feature of tactical significance. Since overlays are to ultimately be combined, their simplicity becomes even more important. The combination of spatially complex overlays could produce a result containing too much information to be effectively understood and utilized. Production of overlays from computer generated thematic maps generally requires simulating the integration and feature selective filtering performed subconsciously by an image interpreter. Specifically it required the straightening of object boundaries, mapping the object in a contiguous manner, and the removal of small or unimportant features.

In Phase 1 of this project, efforts were restricted to the application of currently available software tools, eliminating the possibility of a truly effective theme filter. The need for one was, however, clear in the earliest stages of the project. Being able to "see the forest through the trees" was not a cliché, but was the definition of a major problem. Extracting forest boundaries meant elimination of high resolution microstructure within the desired feature without objectionably modifying the boundary.

The approach used in Phase 1, was basically to apply an image smoothing filter to the image as a preprocessing function. The smoothed image was then used to generate binary masks of various features after the removal of feature microstructure. Then the element extraction processes, which generally required use of microstructure and high resolution was performed on the original data. Themes generated were then restricted

to an area of the mask to eliminate misclassification errors which would otherwise be widespread and unacceptable. While the overall effort was successful, there were serious problems that were left to be solved. The most serious was that smoothing transformed all discontinuities or feature edges into smoothly flowing ramps. Generation of several of the required masks resulted in a situation where adjacent thematic features did not compliment or butt each other properly, with the area near the feature butt line, corresponding to the high slope region of the ramp, usually not being placed into the proper mask. A second problem was that the smoothing treated all objects equally as opposed to considering the spatial characteristics of individual classes. For example, narrow features such as roads, were obliterated by the heavy smoothing needed to map the forest. Lengthy work arounds were used where several degrees of smoothing were applied, but this was not an effective solution.

As a result of the Phase 1 effort, a series of requirements were identified for a post processing filter to work not on grey level images, but on the themes generated from the images. The advantages of the post processing are many and are basically related to the fact that all resolution is maintained and utilized as needed during the element extraction process. Only after the image has been reduced to the desired themes, is the filtering applied. Post processing solves the mentioned ramp problem by totally avoiding the generation of ramps, and the problem of selective filtering is handled more easily after categorization as the features have been identified.

Processing problems to be solved by a post processing filter:

- Theme Simplification - The severity of the filtering action should be user specifiable to achieve the desired amount of boundary smoothing. Capabilities should include extreme low pass filtering.
- Theme Selective Filtering - All themes are not created equal, and the filter must treat individual themes in individual manners. Methods must be provided for the restriction of growth, shrinkage or change for any particular theme.

- Theme Butting - The filter should be able to annihilate areas between adjacent features of interest which commonly result from classification errors or from the presence of incidental features such as shadows.
- Contiguous Area Thresholding - The filter should be capable filtering by object size to allow removal of islands and/or holes from any designated theme.

In the interest of reasonable execution times, the use of box filtering techniques were nearly dictated for the theme filter. The filter must basically count the number of times each theme occurred in each local neighborhood. Each of the themes is then ranked, based on its local occurrence rate in conjunction with operator designated replacement constraints, as a candidate for being replaced and for being a possible replacement for others. The key theme (the one positioned in the center of the filter area at any given time) is then replaced by the theme at the top of the ranking hierarchy. In the absence of operator input constraints the key theme is replaced by the one most common in the neighborhood. To perform this operation means tabulating running areal totals for each of the themes being processed, and could amount to an excessive number of calculations with clumsy implementation. Box filtering, with its ability to operate with filters of any reasonable size without appreciable execution time penalties, is ideally suited to the task. Keeping running totals with box filters even required less computation than their usual use, which is keeping running averages. As with the running averages, the minor artifacting introduced by the use of a square spatial operator, is outweighed by the computational efficiencies.

Figure 13 shows the theme filter main program, and Figure 14 shows the data structure used. Note that the program is set up to handle eight themes, but nine different categories. Theme #0 refers to uncategorized pixels or those not in any of the eight themes. In retrospect, even another category marking multiply classified pixels or those contained in more than one theme would have been useful.

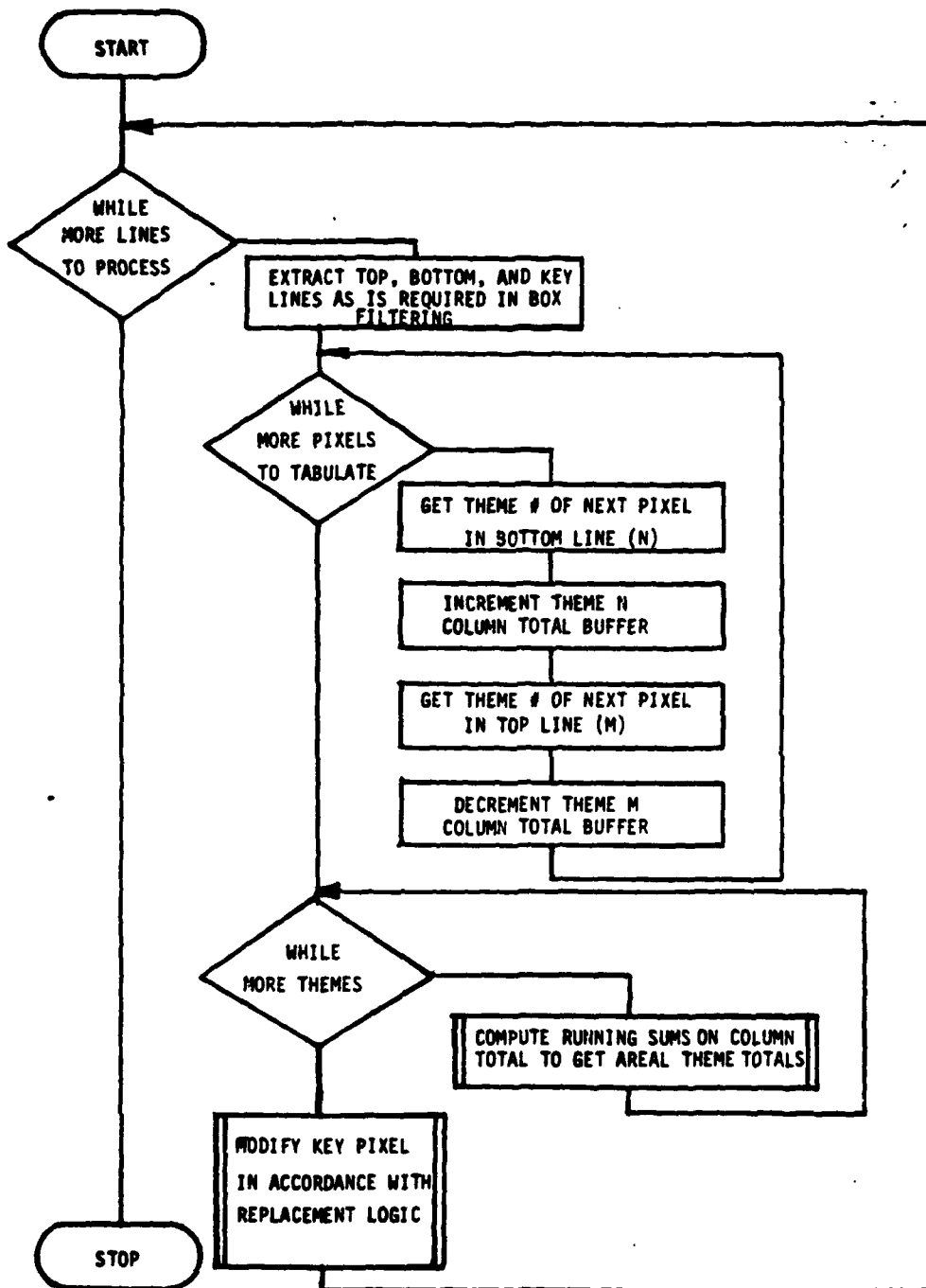


Figure 12. Theme Filter

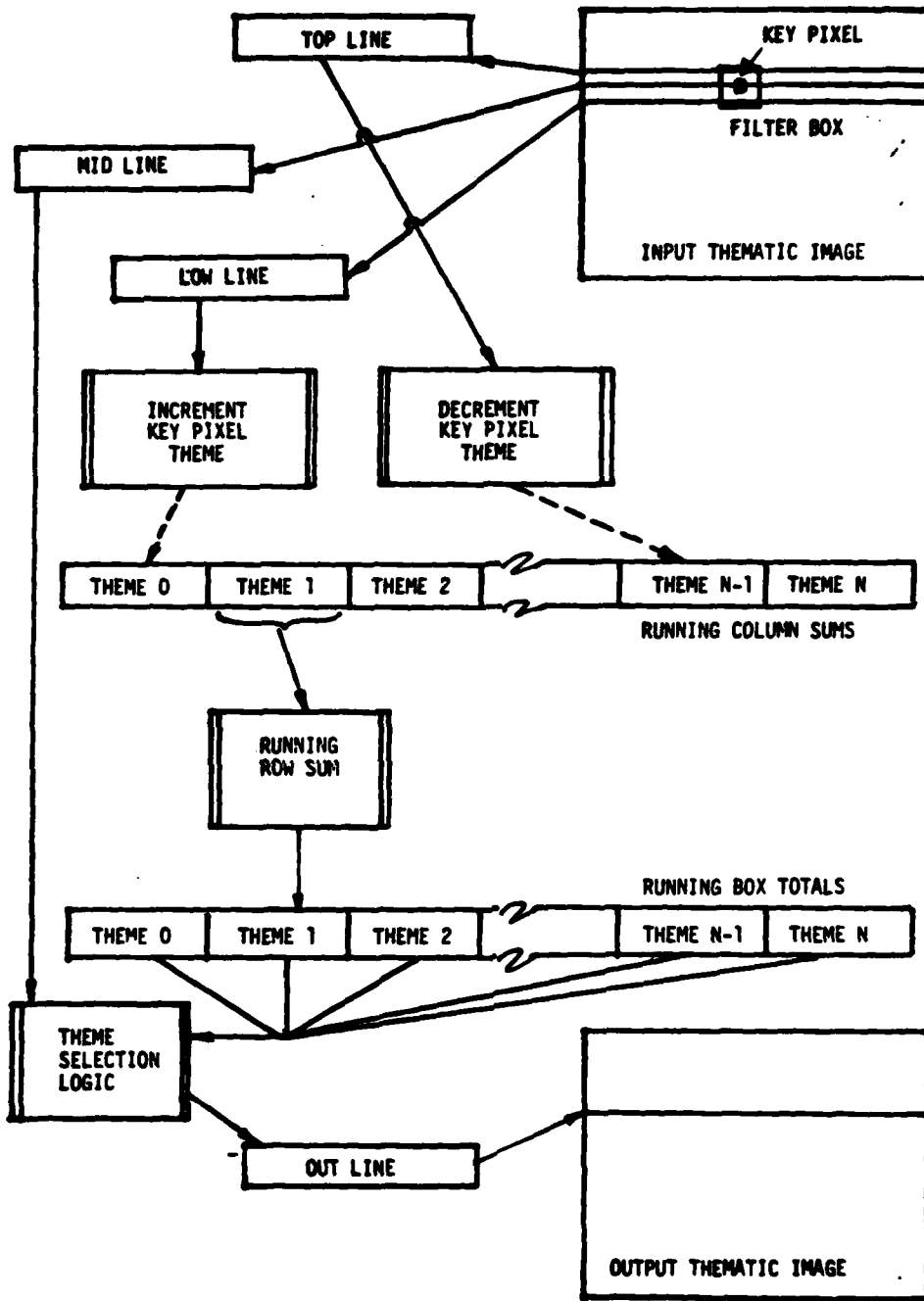


Figure 14. Theme Filter Data Structure

The box filtering approach is not unique in theme processing, but as commonly used it is not sufficient to satisfy the theme processor requirements. Used without replacement constraints, it is not versatile enough to perform all the desired theme manipulation possibilities. Box filtering is simply an effective method of providing information to the real heart of the theme processor, the theme replacement logic.

Interactive control of the theme filter is accomplished by setting theme population thresholds which must be exceeded before a given theme is modified, with separate thresholds for theme growth and for theme shrinkage. The filter can thus be set up to treat themes uniquely when needed. Whenever the shrink threshold is passed, the key pixel is replaced by the most common growable theme. See Figure 15.

Considerable effort went into the operator interface. Its purpose is to assist the operator in the task of setting up fairly complex filtering options. The filter has numerous input parameters, whose input is tedious and has a high potential for error. Also, since the filtering action on a given theme is determined not only by the thresholds directly associated with that theme, but also by the thresholds of its possible replacements, the interaction between these input parameters is not always obvious. The interface is designed to minimize confusion along with tedious repetitive input. The filter is set up to operate in an iterative manner. Allowing either continuing with the filtering or reworking the last operation if undesirable results appear. The filter executes fast enough so that an operator can observe and analyze the result, and then rerun the filter with modified thresholds or box size to better achieve his purpose.

Figure 16 is a verification display for the interface, listing current threshold settings along with a computer generated verbal estimate of the resulting filtering action.

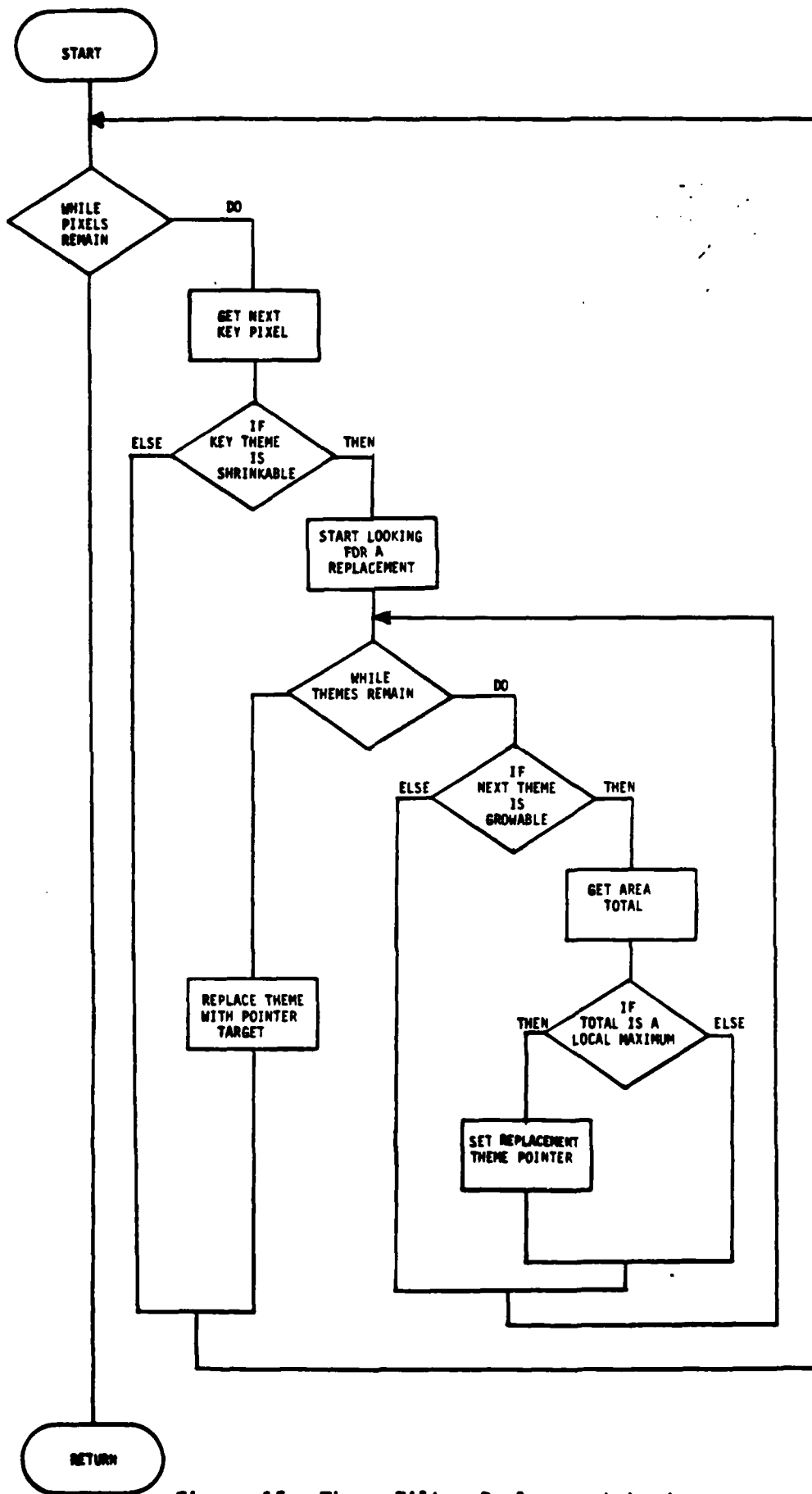


Figure 15. Theme Filter Replacement Logic
37

THE FILTER OPERATES BY TABULATING THE NUMBER OF TIMES EACH INDIVIDUAL THEME OCCURS INSIDE A MOVING RECTANGULAR AREA. FOR EACH FILTER POSITION, IF THE THEME IN THE CENTER OF THE AREA IS ALLOWED TO SHRINK, IT WILL BE REPLACED BY THE MOST COMMONLY OCCURRING AMONG THOSE ALLOWED TO GROW. WHENEVER A SUITABLE REPLACEMENT IS NOT FOUND, THE THEME IS LEFT UNALTERED.

SELECTIVE THEME REPLACEMENT IS CONTROLLED BY ENTERING THRESHOLDS EXPRESSED AS PERCENTILES OF THE FILTER AREA. A SPECIFIC THEME IS ALLOWED TO SHRINK IF THE NUMBER OF TIMES IT OCCURS IS LESS THAN IT'S SHRINK THRESHOLD, AND IS ALLOWED TO GROW IF OCCURANCES EXCEED IT'S GROW THRESHOLD. A NEGATIVE THRESHOLD VALUE CAUSES A THEME TO BE TOTALLY REPLACED.

11 X 11 FILTER 09-FEB-82 10:03:27

THEME THRESHOLD PERCENTILES

THEME NUMBER	SHRINK IF BELOW, %	GROW IF ABOVE, %	FILTERING ACTION	
0	***	***	REPLACE	_____
1	100	0	SMOOTH	_____
2	100	0	SMOOTH	_____
3	0	100	NO CHANGE	_____
4	100	100	SHRINK ONLY	_____
5	0	0	GROW ONLY	_____
6	15	100	REMOVE ISLES	_____
7	0	85	REMOVE HOLES	_____
8	5	95	CLEAN NOISE	_____

CHANGE THEME THRESHOLDS? (Y)ES, OR (N)O >

Figure 16. Theme Filter Operator Interface

D. Use of the Theme Filter

In the pursuit of water resources element extraction it was established that the theme filter was effective in solving a host of image categorization problems. The most severely limiting factor in its usage seems to be the limited ability of the analyst to correctly forecast the action of the filter and to be able to specify the set of interrelated thresholds needed to perform the desired function. Several common image categorization problems, along with suggested and proven solutions making use of the theme filter, are listed below:

1. Capabilities

a. General Theme Smoothing

Application: Spatially simplify theme so that they may be depicted by line maps.

Procedure: Run filter with default thresholds so that each theme is replaced by the locally most common. For best results, run the filter twice to insure no linear artifacts are generated by the rectangular filter.

b. Annihilation of Certain Themes

Application: Often it is desirable to totally eliminate irrelevant themes (shadows for example).

Procedure:

- Enter a negative threshold for the theme to be annihilated.

- Leave all other thresholds default

The theme filter has a replacement flag set by entering a negative threshold. This flag stops the theme from being considered as a possible output.

c. Preserving Certain Themes While Filtering Others

Application: Very accurately mapped themes, or sparse filament themes (roads, etc.) should not be changed.

Procedure:

- Set grow above threshold to 100% so that theme will not grow.

- Set shrink below threshold to 0% so that theme will not shrink.

d. General Noise Cleaner

Application: Most spatially derived themes are improved by eliminating isolated pixels which often represent noise.

Procedure: Set thresholds for all themes to grow above 33% and shrink below 11%, then run a 3 x 3 pixel filter.

The 11% threshold locates totally isolated pixels, while the 33% threshold insures there is a locally dominant class before replacement. By enlarging the filter size this operation can be used to remove larger areas (islands and holes).

e. Resolving Theme Overlap or Underlap

Application: Image classification virtually always results in some error as signatures are so unique. Problem areas often show up as unclassified or multiply classified pixels. A satisfactory solution is to remap these poorly classified pixels on a purely spatial basis.

Procedure:

- Place pixels to be resolved into a separate theme
- Set replacement flag for that theme
- Leave default thresholds for all other themes

2. Processing Example

To demonstrate the filtering capability, one of the panchromatic scenes used in the first phase of this project was recategorized with considerable improvement. The following processing sequence was followed and results are shown in Figure 17.

- a. The scene was categorized on the basis of gray level and microtexture. This task was some what easier than with the previous effort because the filter will contiguously map locally predominant classes allowing less discriminative signatures to be used.



Figure
Theme From Phase 1



Figure
Manual Interpretation
From Phase 1



Figure 17. Comparison of Fort Belvoir/Woodbridge Vegetation Themes Derived by Interactive Computer and Manual Interpretation.

- b. Areas categorized as shadows were eliminated by the filter based on operator interpretation. Two passes were used: (1) when a shadow occurred within a short distance to water, the shadow was changed to water producing considerable improvement in the river shoreline and bridge area, and (2) shadows not near water were mapped to the locally most common class.
- c. The scene was then smoothed by a 5 x 5 pixel filter to produce a categorization comparable to what must be done by a photo interpreter. The "bare" category was restricted from change so that the roads would be preserved. See Figure 18A and 19.
- d. To demonstrate severe filtering, the categories were then processed with an 11 x 11 pixel filter. See Figure 18B and 20. allows virtually any degree of filtering to be achieved with about the same execution time. In working situation, the desired amount of filtering might be determined by scene complexity, desired scale, or intended use.
- e. Both of the products were reduced to line maps via Golay processing (essentially automated). Each theme was outlined, the outlines were combined to produce double width lines, and then the lines were shrunk to a single pixel width.

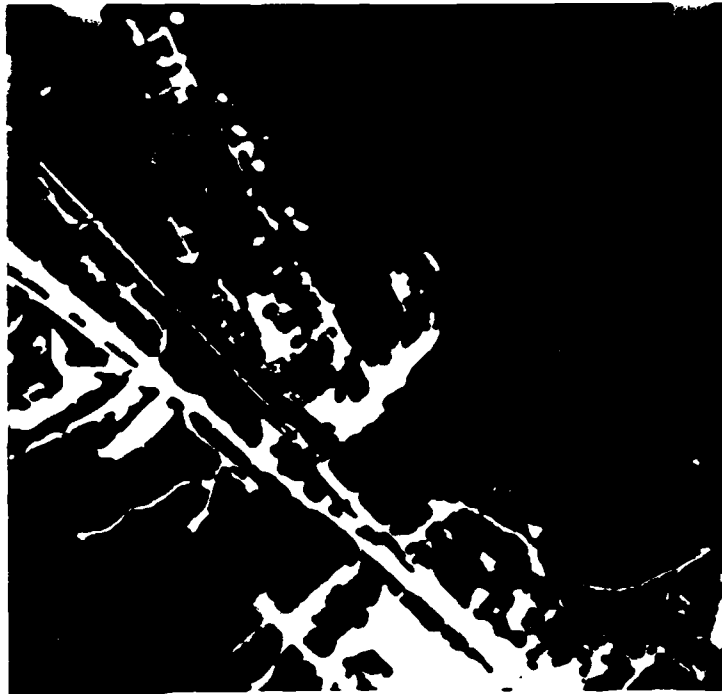


Figure 18A
Theme Filter
Results



Figure 18B
Simplified Themes

Figure 18. Results of Theme Filtering

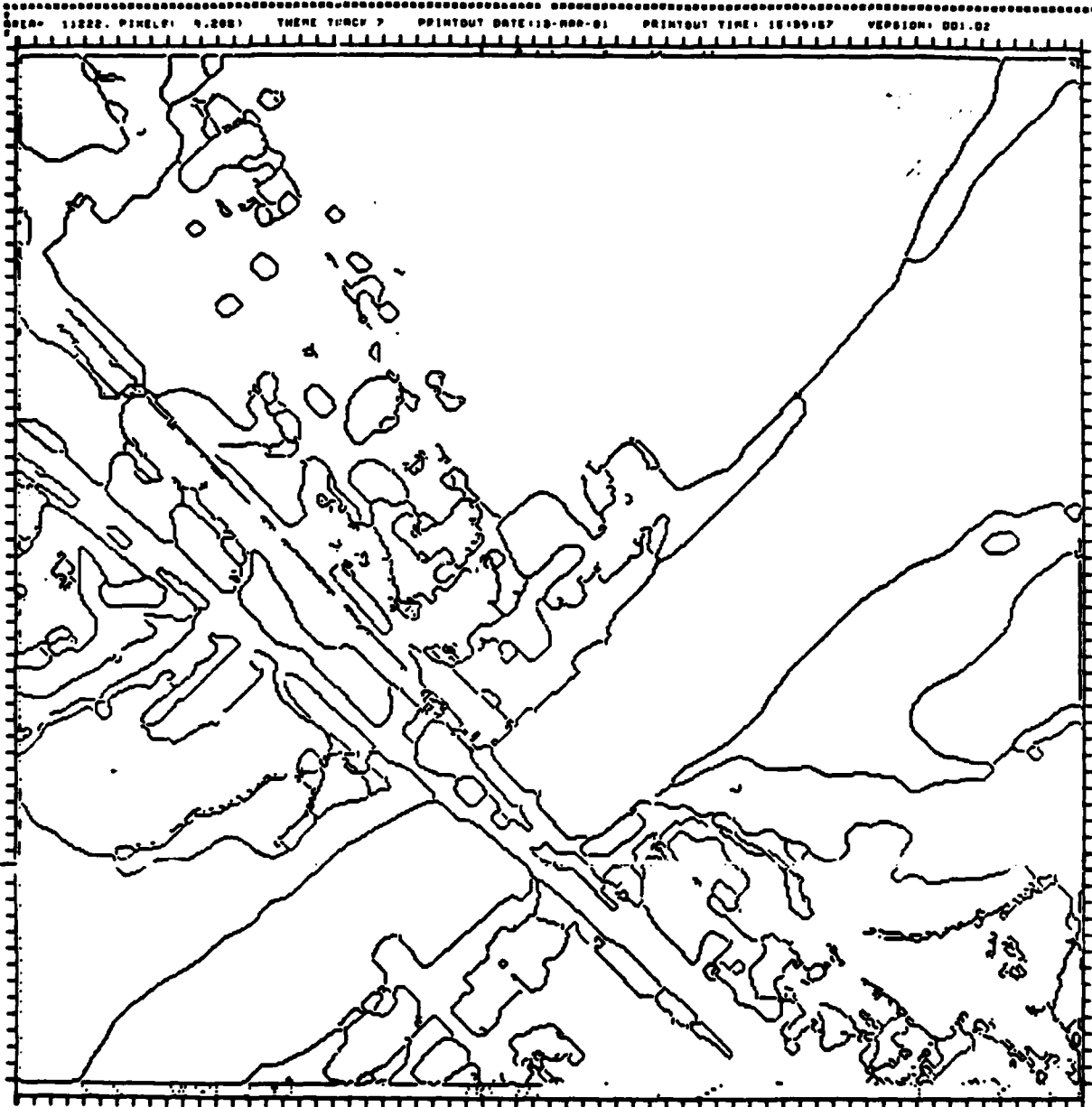


Figure 19. Outlines of Vegetation and Water Themes

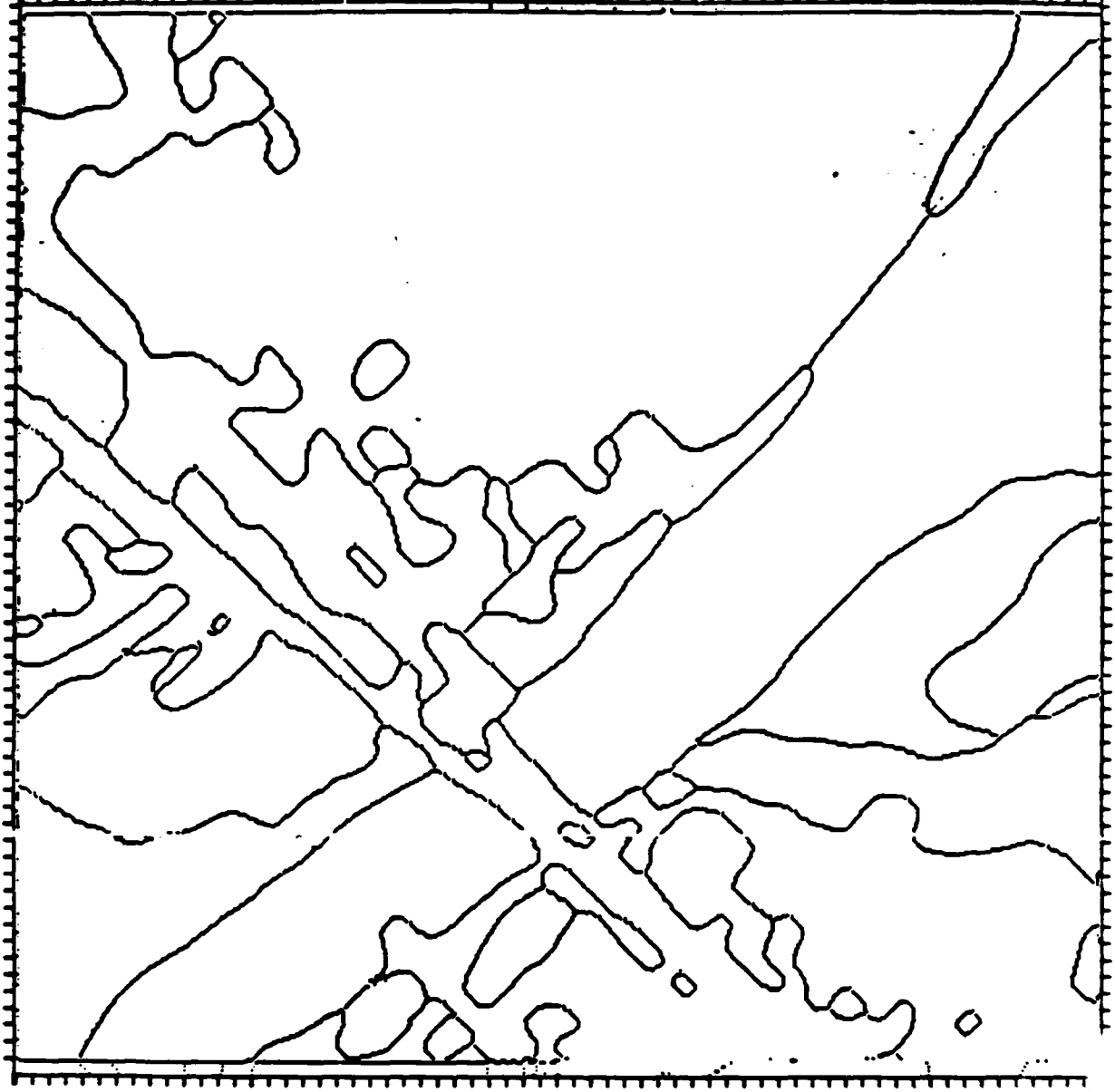


Figure 20. Outlines of Simplified Themes

IV. RESULTS

A. Extraction of Watercourses and Associated Elements

1. Watercourses

a. Extracting Watercourses From Panchromatic and Thermal Imagery

The Knobbs Creek imagery was selected for analysis and development of techniques because it contained data coverage from both thermal IR and panchromatic photographic sensors. Knobbs Creek is a good test for watercourse classification in that the coverage contains a wide range of watercourse widths. It provides a difficult test as the creek is almost 100% surrounded by large trees. This does not provide, however, the best diversity of surrounds with which to test watercourse identification. The ideal test would involve a scene having many watercourses surrounded by a variety of land cover, and having a wide range of watercourse widths.

Knobbs Creek, being surrounded by forest, creates problems for both machine and manual analysis. The creek is obscured by forest when watercourse width is less than approximately 5 meters. In these situations, analyst inferred watercourses will be necessary. Further, the determination of dry gap widths is hindered by the encroaching forest and by shadows from adjacent forest which extend across the watercourse. The shadows create an additional problem for machine analysis by completely or partially covering the watercourse (depending on watercourse width or direction). This requires the machine analyst to derive a signature for water and a signature for shadow which must be added together to enable identification of the entire watercourse. Unfortunately, the signature contains areas which are not watercourses (forest shadows within the forest).

Eight (512 x 512 pixel) subscenes were digitized at GE DIAL from three of the 1:100,000 scale thermal IR transparencies of Knobbs Creek. Four of these were digitized from 4 in. x 4 in. areas in the transparency (ground resolution approximately 2 meter) and four from 1 in. x 1 in. areas (ground resolution approximately 0.5 meter). In addition, four (512 x 512 pixel) subscenes were digitized at GE DIAL from the 1:250,000 scale panchromatic

photographic image of Knobbs Creek. Two of these were digitized from 4 in. x 4 in. areas (ground resolution approximately 4 meter)* and two from 1 in. x 1 in. areas (ground resolution approximately 1 meter)*. These resolutions were chosen for initial tests of watercourse identification to provide an appropriate range of resolution for testing the spatial filters used in watercourse extraction.

The initial step in the investigation was to identify whether gray level density slicing would be sufficient to identify watercourses. Several problems were encountered in this step:

- This thermal IR imagery appears to contain dark and light bands which are apparently caused by automatic gain and offset in the sensor which are adjusted by scan content. This has the undesirable effect of causing forest and watercourse signatures from different areas of the scene to appear similar and thus be recognized simultaneously.
- Both thermal IR and panchromatic photographic digital images exhibit a shading across the image which cause signatures to overlap.

Gray level density slicing of either the thermal or the panchromatic photographic digital images, in order to extract watercourses, produced unacceptable results. Figures 21 and 23 show the watercourse themes extracted from the panchromatic photographic images, and similar poor results were obtained with the thermal images.

At this point, several spatial filters were applied to all digital images. The images resulting from use of these filters have high spatial frequency noise, however, which requires a subsequent smoothing operation. Best smoothing results appear to be produced with the 7 x 7 pixel moving average. Some of the spatial operations tested include: 5-pixel cross min-max texture; the original image minus a 7 x 7 pixel local average; 3 x 3 pixel Laplacian.

* In this report, the term resolution is used to describe the actual dimensions of the smallest resolvable objects in an image.

016

016



Figure 21. Watercourse Extraction via Level Slicing The
1-meter Resolution Panchromatic Photographic Image.

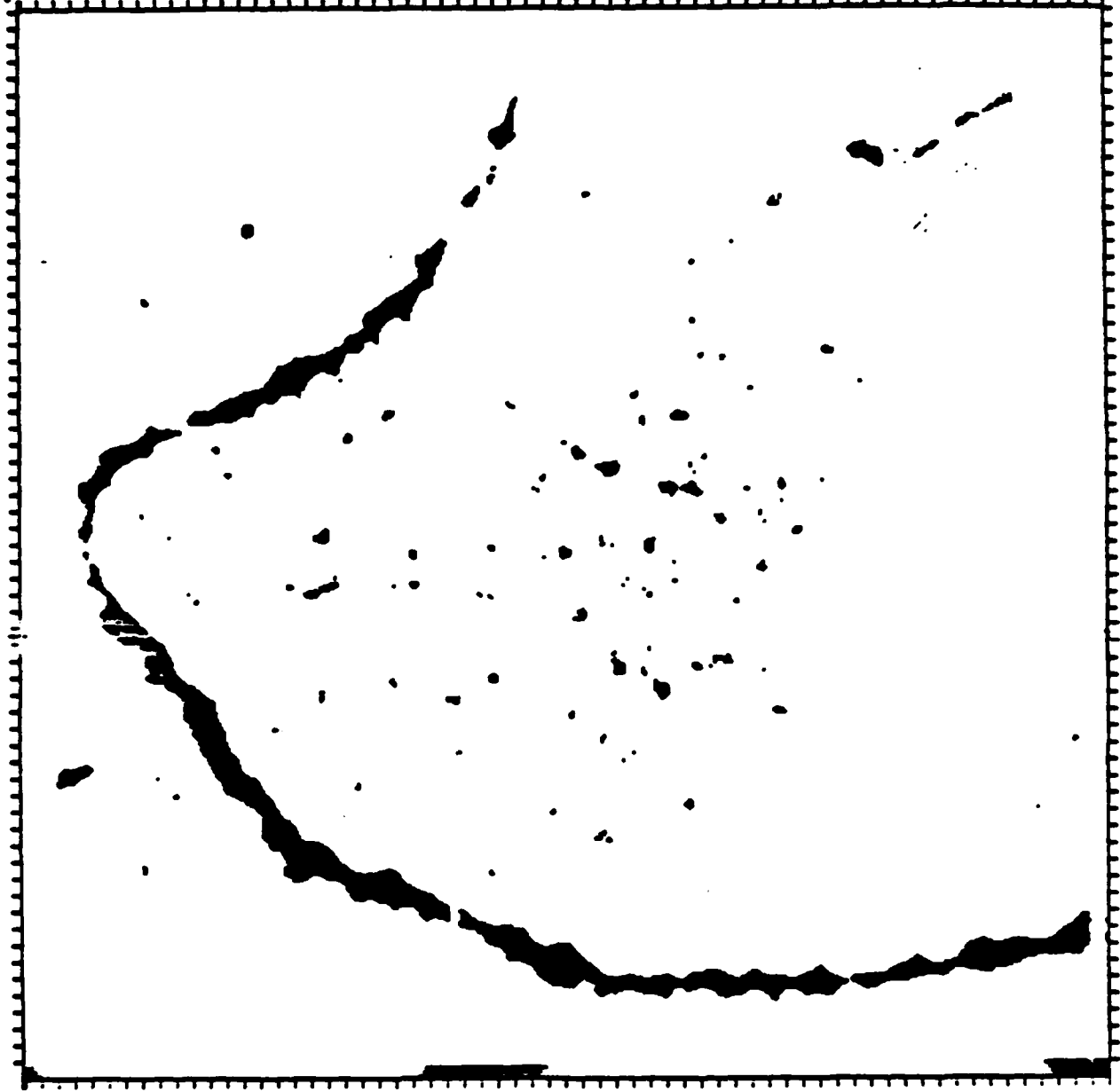


Figure 22. Watercourse Identification via Joint Use of the 1-meter resolution Panchromatic Photographic Image and the Derived Texture Image (Smoothed).



Figure 23. Watercourse Extraction via Level Slicing The 4-Meter Resolution Panchromatic Photographic Image

Results when using the spatially processed images jointly with the original digital images (i.e., two dimensional classification) can be summarized as follows:

- Best results for 1 meter resolution panchromatic photographic and 0.5 meter resolution thermal IR images were obtained with the 5-pixel Cross min-max texture operator.
- Best results for 4 meter resolution panchromatic photographic and 2 meter resolution thermal IR images were obtained with the original image minus the 7 x 7 local average.

Figure 22 shows the result of watercourse identification using the 1 meter resolution panchromatic photographic image and its texture version. In Figure 24, watercourse identification is shown for the case when 4 Meter resolution panchromatic photographic image and the smoothed result of image minus a 7 x 7 pixel local average are used. Figures 23 and 26 show the actual location of the watercourse. Comparison of actual and machine-extracted watercourses indicates shadow areas in both scenes which are falsely included in the watercourse theme.

Similar results were obtained with the digitized thermal imagery.

2. Extracting Watercourses from Radar Imagery

Four subscenes of 512 x 512 pixels depicting X-Band vertical and horizontal polarization; and L-Band vertical and horizontal polarization were made. Figures 25, 26, 27, and 28 are illustrations of the subscenes as displayed on the cathode ray tube (CRT). The data are shown inverted to a negative. Geographic scene locations are shown in Figure 29.

In Figures 30, 31, 32 and 33 theme results were obtained by sensity slicing the data. Figure 34 subscene is the same as Figure 25. It is repeated to allow for comparison relative to the processes subscenes in Figures 35, 36 and 37. In Figure 35, the subscene illustrates the application of the 7 x 7 moving average operator on the X-Band horizontal polarization data. Figure 36 shows the derived associated microtexture image derived from X-Band horizontal polarization data.

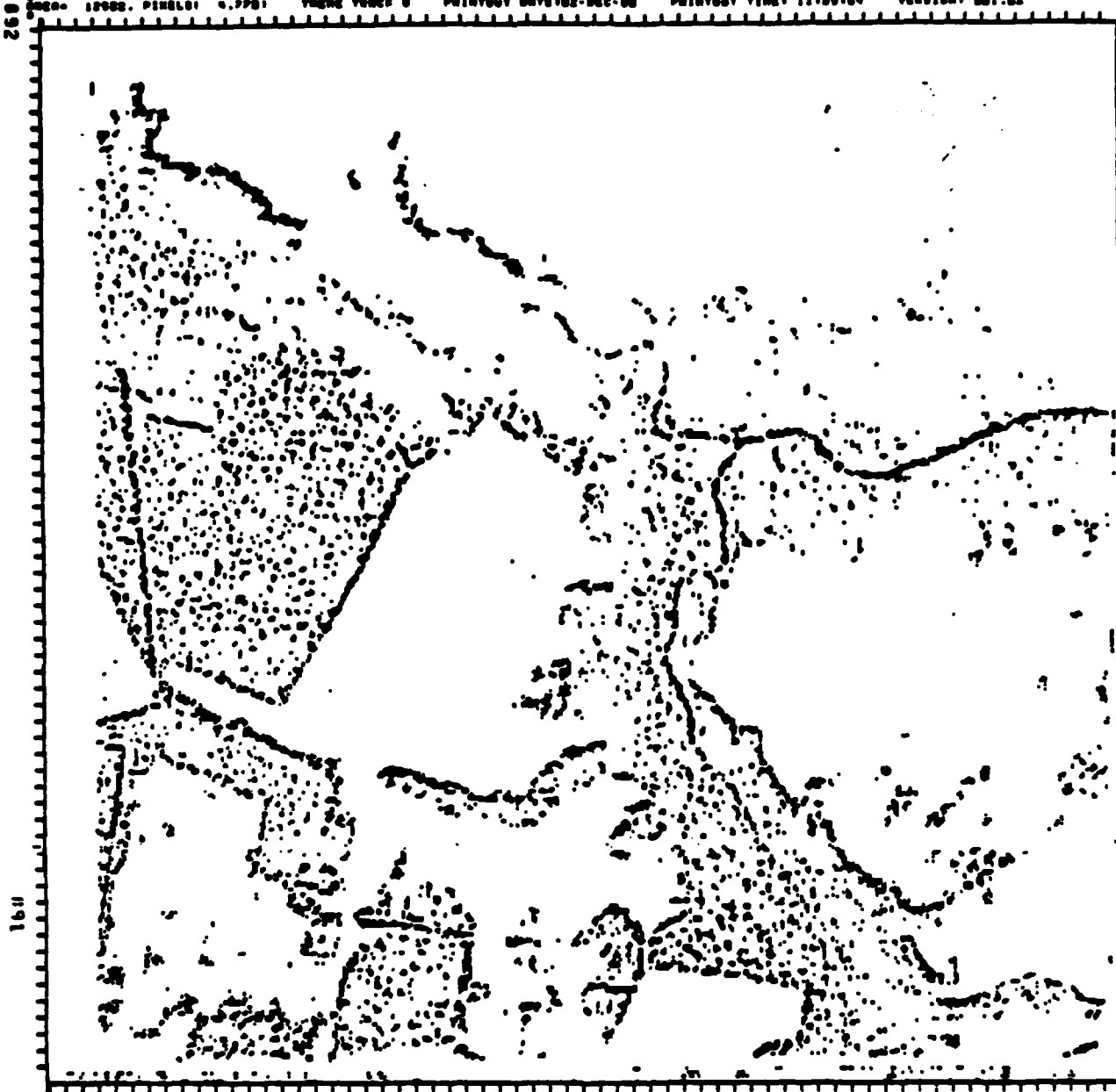


Figure 24. Watercourse Identification via Joint Use of the 4-Meter Resolution Panchromatic Photographic Image and the Result of the Image Minus 7 x 7 Pixel Local Average.

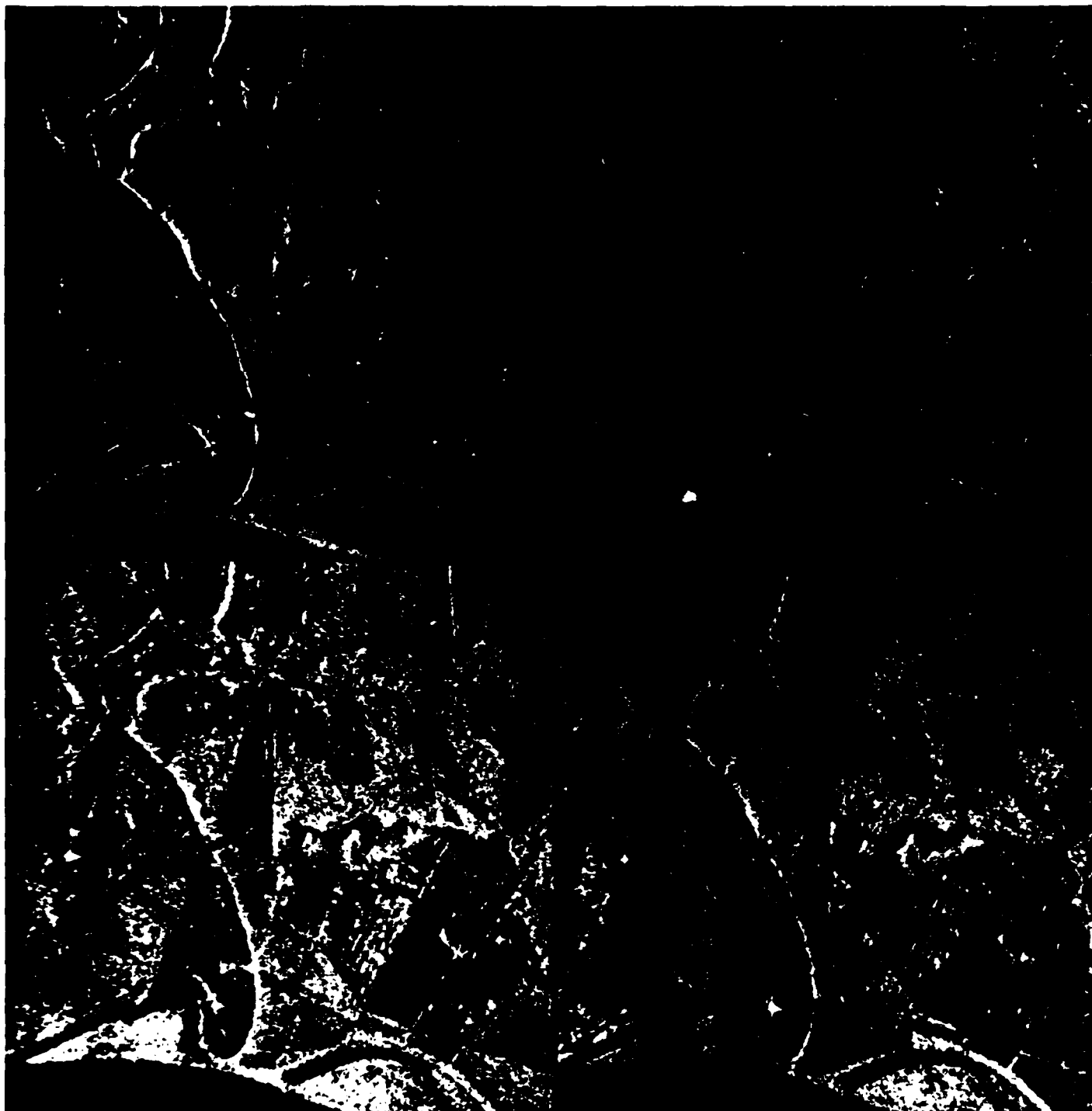


Figure 25 - X-Band Horizontal Polarization
Subscene

Figure 26 - X-Band Vertical Polarization
Subscene

Figure 27 - L-Band Horizontal
Polarization Subscene

Figure 28 - L-Band Vertical
Polarization Subscene

CHARLESTON, WEST VIRGINIA
EASTERN UNITED STATES 1:250,000

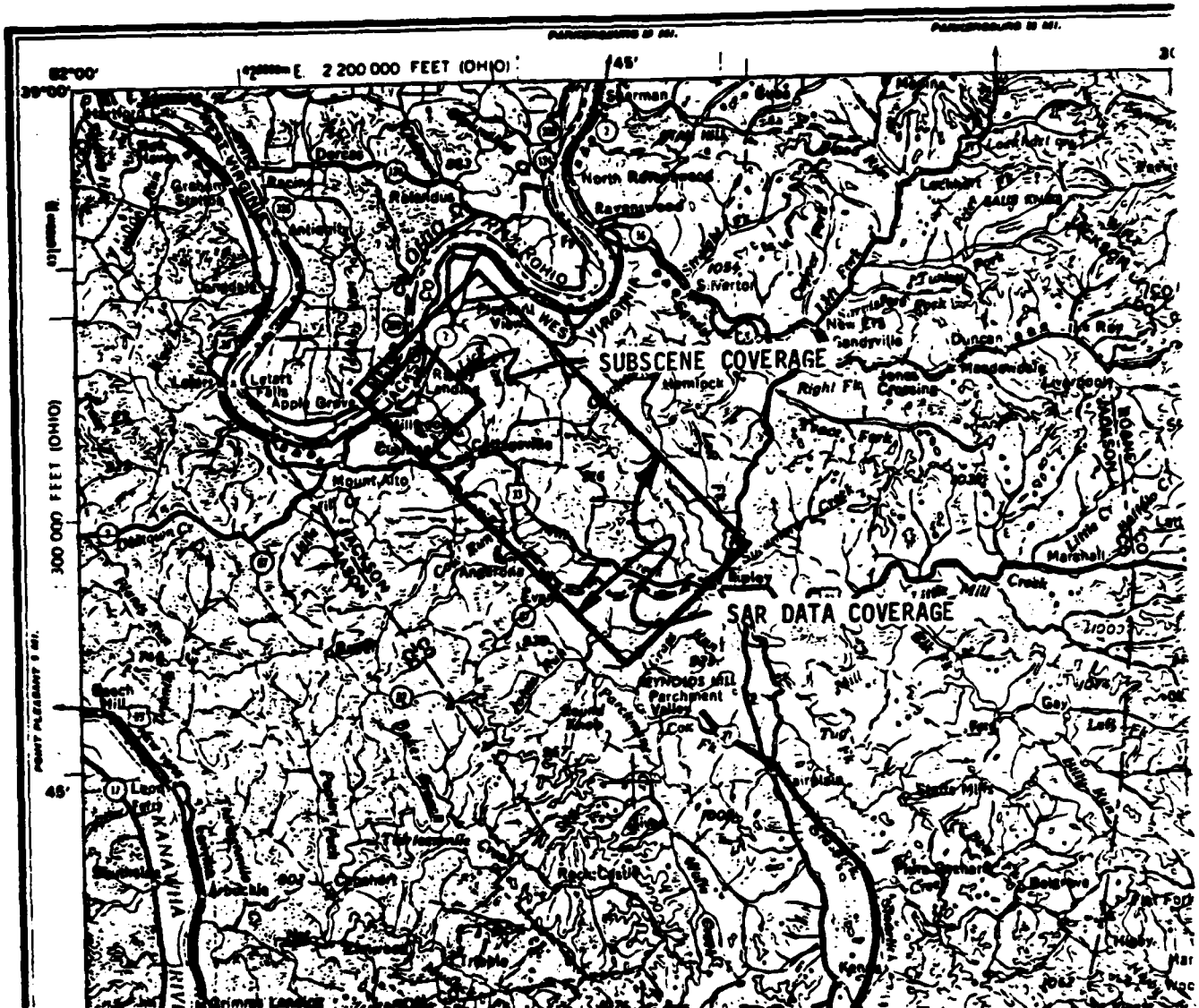


Figure 29. SAR Data and Subscene Locations.

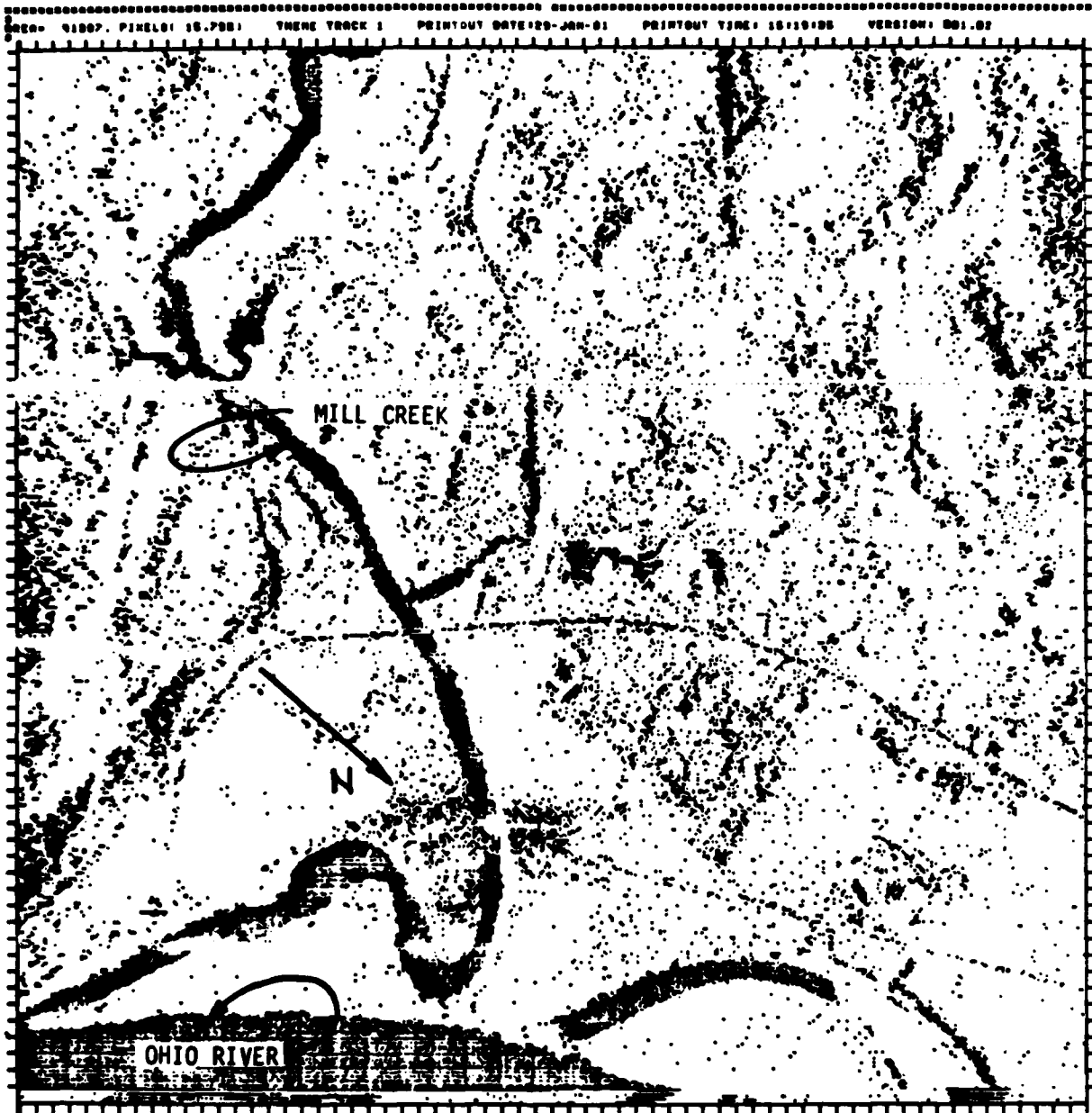


Figure 30. Water Theme Sliced From X-Band SAR w/Horizontal Polarization.



Figure 32. Water Theme Derived From X-Band SAR w/Horizontal Polarization.



Figure 33. Water Theme derived from L-Band SAR w/vertical polarization.



Figure 34 - X-Band Horizontal
Polarization Subscene
(Same as Figure 1)

Figure 35 - X-Band Horizontal
Polarization Data Smoothed
by a 7x7 Pixel Moving
Average Operator

Figure 36 - Associated Microtexture
Image

Figure 37 - Microtexture Image Data
Smoothed by a 7x7 Pixel
Moving Average Operator

Figure 37 shows the results of smoothing the microtexture image data with a 7 x 7 pixel moving average operator. In Figure 38, X-Band SAR horizontal polarization data and the associated microtexture image are shown categorized by a two dimensional partitioning classifier. This procedure appears to provide clearer results than density slicing the SAR data only. Figure 39 illustrates the categorization of water courses and water bodies using the two dimensional partitioning classifier with the X-Band SAR horizontal polarization data and the texture data smoothed by a 7 x 7 pixel moving average operator. It appears that another degree of clarification is achieved with this method. In Figure 40, X-Band SAR horizontal polarization data are smoothed by a 7 x 7 pixel moving average operator and density sliced into a theme for watercourses and water bodies. Apparently this procedure provides the best definition of watercourses and water bodies.

3. Watercourse Extraction Using Combinations of Dissimilar Images

A data set of panchromatic, low resolution Thermal IR and SAR (two flight directions) images were digitized at 5 meter resolution and registered for Little River and New Begun Creek water courses. Different combinations of data from the set, including microtexture derived data (from panchromatic images) were used for the extraction of water course themes. The low resolution Thermal IR data did not have a significant contribution to the derivation of the watercourse themes. From the analysis of a combination of the panchromatic images and the microtexture (derived from the panchromatic images) data a conflict between shadows and water existed. Radar imagery was used to resolve some of the shadow conflict. Derived watercourse themes for Little River are shown in Figures 41, 42 and 43..

A comparison of watercourse themes which were derived from different combinations of Chapel Creek (upper Little River area) panchromatic, radar, and the microtexture data (derived from the panchromatic images) are shown in Figures 44, 45, 46 and 47.

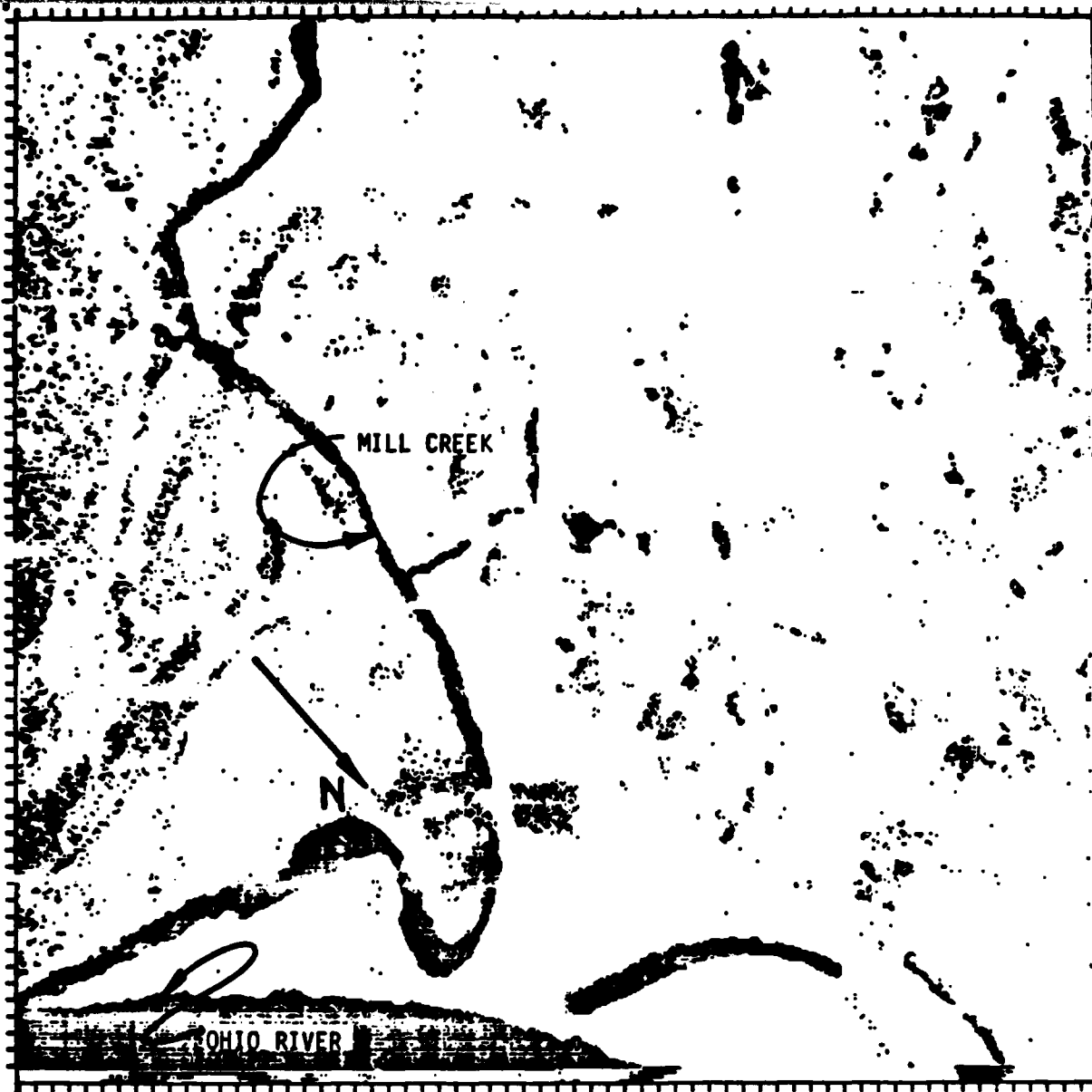


Figure 39. Water Theme Derived From 2-Dimensional Partitioning of X-Band SAR w/Horizontal Polarization Data and Textured Data Smoothed By a 7 x 7 Pixel Moving Average Operator.

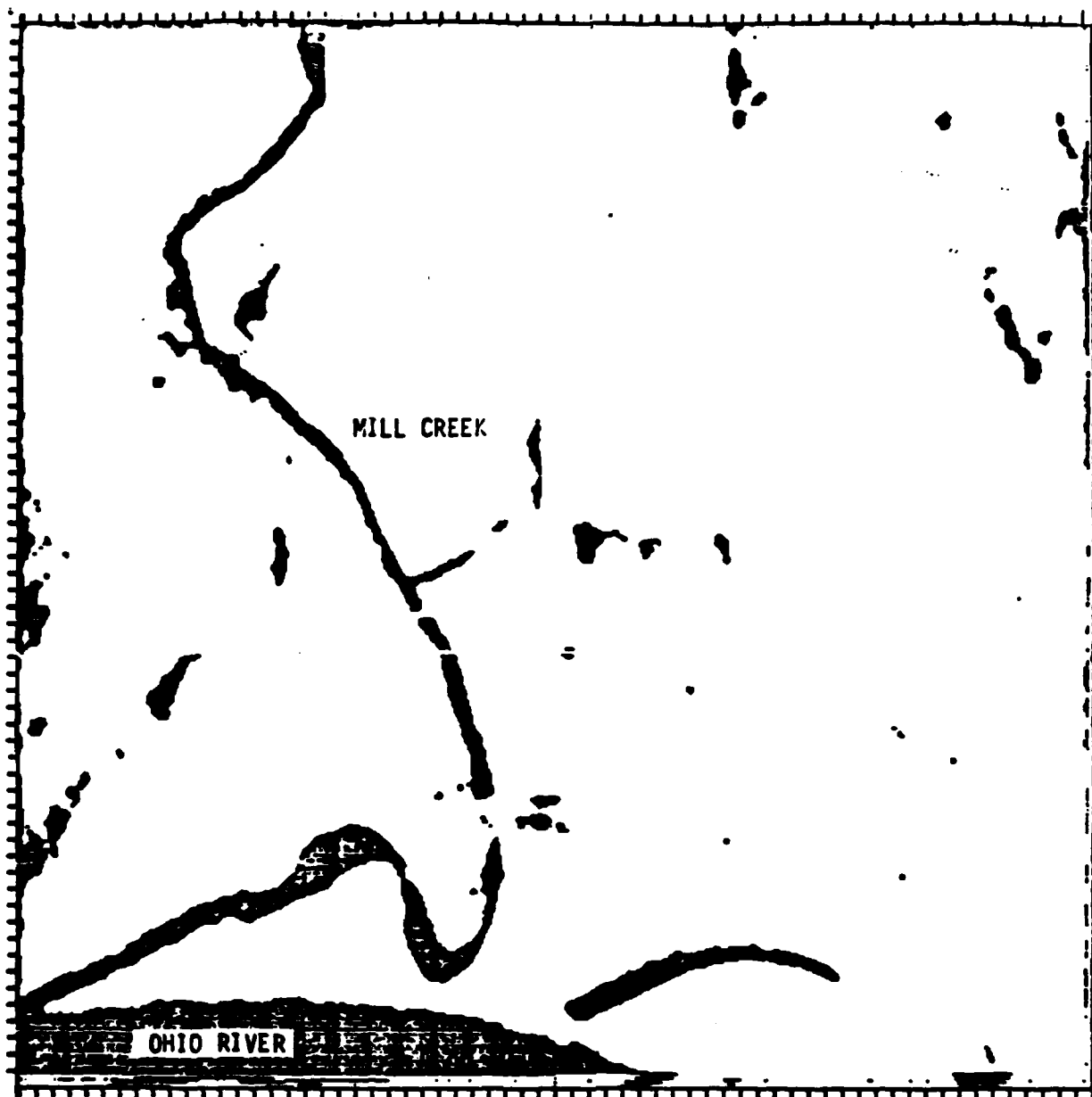


Figure 40. Theme Derived From the 7 x 7 Pixel Smoothed Version of the X-Band SAR Image w/Horizontal Polarization.

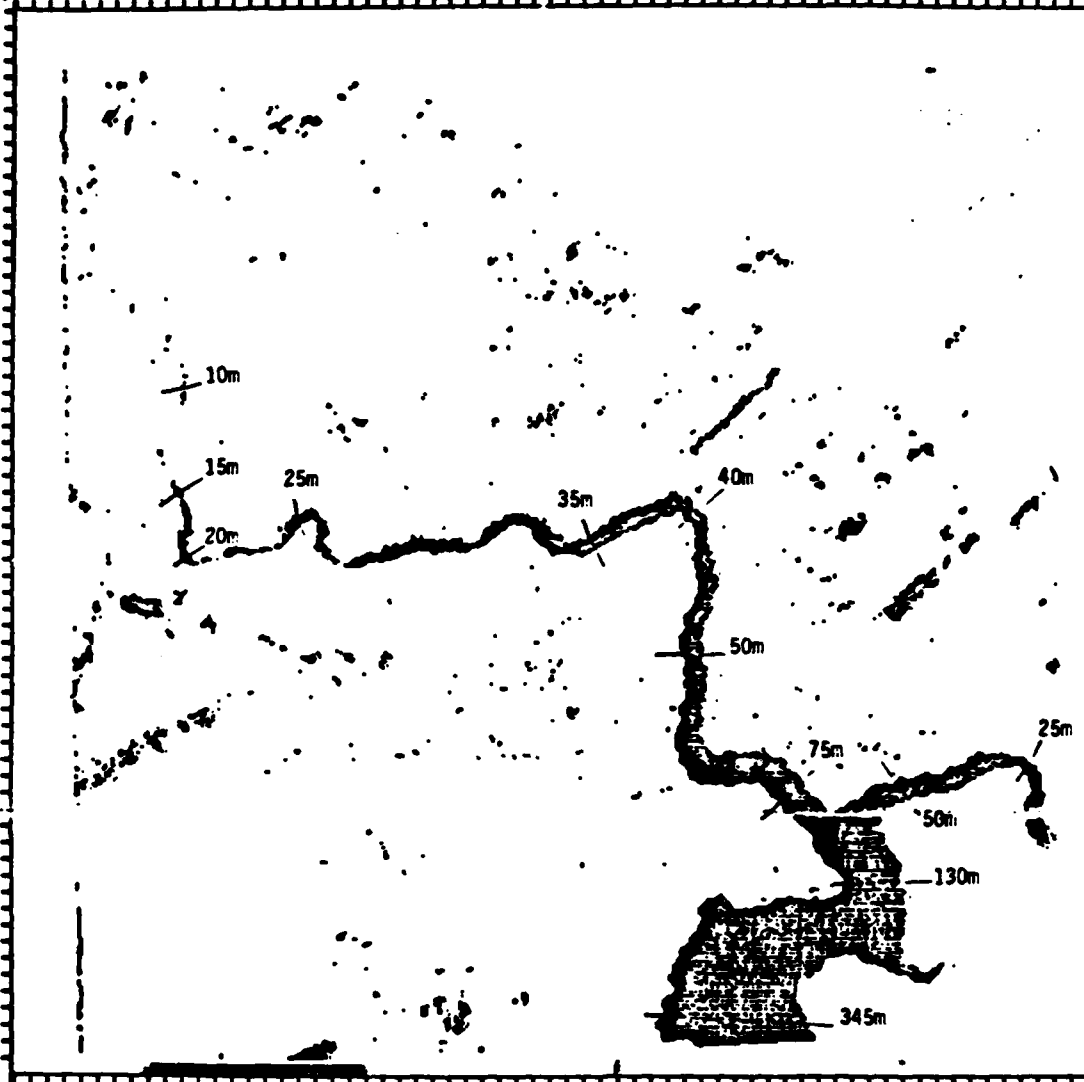


Figure 41. Little River Watercourse Theme Derived from Panchromatic Photography and SAR Data and from the Microtexture Version of the Photography.

.....LITTLE RIVER WATER TREAT.....
11- 11966. 0.750. TIME TARD 2 00:00. DATE:12-FEB-61 00:00. TIME: 18:15:10. VESSEL: DS1.02

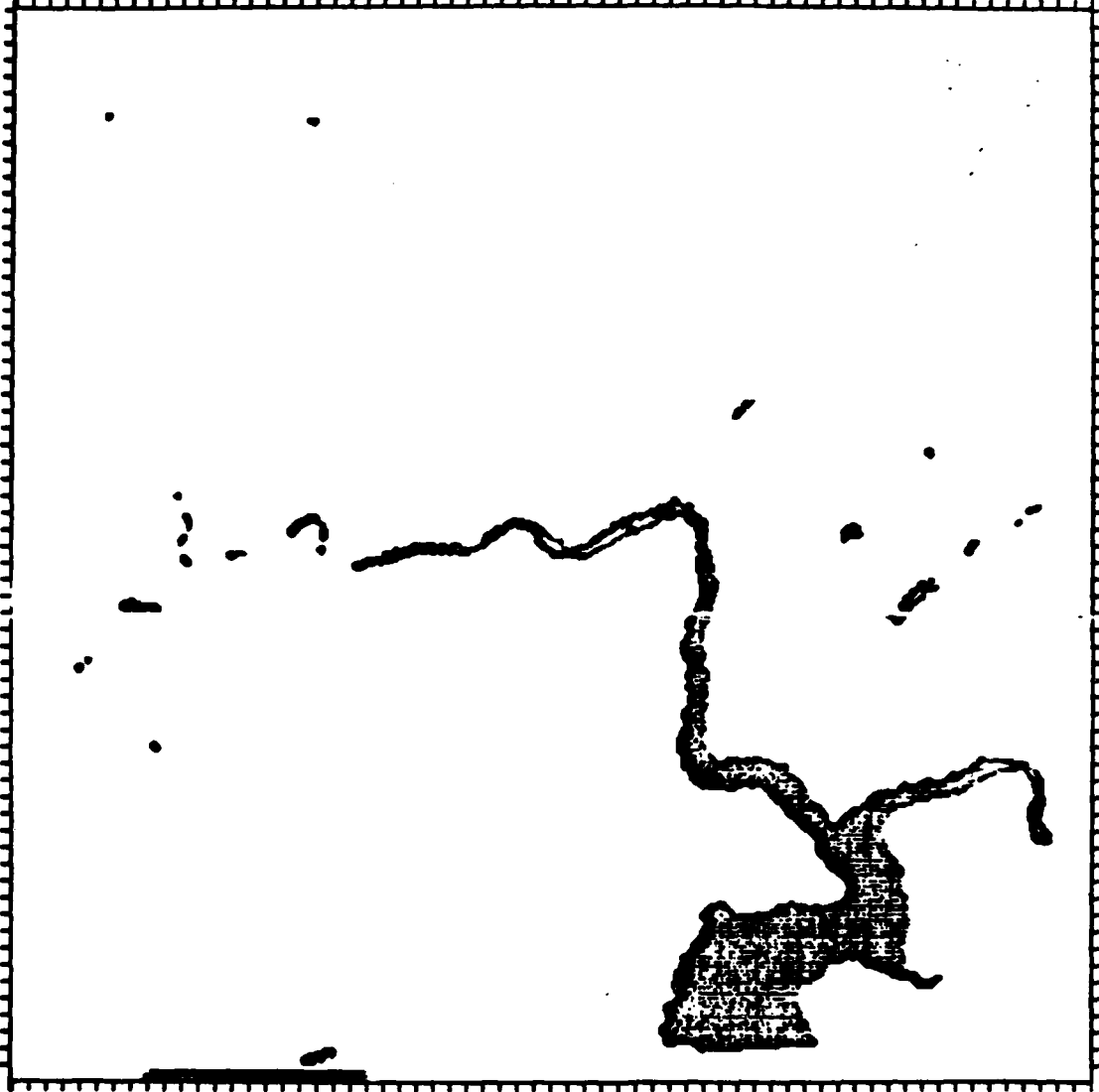


Figure 42. Smoothed Version of Figure 41.

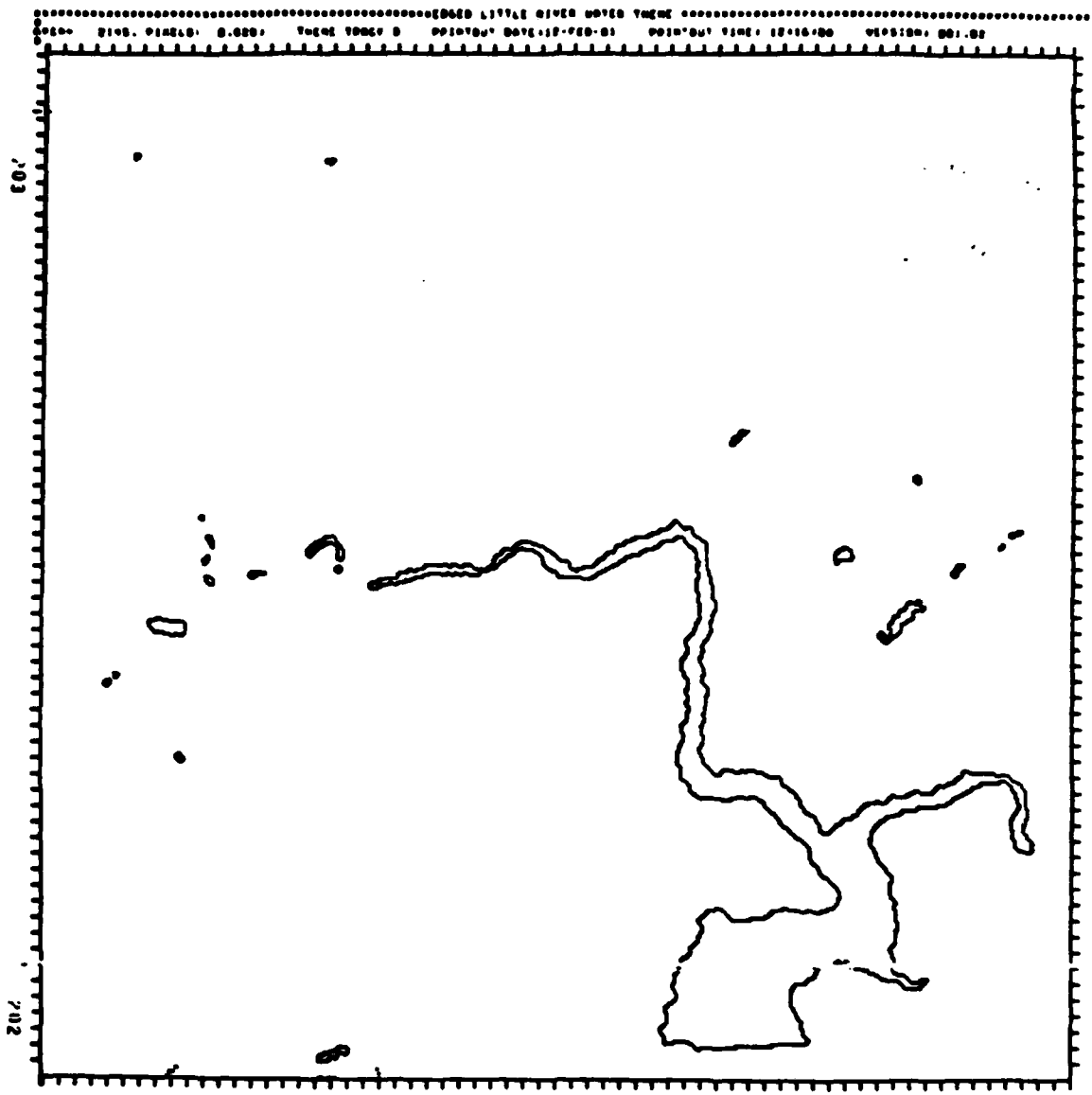


Figure 43. Edged Version of Figure 42

.....CHAPL CREEK WATER COURSE THEME FROM SAR DATA AND PAN CHROM PHOTO.....
FILE: 10000.P10E.01 6.7051 THEME TRACK 0 PRINTOUT DATE: 10-20-81 PRINTOUT TIME: 11:07:50 SECTION: 001.02

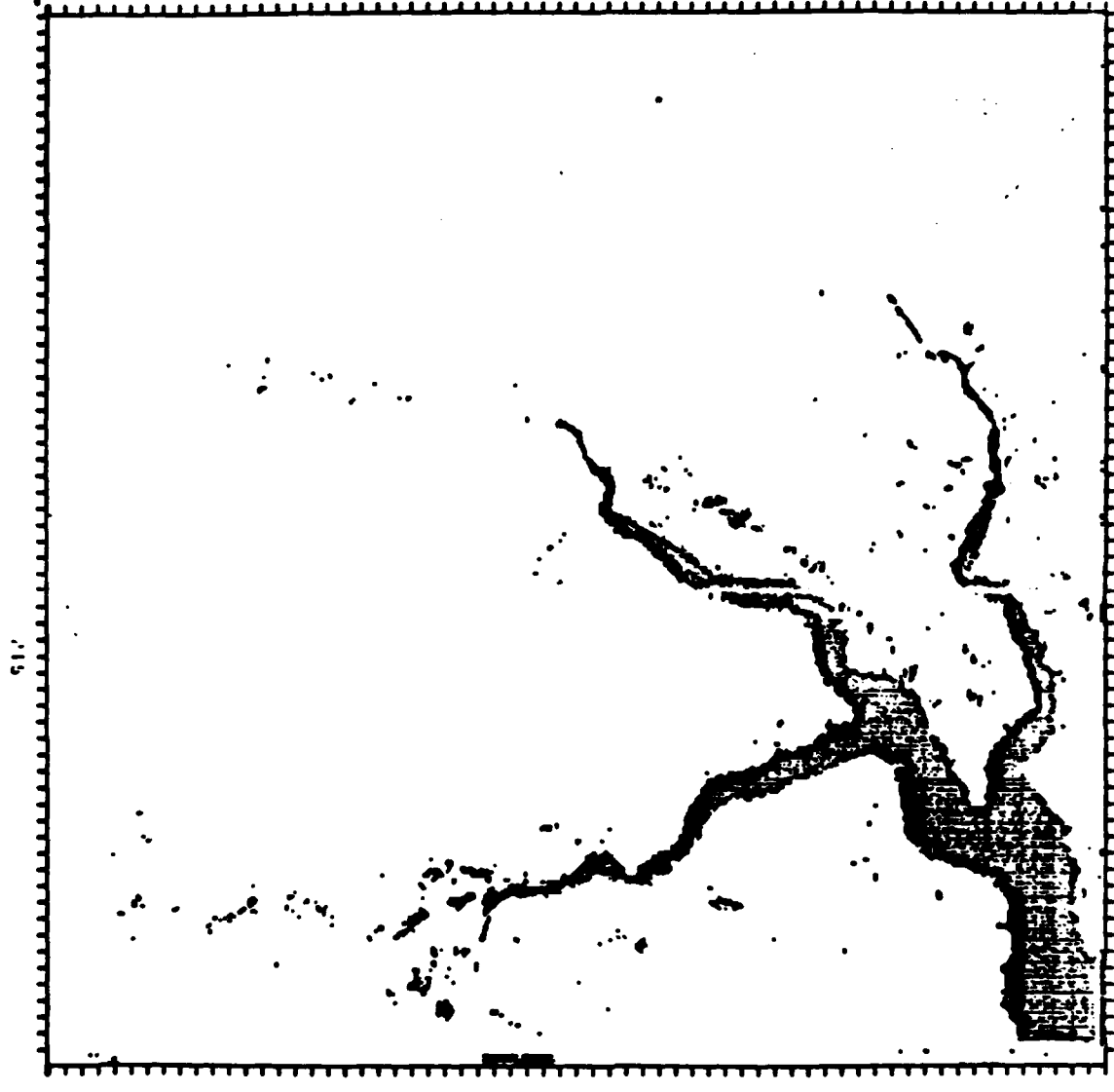


Figure 45. Chapel Creek Watercourse Theme Derived from Panchromatic Photography and SAR Data and from the Smoothed Microtexture Version of the Photography.

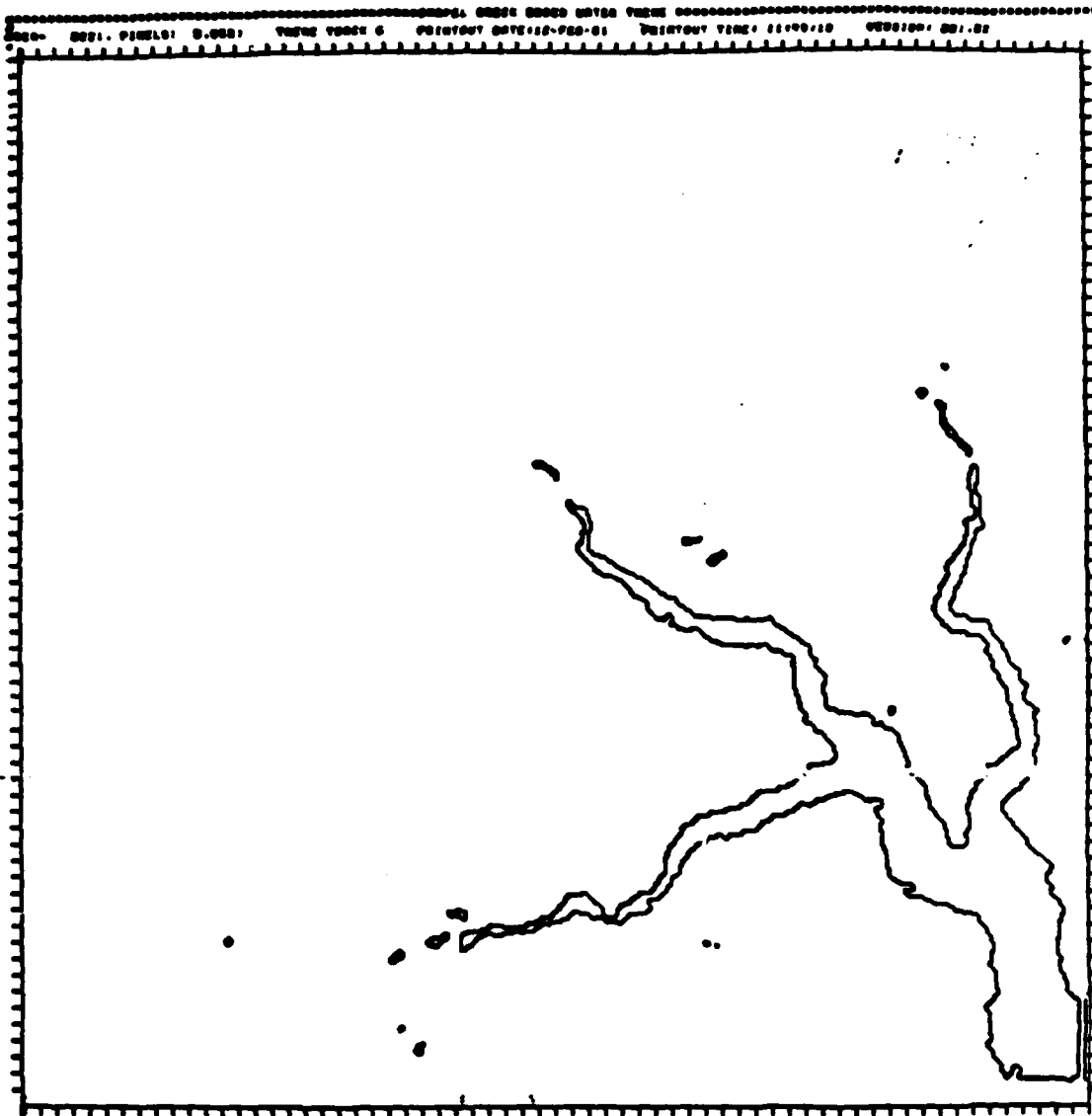


Figure 47. Edged Version of Figure 46.

4. Watercourse Extracting Using Panchromatic Imagery Digitized of Various Scales

A section of the panchromatic image of the Chapel Creek watercourse was digitized at multiple scales to determine what resolution of the derived micro-texture data contains the most separable watercourse class. Preliminary results indicate that panchromatic resolution between 1 and 2 meters yields microtexture data which best discriminates the watercourse class. This also, to some degree, verifies results for vegetation extraction where similar panchromatic resolutions yielded micro-texture data which contained best separability of vegetation classes.

- Alignment of all Water Bodies

Given a suitable thematic classification of a watercourse, the alignment or centerline can be extracted by the methods detailed in Section III. Figure 12 shows the alignment of watercourse as extracted by the Golay method.

- Shore Alignment of all Water Bodies

This is a relatively easy task once water bodies have been categorized. In fact, the recommended procedure is tolerant of classification errors of the omission type and required only a mediocre water body theme. The theme is filtered as required to generate a contiguous map by using the theme filter or a similar operation, and is then outlined by the Golay process. Figure 47 shows a sample alignment theme.

B. Extraction of Dry Gaps and Associated Elements

1. Dry Gaps

Much of the effort involved in producing watercourse factor overlays deals with dry gap width, which is defined as the distance between stream bank shoulders. These width measurements are not usually derivable from a single image. The Terrain Analysis Procedural Guide for Drainage and Water Resources lists two methods of determining dry gap i.e (1) use of topographic information to locate lines of high elevation gradient along the stream bed and (2) locate the shoulder by photo interpretation of stereo pair. While there are interesting possibilities for implementing either of these approaches on an interactive system, no suitable imagery was available resulting in development efforts.

Without elevation information, the extraction of dry gaps is a task best left to the interpreter because of the large amount of judgement required. Dry gap outlines can be quickly entered into digital storage by using a pen/tablet or cursor in conjunction with an image display. Admittedly this is an inelegant solution, but it is also practical. Dry gaps can be located more easily on the basis of vegetation and land form interpretation, than by any unique machine extractable methods.

Once a watercourse and its associated dry gap have been satisfactorily mapped into themes by whatever means, interactive or automated, generation of factor overlays containing watercourse delineation and categorization still entails a series of tasks which are labor intensive and fairly tedious. Tasks which could be done effectively by computer include, the classification of water course segments, dry gap width, encoding of water course delineations by class, and the location of terminal points.

Even with the unfeasibility of extracting dry gaps from the available imagery, it was felt that interactive computer methods could be of value in the extraction of these elements rather than not investigate the elements, some analysis was performed with a manually extracted dry gap as a starting point.

To demonstrate some of the potential tasks, a panchromatic image of Tennessee was used. The scene contained an unusually diverse set of watercourses with dry gap widths ranging from less than 3 to greater than 50 meters. Both the watercourse outline and dry gap outline were classified by manual means with an operator moving a joystick controlled cursor over the image.

While the resulting dry gap theme probably contains errors, it is typical of what can be done on the basis of land form and vegetation analysis with no available elevation data. Figure 48 and 49 show the results

2. Categorizing Watercourse by Dry Gap Width

A sequence which could lend itself to automated processing can be used with watercourse and dry gap themes stored. The first step is to employ a pixel stripping algorithm such as the Golay processor to locate the center line (delineation) of both themes. Figure 50 and 51 show the results

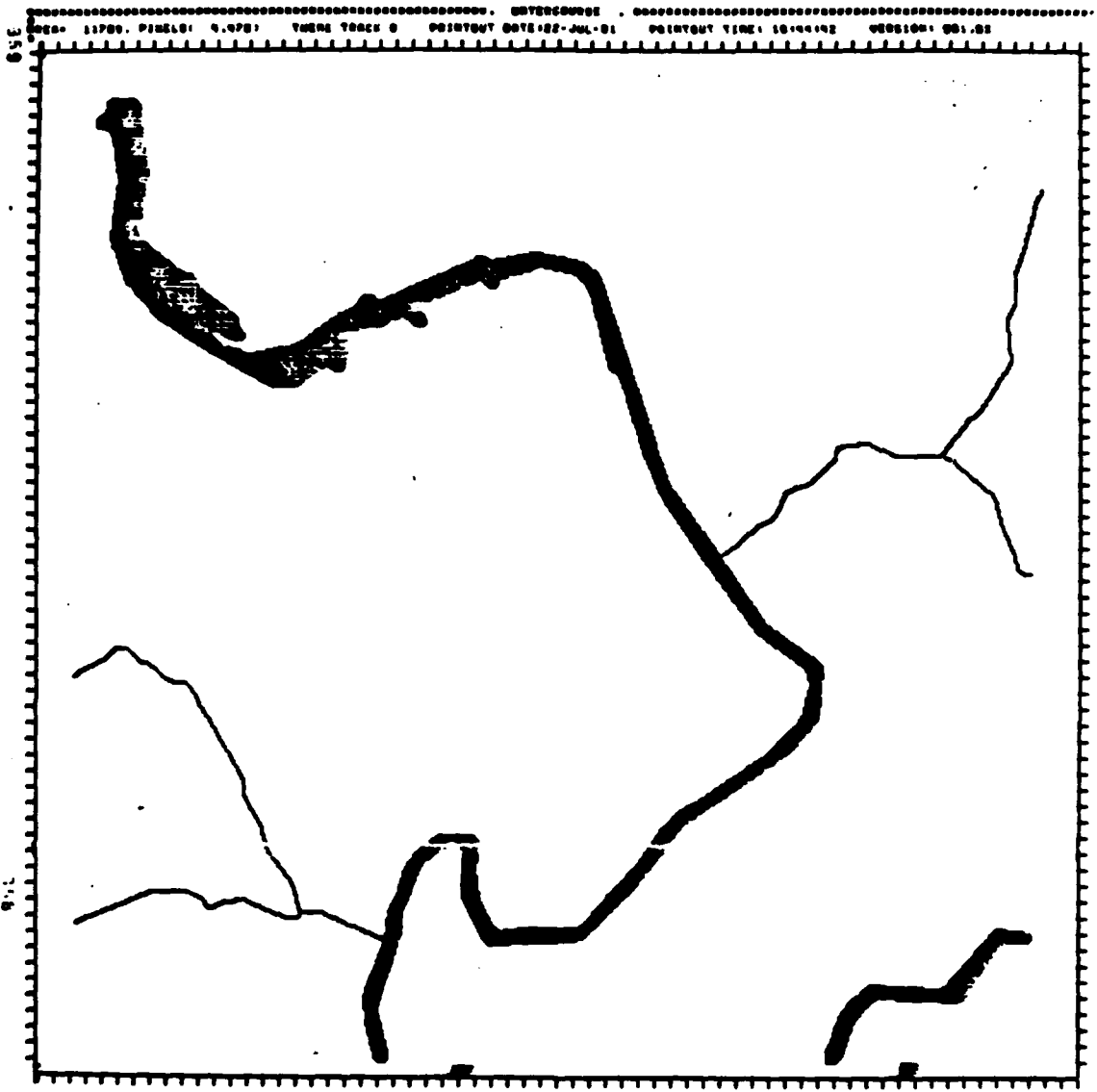


Figure 48. Manually Interpreted Watercourse

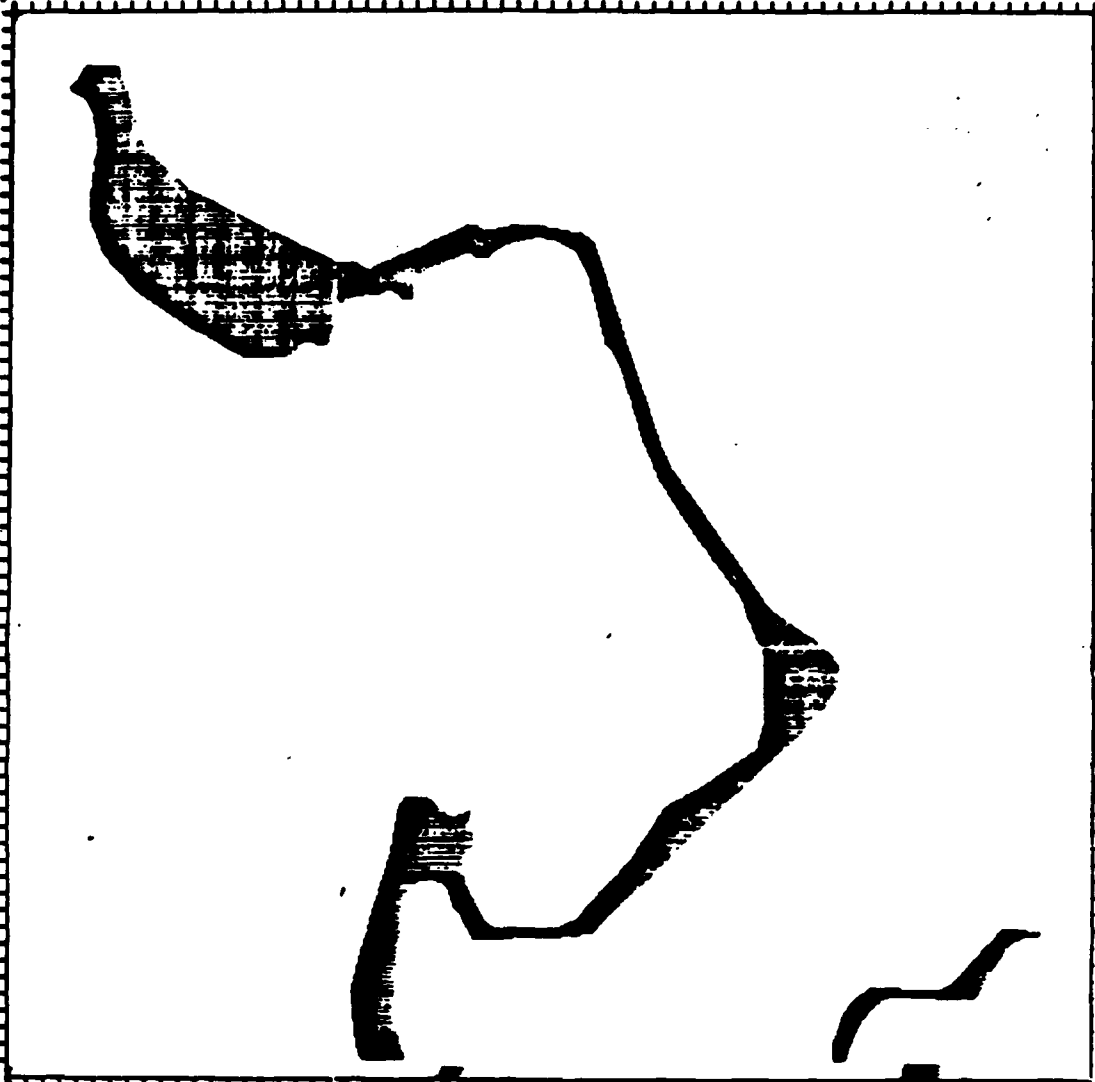


Figure 49. Manually Interpreted Dry Gap

WATERCOURSE SEGMENTS
PREP: 1926 PIXELS: 0.700: THREE TRACK 7 PRINTOUT DATE: 22-JUL-81 PRINTOUT TIME: 10:18:17 VERSION: 001.02

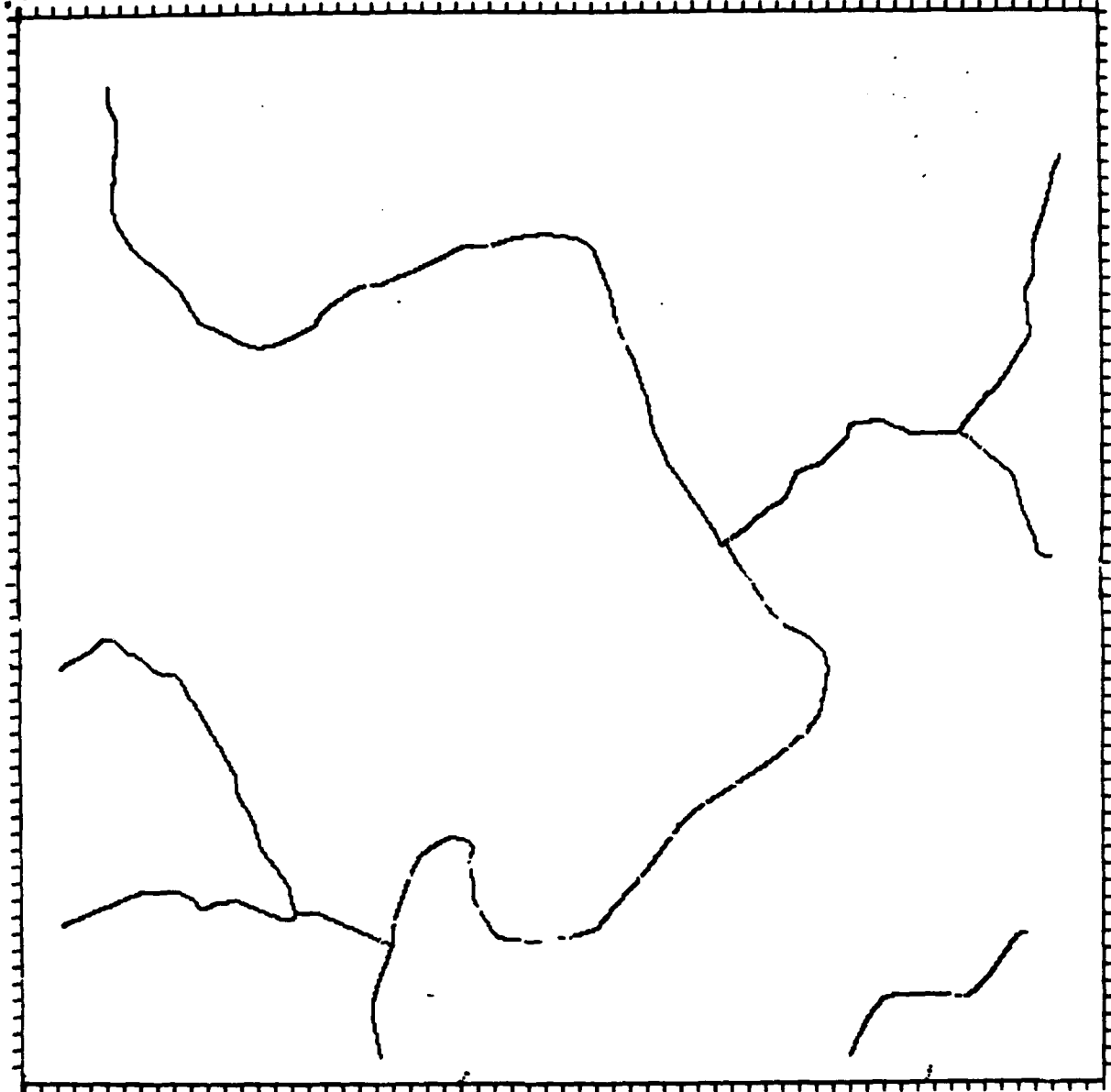


Figure 50. Skeletorized Watercourse

..... DRY GAP CENTERLINE
SER= 1004 PIVELS: 0.418: THREE TRACK 2 PRINTOUT DATE: 22-JUL-81 PRINTOUT TIME: 10:46:17 VERSION: DD: D2

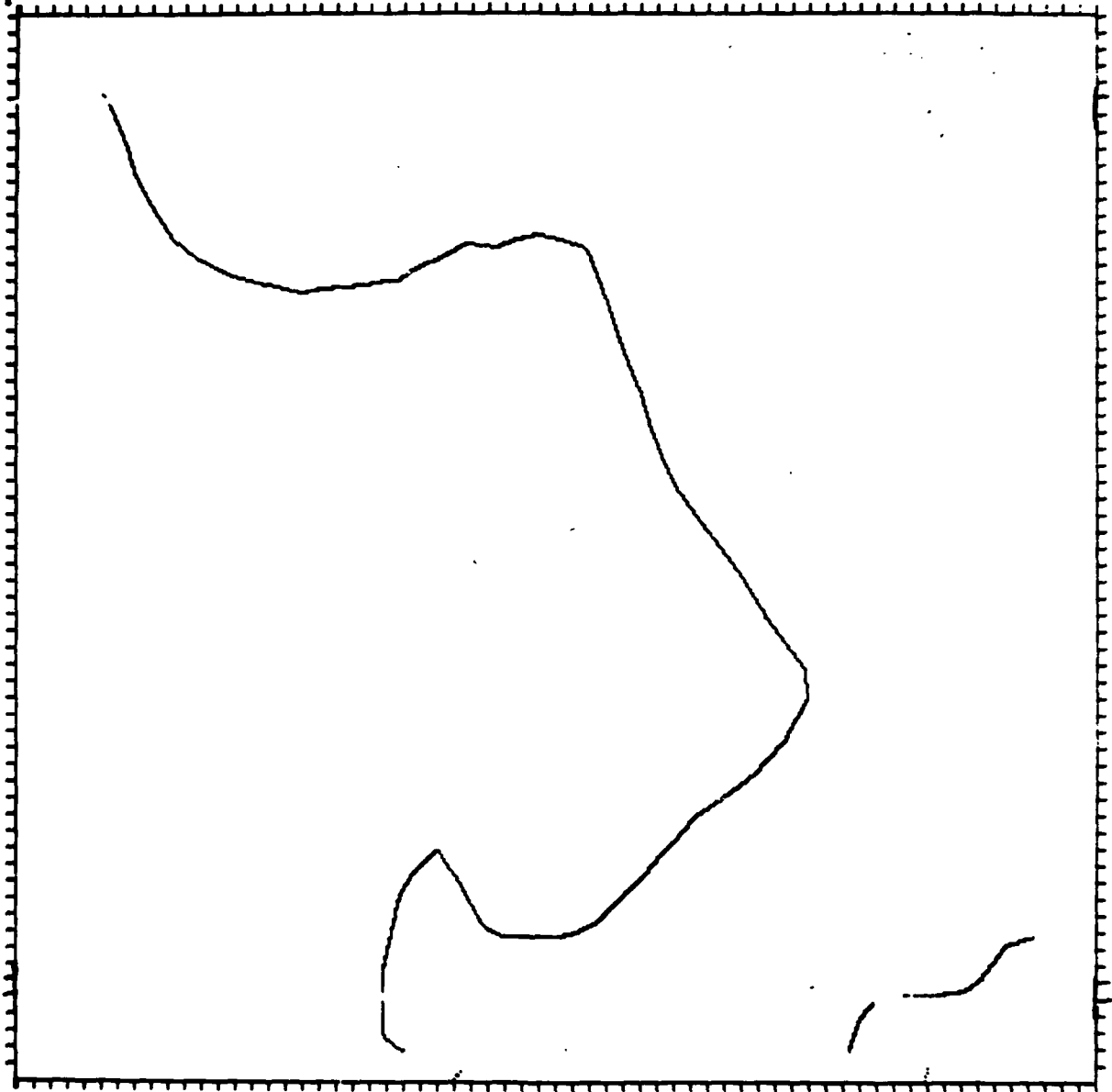


Figure 51. Skeletorized Dry Gap

for this particular image. These results have been touched up to correct some of the previously discussed problems introduced by the Golay skeletonizing algorithm as it is currently implemented on GE DIAL. A correction is needed to the algorithm which assures skeletonizing to a single pixel width line with no loss of connectivity.

The next step basically measures dry gap width along the entire water course by reapplying the Golay process to strip from the dry gap theme, annular rings of pixels corresponding to the desired width increments. In contrast to the previous process, the dry gap is not skeletonized but is allowed to disappear as the stripping continues. Five sequential applications of stripping can be used to yield a set of themes showing areas of dry gap greater than 3, 10, 18, 25 and 35 meters.

These themes are then logically combined with the dry gap center line producing a series of center line segments where each segment maps a dry width interval. Figure 52 is a flow chart describing the algorithm segments.

The desired final result is, of course, not a categorized dry gap center line, but is a categorized watercourse center line. The final steps of the sequence address this problem. A new set of themes must be generated, which categorize the entire area covered by the dry gap into segments by width interval. This is accomplished by applying the theme filter to the dry gap and its center line segments. Filter constraints are set up so that the dry gap theme is totally replaced with the theme corresponding to the closest center line segment. Finally, this set of themes is logically combined with the watercourse center line (or boundaries) to produce the desired categorization. See Figures 53 and 54.

3. Terminal Points of Watercourse Segments

At this point in the suggested processing sequence, we have supposedly performed a successful categorization of the watercourse based upon its associated dry gap width. The result is a series of themes, each containing segments of the watercourse delineation (centerline) corresponding to a given dry gap width increment. From any of these themes it is a simple matter to locate end points of the segments.

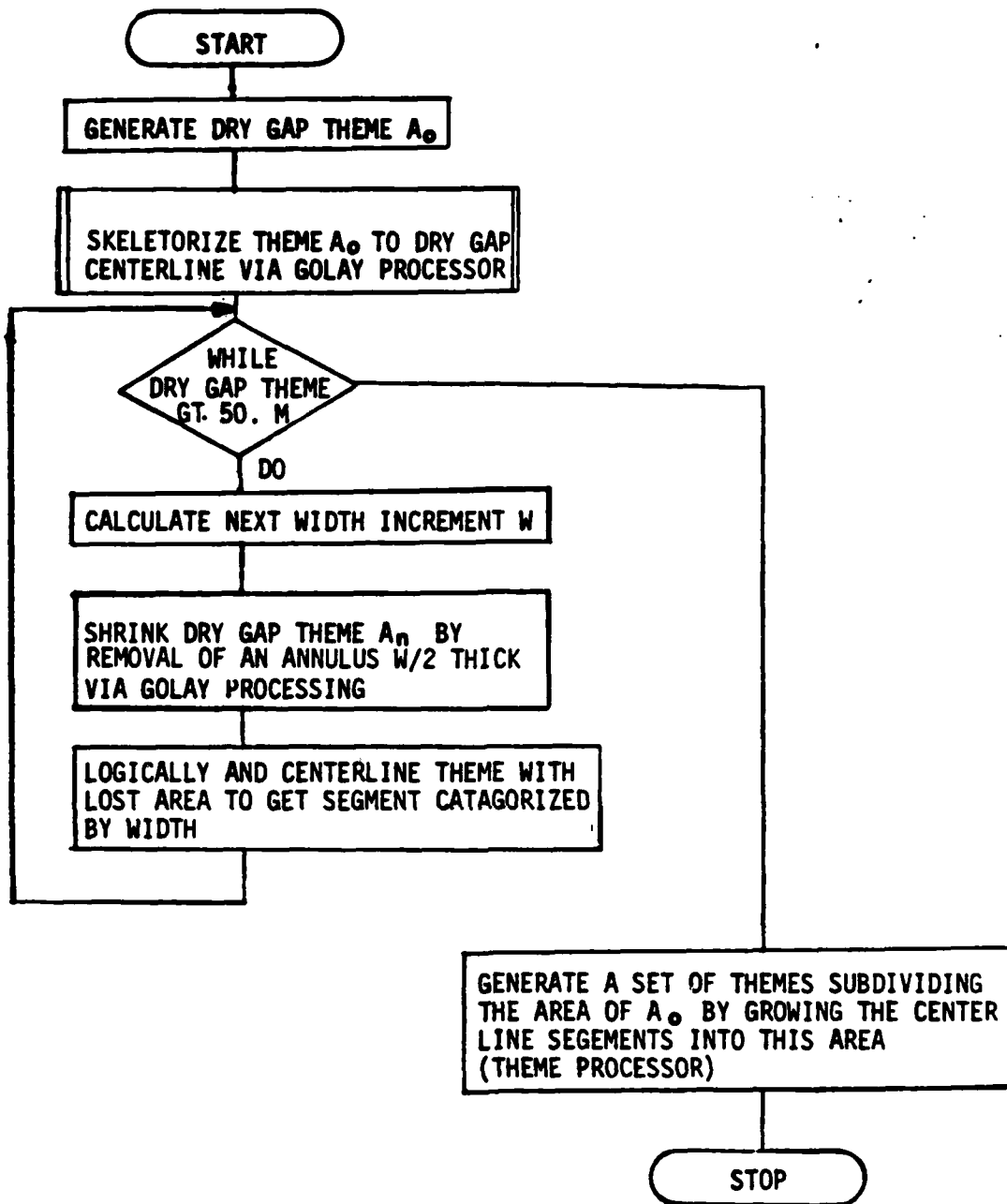


Figure 52. Categorizing a Dry Gap By Width

..... DRY GAP CATEGORIZED BY WIDTH
Area: 2727. PIMSL: 1.045: THERM TRACER 2 PRINTOUT DATE: 02-JUL-01 PRINTOUT TIME: 10:01:10 VERSION: 001.02

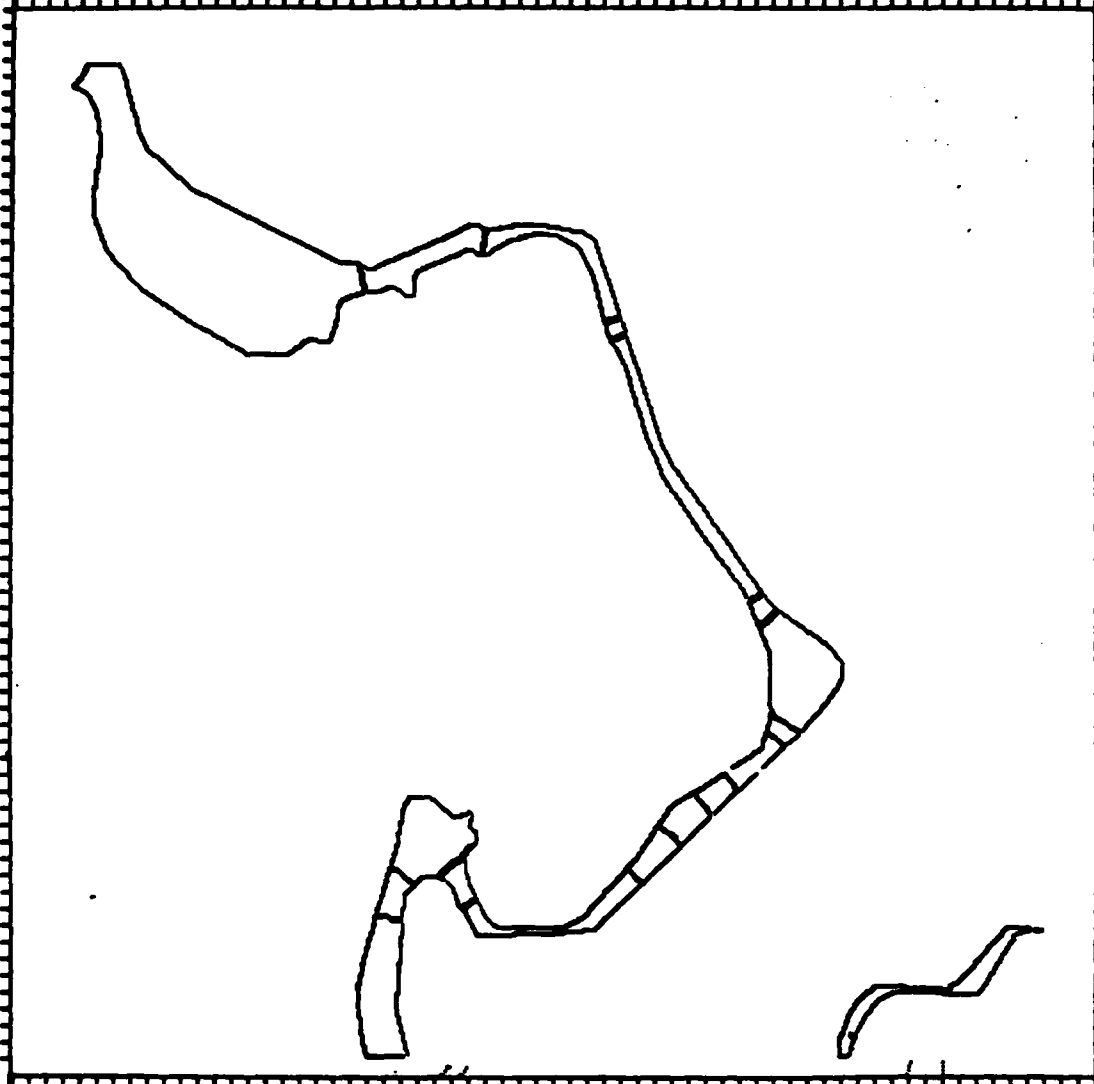


Figure 53. Dry Gap Categorized by Width and Displayed as Polygons

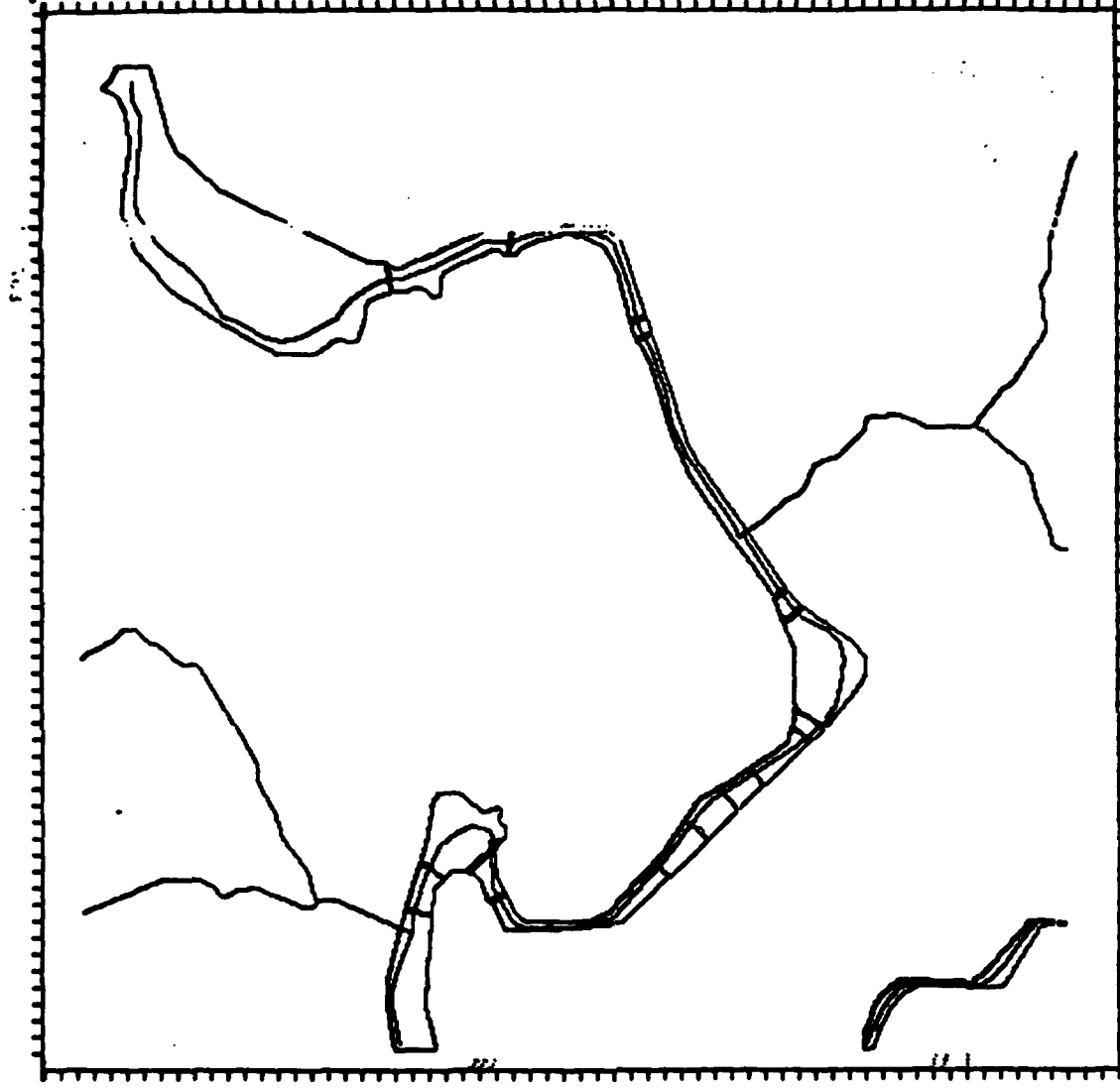


Figure 54. Combining Watercourse Center Line With Dry Gap Categories. Performing a Logical Intersection Produces Watercourse Segments Categorized Width.

Golay processing is a very straight forward means of finding segment end points. The procedure is to operate on the desired width increment theme by deleting all pixels with surrounds containing more than one pixel. Using the numbering convention in Figure 10, surrounds 2 thru 4 would be set for deletion. The output theme contains only the end points of connected segments in the input theme. The operation is completed in one pass over the image and is independent of the number of segments being processed.

For the end points of watercourse segments with less than three meter dry gaps an additional operation may be desirable. In this case, both terminal points of the segments are not used in factor overlays, only the points where the watercourse transitions to greater than three meter dry gap width. The same Golay operation applied to the entire watercourse delineation serves to locate points where the stream disappears. Subtracting these stream end points from the three meter segment end point leaves only the desired transitional points.

4. Encoding Watercourse Delineation

After categorization, the encoding of watercourses by line dashing, can and should be done by automated means. All the dry gap width symbols specified in the Terrain Analysis Procedural Guide for Drainage and Water Resources can be generated from a series of points equally spaced along the watercourse center line. Figure 55 is an algorithm (untested) which could be used to generate these points and Figure 56 lists methods by which the dot pattern could be used to encode lines with the desired patterns. Note that the feature encoding procedures are all raster (line by line) algorithms as opposed to line following algorithms. The raster processing approach has the advantages of neither being affected by image size, complexity, nor an occasional broken or non contiguous watercourse.

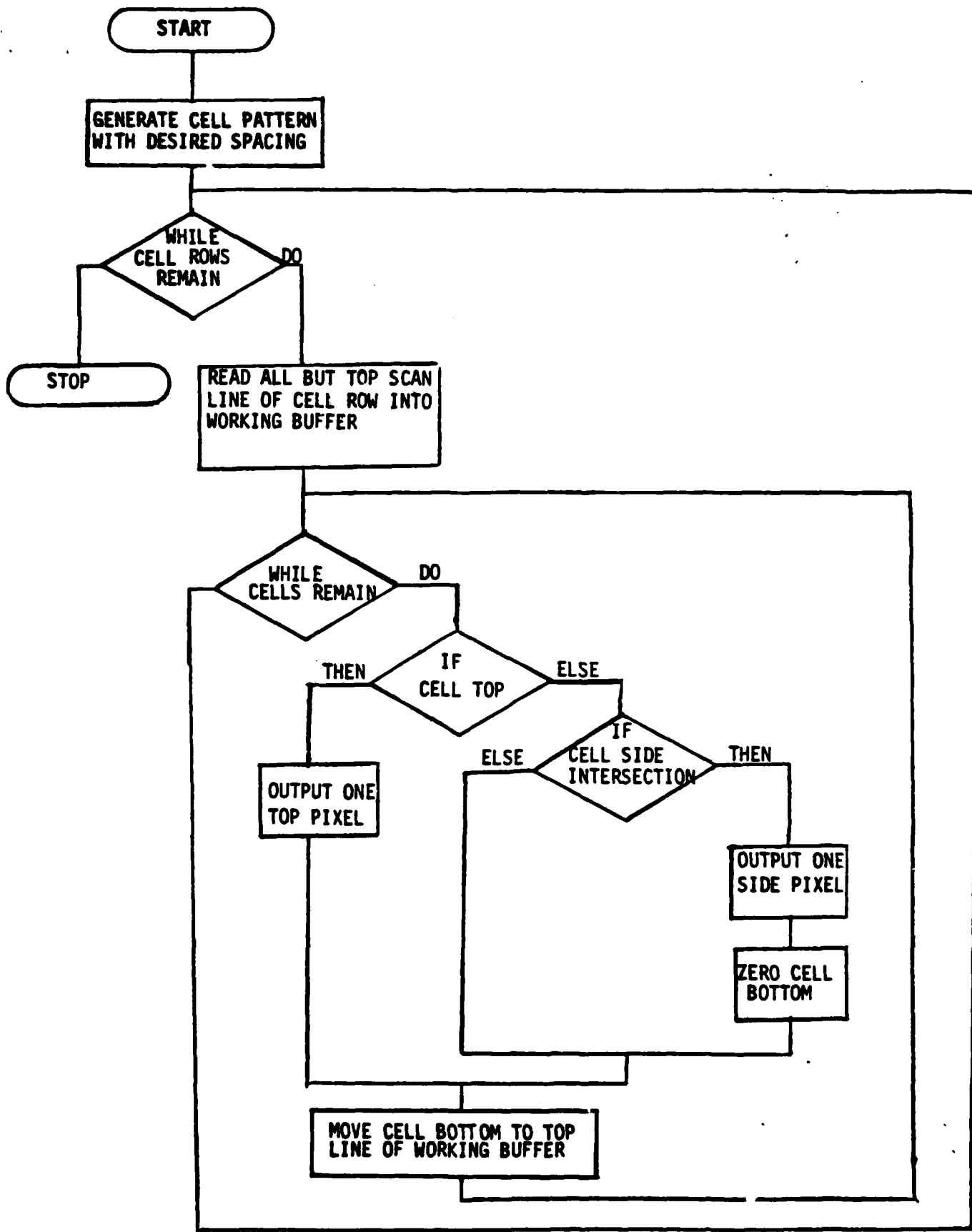


Figure 55. Raster Operating Line Dotting Algorithm

DRY GAP WIDTH (meters)	ENCODING PROCEDURE	RESULT
$W < 3$	<ol style="list-style-type: none"> GENERATE CLOSELY SPACED DOTS FROM SINGLE PIXEL WIDTH WATER COURSE CENTER LINE. GROW DOTS INTO SMALL CIRCULAR AREAS VIA GOLAY PROCESSING 	
$3 < W < 10$	<ol style="list-style-type: none"> GENERATE WIDELY SPACED DOTS FROM CENTER LINE GROW DOTS INTO CIRCULAR AREAS VIA GOLAY PROCESSING OUTLINE CIRCULAR AREAS VIA GOLAY PROCESSING SUBTRACT OUTLINE FROM CENTER LINE 	
$10 < W < 18$	<ol style="list-style-type: none"> GENERATE MODERATELY SPACED DOTS FROM CENTER LINE GROW DOTS TO SMALL CIRCULAR AREAS SUBTRACT AREAS FROM CENTER LINE 	
$18 < W < 25$	<ol style="list-style-type: none"> USE UNMODIFIED CENTER LINE 	
$25 < W < 35$	<ol style="list-style-type: none"> USE PROCEDURE FOR $10 < W < 18$ GROW DASHED LINE VIA GOLAY PROCESSING 	
$35 < W < 50$	<ol style="list-style-type: none"> GROW CENTER LINE VIA GOLAY PROCESSING 	

Figure 56. Encoding Watercourse Delineation

V. CONCLUSIONS

Phase 1 and Phase 2 of the study have concentrated on the feasibility and use of man-machine interactive digital processing techniques for the extraction of terrain data elements from aerial imagery. Interactive digital techniques were developed for vegetation/land cover boundary extraction and for extraction of forest-related data elements. These techniques included the use of spatial crown and texture operators.

Applications of these operators during Phase 1, enabled the extraction of feature roughness at several scales; extraction of vegetation types, wet areas and water bodies; and the determination of forested areas, forest canopy and locations of individual trees. The overall success of Phase 1 suggested the need to further test the operators relative to water resources data element and to improve and test the spatial crown and texture operators at different scales.

The development and coding of software included programs in image resolution and scale changing, feature skeletonizing, and theme smoothing. These programs are implemented on the GE DIAL system, primarily in Fortran language. Limited testing of the programs have been completed in Phase 2 to demonstrate usefulness. The implementation of these programs on systems similar to GE DIAL, is feasible.

Extraction of water resources data elements requires outlining features such as vegetation boundaries and shore line delineation of water bodies. Through spatial binary processing, or Gokay processing, features were quickly and flawlessly outlined.

A slight variation of outlining, line stripping, provided a feasible, but somewhat lengthy (multiple pass) mean of categorizing watercourses by the width of their manually mapped dry gaps. Terminal points of watercourse segments can also be located by one pass Gokay processing. Efforts to delineate watercourses, the prime water resources element, were only partially successful. The seemingly simple problem was to connect separated pixels of a watercourse theme, and then shrink or skeletonize the feature into a single pixel width centerline. Satisfactory delineation was achieved for certain cases but no universally applicable techniques were found.

AD-A114 815

GENERAL ELECTRIC CO LANHAM MD SPACE SYSTEMS DIV F/8 9/2
INTERACTIVE DIGITAL IMAGE PROCESSING FOR TERRAIN DATA EXTRACTIO--ETC(U)
SEP 81 T F WESCOTT, W C DALLAM, C J PETERSON DAAK70-79-C-0153
ETL-0277 NL

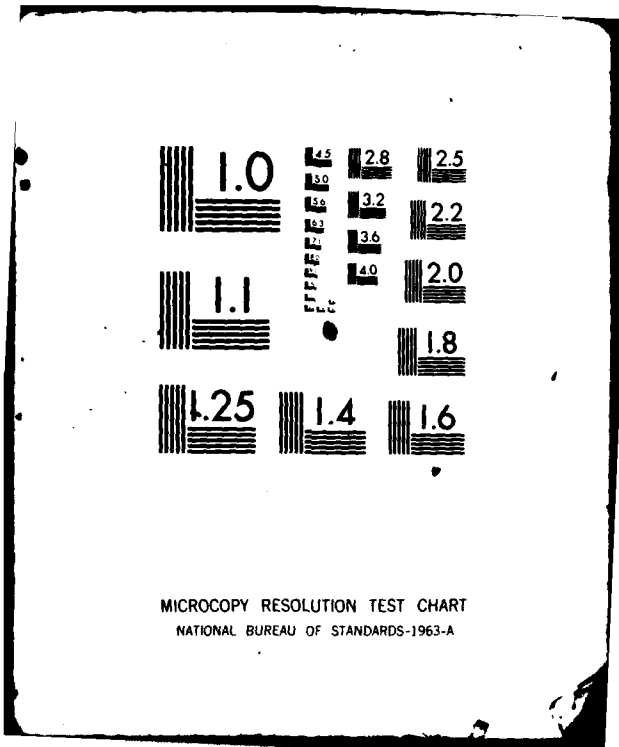
UNCLASSIFIED

2 - 2

AD-A114 815



END
DATE
FILMED
6-82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The production of usable factor overlays, which are the desired result of terrain element extraction was addressed and handled by writing what is thought to be an important theme filtering program. Its basic purpose was to perform spatial low pass filtering on a categorized image as required to generate overlays which resemble those made by photointerpretation and which are spatially simple enough to be combined into complex overlays. Provisions are made to restrict change of critical or exceptionally accurate themes and to remap on a spatial basis those which are incidental or plagued with categorization errors.

VI. RECOMMENDATIONS

In the performance of Phase 2, areas were identified as candidates for further investigation. Generally, these candidates include work described as testing of existing software/techniques and development of new software/techniques. The recommended areas for future work are:

- Continue to perform sufficient testing of techniques developed in both Phase 1 and Phase 2. The testing should include various sites and ground data to radiate the techniques.
- Limited results were obtained for delineation of line features, i.e., streams and roads. It is believed that the existing Golay algorithm, an already invaluable tool, could be revised and made to perform error free skeletonizing.
- Further testing of the theme filtering program is suggested. This testing is relative to the production of factor overlap, i.e. a comparison of manual interpretation vs. computer. The theme filtering program has proven valuable as a cosmetic noise cleaner and as a tool in the feature extraction process. It allows satisfactory results to be filtered from otherwise unacceptable themes. Although not tested in this respect, it is also seen as a promising means of digitally combining selected themes into complex overlays. While highly effective, the program is somewhat cumbersome to set up, and the operator interface should be rewritten so that commonly used filtering options are menu selectable. With the minor change the program is recommended for general interactive use.
- To date, processing of imagery data has been restricted to small geographical areas. The testing of developed techniques from Phase 1 and 2 to mosaicked images is needed relative to the practical uses of terrain analysis. It is recommended that such work be performed to allow for the expansion of the techniques to larger geographical areas.

APPENDIX A
IMAGE RESOLUTION AND
SCALE CHANGING PROGRAM

CC
CC

IMAGE SCALE AND RESOLUTION CHANGING

THIS PROGRAM PROVIDES A MEANS OF MODIFYING IMAGE RESOLUTION TO SIMULATE ANY SENSOR WITH LOWER RESOLUTION THAN THAT INHERENT IN THE ORIGINAL IMAGE DATA, AND TO PERFORM GEOMETRIC SCALING.

ITS MAIN PURPOSE IS TO MODIFY IMAGES IN A MANNER THAT ALLOWS STANDARD SPATIAL OPERATORS TO BE APPLIED TO ANY IMAGE AT A VARIETY OF SCALES. THIS EXTRACTS SPATIAL INFORMATION AT A VARIETY OF FREQUENCIES WITH MINIMAL COMPUTATION. THE SUGGESTED PROCESSING SEQUENCE IS TO USE THIS PROGRAM TO DEGRADE RESOLUTION TO THE POINT WHERE A DESIRED FEATURE DISPLAYS UNIQUE SPATIAL PROPERTIES. APPLY A STANDARD OPERATOR, AND THEN RESCALE THE OUTPUT BACK TO ITS ORIGINAL SIZE FOR SUBSEQUENT FEATURE EXTRACTION.

FORTMAN PROGRAM MODULES:

- EGANDO ---A DRIVER THAT CALLS THE MODULES NECESSARY TO PERFORM ANY SELECTED SCALING OPERATION.
- INDDO ---PROVIDES AN INTERACTIVE METHOD OF DEFINING INPUT AND OUTPUT IMAGE SEGMENTS. MAKES USE OF THE IMAGE-100 JOYSTICK CONTROLLED RECTANGLE/LAP CURSOR.
- SELECT ---ACCEPTS OPERATOR SELECTIONS OF INPUT AND OUTPUT IMAGE PLANES OF CHANNELS IN THE IMAGE 100 REFRESH MEMORY.
- MMANDO ---PERFORMS IMAGE MAPPING VIA NEAREST NEIGHBOR INTERPOLATION.
- AMANDO ---PERFORMS IMAGE MAPPING VIA COMPUTATION OF LOCAL AREAL AVERAGES. USED TO SMOOTH OR SHRINK IMAGES.
- CMANDO ---PERFORMS IMAGE MAPPING VIA A SUBPIXEL CONTEXT DEPENDANT INTERPOLATION. USED TO ENLARGE IMAGES.

PDP-11 ASSEMBLER MODULES:

- LNOSCAL ---CALLED BY MMANDO TO REMAP PIXELS ALONG A SCAN LINE
- COLSUM ---CALLED BY AMANDO TO MAINTAIN RUNNING SUMS ALONG IMAGE COLUMNS
- ROWSUM ---CALLED BY CMANDO TO COMPUTE RUNNING AVERAGES ALONG IMAGE ROWS

IMAGE 100 LIBRARY SUBROUTINES:

- FRONT ---FREE FORMAT INPUT ROUTINES
- IPK ---CURSOR CONTROL ROUTINES
- IPU


```

0001 C SUBROUTINE BGANDU
0002 C
0003 C
0004 C
0005 C
0006 C
0007 C
0008 C
0009 C
0010 C
0011 C
0012 C
0013 C
0014 C
0015 C
0016 C
0017 C
0018 C
0019 C
0020 C
0021 C
0022 C
0023 C
0024 C
0025 C
0026 C
0027 C
0028 C
0029 C
0030 C
  
```

CCCCCCCCCCCCCCCCCC GENERAL * ELECTRIC CCCCCCCCCCCCCCCCCC
 GENERAL CHANNEL TO CHANNEL WINDOWING PROGRAM
 CCCCCCCCCCCCCCCCCC
 IMPLICIT INTEGER(A-G,S-Z)
 DIMENSION BUF(40)
 LOGICAL INSERT
 COMMON /WINDO/ IX1,IY1,IX2,IY2,OX1,OY1,OX2,OY2,RTAG,INSERT
 COMMON /CHNL/ INCHS(5), OUCHS(5)
 DATA UTY /6/
 DATA N'N' /, Y'Y' /
 DATA I'1' /, C'C' /, IR'R' /
 CCCCCCCCCCCCCCCCCC
 CLEAR DISPLAY
 CALL OUTRIT(27,12,7)
 DETERMINE WINDOW MAPPING
 CALL WNDCH
 CALL OUTPUT(27,12,7)
 WRITE(UTY,999)IX1,IY1,IX2,IY2,OX1,OY1,OX2,OY2,RTAG
 999 FORMAT(/, 'MOVE INPUT', ('13',,13,'); ('13',,13,') /
 , ' TO OUTPUT', ('13',,13,'); ('13',,13,') /
 , ' MAGNIFICATION = ',FS.2,/)

SELECT CHANNELS
 CALL SELECT(INCHS,OUCHS)
 ASK FOR MAPPING TECHNIQUE
 TECHNI=N
 IF(RTAG.EQ.1.0) GO TO 120
 WRITE(UTY,1010)
 1010 FORMAT(/,'\$(C)COMPUTED OR (N)NEAREST NEIGHBOR INTERPOLATION >')
 READ(UTY,1020)BUF
 1020 FORMAT(40A2)
 CALL FRONT(BUF)
 IF((BUF(1).NE.N).AND.(BUF(1).NE.C)) GO TO 101
 TECHNI=BUF(1)
 120 CONTINUE
 ASK FOR OUTPUT METHOD
 DELTAX=OX2-OX1
 DELTAY=OY2-OY1
 INSERT=.FALSE.

report window
corner locations

prompt for desired
resampling techniques

VERSION 10 0010-03
DATE 12-11-81 14:08:57

DB:BGANDD.LP:DB:BGANDD

```
5 08  
1 0001 C IF((DELTA.GT.510).AND.(DELTA.GT.360)) GO TO 130  
1 0002 IF(ITE(JTY,1290)  
1 0003 PUNTY/,(1)INSERT OR (R)EPLACE OUTPUT >')  
1 0004 1290 FEND(JTY,1020)BUF  
1 0005 CALL FRONT(BUF)  
1 0006 .IF((BUF(1).NE.1).AND.(BUF(1).NE.IR)) GO TO 129  
1 0007 .IF(BUF(1).EQ.1) INSERT=.TRUE.  
1 0008 130 CONTINUE  
1 0009 C DETERMINE PROCESSING CASE  
1 0010 C NEAREST NEIGHBOR CASE? CALL NNAND  
1 0011 C IF((TECHN.EQ.N).OR.(RFRG.EQ.1.0))  
1 0012 C INTERPOLATION CASE? CALL CDAND  
1 0013 C IF((TECHN.EQ.C).AND.(RFRG.GT.1.0))  
1 0014 C AVERAGING CASE? CALL AAAND  
1 0015 C IF((TECHN.EQ.C).AND.(RFRG.LT.1.0))  
1 0016 C RETURN  
1 0017 C STOP  
1 0018 C END
```

prompt for
output method

call appropriate
resampling module...

SUBROUTINE WINDOW

CCCCCCCCCCCCCCCC GENERAL * ELECTRIC CCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCC PROCESSING WINDOW DEFINITION VIA CURSOR POSITIONING

CCCCCCCCCCCCCCCC CCC TFM 11/80

0002 C IMPLICIT INTEGER(A-D,S-Z)

0003 C DIMENSION BUF(40)

0004 C DIMENSION INDIR(S), OUTCUR(S)

0005 C REAL FLOAT

0006 C COMMON /WINDO/ IX1,IY1,IX2,IY2,OX1,OY1,OX2,OY2,RHAG,INSERT

0007 C DATA UTY /6/

0008 C DATA N/'N', Y/'Y' /

0009 C DATA NULL,' ' /

CCCCCCCCCCCCCCCC CCC

0010 C ASK OPERATOR FOR MAGNIFICATION

0011 50 WRITE(UTY,5000)

0012 5000 FORMAT(/,'ENTER MAGNIFICATION (EG. 2:1) >')

0013 100 READ(UTY,1000)BUF

0014 1000 FURTH(40#)

0015 C IF(BUF(1).EQ.X) CALL EXIT

0016 C P=0

0017 C CALL INTFF(P,BUF,EO,NUMBER)

0018 C CALL INTFF(P,BUF,EO,DENOM)

0019 C TAKE CARE OF INPUTS OF ZERO

0020 C IF(NUMBER.EQ.0) GO TO 30

0021 C IF(DENOM.EQ.0) DENOM=1

0022 C RHAG=(FLOAT(NUMBER))/FLOAT(DENOM)

0023 C LOCATE WINDOWS WITH CURSOR

0024 C IF(RHAG.LT.1.0) GO TO 300

0025 C CASE #1 WINDOW WILL BE ENLARGED

0026 C ODX=512

0027 C ODY=512

0028 C OUTCUR(2)=256

0029 C OUTCUR(3)=ODX/2

0030 C OUTCUR(4)=256

0031 C OUTCUR(5)=ODY/2

0032 C WRITE(UTY,2010)ODX,ODY

0033 C FORMAT(/,'OUTPUT WINDOW = ',13,' X ',13,' >')

0034 C READ(UTY,1000) BUF

0035 C IF(BUF(1).EQ.X) CALL EXIT

0036 C IF(BUF(1).EQ.NULL) GO TO 210

0037 C ELSE REPLACE FULL SIZE OUTPUT DEFAULTS

Note: This module must
 be replaced for non
 Image 100 implementation


```
0000 CALL IWK(INCUR)  
0001 WRITE(UTY,3050)  
0002 FORMAT(' $POSITION CURSOR FOR INPUT WINDOW, THEN CR >')  
0003 READ(UTY,1000)NOTHIN  
0004 CALL IWK(INCUR)  
0005 CONTINUE
```

310 C C
C C
SIZE CURSOR TO OUTPUT AREA

```
0094 ODX=(FLOAT(IDX))*XR/MAG  
0095 ODY=(FLOAT(IDY))*YR/MAG  
0096 CALL IWK(OUTCUR)  
0097 OUTCUR(3)=ODX/2  
0098 OUTCUR(5)=ODY/2  
0099 CALL IWK(OUTCUR)  
0100 WRITE(UTY,3150)  
0101 FORMAT(' $POSITION CURSOR FOR OUTPUT WINDOW, THEN CR >')  
0102 READ(UTY,1000)NOTHIN  
0103 CALL IWK(OUTCUR)  
0104 CONTINUE
```

400 C C
C C
SET WINDOWS IN COMMON

```
0105 IX1=INCUR(2)-IDX/2  
0106 IY1=INCUR(4)-IDY/2  
0107 IX2=IX1+IDX  
0108 IY2=IY1+IDY  
0109 OX1=OUTCUR(2)-ODX/2  
0110 OY1=OUTCUR(4)-ODY/2  
0111 OX2=OX1+ODX  
0112 OY2=OY1+ODY  
0113 RETURN  
0114 END
```


Image loop

```
0037 DO 500 BPTR = 1.5  
0038 BNDI = INCHS(BPTR)  
0039 BNDU = OUCHS(BPTR)  
C C C  
C C C C  
0040 SKIP UNWANTED BANDS  
IF(BNDI.EQ.0) GO TO 450  
, ELSE PROCESS LINES  
SET UP TABLE LOOKUP  
MULTIP=1  
DO 205 J=0,255  
TEMP=MULTIP*KJ  
TEMP=MIN0(TEMP,255)  
TAB(J+1)=TEMP  
CONTINUE  
205 C  
C C  
C C  
0042 IF(INSERT) GO TO 211  
0043 ELSE ERASE LINES ABOVE OUTPUT WINDOW  
0044 DO 210 LINE = 0,OY11-1  
0045 CALL IAU(BNDU,LINE,ZEROS)  
0046 CONTINUE  
0047 END IF  
C C  
C C  
0050 CONTINUE  
0051 LASTDY = -1  
0052 DO 400 LINE=OY11,OY21  
0053 NEWDY = DY(LINE)  
C C C  
C C C  
0057 SKIP PROCESSING FOR DUPLICATED LINES  
IF((NEWDY.EQ.LASTDY).AND.(.NOT.INSERT))GO TO 350  
ELSE PROCESS NEW OUTPUT LINE  
0059 CALL IAU(BNDI,NEWDY,IB)  
MOVE DATA TO OUTPUT BUFFER  
C C C  
C C C  
0060 IF (INSERT) CALL IAU(BNDU,LINE,OB)  
0062 CALL LNSCAL(1B,TAB,OB,DX,DX1,LENGTH)  
END IF  
0063 CONTINUE  
C 350  
C C C  
C C C  
0064 WRITE OUTPUT LINE TO IAC  
0065 CALL IAU(BNDU,LINE,OB)  
0066 LASTDY = NEWDY  
CONTINUE  
C 400  
C C  
C C  
0067 IF(INSERT) GO TO 411  
ELSE ERASE LINES BELOW OUTPUT WINDOW  
DO 410 LINE = OY21+1,511  
CALL IAU(BNDU,LINE,ZEROS)  
CONTINUE  
END IF  
0072 CONTINUE  
C 411
```

clear memory above window

perform nearest
neighbor resampling

clear memory below window

THU 12-NUJ-81 14:09:30 PAGE 003
DB:BGANDG.LP.:DB:BGANDG

001C-03
CORE-11K. UIC-(127.4)

0073 C 450 END IF
0074 C 500 CONTINUE
0075 C RETURN
0076 C END

compute column sums

```
00658 CALL IRV(BNDI,NEB,INI,NEAB)  
00659 CALL COLSUM(NEAB,OLDB,SUMB,NUMPIX)
```

SET STARTING INPUT LINE FOR FUTURE LOOP

```
0070 TOPVAL=BOTVAL +1  
0071 CONTINUE
```

PROCESS OUTPUT LINE

```
0072 IF(INSERT) CALL IRV(BNDO,LINE,OUTB)  
0073 CALL ROWSUM(SUMB,ROWSB,NUMPIX,NCOL,AREA)  
0074 CALL LNSCAL(ROWSB,TAB,OUTB,DX,OX1,LEN)  
0075 CALL IIV(BNDO,LINE,OUTB)  
0076 CONTINUE
```

compute area average
and write output

```
0077 IF(INSERT) GO TO 411  
0078 ELSE ERASE LINES BELOW OUTPUT WINDOW
```

```
0079 DO 410 LINE = OY21+1,511  
0080 CALL IIV(BNDO,LINE,ZEROS)
```

```
0081 CONTINUE  
0082 END IF
```

```
0083 CONTINUE  
0084 END IF
```

```
0085 CONTINUE  
0086 CONTINUE  
0087 RETURN
```

```
0088 END
```

SUBROUTINE COMDO

SCALES AND LOADS A WINDOW VIA CONTEXT DEPENDANT INTERPOLATION

IMPLICIT INTEGER (A-D,S-Z)

REAL FLOAT
 DIMENSION DX(512,2), DY(512,2)

BYTE IB(512,2), OB(512), ZEROS(512)
 BYTE BYTE

LOGICAL INSERT

COMMON /WINDO/ IX1,IY1,IX2,IY2,OX1,OY1,OX2,OY2,PMAG,INSERT
 COMMON /CHANL/ INCHS(S),QUICK(S)

EQUIVALENCE (BYTE,WORD)

SET UP MAPPING FROM INPUT TO OUTPUT COORDINATES

RXINC=FLOAT(IX2-IX1+1)/FLOAT(OX2-OX1+1)
 RYINC=FLOAT(IY2-IY1+1)/FLOAT(OY2-OY1+1)

OY11=OY1+0
 OY21=OY2+0
 K=0
 IF(RYINC.LT.1.0) K=+1.0/(RYINC*2.0)
 L=IY1

DO 20 J=OY11,OY21
 POSIT=R/(INC*K+L)
 INTG=INT(POSIT)
 RFRAC=POSIT-FLOAT(INTG)
 FRACI=128.0*RFRAC
 DY(J,1)=INTG
 DY(J,2)=FRACI
 K=K+1
 CONTINUE

OX11=OX1+1
 OX21=OX2+1
 K=0
 IF(RXINC.LT.1.0) K=+1.0/(RXINC*2.0)
 L=IX1

DO 30 J=OX11,OX21
 POSIT=R*(INC*K+L)
 INTG=INT(POSIT)
 RFRAC=POSIT-FLOAT(INTG)
 FRACI=128.0*RFRAC
 DX(J,1)=INTG
 DX(J,2)=FRACI

Compute mapping tables

```

0040      PX(J,2)=FRACT
0041      K=K+1
0042      30  CONTINUE
0043      C
0044      C
0045      C
0046      C
0047      C
0048      C
0049      C
0050      C
0051      C
0052      C
0053      C
0054      C
0055      C
0056      C
0057      C
0058      C
0059      C
0060      C
0061      C
0062      C
0063      C
0064      C
0065      C
0066      C
0067      C
0068      C
0069      C
0070      C
0071      C
0072      C
  
```

```

      DO 500 BPTR = 1.5
      END1 = INCHS(BPTR)
      ENDO = OUCHS(BPTR)
      SKIP UNWANTED BANDS
      IF(BND1.EQ.0) GO TO 450
      ELSE PROCESS LINES
      IF(INSERT) GO TO 211
      ELSE ERASE LINES ABOVE OUTPUT WINDOW
      DO 210 LINE = 0, OY11-1
      CALL IRV(BNDG, LINE, ZEROS)
      CONTINUE
      END IF
      END IF
      CONTINUE
      SET UP INTERPOLATION LOOP
      LNPNTN=1
      FSTLIN=DY(OY11,1)
      PRELIN=FSTLIN-1
      PRELIN=MAX0(1,PRELIN)
      CALL IRV(BND1,PRELIN,IB(1,LNPNTN))
      LASTDY = -1
      DO 400 LINE=OY11,OY21
      NEADY = DY(LINE,1)
      SKIP READ IF NO NEW INPUT LINE IS REQUIRED
      IF(NEADY.EQ.LASTDY) GO TO 310
      ELSE UPDATE INPUT BUFFERS TO CONTAIN LINES
      ABOVE AND BELOW LINE TO BE INTERPOLATED
      LABOVE=LNPNTN
      LNPNTN=3-LNPNTN
      LBELOW=LNPNTN
      CALL IRV(END1,NEADY,IB(1,LBELOW))
      END IF
      CONTINUE
      PIXEL BY PIXEL INTERPOLATION
      IF(INSERT) CALL IRV(BNDG,LINE,OB)
      DO 350 PIX=OX11,OX21
  
```

read input data

GET 4 INPUT PIXELS SURROUNDING INTERPOLATING POINT

PLEFT=DX(PIX,1)
PRIGHT=PLEFT+1

BYTE=IB(PLEFT,LABOVE)

A=WORD

BYTE=IB(PRIGHT,LABOVE)

B=WORD

BYTE=IB(PLEFT,LBELOW)

C=WORD

BYTE=IB(PRIGHT,LBELOW)

D=WORD

GET FRACTIONAL PART OF INTERPOLATING POINT LOCATION

X=DX(PIX,2)
Y=DY(LINE,2)

SELECT AND INTERPOLATE ON A PLANE CHOSEN BY
LOCAL GRADIENT AND QUADRANT OF POINT IN ABCD

IF(IABS(C-B).LT.IABS(A-D)) CASE=1
IF((CASE.EQ.1).AND.(Y.LT.X)) CASE=3
IF((CASE.EQ.3).AND.(128-Y).LT.X)) CASE=2

GO TO (331,332,333,334) CASE

OB(PIX)=A+(X*(D-C)+Y*(C-A))/128

GO TO 340

OB(PIX)=A+(X*(B-A)+Y*(D-B))/128

GO TO 340

OB(PIX)=A+(X*(B-A)+Y*(C-A))/128

GO TO 340

OB(PIX)=C+(X*(D-C)+(128-Y)*(B-D))/128

END CASES
CONTINUE

WRITE OUTPUT LINE TO IAC

CALL IMA(ENDD,LINE,OB)

LASTDY = NEWDY

CONTINUE

IF(INSERT) GO TO 411

ELSE ERASE LINES BELOW OUTPUT WINDOW

DO 410 LINE = DY?'1+1,511

CALL IMA(ENDD,LINE,ZEROS)

CONTINUE

END IF

CONTINUE

END IF

CONTINUE

CONTINUE

C

0073

0074

C

0075

0076

0077

0078

0079

0080

0081

0082

0083

0084

C

C

C

C

C

C

C

C

C

C

0093

0094

0095

0096

0097

0098

0099

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

compute interpolated
output pixel

write output values

THU 12-NOV-81 14:10:02 PAGE 004
DB:BGUNDO,LP:DB:BGUNDO

RETURN TO UIC-03
CORE-1X, UIC-[127.4]

5 0113 C RETURN
1 0114 C END

APPENDIX B
THEME FILTERING PROGRAM


```

0005 73 CONTINUE
0006 DO 74 I=1,16
0007 LLL=LL+1
0008 CALL TRN(ENDDI,LLL,BUFFER(1,I))
0009 74 CONTINUE
0010 75 CONTINUE
0011 C
0012 100 CONTINUE
0013 C
0014 C
0015 WRITE(UTY,1000) NROW,NCOL
0016 FORMAT(',$FILTER SIZE = ',13,' X',13,' >')
0017 CALL OUTPUT(?)
0018 READ(UTY,1001)BUF
0019 FORMAT(A92)
0020 IF(BUF(1).EQ.NUL) GO TO 125
0021 IF(BUF(1).EQ.X) GO TO 500
0022 P=0
0023 CALL INTFF(P,BUF,50,NCOL)
0024 CALL INTFF(P,BUF,50,NROW)
0025 IF(NCOL.GT.101).OR.(NROW.LT.1) GO TO 100
0026 IF(NROW.GT.101).OR.(NROW.LT.1) GO TO 100
0027 CONTINUE
0028 AREA=NCOL*NROW
0029 C
0030 C
0031 C
0032 C
0033 C
0034 C
0035 C
0036 C
0037 C
0038 C
0039 C
0040 C
0041 C
0042 C
0043 C
0044 C
0045 C
0046 C
0047 C
0048 C
0049 C
0050 C
0051 C
0052 C
0053 C
0054 C
0055 C
0056 C
0057 C
0058 C
0059 C
0060 C
0061 C
0062 C
0063 C
0064 C
0065 C
0066 C
0067 C
0068 C
0069 C
0070 C
0071 C
0072 C
0073 C
0074 C
0075 C
0076 C
0077 C
0078 C
0079 C
0080 C
0081 C
0082 C
0083 C
0084 C
0085 C
0086 C
0087 C
0088 C
0089 C
0090 C
0091 C
0092 C
0093 C
0094 C
0095 C
0096 C
0097 C
0098 C
0099 C
0100 C
0101 C
0102 C
0103 C
0104 C
0105 C
0106 C
0107 C
0108 C
0109 C
0110 C
0111 C
0112 C
0113 C
0114 C
0115 C
0116 C
0117 C
0118 C
0119 C
0120 C
0121 C
0122 C
0123 C
0124 C
0125 C
0126 C
0127 C
0128 C
0129 C
0130 C
0131 C
0132 C
0133 C
0134 C
0135 C
0136 C
0137 C
0138 C
0139 C
0140 C
0141 C
0142 C
0143 C
0144 C
0145 C
0146 C
0147 C
0148 C
0149 C
0150 C
0151 C
0152 C
0153 C
0154 C
0155 C
0156 C
0157 C
0158 C
0159 C
0160 C
0161 C
0162 C
0163 C
0164 C
0165 C
0166 C
0167 C
0168 C
0169 C
0170 C
0171 C
0172 C
0173 C
0174 C
0175 C
0176 C
0177 C
0178 C
0179 C
0180 C
0181 C
0182 C
0183 C
0184 C
0185 C
0186 C
0187 C
0188 C
0189 C
0190 C
0191 C
0192 C
0193 C
0194 C
0195 C
0196 C
0197 C
0198 C
0199 C
0200 C
0201 C
0202 C
0203 C
0204 C
0205 C
0206 C
0207 C
0208 C
0209 C
0210 C
0211 C
0212 C
0213 C
0214 C
0215 C
0216 C
0217 C
0218 C
0219 C
0220 C
0221 C
0222 C
0223 C
0224 C
0225 C
0226 C
0227 C
0228 C
0229 C
0230 C
0231 C
0232 C
0233 C
0234 C
0235 C
0236 C
0237 C
0238 C
0239 C
0240 C
0241 C
0242 C
0243 C
0244 C
0245 C
0246 C
0247 C
0248 C
0249 C
0250 C
0251 C
0252 C
0253 C
0254 C
0255 C
0256 C
0257 C
0258 C
0259 C
0260 C
0261 C
0262 C
0263 C
0264 C
0265 C
0266 C
0267 C
0268 C
0269 C
0270 C
0271 C
0272 C
0273 C
0274 C
0275 C
0276 C
0277 C
0278 C
0279 C
0280 C
0281 C
0282 C
0283 C
0284 C
0285 C
0286 C
0287 C
0288 C
0289 C
0290 C
0291 C
0292 C
0293 C
0294 C
0295 C
0296 C
0297 C
0298 C
0299 C
0300 C
0301 C
0302 C
0303 C
0304 C
0305 C
0306 C
0307 C
0308 C
0309 C
0310 C
0311 C
0312 C
0313 C
0314 C
0315 C
0316 C
0317 C
0318 C
0319 C
0320 C
0321 C
0322 C
0323 C
0324 C
0325 C
0326 C
0327 C
0328 C
0329 C
0330 C
0331 C
0332 C
0333 C
0334 C
0335 C
0336 C
0337 C
0338 C
0339 C
0340 C
0341 C
0342 C
0343 C
0344 C
0345 C
0346 C
0347 C
0348 C
0349 C
0350 C
0351 C
0352 C
0353 C
0354 C
0355 C
0356 C
0357 C
0358 C
0359 C
0360 C
0361 C
0362 C
0363 C
0364 C
0365 C
0366 C
0367 C
0368 C
0369 C
0370 C
0371 C
0372 C
0373 C
0374 C
0375 C
0376 C
0377 C
0378 C
0379 C
0380 C
0381 C
0382 C
0383 C
0384 C
0385 C
0386 C
0387 C
0388 C
0389 C
0390 C
0391 C
0392 C
0393 C
0394 C
0395 C
0396 C
0397 C
0398 C
0399 C
0400 C
0401 C
0402 C
0403 C
0404 C
0405 C
0406 C
0407 C
0408 C
0409 C
0410 C
0411 C
0412 C
0413 C
0414 C
0415 C
0416 C
0417 C
0418 C
0419 C
0420 C
0421 C
0422 C
0423 C
0424 C
0425 C
0426 C
0427 C
0428 C
0429 C
0430 C
0431 C
0432 C
0433 C
0434 C
0435 C
0436 C
0437 C
0438 C
0439 C
0440 C
0441 C
0442 C
0443 C
0444 C
0445 C
0446 C
0447 C
0448 C
0449 C
0450 C
0451 C
0452 C
0453 C
0454 C
0455 C
0456 C
0457 C
0458 C
0459 C
0460 C
0461 C
0462 C
0463 C
0464 C
0465 C
0466 C
0467 C
0468 C
0469 C
0470 C
0471 C
0472 C
0473 C
0474 C
0475 C
0476 C
0477 C
0478 C
0479 C
0480 C
0481 C
0482 C
0483 C
0484 C
0485 C
0486 C
0487 C
0488 C
0489 C
0490 C
0491 C
0492 C
0493 C
0494 C
0495 C
0496 C
0497 C
0498 C
0499 C
0500 C
0501 C
0502 C
0503 C
0504 C
0505 C
0506 C
0507 C
0508 C
0509 C
0510 C
0511 C
0512 C
0513 C
0514 C
0515 C
0516 C
0517 C
0518 C
0519 C
0520 C
0521 C
0522 C
0523 C
0524 C
0525 C
0526 C
0527 C
0528 C
0529 C
0530 C
0531 C
0532 C
0533 C
0534 C
0535 C
0536 C
0537 C
0538 C
0539 C
0540 C
0541 C
0542 C
0543 C
0544 C
0545 C
0546 C
0547 C
0548 C
0549 C
0550 C
0551 C
0552 C
0553 C
0554 C
0555 C
0556 C
0557 C
0558 C
0559 C
0560 C
0561 C
0562 C
0563 C
0564 C
0565 C
0566 C
0567 C
0568 C
0569 C
0570 C
0571 C
0572 C
0573 C
0574 C
0575 C
0576 C
0577 C
0578 C
0579 C
0580 C
0581 C
0582 C
0583 C
0584 C
0585 C
0586 C
0587 C
0588 C
0589 C
0590 C
0591 C
0592 C
0593 C
0594 C
0595 C
0596 C
0597 C
0598 C
0599 C
0600 C
0601 C
0602 C
0603 C
0604 C
0605 C
0606 C
0607 C
0608 C
0609 C
0610 C
0611 C
0612 C
0613 C
0614 C
0615 C
0616 C
0617 C
0618 C
0619 C
0620 C
0621 C
0622 C
0623 C
0624 C
0625 C
0626 C
0627 C
0628 C
0629 C
0630 C
0631 C
0632 C
0633 C
0634 C
0635 C
0636 C
0637 C
0638 C
0639 C
0640 C
0641 C
0642 C
0643 C
0644 C
0645 C
0646 C
0647 C
0648 C
0649 C
0650 C
0651 C
0652 C
0653 C
0654 C
0655 C
0656 C
0657 C
0658 C
0659 C
0660 C
0661 C
0662 C
0663 C
0664 C
0665 C
0666 C
0667 C
0668 C
0669 C
0670 C
0671 C
0672 C
0673 C
0674 C
0675 C
0676 C
0677 C
0678 C
0679 C
0680 C
0681 C
0682 C
0683 C
0684 C
0685 C
0686 C
0687 C
0688 C
0689 C
0690 C
0691 C
0692 C
0693 C
0694 C
0695 C
0696 C
0697 C
0698 C
0699 C
0700 C
0701 C
0702 C
0703 C
0704 C
0705 C
0706 C
0707 C
0708 C
0709 C
0710 C
0711 C
0712 C
0713 C
0714 C
0715 C
0716 C
0717 C
0718 C
0719 C
0720 C
0721 C
0722 C
0723 C
0724 C
0725 C
0726 C
0727 C
0728 C
0729 C
0730 C
0731 C
0732 C
0733 C
0734 C
0735 C
0736 C
0737 C
0738 C
0739 C
0740 C
0741 C
0742 C
0743 C
0744 C
0745 C
0746 C
0747 C
0748 C
0749 C
0750 C
0751 C
0752 C
0753 C
0754 C
0755 C
0756 C
0757 C
0758 C
0759 C
0760 C
0761 C
0762 C
0763 C
0764 C
0765 C
0766 C
0767 C
0768 C
0769 C
0770 C
0771 C
0772 C
0773 C
0774 C
0775 C
0776 C
0777 C
0778 C
0779 C
0780 C
0781 C
0782 C
0783 C
0784 C
0785 C
0786 C
0787 C
0788 C
0789 C
0790 C
0791 C
0792 C
0793 C
0794 C
0795 C
0796 C
0797 C
0798 C
0799 C
0800 C
0801 C
0802 C
0803 C
0804 C
0805 C
0806 C
0807 C
0808 C
0809 C
0810 C
0811 C
0812 C
0813 C
0814 C
0815 C
0816 C
0817 C
0818 C
0819 C
0820 C
0821 C
0822 C
0823 C
0824 C
0825 C
0826 C
0827 C
0828 C
0829 C
0830 C
0831 C
0832 C
0833 C
0834 C
0835 C
0836 C
0837 C
0838 C
0839 C
0840 C
0841 C
0842 C
0843 C
0844 C
0845 C
0846 C
0847 C
0848 C
0849 C
0850 C
0851 C
0852 C
0853 C
0854 C
0855 C
0856 C
0857 C
0858 C
0859 C
0860 C
0861 C
0862 C
0863 C
0864 C
0865 C
0866 C
0867 C
0868 C
0869 C
0870 C
0871 C
0872 C
0873 C
0874 C
0875 C
0876 C
0877 C
0878 C
0879 C
0880 C
0881 C
0882 C
0883 C
0884 C
0885 C
0886 C
0887 C
0888 C
0889 C
0890 C
0891 C
0892 C
0893 C
0894 C
0895 C
0896 C
0897 C
0898 C
0899 C
0900 C
0901 C
0902 C
0903 C
0904 C
0905 C
0906 C
0907 C
0908 C
0909 C
0910 C
0911 C
0912 C
0913 C
0914 C
0915 C
0916 C
0917 C
0918 C
0919 C
0920 C
0921 C
0922 C
0923 C
0924 C
0925 C
0926 C
0927 C
0928 C
0929 C
0930 C
0931 C
0932 C
0933 C
0934 C
0935 C
0936 C
0937 C
0938 C
0939 C
0940 C
0941 C
0942 C
0943 C
0944 C
0945 C
0946 C
0947 C
0948 C
0949 C
0950 C
0951 C
0952 C
0953 C
0954 C
0955 C
0956 C
0957 C
0958 C
0959 C
0960 C
0961 C
0962 C
0963 C
0964 C
0965 C
0966 C
0967 C
0968 C
0969 C
0970 C
0971 C
0972 C
0973 C
0974 C
0975 C
0976 C
0977 C
0978 C
0979 C
0980 C
0981 C
0982 C
0983 C
0984 C
0985 C
0986 C
0987 C
0988 C
0989 C
0990 C
0991 C
0992 C
0993 C
0994 C
0995 C
0996 C
0997 C
0998 C
0999 C
1000 C
1001 C
1002 C
1003 C
1004 C
1005 C
1006 C
1007 C
1008 C
1009 C
1010 C
1011 C
1012 C
1013 C
1014 C
1015 C
1016 C
1017 C
1018 C
1019 C
1020 C
1021 C
1022 C
1023 C
1024 C
1025 C
1026 C
1027 C
1028 C
1029 C
1030 C
1031 C
1032 C
1033 C
1034 C
1035 C
1036 C
1037 C
1038 C
1039 C
1040 C
1041 C
1042 C
1043 C
1044 C
1045 C
1046 C
1047 C
1048 C
1049 C
1050 C
1051 C
1052 C
1053 C
1054 C
1055 C
1056 C
1057 C
1058 C
1059 C
1060 C
1061 C
1062 C
1063 C
1064 C
1065 C
1066 C
1067 C
1068 C
1069 C
1070 C
1071 C
1072 C
1073 C
1074 C
1075 C
1076 C
1077 C
1078 C
1079 C
1080 C
1081 C
1082 C
1083 C
1084 C
1085 C
1086 C
1087 C
1088 C
1089 C
1090 C
1091 C
1092 C
1093 C
1094 C
1095 C
1096 C
1097 C
1098 C
1099 C
1100 C
1101 C
1102 C
1103 C
1104 C
1105 C
1106 C
1107 C
1108 C
1109 C
1110 C
1111 C
1112 C
1113 C
1114 C
1115 C
1116 C
1117 C
1118 C
1119 C
1120 C
1121 C
1122 C
1123 C
1124 C
1125 C
1126 C
1127 C
1128 C
1129 C
1130 C
1131 C
1132 C
1133 C
1134 C
1135 C
1136 C
1137 C
1138 C
1139 C
1140 C
1141 C
1142 C
1143 C
1144 C
1145 C
1146 C
1147 C
1148 C
1149 C
1150 C
1151 C
1152 C
1153 C
1154 C
1155 C
1156 C
1157 C
1158 C
1159 C
1160 C
1161 C
1162 C
1163 C
1164 C
1165 C
1166 C
1167 C
1168 C
1169 C
1170 C
1171 C
1172 C
1173 C
1174 C
1175 C
1176 C
1177 C
1178 C
1179 C
1180 C
1181 C
1182 C
1183 C
1184 C
1185 C
1186 C
1187 C
1188 C
1189 C
1190 C
1191 C
1192 C
1193 C
1194 C
1195 C
1196 C
1197 C
1198 C
1199 C
1200 C
1201 C
1202 C
1203 C
1204 C
1205 C
1206 C
1207 C
1208 C
1209 C
1210 C
1211 C
1212 C
1213 C
1214 C
1215 C
1216 C
1217 C
1218 C
1219 C
1220 C
1221 C
1222 C
1223 C
1224 C
1225 C
1226 C
1227 C
1228 C
1229 C
1230 C
1231 C
1232 C
1233 C
1234 C
1235 C
1236 C
1237 C
1238 C
1239 C
1240 C
1241 C
1242 C
1243 C
1244 C
1245 C
1246 C
1247 C
1248 C
1249 C
1250 C
1251 C
1252 C
1253 C
1254 C
1255 C
1256 C
1257 C
1258 C
1259 C
1260 C
1261 C
1262 C
1263 C
1264 C
1265 C
1266 C
1267 C
1268 C
1269 C
1270 C
1271 C
1272 C
1273 C
1274 C
1275 C
1276 C
1277 C
1278 C
1279 C
1280 C
1281 C
1282 C
1283 C
1284 C
1285 C
1286 C
1287 C
1288 C
1289 C
1290 C
1291 C
1292 C
1293 C
1294 C
1295 C
1296 C
1297 C
1298 C
1299 C
1300 C
1301 C
1302 C
1303 C
1304 C
1305 C
1306 C
1307 C
1308 C
1309 C
1310 C
1311 C
1312 C
1313 C
1314 C
1315 C
1316 C
1317 C
1318 C
1319 C
1320 C
1321 C
1322 C
1323 C
1324 C
1325 C
1326 C
1327 C
1328 C
1329 C
1330 C
1331 C
1332 C
1333 C
1334 C
1335 C
1336 C
1337 C
1338 C
1339 C
1340 C
1341 C
1342 C
1343 C
1344 C
1345 C
1346 C
1347 C
1348 C
1349 C
1350 C
1351 C
1352 C
1353 C
1354 C
1355 C
1356 C
1357 C
1358 C
1359 C
1360 C
1361 C
1362 C
1363 C
1364 C
1365 C
1366 C
1367 C
1368 C
1369 C
1370 C
1371 C
1372 C
1373 C
1374 C
1375 C
1376 C
1377 C
1378 C
1379 C
1380 C
1381 C
1382 C
1383 C
1384 C
1385 C
1386 C
1387 C
1388 C
1389 C
1390 C
1391 C
1392 C
1393 C
1394 C
1395 C
1396 C
1397 C
1398 C
1399 C
1400 C
1401 C
1402 C
1403 C
1404 C
1405 C
1406 C
1407 C
1408 C
1409 C
1410 C
1411 C
1412 C
1413 C
1414 C
1415 C
1416 C
1417 C
1418 C
1419 C
1420 C
1421 C
1422 C
1423 C
1424 C
1425 C
1426 C
1427 C
1428 C
1429 C
1430 C
1431 C
1432 C
1433 C
1434 C
1435 C
1436 C
1437 C
1438 C
1439 C
1440 C
1441 C
1442 C
1443 C
1444 C
1445 C
1446 C
1447 C
1448 C
1449 C
1450 C
1451 C
1452 C
1453 C
1454 C
1455 C
1456 C
1457 C
1458 C
1459 C
1460 C
1461 C
1462 C
1463 C
1464 C
1465 C
1466 C
1467 C
1468 C
1469 C
1470 C
1471 C
1472 C
1473 C
1474 C
1475 C
1476 C
1477 C
1478 C
1479 C
1480 C
1481 C
1482 C
1483 C
1484 C
1485 C
1486 C
1487 C
1488 C
1489 C
1490 C
1491 C
1492 C
1493 C
1494 C
1495 C
1496 C
1497 C
1498 C
1499 C
1500 C

```

operator selection of
 processing parameters

selection of processing
 window

GET THEME THRESHOLDS TO CONTROL FILTERING ACTION

CALL TRNLD(NCOL,NROW,NUTYM,GROPT,GROCT,SHRPT,SHRONT)

DETERMINE PROCESSING WINDOW

WRITE(UTY,1400)
 FORMAT(',\$PROCESS (C)CURSORED AREA OR (S)SUBSCENE? S')
 CALL OUTPUT(?)

READ(UTY,1001)BUF
 IF(BUF(1).EQ.X) GO TO 500
 IF(BUF(1).EQ.C) GO TO 145

ELSE GET LIMITS FROM I-100 COMMON
 OX1=IMAXC(1,1)+1
 OX1=MAX(OX1,MINPIX)
 OY1=IMAXC(2,1)+1
 OY1=MAX(OY1,MINLIN)
 OX2=IMAXC(1,2)+1
 OX2=MIN(OX2,MAXPIX)
 OY2=IMAXC(2,2)+1
 OY2=MIN(OY2,MAXLIN)
 GO TO 149

THEN GET LIMITS FROM CURSOR POSITION
 CONTINUE
 CALL TRN(CURSOR)
 OX1=CURSOR(2)+1
 OX2=CURSOR(2)+12
 OY1=CURSOR(4)+1
 OY2=CURSOR(4)+12

END IF
 CONTINUE

initialize summing buffers

CALL TALK=OX2-OX1

DEFO SUMMING BUFFERS BEFORE ITERATION

DO 160 I=1,BUFLEN(CR,THM1)

CALL C

CALL C

CONTINUE

DO 161 I=1,BUFLEN(B

LINES(I)=0

CONTINUE

SET UP NEGATIVE START LOCATIONS FOR BOX

BOTSTP=OY1-(AKUN+1)/2

TOPSTR=BOTSTR+AKUN

LINSTR=BOTSTR+AKUN/2

BOTAL=BOTSTP

TOPAL=TOPSTR

DO 399 LINE=LINSTR,OY2

READ TOP,BOT, AND MID LINES OF BOX (WITH REFLECTION)

BLREAD=BOTAL

IF(BOTAL.LT.OY1) BLREAD=-BOTAL+OY1+OY1-1

IF(BOTAL.GT.OY2) BLREAD=-BOTAL+OY2+OY2

TLREAD=TOPAL

IF(TOPAL.LT.OY1) TLREAD=-TOPAL+OY1+OY1-1

CALL TRUCRNDI,TLREAD,OLDB

CALL TRUCRNDI,LINE,NEWB

CALL TRUCRNDI,BLREAD,NEWB

IF(TOPAL.GE.BOTSTR) GO TO 305

ELSE PREP TOP LINE

CALL TALOOK(OLDB,ZERO,OLDB,512)

END IF

CONTINUE

BOTAL=BOTAL+1

TOPAL=TOPAL+1

IF CODE INPUT VALUES TO 0-MAXTHM

CALL TALOOK(LINES,ENTRIB,LINES,1536)

UPDATE STORED COLUMN SUMS

IF(TOPAL.LT.BOTSTR) GO TO 310

GO TO 320

THEN PRELOAD SUMMING BUFFERS

CONTINUE

DO 315 PIX=OY1,OY2

B=NEWSB(PIX)

310

0160

0161

0162

read theme data
from memory

FURMAN IV 0010-03
CORE-11K, UIC-(127.4)

THU 19-NOV-81 21:10:17 PAGE 008
DB:THMFIL,LP,=DB:THMFIL

C348 CONTINUE
C350 CONTINUE
CC END OF REPLACEMENT
C CCCCCCCCCC

0176 NTHEME=NUMTHM+1
0177 CALL REPLAC(MIDB,TS,OX1,NUMPIX,NTHEME,
 &ROCONT,SHRCNT,BUFLEN)

C DECODE AND WRITE OUTPUT LINE
C CALL TALOOK(MIDB,DETAB,MIDB,S12)
C CALL IIAV(BNDO,LINE,MIDB)

write output time

0180 END IF
0181 CONTINUE
C CONTINUE

0182 WRITE(UTY,4000)
0183 FORMAT('/',\$(C)HANGE THRESHOLDS, (R)ECYCLE, OR E(X)IT >')
0184 CALL OUTPUT(7)
0185 READ(UTY,1001)BUF
0186 IF(BUF(1).EQ.C) GO TO 100
0187 IF(BUF(1).EQ.P) GO TO 70
0190 IF(BUF(1).EQ.N) GO TO 900
0192 GO TO 400

C C
C 900 CONTINUE
0194 RETURN
C END


```

1 .....
2 .....
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
10 .....
11 .....
12 .....
13 .....
14 .....
15 .....
16 .....
17 .....
18 .....
19 .....
20 .....
21 .....
22 .....
23 .....
24 .....
25 .....
26 .....
27 .....
28 .....
29 .....
30 .....
31 .....
32 .....
33 .....
34 .....
35 .....
36 .....
37 .....
38 .....
39 .....
40 .....
41 .....
42 .....
43 .....
44 .....
45 .....
46 .....
47 .....
48 .....
49 .....
50 .....
51 .....
52 .....
53 .....
54 .....
55 .....
56 .....
57 .....

```

```

.TITLE THMSUM
.GLOBAL THMTOT
.GLOBAL REPLIC
.GLOBAL ROUNT

```

```

THREE ROUTINES FOR RAPID EXECUTION OF THEME FILTERING

```

```

FIRST ROUTINE DOES RUNNING SUM OF THEMES INTO A
SET OF ACCUMULATING BUFFERS, ONE PER THEME

```

```

FORTRAN CALLING SEQUENCE:
CALL THMSUM(NEWB,OLDB,SUMB,START,NUMPIX,BUFLEN)

```

```

NEWB = BYTE ARRAY CONTAINING NEXT LINE FOR AVERAGING AREA
OLDB = BYTE ARRAY CONTAINING LINE TO BE SUBTRACTED FROM AREA
SUMB = WORD ARRAY SUB(BUFLEN,NTHM) CONTAINS COLUMN SUMS OF THEMES
BUFLEN = NUMBER OF PIXELS IN THE LINES

```

```

THMTOT:
UPDATE SUMB BY ADDING NEW LINE AND SUBTRACTING OLD

```

```

MOV 2(R5),R0 ;NEW LINE ADDRESS
MOV 4(R5),R1 ;OLD LINE ADDRESS
MOV 6(R5),R2 ;SUM BUFFER ADDRES
MOV @10(R5),R3 ;GET START PIXEL
MOV @14(R5),R4 ;GET BUFFER LENGTH
ASL R4 ; IN WORDS
MOV @12(R5),R5 ; SET LOOP COUNTER

```

```

ADD R3,R0 ;POINTS TO 1ST NEWB PIXEL
ADD R3,R1 ;POINTS TO 1ST OLDB PIXEL
ADD R3,R2 ;POINTS TO 1ST SUMB PIXEL

```

```

LOOP:
MOVSB (R4)+,R3 ;GET-THEME NUMBER FROM OLD LINE
MUL R4,R3 ;R3=BUFLEN*THMINUM
ADD R2,R3 ;TARGET ADDRESS
DEC (R3) ;DECREMENT THE SUM

```

```

MOVSB (R4)+,R3 ;THEME NUMBER FROM NEW LINE
MUL R4,R3
ADD R2,R3 ;TARGET ADDRESS
INC (R3) ;INCREMENT THE SUM

```

```

TST (R2)+
SMB R2,LOOP
PTS
EXIT:

```



```

115 000242 002030          ; JUMP OUT TO PRESERVE KEY THEME
116
117
118 000244 005000          ; INIT MAXIMUM
119 000246 005001          ; INIT THEME NUMBER
120 000250 016702 000076  ; INIT GROW THRESHOLD POINTER
121 000254 016703 000062  ; PUT SUMB(PIX,0) INTO WORKING REGISTER
122
123 000260 012206          ; GET THEME GROW THRESHOLD
124 000262 111304          ; GET THEME SUM
125 000264 042704 174400 --
126
127 000270 066703 000062  ; SET SUM POINTER TO SUMB(PIX,NEXTHM)
128
129 000274 020004          ; JUMP OUT IF THEME NOT A MAXIMUM
130 000276 002006
131
132 000300 020604          ; JUMP OUT IF TOTAL BELOW GROW THRESHOLD
133 000302 003004
134
135 000304 005706          ; JUMP OUT IF THEME IS TO BE REPLACED
136 000306 002402
137
138 000310 010400          ; CHANGE MAXIMUM TO NEW MAX
139 000312 110115          ; AND OUTPUT THE THEME NUMBER
140
141 000314 005201          ; RESTORE STACK POINTER
142 000316 020157 000026  ;
143 000322 002756
144
145 000324 020557 000030  ;
146 000330 002730
147
148 000332 016701 000002  ;
149 000334 000000 000000  ;
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

```

```

172 DATA NEEDED TO AVOID LINE SHORTENING
173 DOUBLE PRECISION ARITHMETIC TO AVOID OVERFLOW
174
175 .TITLE ROWTOT
176 .GLOBAL ROWTOT
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228

```

FORTRAN CALL: CALL ROWTOT(IBUF, OBUF, LENGTH, AVELEN, DIVISR)

IBUF = WORD ARRAY CONTAINING INPUT DATA
OBUF = BYTE OUTPUT ARRAY
LENGTH = NUMBER OF PIXELS TO BE OUTPUT
AVELEN = NUMBER OF ELEMENTS TO BE AVERAGED
DIVISR = DIVISOR TO COMPUTE AVERAGE

```

      ROWTOT = 0
      I = 0
      O = 0
      L = 0
      A = 0
      D = 0
      SP = 0
      CALL ROWTOT( IBUF, OBUF, LENGTH, AVELEN, DIVISR )
      I = I + 1
      O = O + 1
      L = L + 1
      A = A + 1
      D = D + 1
      SP = SP + 1

```

SET LOOP COUNTERS

```

      MOV R0, I
      INC R0
      MOV R1, O
      INC R1
      MOV R2, L
      INC R2
      MOV R3, A
      INC R3
      MOV R4, D
      INC R4
      MOV R5, SP
      INC R5

```

INITIALIZE POINTERS

```

      MOV R0, I
      MOV R1, O
      MOV R2, L
      MOV R3, A
      MOV R4, D
      MOV R5, SP

```

STARTING ADDRESS FOR LEADING ELEMENT

```

      MOV R0, I
      MOV R1, O
      MOV R2, L
      MOV R3, A
      MOV R4, D
      MOV R5, SP

```

STARTING ADDRESS FOR TRAILING ELEMENT

```

      MOV R0, I
      MOV R1, O
      MOV R2, L
      MOV R3, A
      MOV R4, D
      MOV R5, SP

```

STARTING ADDRESS FOR OUTPUT BYTE

```

      MOV R0, I
      MOV R1, O
      MOV R2, L
      MOV R3, A
      MOV R4, D
      MOV R5, SP

```

TO ACHIEVE LOW END REFLECTION, SUBTRACT ADDRESS FROM OFFSET

```

      MOV R0, I
      MOV R1, O
      MOV R2, L
      MOV R3, A
      MOV R4, D
      MOV R5, SP

```

SET THE POINTERS

```

      MOV R0, I
      MOV R1, O
      MOV R2, L
      MOV R3, A
      MOV R4, D
      MOV R5, SP

```

ROUTUT

```

229 000576 005002
230 000600 005003
231
232
233
234
235 000602 010400
236 000604 062704 000002
237
238 000610 020067 000210
239 000614 003003
240 000616 006400
241 000620 006700 000204
242 000624 061002
243 000626 005503
244
245 000630 010600
246 000632 062706 000002
247
248 000636 020067 000154
249 000642 002420
250
251 000644 020067 000154
252 000650 003003
253 000652 005400
254 000654 066700 000150
255 000660 161002
256 000662 005503
257
258 000664 020667 000136
259 000670 002405
260 000672 010201
261 000674 010300
262 000676 071067 000100
263 000702 110015
264
265 000704 005205
266 000706 020667 000112
267 000712 003733
268
269
270
271
272 000714 162602
273 000716 005603
274
275 000720 062402
276 000722 005503
277
278 000724 010201
279 000726 010300
280 000730 071067 000046
281
282 000734 110025
283
284 000736 036704 000044
285 000742 003364

```

```

CLR R2
CLR R3

PROCESSING LOOP FOR THE FIRST SEGMENT OF THE LINE
MOV R4,R0
ADD #2,R4
; LEADING ELEMENT ADDRESS

CMP R0,IBUF
BGT FORK
; DOES IT POINT TO VALID DATA YET?
; IF NOT, REFLECT ADDRESS
; ADD LEADING ELEMENT INTO ACCUMULATORS
; WITH DOUBLE PRECISION
ADD #R0,R2
; TRAILING ELEMENT ADDRESS

MOV SP,R0
ADD #2,SP

CMP R0,FORSTR
BLT TEST1
; ARE WE STILL PRELOADING ACCUMULATORS?

CMP R0,IBUF
BGT BACK
; INTO VALID DATA YET?
; IF NOT, REFLECT ADDRESS
; SUBTRACT TRAILING ELEMENT FROM ACCUMULATORS
; WITH DOUBLE PRECISION
SUB #R0,R2
SBC R3

CMP R5,OBUF
BLT TEST1
; IS OUTPUT POINTER INTO OBUF YET?
; IF YES, COMPUTE AVERAGE
MOV R2,R1
MOV R3,R0
DIV DIVISR,R0
MOV R0,(R5)
; AND PUT IT INTO OBUF
; INCREMENT OUTPUT POINTER
INC R5
; IS ALL INPUT PAST IBUF YET?
CMP SP,IBUF
BLE LOOP1

```

```

PROCESSING LOOP FOR THE MAIN SEGMENT OF THE LINE
SUB (SP)+,R2
SBC R3
; SUBTRACT TRAILING ELEMENT NORMALLY

ADD (R4)+,R2
ADC R3
; THEN ADD LEADING ELEMENT NORMALLY

MOV R2,R1
MOV R3,R0
DIV DIVISR,R0
MOV R0,(R5)+
; COMPUTE THE AVERAGE
; OUTPUT THE AVERAGE
CMP STOR,R4
BGT LOOP2

```


FILMED
— 8