ETL-0227

LEVEL

FEATURE TAGGING

G. A. Biecker, D. S. Paden, J. L. Potter
Goodyear Aerospace Corporation
1210 Massillon Road
Akron, Ohio 44315

1 April 1980

Final Report, June 1979 to April 1980

Approved for Public Release:
Distribution Unlimited

Prepared for

U.S. Army Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060

AD A091691

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
ETL-0227	AD-A091691	

4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED
Feature Tagging.	Final Report - June 1979 to Apr 1980
	6. PERFORMING ORG. REPORT NUMBER
	GER-16753

7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
G. A. Biecker J. L. Potter D. S. Paden	DAAK70-79-C-0070

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Goodyear Aerospace Corporation 1210 Massillon Road Akron, Ohio 44315	1 Apr 80

11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE
U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060	March 1980
	13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report)
32	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)

Feature Tagging, Pattern Recognition, Automated Cartography

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
The automated recognition of cartographic symbols such as dual cased roads and railroads would significantly reduce the manual labor involved in generating digital cartographic data bases. The effort described in this report was successful in detection 96.5% of the railroad symbol components. There were only 1.5% false taggings. 98.3% of the dual cased roads were tagged with only .7% false taggings. Goodyear Aerospace Corporation (GAC) believes that minor modifications to the algorithms would produce near perfect

DD FORM 1 JAN 73 1473   EDITION OF 1 NOV 65 IS OBSOLETE

156800

20. (con't).

results for both features. Because of the success of this effort,
GAC feels that the project should be continued to allow evaluation
on existing map sheet data and expansion of the effort to addition-
al cartographic symbols.

Accession For

NTIS CRA&I
DTIC TAB
Unannounced
Justification

By
Distribution/
Availability Codes
Avail and/or
Special

Dist

A

SUMMARY

The Feature Tagging program addressed the problem of automatically identifying cartographic symbols in digitized map data. Universally applicable algorithms were developed for the identification and tagging of dual cased roads and railroads. 96.5% of the railroads were cor..ctly tagged while only 1.5% of the vectors tagged as railroads were not railroads. 98.3% of the dual cased roads were correctly tagged with 0.7% false taggings. Shown below is a portion of the original image and the corresponding railroad and dual casing road separations. The complete image and separations are shown in the Conclusions and Recommendations section. Goodyear Aerospace feels that the results of this effort warrant its continuation so that the algorithms developed can be further refined using actual map sheet data. In addition, the effort should be expanded to include the development of algorithms for additional cartographic symbols.
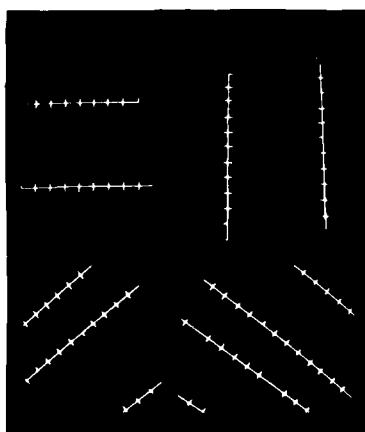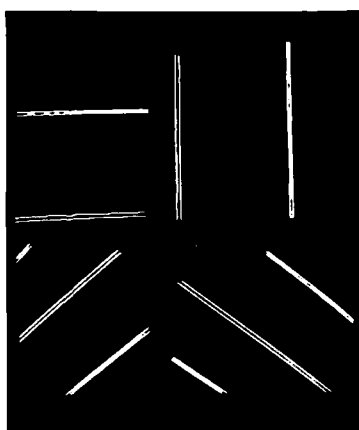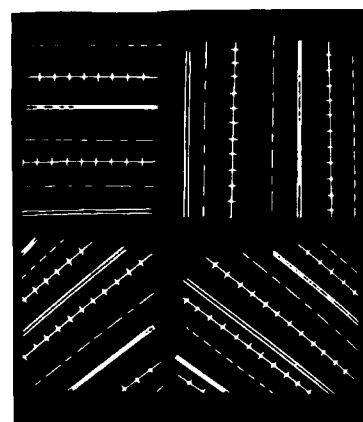
RAILROADS   DUAL CASED ROADS   ORIGINAL

PREFACE

This document is generated under Contract DAAK-79-C-0070
for the U.S. Army Engineer Topographic Laboratories,
Fort Belvoir, Virginia 22060 by Goodyear Aerospace
Corporation (GAC), Akron, Ohio and submitted by GAC as
GER-16753. The Contract Officer's Representative was
Mr. Richard Clark.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS (cont.)

LIST OF TABLES

SECTION 1 - INTRODUCTION

1.1 BACKGROUND

Since 1974, Goodyear Aerospace Corporation (GAC) has been actively involved, with ETL, in the investigation of STARAN* type processors for use in Automated Cartography.

By 1976, GAC had demonstrated that STARAN could use raster scanned pencil drawing inputs and produce clean line and areal symbologies (it also was used to spot, scale, and orient pointer symbologies). To accomplish the above tasks, STARAN raster-to-vector software with auto-edit capabilities was developed. As required for this procurement, the software was written to tolerate lines with variable line width when converting raster scanned sheet data to vector data. To develop the final graphics output product, software was written to convert vector data to a raster form, and to embellish it with line symbol attributes. Thus, thin line vector coordinate data, along with a symbol type descriptor, were found to be sufficient for automatically producing such line symbologies as double casement roads, intermittent streams, and railroads, as well as areal symbologies (that is, swamp regions). For line symbologies, junctions that conformed to cartographic esthetic standards were automatically generated.

During 1976 to 1977, software developed earlier to demonstrate STARAN's cartographic processing power was made more flexible. In particular, the software was altered to allow variable scan resolutions, different sheet sizes, a greater range of line widths, etc.

---

*T.M. Goodyear Aerospace Corporation, Akron, Ohio 44315.

Input/output formats were set up to be compatible with either the DMA RAPS or ETL-IBM raster devices. This software allowed STARAN to perform cartographic tasks in a stand-alone mode.

In 1978, the cartographic software cited above was set up to be used at ETL in conjunction with the CDC 6400 computer. The setup accounted for the file structures, file sizes, and I/O needed for the large-scale conversion of sheet graphics raster data to a vector form and was used to make raster-to-vector format conversion timing tests. The STARAN Raster Processing Software (STRAPS) developed allowed map sheet region windowing, limited scaling, and the plotting of vector data (developed from the raster form) on CALCOMP, CALMA/NOVA, and Gerber plotters. The STRAPS execution time performance tests showed the extraordinary processing speed of STARAN; it demonstrated its reliability and effectiveness.

1.2 PURPOSE

The purpose of this contract was to investigate, develop, and implement algorithms for the automatic detection of cartographically symbolized dual cased roads, railroads, and broken lines (trails and supplementary contours). Automatic detection will provide a significant reduction in the manual effort required to generate digital data bases from existing map sheets.

1.3 APPROACH

The feature tagging effort falls into that area of image processing where pattern recognition and scene analysis overlap. It is not possible to classify the feature types based on isolated pixel values as is frequently done in classical (statistical) pattern recognition. Yet the syntax of the imagery is not sufficiently sophisticated to require the full power of scene analysis techniques as typified by linguistic pattern recognition. However,

-2-

the more sophisticated approach provides flexibility that is often needed in R and D efforts. For this reason, the Feature Tagging software package was patterned after scene analysis techniques.

The Feature Tagging program is organized into four basic tasks; preprocessing, attribute extraction, feature identification and post-processing. Since the attribute extraction and feature identification tasks are performed separately, the implementation of the attribute extraction algorithms is not impaired. Thus, attribute extraction algorithms can be developed to operate on raster data, vector data, or both, whatever combination is most effective. Moreover, the attribute values can be extracted in any order. Only after all attributes have been extracted are the classification algorithms applied. The preprocessing and post-processing tasks provide the basic data input and output functions.

1.4   FACILITIES   -

Two main systems were used for this work: 1) the DMA/ETL Digital Image Analysis Laboratory (DIAL) and 2) the GAC STARAN Evaluation and Training Facility (SETF), Akron, Ohio. The main parts of the above equipment used were:

1) At ETL, the CDC 6415-8 processor including the 98K core storage, seven and nine-track magnetic tapes, large capacity disks along with the Command Channel interface link to the STARAN (4-arrays) and the PDP-11/20 supported by two RK05 disks were used. The CONTAGRID Software package available at ETL was used to produce the vector data.

2) The GAC SETF, at Akron, Ohio, was utilized for algorithm development and checkout of the STARAN processing.

3) The COMTAL (512x512) Image and Graphics Display Equipment, which is part of the SETF equipment was extensively used as a debugging aid (along with additional software) in allowing the display of vector data.

-3-

SECTION 2 - DISCUSSION

2.1 GENERAL

In the paragraphs that follow, details of the algorithms and test results are given. The software required to input, vectorize, and build master vector files was derived by modifying similar modules resulting from the CONTAGRID program under contract DAAK7C-77-C-0223.

2.2 INPUT DATA

Figure 1 shows the non mil-spec test graphic used for algorithm testing. This graphic was scanned and supplied to GAC by ETL. The graphic shown contains symbology which exceeds the line thickness parameters of the vectorization program. However, since a substantial portion of the image could be processed, and the purpose of this effort was feature detection, not vectorization, a reduced data base was used (see Figure 12).

The input data to the Feature Tagging software consists of a set of array vector data and master vector headers. During the raster-to-vector process, small areas (array load) of the raster data are vectorized resulting in array vectors. The array vectors are then organized into master vectors which are the logical entities that are tagged as railroads, dual cased roads, etc.

It is important to understand that the intersection of two or more lines results in the termination of vectors. For example, the line drawings in Figure 2a will result in the master vectors shown in Figure 2b. The feature tagging algorithms were designed to operate on vector data which has been organized in this manner.

There are occasions when it is more expedient to work with data in a raster instead of vector format. In these cases, the vector data is converted to raster data before processing. It is a straight-forward conversion and the compactness of the basic vector format saves considerable I/O and storage space.
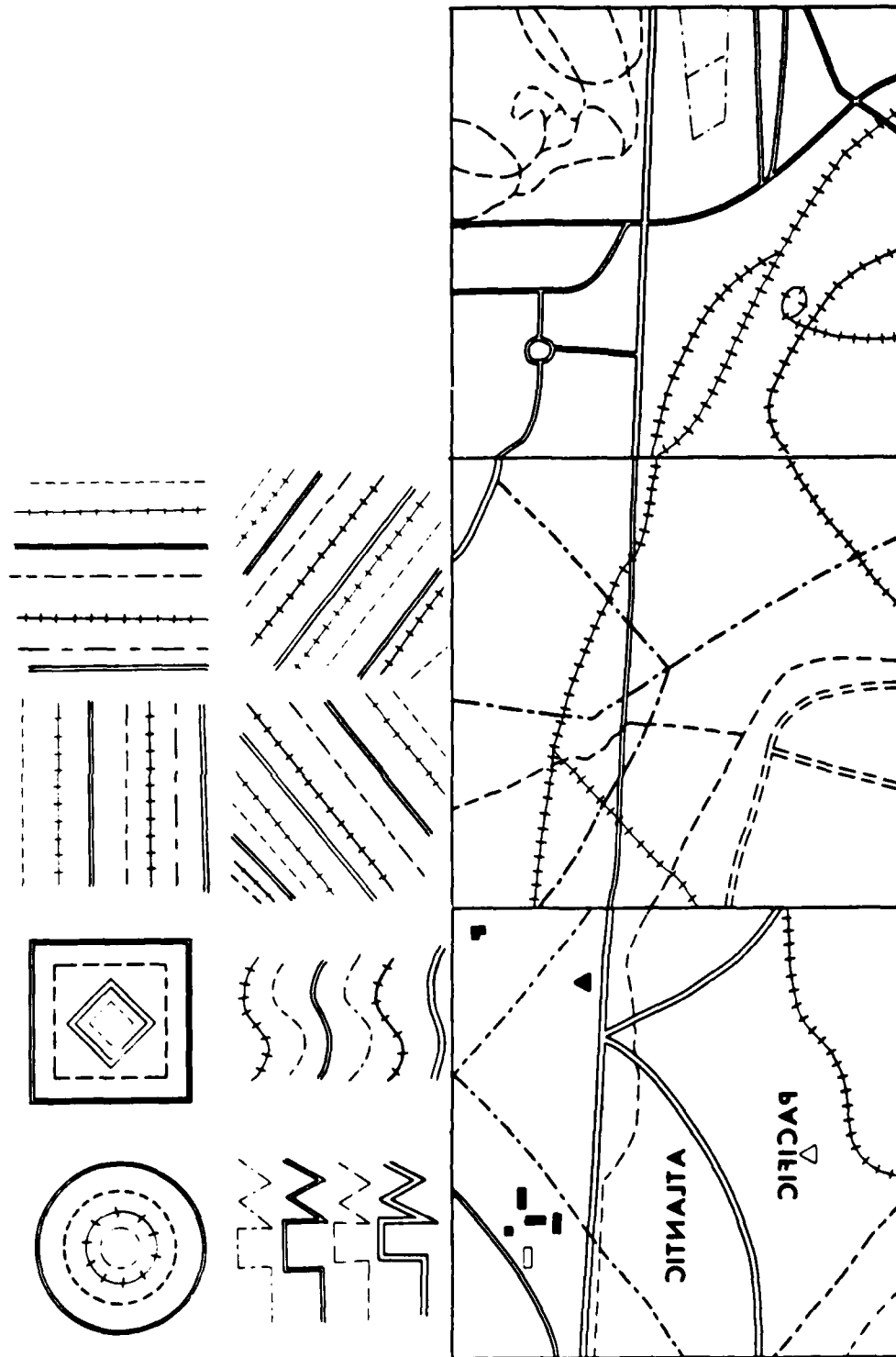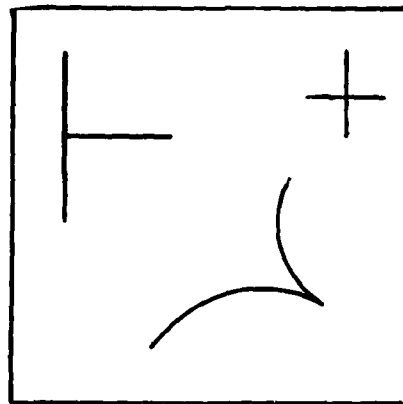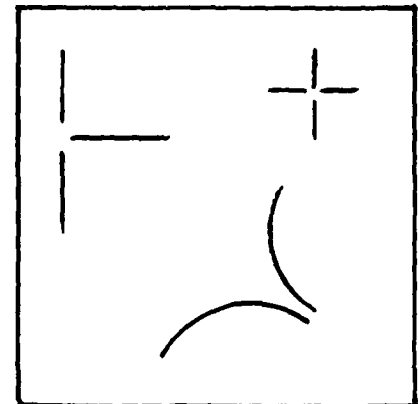
-4-

Figure 1 - Feature Tagging Test Image

-5-

a.   b.

Figure 2 - Master Vectors

2.3   SOFTWARE ORGANIZATION

The Feature Tagging program structure is shown in Figure 3. By
separating the attribute extraction from the feature classification
modules, the structure is extremely flexible and allows for the
easy addition of new attribute extraction and feature identifica-
tion modules independent of one another as they are developed.
Since each attribute is extracted independently of the others, it
need be obtained only once. Finally, the hierarchic structure
of the feature classification branch leaves final assessment of
feature type to the top level module. In this way, multiple or
conflicting classifications made at the lower levels can be re-
solved at the top, where a more global view exists.

2.4 ATTRIBUTE EXTRACTION ALGORITHMS

Figure 3 identifies six attributes that were hypothesized to be useful for feature tagging. These are: curvature, length, directivity, thickness, parallelism and junction characterization. These attributes were intended to be general in nature, useful for a number of different features and are included in Figure 3 to illustrate the ease with which new attributes could be added to the system. The three attributes, length, parallelism and junction characterization required to classify railroads and dual cased roads, are discussed below.

2.4.1 Parallelism

The "degree of parallelism" of a master vector is determined by the percentage of array vectors comprising the master vector which have parallel components. Thus, in Figure 4 the array vector components of master vector m and n are parallel in the array load b. In the final analysis, however, master vector m is "not very parallel" to master vector n. Nevertheless, the "degree of parallelism" is measured and associated with master vectors m and n. The intent of this algorithm is to measure a parallelism attribute; not to make a parallel label assignment.

The algorithm begins by generating a raster representation of the vector data (see Figure 5). Then, each vector is thickened by an amount determined by a parameter. Next, the entire array is complimented leaving the (previously) thickened vectors now as thick gaps. This is the first major step toward extracting parallelism. Not only are the vectors now presented as gaps, but the center of any parallel lines now appear as thick lines. The array is then processed with a "thin" routine until the center lines of the parallel vectors are reduced to a single pixel wide. A routine is then executed which eliminates pixels that are not contained in a line one pixel wide. The desired pixel patterns are shown in Figure 6 where the central pixel represents the pixel under test, while the others represent the test pixel neighbors. All pixel combinations not shown in Figure 6 are invalid. If the pixel is deemed valid, it is left alone (i.e., remains set). If, on the other hand, the
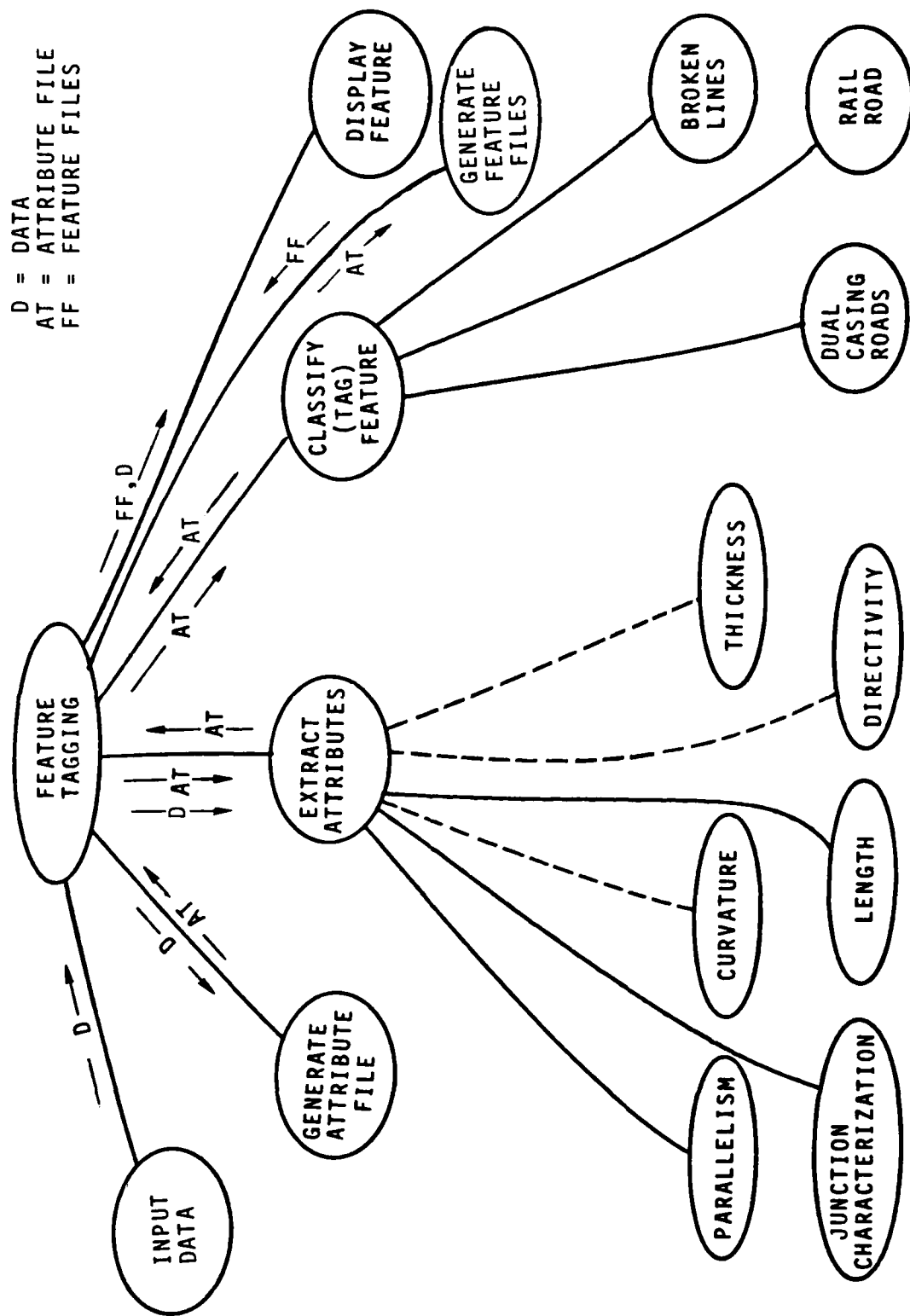
Figure 3 - Feature Tagging Structure Chart

D = DATA
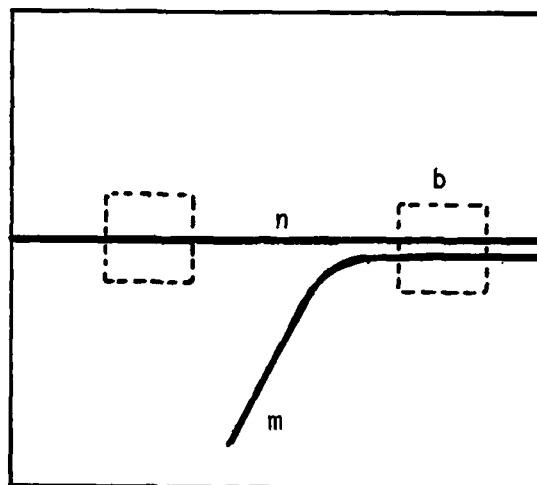AT = ATTRIBUTE FILE
FF = FEATURE FILES

Figure 4 - Partial Parallelism

pixel is determined to be invalid, it is set to a 0. What remains after this process are the pixels designating a centerline of the desired parallel lines.

This centerline representation of the parallel lines is then thickened to approximately 2 pixels wider than the original parallel vectors. Finally, the entire array load is ANDed with the original image. The results are vectors parallel to each other.

From this point a test is made to detect which array vectors remain after the parallel extraction. The start and end point of each vector in the array load is accessed from core memory. A "window" is drawn around both ends of the vector as shown in Figure 7. If a bit is set somewhere within both "windows" of the vector's end points, the vector is determined to be parallel. The length (in pixels) of the parallel array vectors are added to the running total of the corresponding master vectors.
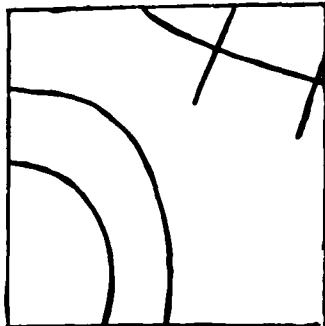
A. ORIGINAL   B. THICKEN   C. NEGATE
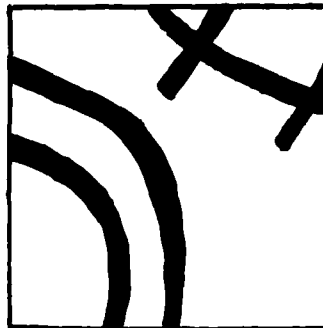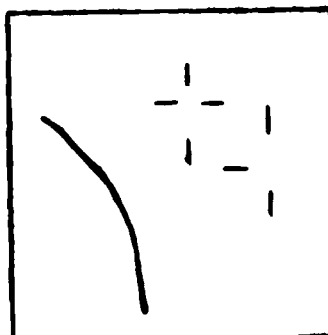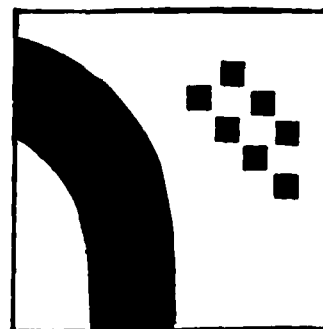
D. THIN   E. SEARCH FOR VALID PIXELS   F. THICKENED

G. THICKENED "AND"ed WITH ORIGINAL

Figure 5 - Extracting Parallelism

```
0|1|0   0|1|0   0|1|0   0|1|0   0|1|0   0|0|1
0|1|1   0|1|0   0|1|0   0|1|0   1|1|0   0|1|0
0|0|0   0|0|1   0|1|0   1|0|0   0|0|0   0|0|1


0|0|1   0|0|1   0|0|1   0|0|0   0|0|0   0|0|0   1|0|0
0|1|0   0|1|0   1|1|0   0|1|1   0|1|1   1|1|1   0|1|1
0|1|0   1|0|0   0|0|0   0|1|0   1|0|0   0|0|0   0|0|0


0|0|0   0|0|0   1|0|0   0|0|0   1|0|0   1|0|0   1|0|1
0|1|0   1|1|0   0|1|0   1|1|0   0|1|0   0|1|0   0|1|0
1|0|1   0|0|1   0|0|1   0|1|0   0|1|0   1|0|0   0|0|0
```

Figure 6 - Valid Pixel Types

```
X X X X X
X X X X X
X X . X X
X X X . X
X X X X .
            . . .
                  .
                      .
              . . . .   .
                          .                     X X X X X
                              .                 X X X X X
                                . . . . . . .   . X X
                                                X X X X X
                                                X X X X X
```

Figure 7 - End Point Windows

-11-

The parallel extraction process is repeated for all array vectors. Once all of the data has been processed, the master vector running totals are inserted into the attribute file as a measure of parallelism for each master vector.

Note in Figure 5 that an artifact of the thinning process causes "false" center lines to be generated in "corners" (see Figure 8). Normally, these false centers are eliminated when the end point test is made. However, occasionally short vectors perpendicular to larger vectors are attributed a large percentage of parallelsim due to this process.

A second approach to parallelism extraction was generated in which the slope of a vector at any point is measured by means of a template match and used to select a second template which looks for parallel points (see Figure 9). This approach was tested on selected array loads of data; since it avoids the false center problem, it appears to be worth further investigation.

2.4.2 Junction Classification

During the raster-to-vector classification process, the CONTAGRID software records the presences or absence of junction points. For example, the input, as shown in Figure 10a, will produce the master vectors and junction points as shown in Figure 10b.
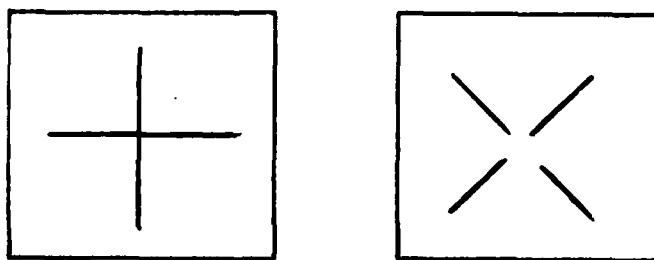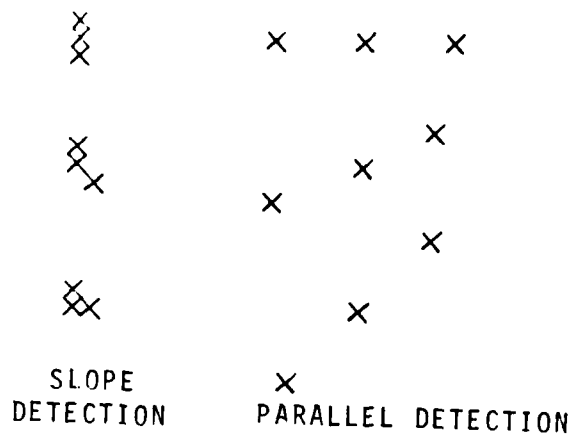

Figure 8 - False Centers

SLOPE
DETECTION

PARALLEL DETECTION

Figure 9 - Correspondence Between Slope Detector and
Parallel Detector Templates

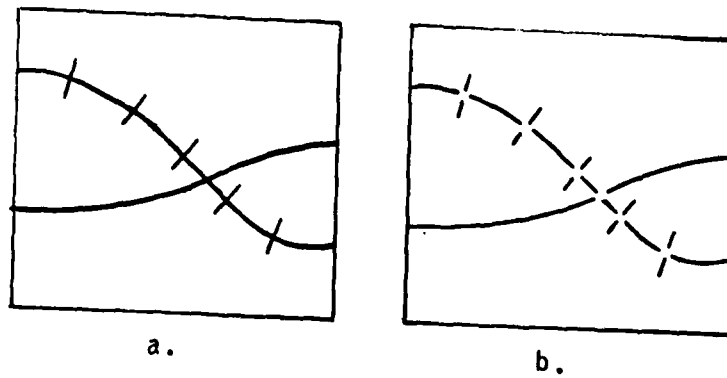a.   b.

Figure 10 - Junctions

If a vector stops without joining another vector, that end of the vector is said to be open. Vectors can have two junction end points, two open end points, or one open and one junction point. All of this information is stored in the master vector attribute file when it is created.

When input raster data similar to Figure 1 is vectorized, the junctions are not always well formed and several different junction patterns are possible. Figure 11 shows all that were detected in the test data. All patterns except 11e (L junction) and 11f (Double ties)were expected. The double ties pattern apparently is generated when a tie has a "hole" in the rasterized representation. This hole causes the vectorization process to generate two vectors, one on each side of the hole. These vectors have the same starting and end points and usually the same length.

2.4.3 Length

During the vectorization process the pixel length of each vector is calculated. The pixel length is the number of pixels in the vector and is not related to the distance between the start and end points. For example, a closed contour may have a length of several thousand pixels, but the start and end points will be adjacent.

2.5 FEATURE CLASSIFICATION ALGORITHMS

2.5.1 Dual Cased Road Tagging

The Dual Cased Road classification routine searches the attribute file for master vectors with a "large" (a parameter) percentage of parallelism. The parallelism attribute extracted is a cumulative length measure of parallel array vectors. The length of the parallel part of a master vector is compared to its total length to determine the percentage of parallelism.
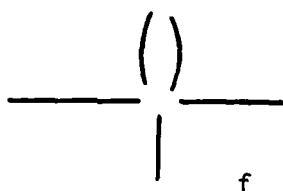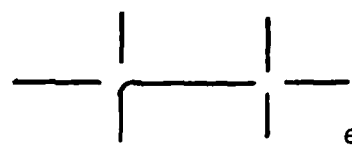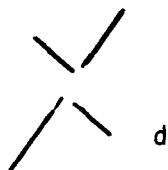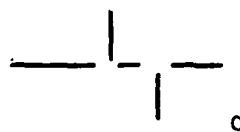
-14-

Figure 11 - Junction Patterns

If a vector has sufficient parallelism, is not part of a railroad vector, and is a minimum length, it is tagged as a Dual Cased Road vector.

2.5.2 Railroad Tagging

The railroad tagging routine searches the attribute file for master vectors which show the junction characteristics of a railroad. First, the "ties" of a railroad are found by searching for a vector with minimum and maximum length attributes and one open and one closed (junction) end point. Then, the junction end point of each vector is found and its X and Y coordinate values extracted (the junction point). Next, a search is made for all vectors with end points in the vicinity of the junction point. If exactly four vectors are found, two ties (a vector with one open end point and one closed end point) and two rails (a vector with two closed end points); then, all four vectors are tagged as railroads.

After all railroad vectors have been found in this manner, all the rails are processed for linkages. The junction number of each rail is determined and searched for in the attribute file. Any vector with the same junction number is tagged as a railroad.

This simple, but refined approach to railroad tagging, successfully finds virtually all of the junction patterns resulting from vectorization. The one pattern not found (the "L" junction) is shown in Figure 11e. Junctions with this pattern can be found by determining the rail of the neighboring junction.shown on the right in Figure 11e. The length and position of the rail entering the junction can be used to predict the location of the "L" junction. A search for vectors with end points in the vicinity would then "find" the missing rail and tie of the junction and all vectors could be tagged as railroads. Unfortunately, time did not permit the implementation of this algorithm. Fortunately, it is not an extremely common junction pattern and the linking process is capable of tagging all the vectors at such junction patterns in this set of test data. This algorithm should be included, however, in any production situation.

-16-

2.6 ALGORITHM DEVELOPMENT SUPPORT SOFTWARE

In order to effectively develop Feature Tagging algorithms, a number of support routines had to be developed. These routines provided the ability to display the master vectors and array vectors on the Comtal so that the original data base could be studied and the effects of algorithm changes could be determined interactively.

The first routine allowed the display of up to 4 array loads of data on the Comtal in raster form. By selecting the array load and swath number, any portion of the test data can be observed. This capability was crucial in determining the basic parameters to be used in the various algorithms.

The array vector display routine, however, could only display a relatively small area. Thus, a "shrink" routine was developed which could reduce the "size" of the array loads so that an area 13 array loads by 16 array loads could be displayed. This corresponds to an area about 2½" by 2½" on the original image.

Finally, a routine was written to produce a list of master vectors that were tagged as Dual Cased Roads or Railroads. These lists provide the input for a routine which will selectively display only those array vectors which are components of master vectors on the list. These routines were invaluable in determining the effectiveness of the attribute extraction and feature classification programs. They were used to produce all the imagery shown in this report.

SECTION 3 - CONCLUSIONS AND RECOMMENDATIONS

The purpose of the Feature Tagging effort was to prove the feasibility of developing algorithms that could automatically identify and tag cartographic features. In particular railroads and dual cased roads, were studied*. The results of this effort are quite impressive. In the case of railroads, the algorithms developed, programmed, and tested are capable of detecting and correctly tagging 96.5% of the railroad vectors. Only 8 non-railroad vectors were falsely tagged. For dual cased roads, only one vector out of 59 was missed while seven were falsely tagged. Table I summarizes the classification results. Figures 12, 13, and 14 show the original test area, the railroad separation, and the dual cased road separation, respectively.

In addition to the actual development of the algorithms, the Feature Tagging program has resulted in a set of software that is a powerful, economical tool for the development of future algorithms for the automatic detection of the entire set of cartographically symbolized features. The techniques of structured design and programming, and the strategy of keeping attribute extraction separated from feature classification were used to achieve this result.

The use of near-neighbor pixel comparisons on thinned, vectorized scan data has been shown to be a significant algorithmic approach to attribute extraction. The parallel processing capabilities of STARAN results in a speed factor to make this approach viable for the large data sets present in cartographic processing. However, since the STARAN has a single instruction stream, like conventional computers, the algorithms and software organization developed under this contract can be universally applied to all sequential computers.

*Broken lines were not included in the present investigation so that railroads and dual cased roads could be thoroughly studied.

TABLE I - CLASSIFICATION RESULTS

NUMBER OF:	ALL TYPES	RAILROADS	DUAL CASED ROADS
VECTORS	1109	574	59
VECTORS CORRECTLY IDENTIFIED	612	554 (96.5%)	58 (98.3%)
VECTORS FALSELY TAGGED	15	8 (1.5%**)	7 (0.7%*)
VECTORS TAGGED	627	562	65
VECTORS NOT TAGGED	21	20 (3.5%)	1 (1.7%)

*PERCENTAGE FIGURED AS A RATIO OF VECTORS FALSELY TAGGED TO ALL VECTORS NOT OF THE CLASS.
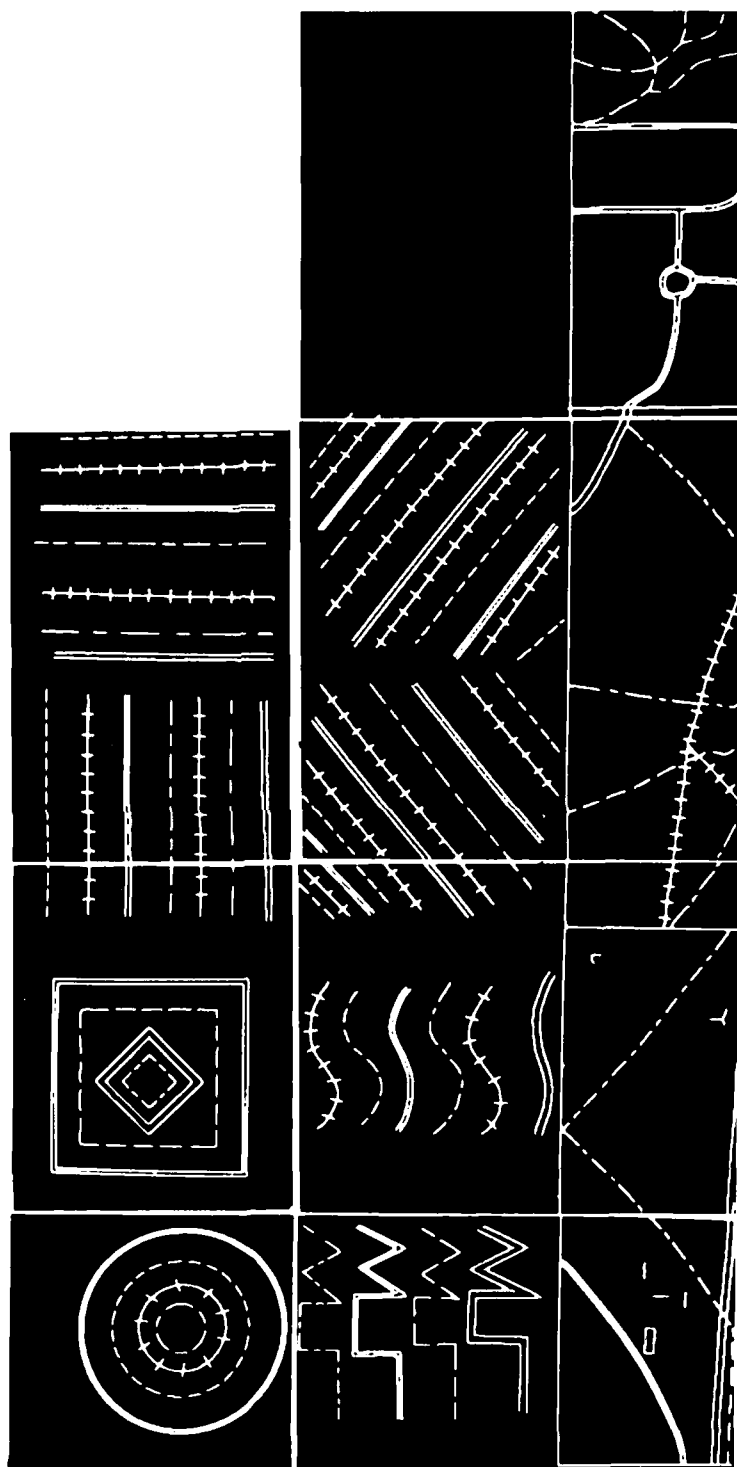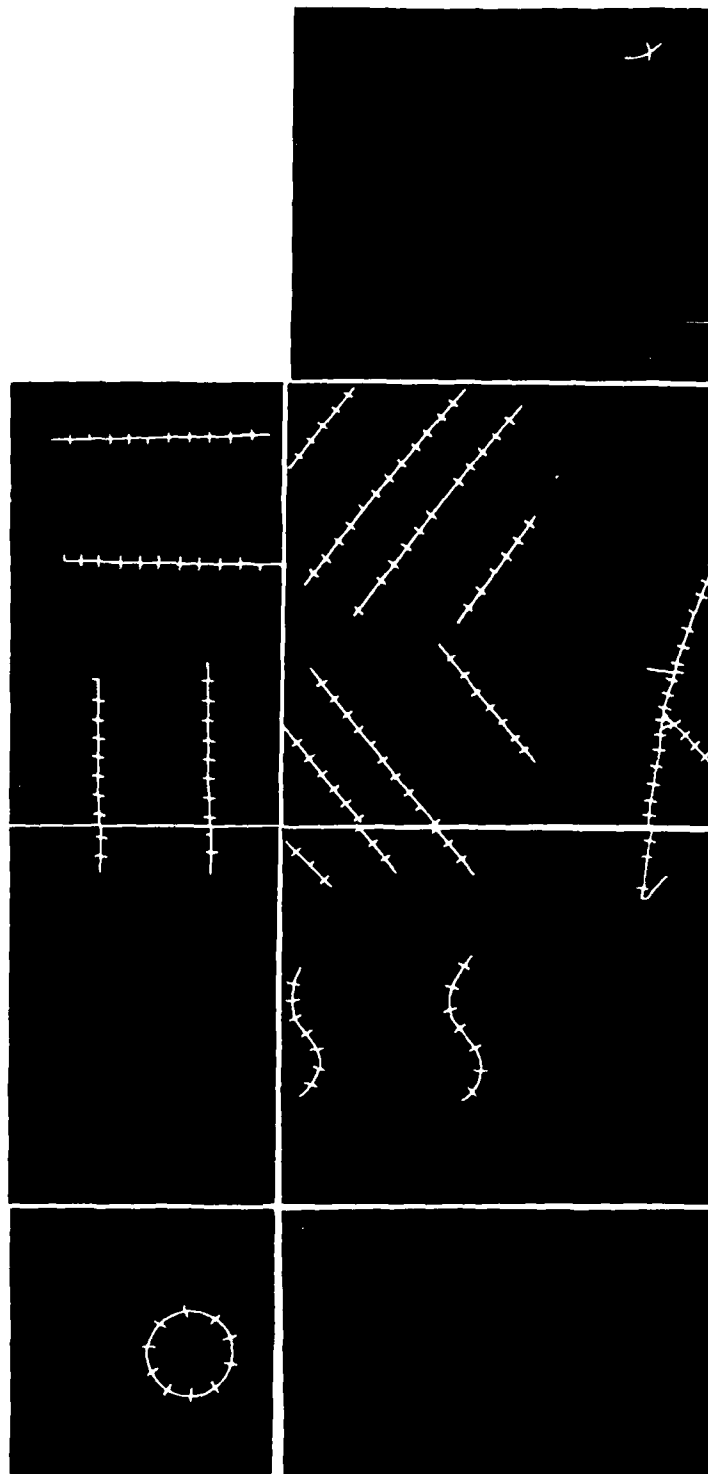
Figure 12 - Original Test Image
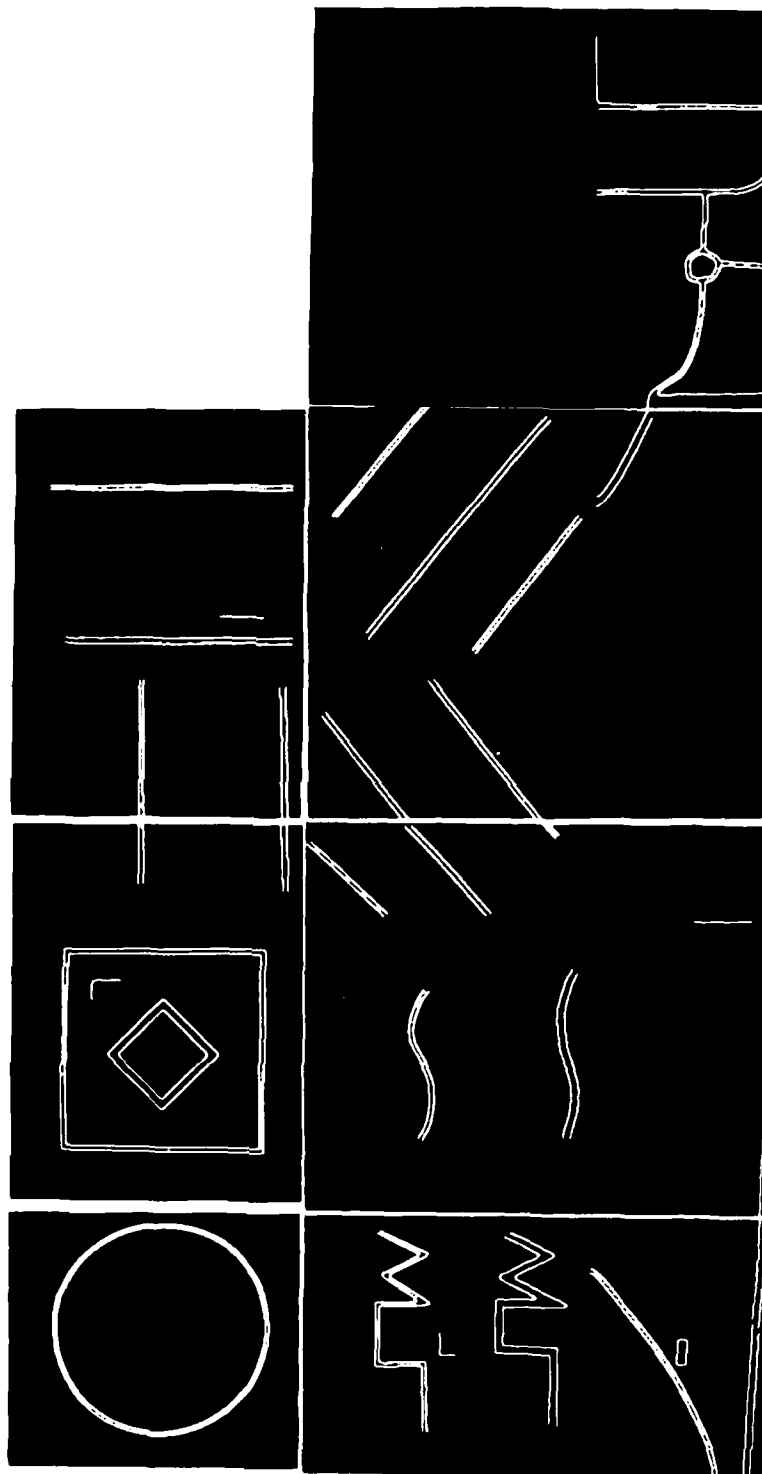
Figure 13 - Railroad Features

Figure 14 - Dual Cased Road Features

A significant conclusion of this investigation is that both raster and vector domain processing are required to efficiently extract the attributes required for automatic tagging. For example, "parallelism" is easier to measure in the raster (pixel) domain where the true two-dimensional nature of the original data is preserved, while the junction information required for tagging railroads is easier to extract from the vector data itself.

The success of this first effort warrants the further development of feature tagging techniques along several fronts. First, the algorithms developed should be tested against actual map sheet data so that an accurate estimate of the effectiveness of the algorithms on data relevant to the actual needs of DMA can be determined.

Then, analysis should be performed to determine the cost benefit of such algorithms and the most cost effective approach to implementing the algorithms in a production environment. For example, no algorithm will be capable of perfect tagging; therefore, should the algorithm parameters be set to error on the side of undertagging or overtagging? Since initial algorithm development efforts commonly emphasises techniques instead of speed (as did the effort reported here). A portion of this analysis should include timing.

In addition, the feature tagging effort should be expanded to develop algorithms for additional cartographic symbols such as broken lines. These algorithms should not only be classified by symbol type (broken lines), but by symbolic meaning (political boundary, trail, supplementary contours, etc.) also.

Goodyear Aerospace feels that the Feature Tagging effort not only has proven the feasibility of automatic feature tagging, but has provided a clue to automatic ridge-stream line generation. The "false center" line artifact produced in the parallelism attribute extraction algorithm (see Figure 8) could be used to find the "center" (i.e., stream line) between two contour lines (see Figure 15). The potential of this approach to automatic R/S line generation is worth further investigation.
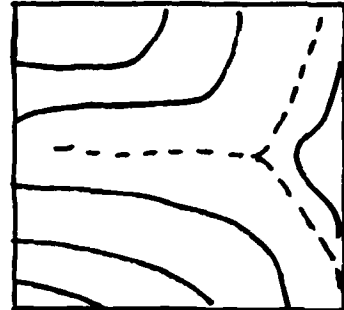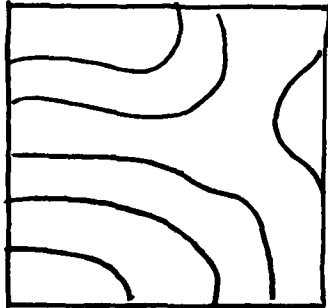
Figure 15 - Ridge-Stream Line Generation