INFORMATION TO USERS

While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. For example:

- Manuscript pages may have indistinct print. In such cases, the best available copy has been filmed.

- Manuscripts may not always be complete. In such cases, a note will indicate that it is not possible to obtain missing pages.

- Copyrighted material may have been removed from the manuscript. In such cases, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or as a 17”x 23” black and white photographic print.

Most photographs reproduce acceptably on positive microfilm or microfiche but lack the clarity on xerographic copies made from the microfilm. For an additional charge, 35mm slides of 6”x 9” black and white photographic prints are available for any photographs or illustrations that cannot be reproduced satisfactorily by xerography.
Detection of critical points: The first step to automatic line generalization

Thapa, Khagendra, Ph.D.
The Ohio State University, 1987

Copyright ©1987 by Thapa, Khagendra. All rights reserved.
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark √.

1. Glossy photographs or pages ______
2. Colored illustrations, paper or print ______
3. Photographs with dark background ______
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy /
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages /
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print ______
11. Page(s) ________ lacking when material received, and not available from school or author.
12. Page(s) ________ seem to be missing in numbering only as text follows.
13. Two pages numbered ______. Text follows.
14. Curling and wrinkled pages ______
15. Dissertation contains pages with print at a slant, filmed as received ______
16. Other ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

University
Microfilms
International
DETECTION OF CRITICAL POINTS: THE FIRST STEP TO AUTOMATIC LINE GENERALIZATION

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree DOCTOR OF PHILOSOPHY in the Graduate School of The Ohio State University

By

KHAGENDRA THAPA, B.Sc., B.Sc(HONS.) CNAA, M.SC.E., M.S.

* * * *

The Ohio State University

1987

Dissertation Committee:
Professor JOSEPH C. LOON
Professor DEAN C. MERCHANT
Professor ANTON F. SCHENK
Professor EDWIN TRIPP

Approved by
Dr. DEAN C. MERCHANT
CO-ADVISER
Dr. Joseph C. LOON
CO-ADVISER
DEDICATION

TO MY PARENTS RANADHOJ and KRISHNA KUMARI THAPA
AND
MY WIFE AND SONS RAJANI, SAMRAT, AND BIRAT.
ACKNOWLEDGEMENTS

I would like to express my appreciation to the co-advisors Dr. D. C. Merchant and Dr. J. C. Loon. I have had very useful discussions with both of them. I shall never forget the moral support that prof. Merchant provided in difficult times. In addition, the USGS maps at different scales were made available by Dr. Loon.

The dissertation advisory committee was formed very early. This was really very beneficial to me. Dr. Schenk was very interested in my work he provided not only the useful guidance but also helped me by providing the Research Associateship. Sometimes I had the tendency of trying to get away with less than perfect results but Dr. Schenk would not give in. I greatly appreciate his financial assistance and scientific insistence.

It was indeed to my benefit to have Dr. Tripp from Computer and Information Science Dept. in my advisory committee. He was very helpful and willing to listen to my problems. I am very much grateful to Dr. Tripp for serving in my advisory and reading committee.

- iii -
During the course of my study here at OSU, I have also ben-
efitted from the legendary professors Mueller, Uotila, and
Rapp.

Dr. Anderson of Math. Dept., Utah State University helped
me by sending some literature as well as software on their
method of vertex detection in digitized curves. The soft-
ware was not used. I thank him for his help. I would also
like to thank Karina Wall of Sweden for clarifying her
algorithm for curve segmentation.

My thanks are also due to Dr. Dunham of Dept. of Electrical
Engineering, Southern Methodist University, for sending his
software on Optimum Method of Curve Segmentation. I also
thank Dr. Robert Bolles of SRI International for sending
his software on critical point detection. I am grateful to
Dr. Mokhtarian of Dept. of Computer Science, British Colom-
bia University, for sending his software on Scale Space
Computation and image matching. However, it should be not-
ed that none of the software sent by others was useful to
me. Therefore, none of the software written by others was
used.

I thank my wife for taking care of our sons when I have had
no time to spend with them even on weekends and evenings.
I express my appreciation to all those who participated in the manual critical points detection of my test curve. I also thank the librarians H. Mercado, Mrs. K, Mary Amorose, and Ron Cole for helping me to find many articles and books. Larry Shubat was the first to read and find some typing errors.

The computer funds for performing this research were provided by the Instruction and Research Computer Center, The Ohio State University.
VITA


1978-1980 .............. Research and Teaching Assistant Dept. of Surveying Engineering, University of New Brunswick, Fredericton, Canada.


1982-1984 .............. Graduate Teaching Associate, Department of Mathematics, The Ohio State University.

1984- Present ........... Graduate Teaching and Graduate Research Associate, Dept. of Geodetic Science and Surveying, The Ohio State University.

1985 ...................... M. S. Dept. of Geodetic Science and Surveying, The Ohio State University.

- vi -
PUBLICATIONS


FIELDS OF STUDY

Major Field: Geodetic Science and Surveying


Studies in Inertial Geodesy: Professor K. P. Schwarz.


- vii -
Studies in Statistics and Numerical Analysis
Professors: A. Dean, R. Sabat, and D. B. Chhetry.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDICATION</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>VITA</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xi</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>I. CRITICAL POINT DETECTION AND LINE GENERALIZATION</td>
<td>9</td>
</tr>
<tr>
<td>1.1 Definition of Critical Points</td>
<td>9</td>
</tr>
<tr>
<td>1.2 Definition of Line Generalization</td>
<td>12</td>
</tr>
<tr>
<td>1.3 The Need for Line Generalization</td>
<td>16</td>
</tr>
<tr>
<td>1.4 Importance of Critical Point Detection in Line Generalization</td>
<td>19</td>
</tr>
<tr>
<td>1.5 Advantages of Critical Point Detection</td>
<td>23</td>
</tr>
<tr>
<td>1.6 Proposition for Critical Points Detection and Line Generalization</td>
<td>24</td>
</tr>
<tr>
<td>II. LITERATURE REVIEW</td>
<td>26</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>26</td>
</tr>
</tbody>
</table>
### 2.2 Algorithms for Finding Critical Points

- **2.2.1 Classification of the Critical Point Detection Algorithms**
- **2.2.2 Local Algorithms**
- **2.2.3 Global Algorithms**

### 2.3 Algorithms for Line Generalization

### 2.4 Algorithms for Line Smoothing

- **2.4.1 Smoothing Methods**
- **2.4.2 Interpolating Methods**

### 2.5 Existing Methods for Evaluation of Line Generalization

### 2.6 Summary

### III. DESCRIPTIONS OF THE METHODS OF FINDING CRITICAL POINTS

- **3.1 Introduction**
- **3.2 Critical Points Detection Using Zero-crossings**
  - **3.2.1 Vector to Raster Conversion**
  - **3.2.2 Chain Encoding**
  - **3.2.3 Second Derivative of The Gaussian**
  - **3.2.4 Convolution**
  - **3.2.5 Post Processing Using a Dynamic Mask**
  - **3.2.6 Step by Step Procedure for Critical Points Detection Using Zero-Crossings**
- **3.3 Critical Points Detection Using Normalized Symmetric Matrix**
  - **3.3.1 The Nature of Scatter Matrices and Their Eigenvalues**
  - **3.3.2 Eigenvalues of the Normalized Symmetric Scatter Matrix(NSS)**
  - **3.3.3 Algorithm to Detect Critical Points Using NSS Matrix**

### IV. RESULTS OF DETECTION OF CRITICAL POINTS IN DIGITIZED CURVES

- **4.1 Introduction**
- **4.2 Vector to Raster Conversion and Pixel Size Selection**
- **4.3 Effect of Using Different Signals on Zero-Crossings**
- **4.4 Effect of Changing the Scale of Gaussian**
4.5 Effect of Thresholding the Convoluted Values on Zero-Crossings ... 111
4.6 Effect of Rotation of the Signal on Zero-Crossings .................. 114
4.7 Effect of Smoothing the Signal on Zero-Crossings .................... 117
4.8 Effect of Taking Common Zero-Crossings from Different Channels ... 119
4.9 Raster to Vector Conversion ........................................ 121
4.10 Comparison between Manual and Algorithmic Critical Points
     Detection .................................................................. 124
     4.10.1 Manual Critical Points
            Detection: The Experiment ...................................... 124
4.11 Comparison with the Results of Marino ............................... 128
4.12 Results of Critical Points
     Detection by NSS Matrix ............................................. 133

V. LINE GENERALIZATION USING ZERO-CROSSINGS ...................... 134

5.1 Introduction .................................................................. 134
5.2 Selection of Lines for Testing and Evaluation ....................... 134
5.3 Inapplicability of Forsen's Algorithm To Line Generalization .... 137
5.4 Comparison between Algorithmic and Manual Line Generalization 141
5.5 Effect of Step by Step Generalization ................................ 148
5.6 Line Generalization: Critical Points
     Detection vs Shape Preservation .................................... 150

VI. EVALUATION OF LINE GENERALIZATION ................................. 153

6.1 Introduction .................................................................. 153
6.2 Evaluation by Comparing with the Results Obtained Manually ...... 154
6.3 Evaluation Using McMasters Mathematical Methods ................. 155
6.4 Computation of McMaster's Mathematical Measures ................. 155
6.5 Analysis ..................................................................... 158
6.6 Checking the Algorithmic and Cartographic Criteria of Zero-
     Crossings .................................................................. 160
     6.6.1 Cartographic Criteria ......................................... 160
     6.6.2 Algorithmic Criteria ........................................... 161
<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Illustrates the stability of the critical points</td>
<td>11</td>
</tr>
<tr>
<td>2. Reumann and Witkam's Tube along the Curve</td>
<td>36</td>
</tr>
<tr>
<td>3. Comparison between critical lines formed by ESA</td>
<td>38</td>
</tr>
<tr>
<td>4. The circles about each point $P_i$ define sectors</td>
<td>40</td>
</tr>
<tr>
<td>5. Selection of critical points by Sklansky and Gonzalez</td>
<td>43</td>
</tr>
<tr>
<td>6. Illustration of the Forsen's Algorithm</td>
<td>51</td>
</tr>
<tr>
<td>7. Line and Grid Location for Bresenham's Algorithm</td>
<td>77</td>
</tr>
<tr>
<td>8. Simple Freeman Chain-Code and Pixel Neighbourhood</td>
<td>81</td>
</tr>
<tr>
<td>9. Second derivative of the Gaussian mask for sigma=12</td>
<td>85</td>
</tr>
<tr>
<td>10. Pixel plot of the test figure. Pixel size 0.25 mm.</td>
<td>99</td>
</tr>
<tr>
<td>11. Zero-Crossings of the SFC signal</td>
<td>102</td>
</tr>
<tr>
<td>12. Zero-Crossings of the EFC signal</td>
<td>103</td>
</tr>
<tr>
<td>13. Zero-Crossings of the CEFC signal</td>
<td>103</td>
</tr>
<tr>
<td>14. Zero-Crossings of the SCEFC signal</td>
<td>104</td>
</tr>
<tr>
<td>15. Zero-Crossings of the Curvature signal</td>
<td>104</td>
</tr>
<tr>
<td>16. Connection of the Zero-Crossings of figure 15 by straight lines</td>
<td>105</td>
</tr>
</tbody>
</table>
17. Zero-Crossings for SCEFC Signal with sigma = 3 ..............................................108
18. Zero-Crossings for SCEFC Signal with sigma = 12 ............................................109
19. Zero-Crossings for Curvature Signal with sigma = 3 ......................................109
20. Zero-Crossings for Curvature Signal with sigma = 12 ...................................110
21. Connection of the Zero-Crossings of figure 20 by straight lines .......................110
22. The Results of Thresholding on Zero-Crossings ...........................................112
23. The Results of Thresholding on Zero-Crossings ...........................................113
24. Zero-Crossings which are Common to at least two Channels ..............................121
25. The Results of Raster to Vector Conversion ..............................................123
26. The Results of Connecting the Selected Points of Figure 25 ............................123
27. Points selected by respondants ........................................................................125
28. Results of Critical Points Detection by Zero-Crossings ..................................126
29. Results of Critical Points Selection by Humans .............................................127
30. Results of Critical Points Selection by Humans .............................................129
31. Results of Critical Points Selection ................................................................130
32. Results of Critical Points Selection by NSS Matrix ......................................133
33. Hunting Creek along River Miles at Various Scales .......................................136
34. Comparison between lines generalized by Cartographers ..............................138

- xiv -
35. Result of Enlarging figure 34 (A) by four times ........................................... 139
36. Comparison between lines generalized by Cartographers ........................................... 140
37. Comparison between lines generalized by Cartographers ........................................... 141
38. Result of Enlarging figure 37 (A) by four times ........................................... 143
39. Comparison between lines generalized by Cartographers ........................................... 143
40. Greenwood Creek along River Miles at Various Scales ........................................... 145
41. Comparison between lines generalized by Cartographers ........................................... 146
42. Comparison between lines generalized by Cartographers ........................................... 147
43. Comparison between lines generalized by Cartographers ........................................... 148
44. Comparison between lines generalized by Cartographers ........................................... 149
LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>DESCRIPTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Zero-Crossings for Different Signals</td>
<td>101</td>
</tr>
<tr>
<td>2.</td>
<td>Effect of Changing Sigma on Zero-Crossings</td>
<td>106</td>
</tr>
<tr>
<td>3.</td>
<td>Effect of Rotating the Curve on Zero-Crossings</td>
<td>115</td>
</tr>
<tr>
<td>4.</td>
<td>Effect of Rotating the Curve on Zero-Crossings</td>
<td>115</td>
</tr>
<tr>
<td>5.</td>
<td>No. of Zero-Crossings for Different Levels</td>
<td>118</td>
</tr>
<tr>
<td>6.</td>
<td>Summary of three Stages of Critical Points Detection</td>
<td>131</td>
</tr>
<tr>
<td>7.</td>
<td>Various Mathematical Measures Computed</td>
<td>157</td>
</tr>
<tr>
<td>8.</td>
<td>Various Mathematical Measures Computed</td>
<td>158</td>
</tr>
</tbody>
</table>
INTRODUCTION

This is an information age. We the Geodetic Scientists are experts on spatial information. One of the branches of Geodetic Science viz: Cartography is concerned with the collection, storage, manipulation, and display of spatial information.

The spatial information in Cartography is collected at various scales usually at large scales through different means e.g. by ground survey, by analytical photogrammetric methods, by digitizing (or scanning) existing maps, or from satellite imagery.

Generally the amount of data collected is much more than what is required to adequately represent a particular feature. Therefore, there is a need for data compaction or data compression.

The spatial information that is displayed in Cartography needs to be portrayed at different scales for various purposes. As the scale is reduced, one needs to further compress the data and remove the unwanted and unimportant detail, if one wants the displayed information to be clear,
concise, and uncluttered. This can only be achieved through the process of generalization.

Now-a-days more and more information for Cartographic purposes is gathered in raster mode, for example, the scanning of existing maps, or the information gathered from satellite imagery, or the results of edge detection from photographs. However, graphic plotters usually work on vector mode. In addition, it is much cheaper to store data in vector mode. Therefore, there is a need for an efficient method of raster to vector conversion.

Computers have played and are continuing to play a major role in the field of cartography. However, the process of map production especially when a change in scale is required (which is usually the case) cannot be made automatic despite the concerted effort by many government agencies such as DMA, and USGS as well as many private firms. The bottleneck for this has been the process of line generalization. There does not exist a satisfactory method which can generalize lines automatically as the scale of the map changes. So, even today, generalization of lines in maps is performed interactively or manually.

This dissertation addresses the above mentioned problems of cartography. The problems of data compaction and raster to vector conversion are solved by the process of finding
critical points in the raster data by using the method of zero-crossings of the convoluted values of the second derivative of the Gaussian with the signal derived from the digitized data. The problem of automatic line generalization from any larger scale to any smaller scale is also solved by finding the zero-crossings. The data compaction in vector data is achieved by analyzing the eigenvalues of the normalized symmetric scatter matrix derived from the data.

The evaluation of the generalized lines will be performed by comparing the corresponding lines generalized by zero-crossings algorithm with those performed by cartographers. In addition, some mathematical measures proposed to evaluate line generalization are examined.

By finding the zero-crossings of the convoluted values of second derivative of the Gaussian with the various signals (based on chain code and its derivatives), it has been possible to solve the above problems. Dramatic results in the process of data compaction by finding the critical points of the digitized lines have been found. Moreover, the algorithm is found to be very useful for raster to vector conversion. Furthermore, the lines generalized by zero-crossings algorithm presented in this research are compared with those performed by other cartographers. It was found
that the lines generalized by zero-crossings algorithm are almost identical to those produced by cartographers.

As described in chapter three, there is a plethora of literature on the methods of line generalization. However, none of them work satisfactorily because the existing methods only perform either data compaction or data smoothing. Therefore, the zero-crossings method of line generalization presented here is superior to the existing methods for the following reasons:

1. Zero-crossings method achieves line smoothing and data compaction in one step.
2. When one performs smoothing using the existing methods such as weighting, the smoothed coordinates are different from the existing points. These points might be completely off the original points and therefore may violate National Map Accuracy Standards. Unlike the existing methods zero-crossings algorithm does not compute any new coordinates.
3. The zero-crossings method of line generalization gives results which are very close to those obtained by conventional cartographers. However, the only disadvantage is that it cannot introduce the errors due to negligence and fatigue that a cartographer is likely to make. In other words, the results obtained by zero-
crossings algorithm are consistent, unlike those of cartographers.

4. The zero-crossings method of critical points detection can mimic humans.

The following are the most important results of my research and my contribution to the scientific community:

1. I have covered extensive literature search from many disciplines such as Computer Graphics, Pattern Recognition, Computer Vision, Artificial Intelligence, Electrical Engineering, and Cartography.

2. I have modified the zero-crossings algorithm by using different signals and by introducing the idea of thresholding. The previous researchers e.g. Mokhtarian and Mackworth (1986) and Asada and Brady (1986) used common zero-crossings from at least two channels instead of thresholding the zero-crossings of one channel.

3. Using the zero-crossings of the convoluted values of the second derivative of the Gaussian with the curvature signal derived from the digitized data, I was able to automatically generalize lines even when there was a drastic change in scale. Moreover, the results of generalized lines obtained by the zero-crossings algorithm are very close to those obtained by conventional cartographers. These results were achieved
using existing USGS maps at various scales. The practical significance of these results to the mapping community are rather obvious.

4. The zero-crossings of the convoluted values of the second derivative of the Gaussian with the signals derived from the chain code provide an efficient and reliable means of raster to vector conversion. This had never been pointed out before even though the zero-crossings are known to be rich in information (Marr and Hildreth, 1980).

5. The analysis of the eigenvalues of the normalized symmetric scattered matrix provides an useful way of detecting critical points and data compaction in vector data. This is also a new technique found by me during the course of my research.

6. By using different channels and by thresholding, I was able to automatically detect the same set of critical points as those selected by cartographers and non-cartographers in digitized curves. To my knowledge, such an algorithm did not exist before.

7. I found that when the process of line generalization involves a drastic change in scale e.g. 1:24000 to 1:100000 or smaller, what one needs to preserve are the basic shape and general character of a line and not the critical points. The latter conclusion is con-
trary to the findings and beliefs of some of the previous researchers.

In chapter one, various definitions of line generalization and critical points detection are given. In addition, the importance of critical points detection and the need for line generalization are outlined.

In chapter two, an extensive literature review on critical points detection, line generalization, and line smoothing is given. I have covered literature from many disciplines including Artificial Intelligence, Computer Vision, Image Processing, Pattern Recognition, Computer Graphics, and Cartography.

In chapter three, the methodology for critical points detection and line generalization is explained.

In chapter four, the results of critical points detection, vector to raster conversion, and effects of using different signals in the convolution are examined. Moreover, the critical points detected by zero-crossings algorithm are compared with those detected by humans. In addition, various tests performed such as the effect of smoothing the signal to check the robustness of zero-crossings method of finding critical points detection are included.
In chapter five, the inapplicability of the existing methods of line generalization, when there is a large change in the scale, is demonstrated. Further, comparisons between the lines generalized by zero-crossings algorithm and those made by cartographers using maps at different scales but of the same area, are made.

In chapter six, visual as well as mathematical methods of evaluating line generalization are examined.
CHAPTER I
CRITICAL POINT DETECTION AND LINE GENERALIZATION

In this chapter the definitions of a critical point, and of line generalization are given. In addition, the need for line generalization and the importance of critical points detection in line generalization are outlined.

1.1 DEFINITION OF CRITICAL POINTS

Before defining the critical points, it should be noted that critical points in a digitized curve are of interest not only in the field of Cartography but also in other disciplines such as Pattern Recognition, Image Processing, Computer Vision, and Computer Graphics. Marino (1979) defined critical points as "Those points which remain more or less fixed in position, resembling a precis of the written essay, capture the nature or character of the line".

In Cartography one wants to select the critical points along a digitized line so that one can retain the basic character of the line. Researchers both in the field of Computer Vision and Psychology have claimed that the maxi-
ma, minima and zeroes of curvature are sufficient to preserve the character of a line. In the field of Psychology, Attneave (1954) demonstrated with a sketch of a cat that the maxima of curvature points are all one needs to recognize a known object. Hoffman and Richards (1982) suggested that curves should be segmented at points of minimum curvature. In other words, points of minimum curvature are the critical points. They also provided experimental evidence that humans segmented curves at points of curvature minima.

Because the minima and maxima of a curve depend on the orientation of the curve, the following points are considered as critical points:

1. curvature maxima
2. curvature minima
3. end points
4. points of intersection

It should be noted that Freeman (1978) also includes the above points in his definition of critical points.

Hoffman and Richards (1982) state that critical points found by first finding the maxima, minima, and zeroes of curvature are invariant under rotations, translations, and uniform scaling. Marimont (1984) has experimentally proved that critical points remain stable under orthographic projection as shown in figure 1.
Figure 1: Illustrates the stability of the critical points under orthographic projections. Left are the critical points of a plane curve. On the right the curve is projected orthographically at various orientations and the critical points of the resulting curve are marked (derived from Marimont, 1984). (Note: The symbols used to mark the critical points are slightly offset from the points.)

The use of critical points in the fields of Pattern Recognition, and Image Processing has been suggested by Brady (1982), and Duda and Hart (1973). The use of critical points in Cartography was proposed by Solovitskiy (1974), Marino (1978), McMaster (1983), and White (1985).
1.2 DEFINITION OF LINE GENERALIZATION

The manual process of line generalization is highly subjective, so much so that the same person cannot achieve identical results when he/she performs generalization of a line at different times. Perhaps it is due to the subjective nature of this process that there is no single definition of line generalization in the literature. Keates (1981) correctly claims that there is no universal agreement on the definition of line generalization. Everyone tends to define line generalization the way they view it. Some Cartographers for example Robinson and Petchenik (1976) have gone to the extent of calling line generalization as elusive as anything in print.

The fact that line generalization is subjective and depends on the personal feelings of the Cartographer was recognized as far back as 1908. In 1908 Eckert stated, "Generalization depends on personal and subjective feelings and was part of the art of map making process" (Clayton, 1985).

Probably the definition which explains what is really involved in the practice of line generalization is the one given by Hettner (McMaster, 1983). According to Hettner "Generalization of a map is first of all a question of restriction and selection of source material. This is achieved partly by simplification of the objects on map, partly by omitting the small or less interesting objects". 
Part of the reason for so many definitions and interpretations of line generalization could be due to the lack of any "rules" for this process. So Lundquist (1959) attempted to devise the "rules" for line generalization and came up with the following:

1. Purpose of the map
2. Reduction factor (scale)
3. Objective evaluation using an examination of relevant data
4. Conscious effort to avoid personal prejudice
5. Attempt at uniformity
6. Local importance factor stressing a need for regional knowledge.

In a way this was an attempt to make the process of line generalization objective.

Steward (1974) takes a broader view of generalization and states that it is a conceptual term of universal significance and therefore applicable to all branches of intellectual activity and scholastic discipline. Line generalization in particular is an aspect of pattern analysis which is a derivative of an inductive process.

Freeman (1973) claims that line drawing is a medium of communication. Therefore, when one considers line generalization he/she is essentially referring to the processing of information depicted by line drawings.
According to Robinson et al. (1985) the processing of information portrayed by the line depends on the following factors:

1. Purpose of the map
2. Scale of the map
3. Graphic limits
4. Quality of data.

It is interesting to note that Lundquist (1959) is more explicit about these factors than Robinson et al. (1985). Moreover, it should be pointed out that researchers working in the field of Computer Vision, for example, Fischler and Bolles (1986, 1983) also realize that curve partitioning or curve segmentation depends on the purpose and scale. This is not unexpected because line generalization is in a way a problem of curve segmentation.

US Defense Mapping Agency (DMA) is one of the major producers of maps. DMA defines map generalization in the Glossary of Mapping Charting and Geodetic Terms as "Smoothing of the character of features without destroying their visible shape. Generalization increases as map scale decreases". In addition, the DMA manual compilation guide (Zoraster et al., 1984) states "...contours should be generalized by smoothing the fine detail but retaining the basic shape".
Morrison (1974) and Robinson et al. (1985) define map generalization as involving the processes of simplification, classification, symbolization and induction. For an explanation of these terms refer to Robinson et al. (1985).

Pannekoek (1962) emphasizes the fact that the essential character of the line should be retained. "As a general rule, it should be said that the features that determine the essential character of the terrain should be stressed and non-essentials should remain subordinate to them or omitted altogether" (Pannekoek, 1962).

Keates (1973) also notes the fact that the major elements of the shape of the line should be retained. "The first step is to select the individual features which are to be retained at the smaller scale, which at the same time will continue to represent the general characteristics of the ... area. In addition, each individual feature has to be simplified in form by omitting minor irregularities and retaining only the major elements of shape" (Keates, 1973).

An examination of the above definitions of generalization enables one to observe the following facts:
1. Currently no unique and standard definition of line generalization exists.
2. It is difficult to list a set of objective criteria for evaluating line generalization.
3. Many researchers are not exactly sure as to what is really involved in line generalization.
4. Retention of the basic shape (character) seems to be the right way to approach line generalization.

1.3 THE NEED FOR LINE GENERALIZATION

Unless one wants to depict the environment at a scale of 1:1 in terms of location and description, one has to approximate reality. Moreover, no one can portray all the natural features present in the environment. In addition, one has to use different symbols to represent different features.

It should be noted that one major approximation of reality (in a sense generalization) takes place when one decides to represent the complex three dimensional earth (Steward, 1974) into two-dimensional plane. Further, a second approximation or generalization is carried out by Surveyors and Photogrammetric compilers when they decide as to which information to collect and how much detail to retain. Their decision is based on the purpose and scale of the map.

In fact the scale of the map governs the type of instruments and the methods of surveying used at the stage of data collection. For example, it would be a waste of time, money, and energy if one used precise methods of detail
survey when mapping at a scale of 1:50000 or smaller is required.

The purpose of the map plays an important role in deciding which features to retain in a map. For example, if one is planning to produce road maps, one will not be too worried about the accuracy of depiction of rivers, or lakes, or contour lines.

The extent of the area to be covered plays an important role in deciding which method of mapping to use. For example, if one is planning to map only a small area at large scale, it would not be a good idea to use Photogrammetric methods. On the other hand, mapping of large areas is almost invariably performed by Photogrammetric methods. But again, the scale of the photographs to be taken and the type of restitution instrument used for compilation depends very much on the scale of the map to be produced. Photogrammetrists such as Burnside(1985) suggest that the ratio of scale of photographs to the scale of maps should not be less than 1 is to 6.

The amount and type of detail collected by Photogrammetrists and Surveyors also depends on the scale and purpose of the map. For example, if the map is to be produced at large scale for an engineering purpose, almost all the detail including individual trees will be depicted at their
correct planimetric location. However, if the map is to be produced at a scale of 1:50000 a lot of detail will be ignored for the following two reasons (Zoraster et al, 1984):

1. Reduction of natural and cultural features in accordance with the scale of the map and difficulty of symbolization of many of these features at small scale.

2. Communication of the relationships underlying observations of geographic phenomenon through elimination of unnecessary detail or exaggeration of important detail.

The purpose of the above discussion is two fold namely:

1. Generalization is not performed just by cartographers.

2. Generalization takes place even at the data collection level depending on scale and purpose of the map.

In summary, line generalization in Cartography is required because as the scale is reduced one can depict less and less number of features since there is less space available on the map. Moreover, as the scale is reduced lines representing different linear features such as roads, railway lines, transmission lines, rivers, boundaries and coastlines become highly cluttered unless they are simplified. A cluttered map will be aesthetically bad looking and, therefore, will not serve its purpose as a means of graphic communication.
1.4 IMPORTANCE OF CRITICAL POINT DETECTION IN LINE GENERALIZATION

An examination of the various definitions of line generalization given in section 1.2 shows that Cartographic line generalization has hitherto been a subjective process. When one wants to automate a process which has been vague and subjective, many difficulties are bound to surface. Such is the situation with Cartographic line generalization.

One way to tackle this problem would be to determine if one can quantify it (i.e. make it objective) so that it can be solved using a digital computer. Many researchers such as Solovitskiy (1974), Marino (1979), and White (1985) agree that the way to make the process of line generalization more objective is to find out what Cartographers do when they perform line generalization by hand? In addition, find out what in particular makes the map lines more informative to the map readers. Find out if there is anything in common between the map readers and map makers regarding the interpretation of line character.

Marino (1979) carried out an empirical experiment to find if Cartographers and non-cartographers pick up the same critical points from a line. In the experiment she took different naturally occurring lines representing various features. These lines were given to a group of Cartogra-
phers and a group of non-cartographers who were asked to select a set of points which they consider to be important to retain the character of the line. The number of points to be selected was fixed so that the effect of three successive levels or degrees of generalization could be detected. She performed statistical analysis on the data and found that cartographers and non-cartographers were in close agreement as to which points along a line must be retained so as to preserve the character of these lines at different levels of generalization.

The problem of retaining the character of a line or the basic shape of a line arises not only in Cartography but also in Pattern Recognition, Computer Graphics, Image Processing, and Computer Vision (Freeman, 1978), (Marimont, 1984), (Fischler and Bolles, 1986, 1983).

When one says one wants to retain the character of a line what he/she really means is that he/she wants to preserve the basic shape of the line as the scale of representation decreases. The purpose behind the retention of the basic shape of the line is that the line is still recognized as a particular feature—river, coastline or boundary despite of the change in scale. The assumption behind this is that the character of different types of line is different. That is to say that the character of a coastline is different from
that of a road. Similarly, the character of a river would be different from that of a transmission line and so on.

The fact that during the process of manual generalization one retains the basic shape of the feature has been stated by various veteran Cartographers. For example, Keates (1973) states, "... each individual feature has to be simplified in form by omitting minor irregularities and retaining only the major elements of the shape".

Solovitskiy (1974) identified the following quantitative and qualitative criteria for a correct generalization of lines:

1. The quantitative characteristics of a selection of fine details of a line.
2. Preservation of typical tip angles and corners
3. Preservation of the precise location of the basic landform lines.
4. Preservation of certain characteristic points.
5. Preservation of the alternation frequency and specific details.

He further states "The most important qualitative criteria are the preservation of the general character of the curvature of a line, characteristic angles, and corners ..".
In the above list, what Solovitskiy is basically trying to convey is that he wants to retain the character of a feature by preserving the critical points.

Buttenfield (1985) also points out the fact that Cartographers try to retain the basic character of a line during generalization. She states "...Cartographer's attempt to cope objectively with a basically inductive task, namely, retaining the character of a geographic feature as it is represented at various Cartographic reductions".

Boyle (1970) suggested that one should retain the points which are more important (i.e. critical points) during the process of line generalization. He further suggested that these points should be hierarchical and should be assigned weights (1-5) to help Cartographers decide which points to retain.

Campbell (1984) also observes the importance of retaining critical features. He states, "One means of generalization involves simply selecting and retaining the most critical features in a map and eliminating the less critical ones".

The fact that retention of shape is important in line generalization is also included in the definition of line generalization. The DMA definition states as mentioned in section 1.2 "Smoothing the character of features without destroying their visible shape".
Tobler as referenced in Steward (1974) also claims that the prime function of generalization is "... to capture the essential characteristics of... a class of objects, and preserve these characteristics during the change of scale".

The fact that retention of critical points is sufficient to preserve the basic shape of a feature is supported by the research in Psychology by Attneave (1954) and in Computer Vision by Hoffman and Richards (1984) as mentioned in section 1.1.

Boyle (1970) and Fischler and Bolles (1986) also indicate the fact that there is a certain hierarchy of critical points depending on the purpose of curve segmentation in Computer Vision or purpose and scale of the map in Cartography as stated above.

1.5 ADVANTAGES OF CRITICAL POINT DETECTION

According to Pavlidis and Horowitz (1974), Roberge (1984), and McMaster (1983) the detection and retention of critical points in a digital curve has the following advantages:

1. Data compaction as a result plotting or display time will be reduced and less storage will be required.

2. Feature extraction.

4. Problems in plotter resolution due to scale change will be avoided.
5. Quicker vector to raster conversion and vice-versa.
6. Faster choropleth shading. This means shading or color painting the polygons.

Because of the above advantages, research in this area is going on in various disciplines such as Computer Science, Electrical Engineering, Image Processing, and Cartography.

1.6 PROPOSITION FOR CRITICAL POINTS DETECTION AND LINE GENERALIZATION

The basic assumption behind line generalization is that at first the critical points are detected. Different levels of critical points in a curve are detected for different degree of generalization. Once the critical points are detected a line smoothing technique is used to smooth the line.

The zero-crossings of the second derivative of the Gaussian with the signal derived from the data are rich in information (Marr and Hildreth, 1980). Especially when the zero-crossings coinciding spatially to at least two channels are taken (ibid). Note that Marr and Hildreth (1980) discuss this technique with reference to edge detection in image processing. Edges in images are defined as the discontinu-
ities in the first derivative of the gray level. Marr and Hildreth (1980) use gray level as the signal for the convolution with the Laplacian of the Gaussian. They then look for the zero-crossings in convoluted values to detect edges.

This technique may be used to detect critical points in a digitized curve (Asada and Brady, 1986). The level of critical points detection may be changed by simply changing the channel of the Gaussian. The smoothing of the critical points may be achieved by using cubic parametric splines (Spaeth, 1974). Or one can use weighted moving averages.

It was found that it was not necessary to use common zero-crossings from more than one channel in order to detect critical points in a digitized curve. In addition, it was not even necessary to smooth the critical points for line generalization because both smoothing and critical point detection can be done in one step.

It was also found during the course of investigation that when line generalization involves a drastic change in scale say from 1:24000 to 1:250000, it is not the critical points but the general shape of the curve that one needs to retain. This is contrary to the basic assumption made by Cartographers e.g. (White, 1985) and at the beginning of this research.
CHAPTER II
LITERATURE REVIEW

2.1 INTRODUCTION

The proliferation of computers not only has had a great impact on existing fields of studies but also created new disciplines such as Computer Graphics, Computer Vision, Pattern Recognition, Image Processing, Robotics etc. Computers play an ever increasing role in modern day automation in many areas. Like many other disciplines, Mapping Sciences in general and Cartography in particular have been greatly changed due to the use of computers.

It is known from experience that more than 80% of a map consists of lines. Therefore, when one talks about processing maps, one is essentially referring to processing lines. Fortunately, many other disciplines such as Image Processing, Computer Graphics, and Pattern Recognition are also concerned with line processing. They might be interested in recognizing shapes of various objects, industrial parts recognition, feature extraction or electrocardiogram analysis etc.
Whatever may be the objective of line processing and whichever field it may be, there is one thing in common viz: it is necessary to retain the basic character of the line under consideration. As mentioned above one needs to detect and retain the critical points in order to retain the character of a line.

There is a lot of research being carried out in all the above disciplines as to the detection of critical points. Because the problem of critical points detection is common to so many disciplines, it has many nomenclatures. A number of these nomenclatures (Wall and Danielson, 1984), (Dunham, 1986), (Imai and Iri,1986), (Anderson and Bezdek, 1984), (Herkommer, 1985), (Freeman and Davis, 1977), (Rosenfeld and Johnston, 1973), (Rosenfeld and Thurston, 1971), (Duda and Hart, 1973), (Opheim, 1982), (Williams, 1980), (Roberge, 1984), (Pavlidis and Horowitz, 1974) (Fischler and Bolles, 1983,1986), (Dettori and Falcidieno,1982), (Reumann and Witkam, 1974), (Sklansky and Gonzlaz, 1980), (Sharma and Shanker,1978), (Williams,1978) are listed below:

1. Planer curve segmentation
2. Polygonal Approximation
3. Vertex Detection
4. Piecewise linear approximation
5. Corner finding
6. Angle detection
7. Line description
8. Curve partitioning
9. Data compaction
10. Straight line approximation
11. Selection of main points
12. Detection of dominant points
13. Determination of main points

Both the amount of literature available for the solution of this problem and its varying nomenclature indicate the intensity of the research being carried out to solve this problem. It is recognized by various researchers (e.g. Fischler and Bolles, 1986) that the problem of critical points detection is in fact a very difficult one and it still is an open problem.

Similarly, the problem of line generalization is not very difficult if carried out manually but becomes difficult if one wants to do it by computer. It has been noted in section 1.3 that because of the subjective nature of this problem and because of lack of any criteria for evaluation of line generalization, it has been very difficult to automate this process. Recently some researchers for example (Marino, 1979) and (White, 1985) have suggested that one should first find critical points and retain them in the process of line generalization.
In this section a brief review of existing literature for the detection of critical points in a digitized curve is given. In addition, existing literature in line generalization and smoothing is also examined. Finally, present methods of evaluating line generalization are given.

2.2 ALGORITHMS FOR FINDING CRITICAL POINTS

As noted in the previous section, there are many papers published on critical points detection which is identified by different names by different people. It should, however, be noted that the detection is not generic but, as indicated by Fischler and Bolles (1986) depends on the following factors:

1. purpose
2. vocabulary
3. data representation
4. past experience of the 'partitioning instrument' In cartography it would mean the past experience of the cartographer.

It is interesting to note that the above four factors are similar to the controls of line generalization that Robinson et al. (1985) have pointed out (see section 1.2). However, the fourth factor viz: past experience and mental stability of the Cartographer is missing from the latter list.
2.2.1 Classification of the Critical Point Detection Algorithms

One can classify these algorithms using different criteria. Pavlidis (1982) classifies these algorithms into 'scan along' and 'hop along'. A scan along algorithm processes the points one at a time and increases the length of the line segment as long as some error criteria is satisfied. On the other hand, a hop along algorithm deals with the subarcs of the curve, and splits them up if some error criteria is not satisfied. The hop along algorithm is also called split-and-merge algorithm. In the merge part of the algorithm almost collinear segments are merged because line segments longer than the step size cannot occur.

Another broad way of classifying these algorithms is the following:
1. Direct
2. Iterative

Direct methods will be able to detect the critical points in one pass through the data whereas iterative methods need more than one pass through the data.

The most fundamental classification of the algorithm, however, would be to classify them as follows:
1. local algorithms
2. global algorithms
Local algorithms only look at a few points at a time to detect critical points whereas global algorithms take into consideration all the points.

2.2.2 Local Algorithms

Rosenfeld and Johnston (1973) proposed a technique for finding significant curvature maxima (i.e. critical points) on a digital curve. The procedure they claim is similar to an edge detection technique reported in Rosenfeld and Thurston (1971) which detects significant maxima in the rate of change of average gray level.

The procedure may be described as follows:

Let \((x(i), y(i)), i=1,N\) be a set of digitized coordinates of a curve.

Then define

\[
\begin{align*}
\mathbf{a} &= (X, Y) \quad \text{where } X = x(i) - x(i+k) \\
\mathbf{b} &= (X', Y') \quad \text{where } Y = y(i) - y(i+k) \\
\end{align*}
\]

\[
\begin{align*}
X' &= x(i) - x(i-k) \\
Y' &= y(i) - y(i-k) \\
\end{align*}
\]

Then the cosine of the angle between the vectors 'a' and 'b' is given by:

\[
C_i = \frac{(a \cdot b)}{|a| \cdot |b|}
\]

where
-1 ≤ C ≤ 1

|a| and |b| are the magnitudes of 'a' and 'b' respectively.

The value of C will be close to 1 if the angle between 'a' and 'b' is close to zero, and it will be smaller if the angle is larger. In other words, C will be larger if the curve is turning rapidly and smaller if the curve is relatively straight. This fact is used to detect the curvature maxima.

At each point (x(i),y(i)) compute \( C_1(1), C_1(2), \ldots, C_m \) for some fixed m. Then assign size 'h' to point (x(i),y(i)) and the cosine value \( C_h(i) \) for the largest 'h' such that

\[ C_1(m) < C_1(m-1) < \ldots < C(h) < C(h-1) \]

After that retain only those points for which the computed value C > C(h)

The problem with this method is that it can very easily miss the critical points because it is taking \( 2^k \) points at a time. According to Rosenfeld and Johnston (1973) the maxima are often one or two points away from where they appear to belong.

Rosenfeld and Weska (1975) tried to improve on the above method by adopting the following averaging scheme on the computed angle C:

\[ Ca(k) = 2 \left\{ \frac{C(k)+C(k-1)+\ldots+C(k/2)}{k+2} \right\}; \text{ for even } k. \]
and

\[ Ca(k) = 2^* \left\{ \frac{C(k)+C(k-1)+...+C(k-1/2)}{k+3} \right\} \text{; for odd } k. \]

It was reported (Rosenfeld and Weska, 1975) that the improved method gives slightly better results.

Davis (1977) used a similar technique to that of Rosenfeld and Johnston (1971) but he modified it so that it can also take into account the sides of the figure as well.

Pavlidis and Horowitz (1974) use a split-and-merge algorithm for segmentation of a plane curve (i.e. lines connecting the critical points). In this method a fixed segment of the curve is selected and the error norm is computed in this segment. If this error is greater than the maximum allowed error norm, the segment is split and similar operation is repeated. Merge the adjacent segments provided that it will produce a new segment with error norm less than maximum allowed. The process is repeated until there will be no change in different segments.

Note that the split operation may be computationally expensive since it requires evaluation of integrals. The segments of the curve produced by this method are not necessarily continuous and might miss critical points.

Freeman and Davis (1977) proposed a corner finding algorithm for chain encoded (Chain codes will be explained in
Chapter 3 section 3.2.2) curves. This method of critical points detection is based on determination of incremental curvature. This algorithm scans the chain code with a moving line segment which connects the end points of the sequence of S links. The angular differences between the successive links are used as a measure of smoothed local curvature.

The algorithm as given in Freeman and Davis (1977) is outlined below:

Suppose that a digitized line has been chain encoded $C_i \{0,1,\ldots,7\}$ as explained in section 3.2.2. Define the straight line segment $D_i$ from the first point code $C_{i-s+1}$ to the last point code $C_i$. The length of the segment is given by:

$$D_i = \sqrt{X_i^*2 + Y_i^*2}$$

where

$$X_i = \sum C_{ix}$$
$$Y_i = \sum C_{iy}$$

$C_{ix}$ and $C_{iy}$ are the X and Y components of the chain code and $C_{ix}, C_{iy} \in \{-1,0,1\}$.

The angle between X-axis and forward direction of line segment is given by:

$$\Theta = \arctan \left( \frac{Y_i}{X_i} \right); \quad \text{if } |X_i| > |Y_i|$$
\[ \Theta = \arccot \left( \frac{x_i}{y_i} \right); \text{ if } |x_i| < |y_i| \]

and \( 0 \leq \Theta \leq 360 \).

To provide some more smoothing, the incremental curvature Delta is defined as twice the mean over two successive angular differences.

That is,

\[ \delta_i = 2 \times \left( \frac{(\Theta_{i+1} - \Theta_i) + (\Theta_{i} - \Theta_{i-1})}{2} \right) \]

or \( \delta_i = \Theta_{i+1} - \Theta_{i-1} \)

Note that the smoothing has taken place in the length \( s \) of the line segment. The smoothing will be heavier for larger values of \( S \).

Freeman (1978) used the above algorithm to detect critical points for the purpose of shape description. He reports that the value of \( \delta \) will normally range from 5 to 113.

The problem with this method is that the critical points may be smoothed over. Further this algorithm does not take into consideration the fact that there are discontinuities in the chain code as explained in chapter 3. These discontinuities may introduce false critical points.

Reumann and Witkam (1974) published a method of curve segmentation in which a strip or tube is shifted over the curve along the direction of its initial tangent until the
tube end intersects with the curve. The point of intersection (see figure 2) is retained as a critical point. The strip is now laid along the initial tangent to the remaining part of the curve starting from the last critical point. The process is repeated until the end of the curve is reached.

Figure 2: Reumann and Witkam's Tube along the Curve. Derived From Reumann and Witkam (1974).

Opheim (1982) reported that Reumann and Witkam algorithm has some shortcomings in that it will not handle sudden bends in curvature in a satisfactory manner.

He was critical about the Reumann and Witkam algorithm in two respects viz:
1. on the shape of the search region
2. on the direction of the search region.
Opheim (1982) believes that the search region of two parallel lines to infinity usually gives bad results. Therefore, he improved the search region by introducing two more parameters $D_{\text{min}}$ and $D_{\text{max}}$ such that the two consecutive critical points should not be further away from each other than a fixed distance $D_{\text{max}}$ and it should not be closer than a small distance $D_{\text{min}}$. Opheim (1982) further suggested that a natural choice of search direction could be the line joining the last two selected critical points.

The above two improvements will not alleviate the problem with the sudden change in the curvature. Therefore, Opheim (1982) asks to use Forsen's algorithm (i.e. Douglas Peucker) for such cases. Forsen's algorithm will be explained in section 2.2.2.

Roberge (1985) states that the Reumann and Witkam algorithm is unstable in the sense that critical line formation is unstable and it is also sensitive to rounding off errors. In order to overcome the problems in the Reumann and Witkam algorithm, Roberge (1985) proposed a new algorithm called Enhanced Strip Algorithm (ESA) which is based on the Reumann and Witkam Algorithm.

Instead of forming the critical line using two consecutive points $P_a$ and $P_{a+1}$, ESA selects the first point, for critical line formation, lying a distance $d$ from point $P_a$, call
Figure 3: Comparison between critical lines formed by ESA and Reumann-Witkam's algorithm and computation of $D_b$ and $r_b$ (derived from Roberge, 1985).

This point $P_f$ (see figure 3.). This implies that every point $P_c$ ($a < c < f$) still lies within distance $d$ of the new critical line. This approach of forming a critical line has the following two advantages to that of Reumann and Witkam (Roberge, 1985):

1. The critical line formation process is less sensitive to minor computational errors since points $P_a$ and $P_f$ are further apart.
2. The critical line tends to approach the general direction of the curve (after point \( P_a \)) rather than the direction of the curve in a restricted neighbourhood about point \( P_a \).

ESA requires the computation of the following quantities in order to detect the critical points (Roberge, 1985):

1. \( d_b \) the distance between critical line and the \( P_b \) (\( f \cdot b \)) see figure 3.
2. \( r_b \) the distance from point \( P_a \) to point \( P'_b \) (\( P'_u \) is the orthogonal projection of point \( P_b \) onto the critical line).

Critical points will be detected whenever the following conditions are satisfied (Roberge, 1985):

1. \( |(D_b)| > d \) is satisfied whenever \( P_b \) is further than distance \( d \) from the critical line
2. \( r_b < r_{b-1} \) is used to detect critical points

Roberge (1985) claims that ESA requires less time and less storage than other methods but does not necessarily give the minimum number of critical points.

Williams (1978) presented an algorithm derived from considerations similar to those of Reumann and Witkam. In this algorithm all the coordinates are first transformed to the polar coordinates \((r, \Theta)\).
Figure 4: The circles about each point $P_i$ define sectors $S_i$ which contain all lines through $P_i$ passing within distance $d$ of $P_i$. Sector $T_3$ contains all lines through $P_i$ which pass within distance $d$ of both $P_2$ and $P_3$.

Then as shown in figure 4., the set $S_i$ of all lines through point $P_1$ which pass within a distance $d$ of another point $P_i$ may be expressed in terms of the angular distance $\delta_i$ (Williams, 1978).

$$S_i = \{ \Theta = \phi \mid \Theta_i - \delta_i < \phi < \Theta_i + \delta_i \}$$

where

$$\Theta_i = \arctan \left( \frac{y_i - y_1}{x_i - x_1} \right)$$

$$\delta_i = \arcsin \left( \frac{d}{r_i} \right)$$

$$r_i = \sqrt{ (x_i - x_1)^2 + (y_i - y_1)^2 }$$
The set of all lines through the point $P_1$ which pass within $d$ of the first $i$ points is given by (Williams, 1978):

$$T_i = \{ \Theta = \phi \mid \Theta_{\text{min}} < \phi < \Theta_{\text{max}} \}$$

where

$$\Theta_{\text{min}} = \max (\Theta_j - \delta_j)$$
$$\Theta_{\text{max}} = \min (\Theta_j + \delta_j)  \quad 2 < j < I$$

$\Theta_{\text{min}}$ and $\Theta_{\text{max}}$ may be found by the following technique (Williams, 1978):

1. Initially
   $$\tan(\Theta_{\text{min}}/2) = \tan((\Theta_2 - \delta_2)/2)$$
   and
   $$\tan(\Theta_{\text{max}}/2) = \tan((\Theta_2 + \delta_2)/2)$$

2. For $i = 3, 4, \ldots \ldots$, and under the condition that
   $$\tan(\Theta_{\text{min}}/2) < \tan(\Theta_{i}/2) < \tan(\Theta_{\text{max}}/2)$$
   $$\tan(\Theta_{\text{min}}/2) = \max\{\tan(\Theta_{\text{min}}/2), \tan(\Theta_{i} - \delta_i/2)\}$$
   and
   $$\tan(\Theta_{\text{max}}/2) = \min\{\tan(\Theta_{\text{max}}/2), \tan(\Theta_{i} + \delta_i/2)\}$$

If the condition in (2) is violated, the line segment $P_i$ to $P_{i-1}$ is of maximal length i.e. $P_{i-1}$ is a critical point and the procedure is repeated from point $P_{i-1}$. 
The algorithm is numerically stable and does not suffer greatly from direction dependencies. But it does not give optimal results either in accuracy or in storage (Williams, 1978).

Williams (1981) devised another algorithm for straight line approximation of digitized curves. This algorithm uses a bounded precision approximation method to produce optimal results. The approximating lines are formed from tangents to the constrained circles drawn about each point. The tangents and circle pass within a specified distance of the curve points. The algorithm produces lines of maximum length that do not pass through the original curve points. Hence it is not a critical points detection technique.

Sklansky and Gonzalez (1980) present a polygonal approximation algorithm. The procedure is implemented as scan-along process which has the advantage of requiring small memory regardless of the number of digitized points. Moreover, the process is very fast. But unfortunately this method does not quite select the critical points.

In figure 5. A, B, C are the selected points by Sklansky and Gonzalez method. Obviously, point B cannot be regarded as critical point in the sense that it lies in the straight line 7C.
Dettori and Falcidieno (1982) describe an algorithm for critical point detection. Their algorithm is based on the fact that for a given subset of consecutive digitized points, if there is a band of width less than epsilon wide containing all of the subset, then only the two extreme points are enough to represent it. The existence of the band is determined by constructing a convex hull. The convex hull is defined by the smallest subset of vertices which define a polygon containing all vertices.

Dettori and Falcidieno's algorithm to select the critical points may be outlined (Dettori and Falcidieno, 1982) as follows:

1. Begin with initial two points, that is, the subset consisting of the essential points.
2. If it exists, join a point to the subset under consideration.

3. Compute the convex hull of this subset.

4. Check the dimensions of the convex hull so constructed by varying the distance of each side from the vertices which are not its extremes.

5. Should at least one side with an acceptable distance exist, we repeat step 2 otherwise, the subset previously recognized is the maximum one can delete, and thus the penultimate point is the second representative point; repeat from step 1, with the last point considered as the initial point.

One advantage of this method is that it can get rid of spikes. It is also demonstrated by Dettori and Falcidieno (1982) that this algorithm is faster than Forsen's (Douglas Peucker) algorithm. Note that the latter algorithm is explained in section 2.2.3.

Tejwani and Jones (1984) describe an algorithm called Adaptive Line of Sight method for detecting the critical points in digitized curves. This method is based on a set of coordinate axes that are dependent on the shape of the curve under consideration.

In order to describe this algorithm one needs to define (Tejwani and Jones, 1984) the following terms:

1. A curve is said to be in Line of Sight of a Point P (LSP) if every point in the curve C can be connected to P without intersecting the curve at any other point.
2. Let \( n \) be the normal projection of a curve \( C \) onto a straight line \( L \). The curve \( C \) is then said to be in Line of Sight of Axis \( L \) (LSA) if all points from \( C \) can be mapped injectively onto \( n \).

The Adaptive Line of Sight Method is based on two passes. In the first pass, the curve is divided into appropriate number of critical points such that all the boundary points lie on the same side of the straight line \( L \) joining any two adjacent critical points. Moreover, the points should be in line of sight of \( L \).

In the second pass, the more critical points from the selected critical points are determined using the derivative of the normal distance of the point from \( L \) with respect to the distance along \( L \).

This method is suitable for shape description because it is based on coordinate axes dependent on the shape of the curve under consideration.

The algorithm presented by Anderson and Bezdek (1984) to detect critical points in a digital curve is based on the determination of curvature and tangential deflection of discrete arcs computed from the scattered matrix pairs. Scattered matrix is explained in section 3.3.1.
The procedure is rather long. Refer to Anderson and Bezdek (1984) for details of this technique. This technique does not give satisfactory results in that in some instances it selects wrong critical points as is clear from the results of Anderson and Bezdek (1984).

Herkommer (1985) describes an algorithm to determine the critical points in a digitized curve based on correlation coefficient. The correlation of the selected data points to the parent data set is controlled by RMIN which is the minimum acceptable correlation between the given data and the line segment. The method may be explained in the following steps (Herkommer, 1985):

1. Input (x,y) coordinates until at least three have been read.
2. Determine the equation of the line that passes through the first and last point being considered.
3. Calculate the correlation coefficient between the current line segment and current data points.
4. If the computed correlation coefficient (R) is greater than RMIN, goto step 1 and add another point to the line segment; else proceed to step 5.
5. Save the penultimate point as the break point because it was the last to pass the test.
6. Make the newly determined breakpoint the initial point in the new segment.
7. Go to step 1 and add another point to the segment.

Like the majority of algorithms reviewed in this section, this technique is applicable only to the vector data. It
also tends to retain more points at the densely digitized sections of a curve. But this is not necessarily a disad-
vantage.

An algorithm based on split-and-merge with a single parameter was suggested by Eklundh and Howako (1984). The algorithm involves the following three steps:

1. Collinearity tests
2. Merge tests
3. Adoption of the step size.

1. Collinearity Tests

First compute the arc length over chord distance $S/L$. Then for some tolerances $t_1$ and $t_2$

1. if $S/L < t_1$, the line is accepted
2. if $S/L > t_2$, the line is rejected
3. if $t_1 < S/L < t_2$, then one has to perform a series of tests of the form $e < t_3$, where $e$ is the maximum absolute error ($L$-norm) and $t_3$ depends on the number of sign changes.

If the curve does not pass this test further test of the form $e < L^4 t_4$ is required. If the curve fails even this test, it is rejected.

2. Merge Tests
Efficient way of performing the merge test is to compute the angle between the approximating lines or perform a collinearity test on the three points.

3. Adaption to the Step Size

The step size is modified in such a way that it will provide the best approximation to the curve.

Marimont (1984) suggests that the output from an edge detector should be smoothed before detecting critical points in it. He uses the Gaussian mask for smoothing the input data from an edge detector.

Marimont (1984) describes his method of critical points detection as follows:

"The method used to find critical points oversamples the smoothed curve at a rate that depends on the range of the intersample distances and computes position, tangent direction, and curvature at each oversampled location. The pattern of sampled curvatures indicate when a critical point lies between samples, and an iterative method is used to find its location as accurately as necessary".

The above process provides a list of points of the curve which may include points which are not critical and as such are not required to represent the curve. Therefore, dynamic programming is used to find the list of critical points which best approximate the curve.

Marimont's method gives good results. However, he points out the fact that critical points on the smoothed curve do
not necessarily lie at points corresponding to the original points.

Fischler and Bolles (1986, 1983) presented an algorithm for critical points detection in a digitized curve. The algorithm is related to but distinct from the mathematical concept of curvature.

The technique labels each point on the curve as belonging to one of the following three groups (Fischler and Bolles, 1986):

1. A point in the smoothed segment
2. A critical point
3. A point in a noisy segment.

The above classification is achieved by analyzing the deviations of the curve from a stick which is iteratively advanced along the curve. If the curve makes a single excursion away from the stick it will be labelled as critical point. If the curve stays close to the stick, points in this interval will be labelled as being in smooth interval. Moreover, if the curve makes two or more excursions from the stick, points in this interval are labelled as being in noisy interval.

The above process is repeated for sticks of different sizes which is equivalent to analyzing the curve at different resolutions.
The process is halted when the points selected are too close to those previously selected.

Fischler and Bolles (1986, 1983) compared the critical points selected by their algorithm with those selected by humans. They found that their algorithm was able to pick up the same points as those selected by humans. It should be pointed out that the algorithm cannot detect points of inflection and points in the smooth section of the curve.

2.2.3 Global Algorithms

Perhaps the most popular method of finding critical points in vector data is the one presented by Duda and Hart (1973). Ramer (1972) and Douglas and Peucker (1973) published the same algorithm as the one by Duda and Hart.

According to Duda and Hart (1973) the algorithm was suggested by G. E. Forsen. All these publications were made at about the same time. But usually it takes longer to publish a book than an article. Therefore, it can be assumed that Forsen was the first who thought about the technique. Accordingly, this technique will be called Forsen's algorithm in this dissertation.

Duda and Hart (1973) explain the Forsen's algorithm as follows:
"Given our usual set of n points, we fit an initial line, call it AB, by merely connecting the end points. The distances from each point to this line are computed, and if all the distances are less than some preset threshold the process is finished. If not, we find the point furthest from AB, call it C, and break the initial line into two new lines AC and CB. This process is then repeated separately on the two new lines, possibly with different thresholds".

Figure 6: Illustration of the Forsen's Algorithm.

Figure 6 illustrates the Forsen's algorithm. In this figure the line AB is broken into AC and CB because C is the furthest point from the line AB. All the points between AC satisfy the threshold criteria but CB is broken into CD and DB. Therefore, the final result is the sequence of connected lines AC, CD, and DB.

Duda and Hart (1973) point out that the algorithm has the following two drawbacks:
1. A single outlier may drastically change the final results

2. A selected critical point may not be the real critical point.

The first problem may be alleviated by preprocessing the data so that it is noise-free and the second problem may be alleviated by post processing where the selected points are shifted around.

Fischler and Bolles (1986, 1983) developed an algorithm to locate the critical points along a curve in such a way that the segments between the critical points would be modelled by either straight line or circular arc.

Basically Fischler and Bolles (1986) approach was to analyze the several views of the curve and construct a list of critical points from each view and select the optimum number of critical points between which the straight lines or the circular arcs may be fitted.

Liao (1981) suggested a two stage approach of finding critical points in a digitized curve. In the first stage a modification of the Forsen’s algorithm is used to locate approximate critical points. In the second stage the algorithm scans along the data points and fits them with conic arcs and straight lines.
The advantages of the Fischler and Bolles (1986) and Liao (1981) algorithms presented in this section are that these algorithms are able to pick up critical points even in the smooth transition between straight line and circular part.

Shanker and Sharma (1978) described an algorithm which detects critical points based on maximum global curvature. The global curvature at each point is computed iteratively as a function of local curvature of its neighbourhood.

Since this method is strongly influenced by the curvature of the neighbourhood, it will fail to detect the genuine critical points.

2.3 ALGORITHMS FOR LINE GENERALIZATION.

As pointed out earlier, the process of manual line generalization is highly subjective and depends not only on the purpose, scale, quality of data, and graphic limits but also on the skill, experience and state of mind of the person who performs generalization. Clearly, if one wants to do such a task by computer it must be made as objective as one possibly can. Perhaps the first attempt to make the process of line generalization more objective was made by Topfer and Pillewiser (1966). They called their method as "Principle of selection". According to this principle of selection, the number of features to be selected in a gen-
eralized line is given by the radical law. In this case features means the number of meanders in a river, the number of islands in a sea, or a number of wooded lots in an area etc. The radical law as given below expresses the number of features \( n_f \) to be retained as a function of number of features \( n_a \) in original scale, scale denominator \( M_a \) of the source map, and the scale denominator \( M_f \) of the generated map:

\[
n_f = n_a * \sqrt{M_a / M_f}
\]

It should, however, be noted that this formula does not provide an algorithm because it tells only how many features to retain but not which ones to retain. This formula was only meant to give some indications of the number of features to be retained to the practicing cartographers.

Srňka (1970) published a mathematical equation which may be used for the selection of the linear features. He calls it the law of selection of linear elements. Its general form is given by the following equation:

\[
n(p_1)\% = e_1 n(p)^{\text{inh}}(p_0)^{\text{gi}}
\]

where

\[n(p) = \text{the number of elements within the reference (unit) P of the base map.}\]

\[n(p_1)\% = \text{the percentage of the original number of elements}\]
represented within the area $p_i$ in the $i^{th}$ derived map.

$\textbf{h}(p_0) =$ length of the linear elements within the reference area $p_0$ of the base map.

$e_i =$ total level of selection

$f_i =$ variable degree of selection at different numbers of linear elements of the base map.

$g_i =$ variable degree of selection as a function of the length of the linear segment.

Like the radical law this method does not say which linear features to retain.

Sukhov (1970) attempted to use information theory to map generalization. He expressed the information loss between the source map and generalized map as relative entropy. Sukhov (1970) claims that generalization results in loss of information if carried out properly. This information loss between the base map and generalized map may be expressed as the difference of entropy of the two maps.

Perhaps the first automatic algorithm for line generalization was introduced by Tobler (1966). This procedure called $N^{th}$ point algorithm sequentially removes every $N^{th}$ point from the data. This procedure is completely arbitrary (Rhind, 1973, Robinson et al., 1985).
This is a crude method or it is not a method at all since it might miss all the critical points. This method should never be used especially when there are so many other methods which are simple and cheap as well (e.g. Reumann-Witkam, 1974).

Zoraster et al. (1984) suggested a more complex approach in which they make use of probability theory in point selection. In this method they suggest eliminating points in a pseudo-random manner. This improvement reduces the dependency on the starting point and guarantees that an unfortunate choice of parameters does not result in points being removed which coincide with some fundamental frequencies in the data.

Solovitskiy (1974) suggested an algorithm for automatic generalization of lines. According to his method the angle at every point is checked and if it is less than certain threshold angle the point (corner) is retained and it is not smoothed. The rest of the points which do not satisfy the criteria are smoothed using a smoothing matrix $Z$. According to Solovitskiy "The smoothing is done under the condition of minimizing the sum of squares of the deviations of the coordinate values of the original points from the regression line".
Jenks as cited in McMaster (1983) and Clayton (1985) suggested an algorithm which is based on the following three parameters:

1. MIN1 = minimum allowable distance between point one and two
2. MIN2 = minimum allowable distance between point one and three
3. ANG = maximum angle between the two lines.

Point two will be rejected if the distance from point one to point two is less than MIN1 or the distance between point one to point three is less than MIN2. If both distances satisfy the above criteria, the angular check is performed. If the angle between the above two lines is less than ANG, point two will be rejected. This method is bound to be very expensive in terms of computation.

Brophy (1973) presented an algorithm which enables the Cartographer to retain significant controls over generalization. His program called ALIGEN can perform feature elimination as well as point elimination. Brophy claims that he has incorporated both subjective and objective criteria of line generalization in his algorithm.

Brophy's program ALIGEN consists of six components that allow for a number of generalization options. Component one enables the removal of smaller features. Whether a particu-
lar feature should be eliminated or not, is decided by considering the scale change between the base map and generalized map, the line weight change between the maps, and degree of generalization desired. Component two redefines the new curve created in component one by converting them "as a series of tangent points of finite width equal to the line weight of the line on the generalized map" (Brophy, 1973). Component three is optional and enables the Cartographer to constrain certain points so that the curve will pass through them. Component four is also optional and enables one to smooth and exaggerate certain features. Finally, component six plots the generalized line. Brophy's algorithm has been well received by Cartographers (e.g. Morrison, 1975). Note that there is a basic difference between Brophy's approach and the approach taken in this research in that the former is an interactive algorithm whereas this research concentrates on developing an automatic algorithm.

Vanicek and Woolnough (1975) developed an algorithm based on the transformation of parameters of a digitized curve into a set of pseudo-hyperbolae such that no point lies outside a tube of a given error tolerance epsilon. It is taken to be equal to plotter resolution.
In this algorithm, first the coefficients of pseudo-hyperbolae

\[ y = \pm \frac{C_1 + C_2}{x + C_3} \]

are determined by taking the average direction of the first three points. After that more successive points are selected until they fail to be within a tube of \( \pm 8 \varepsilon \). "Using the beginning and end points for proper direction a rigorous check is made of points selected so that they lie in a tube of \( \pm \varepsilon \) wide. The number of points selected is altered until this condition is met, at which time a segment length is computed" Vanicek and Woolnough (1975).

More line segments are determined by defining pseudo-hyperbolae with the vertex coinciding with the end of the last segment and with the axis oriented in the direction of the last segment. The whole process is repeated until the end of data is reached.

One limitation of this method is that it cannot be used for closed curves unless they are broken into two or more parts.
2.4 ALGORITHMS FOR LINE SMOOTHING

Line smoothing may be performed in two ways. In one group of methods such as using moving averages, the smoothed line does not necessarily pass through the original digitized points irrespective of whether they are critical points or not. In the other group of methods such as fitting B-splines the fitted curve passes through the given critical points. Accordingly, the smoothing algorithms are divided into two groups viz:
1. Smoothing methods and
2. Interpolating methods.

2.4.1 Smoothing Methods

The smoothing algorithms which do not preserve original points, transform the coordinate pairs into new coordinate pairs using some mathematical functions thereby changing the point locations.

According to White (1985) "Smoothing techniques appear to be most appropriate for data sets already reduced in size, in order to provide a more natural looking line".

Koeman and Van der Weiden (1970) discussed an algorithm for linear smoothing. Essentially they use a moving average technique which is applied to X and Y coordinates separately. Moving average assigns the mean value of a series of
coordinates to the center value. They claim that their method gives good results if the ratio between the distances between points along the smoothed or curved section, and the distance between points on very irregular lines is not too large. This means that the density of digitization should be uniform for the proper application of this method.

Tobler (1966) improved on Koeman and Van der Weiden method by suggesting the use of weighted moving average. However, it should be noted that the sum of the weights used must be equal to unity if the preservation of the scale is desired. One should be careful when selecting weights for this case, since if the larger weights are assigned to the extreme points the character of the line will be changed. On the other hand, the character of the line will be preserved if the larger weights are assigned to the center point.

Note that the above methods of line smoothing do not reduce the number of data points which is clearly a great disadvantage.

Boyle (1970) developed an algorithm for line smoothing and called it the forward look algorithm. Boyle explains his algorithm as "each time one-quarter of the distance of a four point look ahead or one tenth of the ten point look ahead is completed, a new aiming point is created on logic-
al basis". According to Boyle (1970) this method works well for all types of lines and the results are visually pleasing.

Perkel as referenced in Clayton (1985) introduced the most comprehensive and objective method of line smoothing. This method of smoothing involves rolling a circle of radius epsilon along both sides of a curve. The path of the circle defines the smoothed curve. This path of the curve, obviously depends upon which side of the curve the circle is rolled. For a very complex curve a residual zone is left between the two paths thereby introducing an ambiguity on the generalized line. This ambiguity is the most fundamental problem with this method. Moreover, the algorithm is difficult to program.

2.4.2 Interpolating Methods

Akima (1970) published a new mathematical method of line smoothing. The method was designed to pass through the given points and look smooth and natural. The method is based on using piece-wise polynomials of at most degree three. In this method the slopes at each point are determined and the polynomials for each section of the curve are fitted using the coordinates and slopes of the points. It was found by Akima (1970) that the curve drawn by this method is closer to the manually drawn curve than those drawn by other mathematical methods.
The other methods of smoothing curves are fitting splines and Bezier curves through these points (Rogers and Adams, 1976), (Buttenfield, 1985), (Zoraster et al. 1984), (Dierckx, 1982), (Spath, 1974), (Riesenfeld, 1973), (Yamaguchi, 1978), (Yang et al. 1986), (Greyville, 1967), (Cox, 1972), (De Boor, 1972), (Reinsch, 1967), and (Lozover and Preiss, 1981). Splines fit piece-wise polynomials at each subinterval of the curve in such a way that if one fits a polynomial of degree \( M - 1 \) the fitted curve will have \( C^{M-2} \) continuity, that is, the derivatives of order \( M-2 \) are everywhere continuous. Splines have been used in CAD/CAM by many manufacturing industries such as car and aircraft manufacturers, Riesenfeld (1973).

The question of selection of the basis for the representation of splines of fixed order and fixed partition is of fundamental importance. Only the following three kinds of bases for spline spaces have actually been given serious consideration (De Boor, 1972):

1. Truncated power bases
2. Cardinal splines

The truncated power bases are known to suffer from ill-conditioning and cardinal splines are difficult to compute. But B-spline bases are well conditioned for orders less
than or equal to twenty. In addition, B-spline bases are local in the sense that at every point only a fixed number equal to the order of the spline is non-zero (De Boor, 1972). Moreover, B-splines are evaluated easily.

The methods of line smoothing using Fourier Transform have been described by various researchers (Bennet and Mcdonald, 1975), (Oommen and Kashyap, 1983), and (Moellering and Rayner, 1982). However, the methods based on Fourier Transform are not, in general, applicable to Cartography because they apply only to closed figures since the problem formulation requires that the figure under consideration be closed. Nevertheless, one may get round to this problem by augmenting the figure with a closure contour.

2.5 EXISTING METHODS FOR EVALUATION OF LINE GENERALIZATION

As mentioned in section 1.2, line generalization is largely a subjective process. There do not exist any hard and fast rules for performing line generalization. Consequently it is difficult to evaluate such a process. Nevertheless, some researchers (McMaster, 1983), (Marino, 1978), (White, 1985) have attempted to evaluate line generalization or has worked in that direction. Similar problems arise in other
disciplines such as Pattern Recognition, Image Processing, and Computer Vision with reference to curve partitioning and interpretation. Fischler and Bolles (1986) admit that evaluation of such a process is a difficult task. The basic questions are how does one know that a certain algorithm is successful? How does one can compare the results of the two different algorithms?

Fischler and Bolles (1986) suggest that any evaluations must be based on the following considerations:

1. Is there a known correct answer?
2. Is the problem formulated in such a way that there is "provably a correct answer"?
3. How good is the agreement with (generic or "expert") subjective human judgement?
4. What is the tradeoff between "false alarms" and "misses" in the detection of critical points?

Criteria 1 and 2 are difficult if not impossible to fulfil since there are no correct answers and the problem of line generalization cannot be formulated in such a way that there is a provably correct answer.

McMaster (1983) came up with 30 measures for evaluation of line generalization. Out of 30 measures he found that only the following six were statistically independent:

1. Ratio of the change in the number of coordinates
2. Ratio of the change in the standard deviation of the number of coordinates per inch.
3. Ratio of the change in angularity
4. Ratio of the change in the number of curvilinear segments
5. Total length of vector differences per inch.
6. Total areal difference per inch.

It is interesting to note that Pavlidis (1980), and Perkins (1978) proposed similar measures to compare the similarity of two curves. The following are the criteria that Pavlidis (1980) used:

1. Type of curve (arc, circle, straight line etc.)
2. Total length or radius of arc
3. Total angular change
4. Number of lines or arcs.

The other six criteria that he suggested are not applicable to the problem at hand and therefore are not listed here.

Zoraster et al. (1984) have suggested two types of criteria for evaluations of algorithms used for line generalization namely: 1. Cartographic and 2. Algorithmic.

The following are the cartographic criteria (Zoraster et al. 1984):
1. Preservation of map character and accuracy
2. A focus on global features
3. The ability to vary the amount of generalization as a function of feature type.

Algorithmic criteria as suggested by Zoraster et al. (1984) include the following:
1. Predictable reduction in data.
2. Invariant with respect to mathematical operations.
3. Predictably controlled by simple parameters.
4. Modular to meet different map specifications.
5. Computationally fast.

The above criteria seem to be reasonable to evaluate the performance of an algorithm but do not help much in evaluating line generalization. In chapter six criterion 3 of Fischler and Bolles (1986) will be used to evaluate the lines generalized algorithmically and a few of the criteria used by Pavlidis (1980) and McMaster (1983) will also be used.

2.6 SUMMARY

The problem of detecting critical points in a digitized curve is a very difficult one. The problem is not generic but depends on the purpose, scale, and quality of data. Following the revelations from the field of Psychology that the information content of a curve is concentrated at points of high curvature, most of the algorithms for find-
ing critical points tend to locate the points of high curvature.

The amount of literature available in the field indicates the intensity of research going on to solve this problem. The majority of researchers in this area agree that the problem is still an open one and currently no computer algorithm exists which can solve the problem of critical points detection satisfactorily.

The attempts to automatically generalize lines by computer have not been successful. The primary reason for this seems to be due to the lack of any guidelines or rules for line generalization. The recent trend is to make line generalization more objective. One way to make it objective is to first find the critical points in the line to be generalized and then compare these with those critical points selected by humans in the same curve. This has already been done and it was found that one can devise algorithms which can mimic humans in terms of critical points detection. Once the critical points are detected, line generalization may be carried out by either simply joining the critical points or fitting some kind of curves (e.g. splines) depending on the purpose of the map.

There are two types of line smoothing techniques viz: interpolating and smoothing. Both types have been tried in
Cartography albeit the latter method seems to be easier and more common.

Only recently attempts to evaluate line generalization have been made. But there do not exist any acceptable criteria for such a purpose.
3.1 INTRODUCTION

In this chapter two methods of finding critical points are discussed. The first method is based on finding zero-crossings of the convolution of the signal (derived from the digitized data) with the second derivative of the Gaussian. A post processing of the zero-crossings using a dynamic mask is performed to eliminate non-critical points.

The second method of critical points detection is based on first computing the normalized symmetric scattered matrix from the digitized data. The eigenvalues of this matrix are used to detect the linear segments between the critical points.

3.2 CRITICAL POINTS DETECTION USING ZERO-CROSSINGS

Researchers in the field of Pattern Recognition, Image Processing, and Computer Vision are interested in detecting edges in images. The fact that there is a change in gray-
level at the edges of images was long recognized by Rosenfeld and Thurston (1971), and Marr and Hildreth (1980). Consequently, this idea was exploited to detect the edges in images by various researchers.

Marr and Hildreth (1980) suggested that the zero-crossings of the convolution of the gray-level with the Laplacian of the Gaussian across a number of channels are rich in information. This statement is supported by the theoretical work of Logan (1977). In this case zero-crossings across a number of channels means the zero-crossings of the convoluted values of the Laplacian of the Gaussian with the gray level signal for different values of the scale (sigma) of the Gaussian. In addition, it should be noted that the zero-crossings of the second derivative (or the Laplacian) of the Gaussian are the same as the extremas of the first derivative. The collection of zero-crossings across a number of channels is known as the finger-prints. The plot of these zero-crossings against the different channels is known as the scale space image (Mokhtarian and Mackworth, 1986).

Witkin (1983) demonstrated that as the scale parameter of the Gaussian decreases, additional zero-crossings may appear but the existing ones, in general, will not disappear. Witkin claimed that the Gaussian is the only function
which satisfies this property. In addition, the Gaussian is symmetric, and strictly decreasing about the mean and therefore weighting assigned to the signal values decreases smoothly with distance. Furthermore, the Gaussian is infinitely differentiable.

As stated earlier, different number of zero-crossings are obtained for different channels of convolutions i.e. different values of sigma. In this research, channels and sigma of the Gaussian are interchangeably used. Further the Gaussian function is given in section 3.2.3. The problem is how does one combine the zero-crossings from different channels to obtain the optimum information? How many channels are required to accurately detect the edges in images? Marr and Hildreth (1980) speculated that zero-crossings that spatially coincide over several channels are "physically significant". Marr and Hildreth (1980) further suggested that a minimum of two channels, reasonably separated in the frequency domain are required in order to warrant the existence of an edge. The zero-crossings which are common to both channels indicate the presence of an edge. The edges thus detected are called the primal sketch.

Rosenfeld and Thurston (1973) recognized the fact that the technique that can be used to detect edges may also be used to detect the curvature in a digitized curve. Asada and
Brady (1986) proposed the scale space approach to representing the significant changes in curvature. Their approach to represent curvature changes in a curve is termed as curvature primal sketch because of its correspondence with the primal sketch for representing the gray-level changes.

Asada and Brady (1986) introduced two isolated curvature changes viz: the corner and the smooth join. For a corner the tangent to the curve (and hence the curvature) is discontinuous. For a smooth join the tangent is continuous but the curvature is discontinuous.

The above two isolated curvature changes are not sufficient to analyze a digital curve. Therefore, Asada and Brady (1986) introduced the following compound primitives:

1. Crank: a compound change in curvature comprising of two nearby corners of opposite sign.
2. End: a compound change in curvature comprising of two nearby corners of the same sign.
3. Bump or dent: a complex compound change that consists of two nearby crank changes.

Asada and Brady (1986) derived different expressions for the above different types of primitives for performing Gaussian convolution. From Asada and Brady (1986) it is not clear what signal they use in the function. In addi-
tion, it is also unclear what signal they use for convolution. Nevertheless, their results in detecting critical points in the digitized boundaries of knives, hammers, screw drives etc. are impressive.

Mokhtarian and Mackworth (1986) used the scale space image and the Gaussian convolution for the description and recognition of planer curves. They compute the zeroes of curvature by convolving the path length with the first and second derivatives of the Gaussian. They used the coastline of Africa as an example. In this example (refer to Mokhtarian and Mackworth, 1986) it is clearly seen that the method they use does not necessarily lead to the critical points.

In this dissertation, I approach the problem differently. The basic objective is to detect the critical points along a digitized curve whether closed or open.

Unlike Asada and Brady (1986) and Mokhtarian and Mackworth (1986), we convolve the signal only with the second derivative of the Gaussian. I do not use any curvature primitives. The signal (Schenk, 1986) is derived from the Freeman chain code instead of the path length. In the next few sections, I explain exactly how I perform the convolution, what signal I use, and how I carry out the post processing.
3.2.1 Vector to Raster Conversion

The technique of critical points detection using the zero-crossings of the Gaussian convolution with the signal requires that the data be in raster form. If the data is already in raster form, this step will not be required.

According to Rogers (1985) the following are the desirable characteristics of the vector to raster conversion technique:

1. Straight lines should appear as straight lines
2. The algorithm should start and end accurately
3. The algorithm should work fast
4. There should be no gaps or overlaps.

It is difficult to satisfy all the above criteria. However, acceptable approximations can be achieved by selecting sufficiently small pixel sizes. Note that if one is using a display device the pixel size is fixed and one may not be able to reduce it below a certain size.

Basically, vector to raster conversion algorithms use the fact that if one increments the value of \( x \), the corresponding value of \( y \) may be evaluated by using the equation

\[ y = mx + c \]

where \( m \) is the slope of the line and \( c \) is the intercept. However, this equation involves multiplication of \( x \) by \( m \).
This may be eliminated by incrementing $x$ by unity so that a unit change in $x$ corresponds to changes in $y$ by $m$. But if $m > 1$, then the roles of $x$ and $y$ must be reversed and a unit change in $y$ will require $x = 1/m$ change in $x$. This method is called the Incremental Algorithm (Foley and Van Dam, 1982).

In order to make the arithmetic involved in vector to raster conversion as fast as possible, it is preferable to use integer arithmetic in algorithms which perform this task. The incremental algorithm discussed above will give both inaccurate ending points and will also cause overlapping of pixels if a series of lines are considered (Rogers, 1985).

Unlike the algorithm discussed above, the algorithm devised by Bresenham as referenced in (Foley and Van Dam, 1982) and (Rogers, 1985) works using only the integer arithmetic. Bresenham's algorithm selects the optimum raster locations to represent a straight line. To achieve this the algorithm always increments by one unit in either $x$ or $y$ depending on the slope of the line under consideration. The increment on the other variable is either zero or one depending on the distance (called error) between the actual line and the nearest grid intersection.

As shown in Figure 7, let us suppose that at the $i^{th}$ step point $P_i = (x_i, y_i)$ has been selected to be close to the line. Now the next step is to decide whether
it is pixel $S_j (r+1, q)$ or $T_i (r+1, q+1)$. Let $s$ and $t$ represent the distance between the line and the pixels $S$ and $T$ respectively. Bresenham's algorithm requires that one selects pixel $T_i$ if $t < s$ else $S$ will be selected. In other words, select $T_i$ if $t-s < 0$ otherwise select $S_i$ (Foley and Van Dam, 1982).

Suppose that one wants to rasterize the line between points $(x_1, y_1)$ and $(x_2, y_2)$. Again suppose that $(x_1, y_1)$ is closer to origin. The line is then translated to the origin so that it becomes the line from $(0, 0)$ to $(dx, dy)$ where

\[
\begin{align*}
  dx &= x_2 - x_1 \\
  dy &= y_2 - y_1.
\end{align*}
\]
The equation of the line between these points is given by:

\[ y = \frac{dy}{dx} \cdot x \]

From figure 7, one can write the following equations of the line depending on which pixel is selected (Foley and Van Dam, 1982):

\[ q = \frac{dy}{dx} (r + 1) + s \]

or \[ s = \frac{dy}{dx} (r + 1) - q \] \hspace{1cm} (3.1)

\[ q + 1 = \frac{dy}{dx} (r + 1) + t \]

or \[ t = q + 1 - \frac{dy}{dx} (r + 1) \] \hspace{1cm} (3.2)

Subtracting (3.2) from (3.1) one obtains

\[ s - t = 2 \cdot \frac{dy}{dx} (r + 1) - 2q - 1 \] \hspace{1cm} (3.3)

When \( s - t < 0 \) one chooses S otherwise T.

Multiply both sides of 3.3 by \( dx \) and simplify:

\[ dx \cdot (s - t) = 2 (r \cdot dy - q \cdot dx) + 2 \cdot dy - dx \]

Since \( dx \) is positive, one can use \( dx \cdot (s-t) \) as the test for choosing either pixel i.e.

If \( dx \cdot (s-t) > 0 \) one selects \( T_i \).

put \( d_i = dx(s-t) \); then

\[ d_i = 2(r \cdot dy - q \cdot dx) + 2 \cdot dy - dx \]

substitute \( r = x_{i-1} \); and \( q = y_{i-1} \) then;

\[ d_i = 2(x_{i-1} \cdot dx - y_{i-1} \cdot dx) + 2 \cdot dy - dx \] \hspace{1cm} (3.4)
Adding one to each index one gets:

\[ d_{i+1} = 2(x_i * dy - y_i * dx) + 2*dy - dx \]  \hspace{1cm} (3.5)

Subtracting (3.5) from (3.4) one gets:

\[ d_{i+1} - d_i = 2*dy(x_i - x_{i-1}) - 1)* \]
\[ 2*dx(y_i - y_{i-1}) \]  \hspace{1cm} (3.6)

But the increment in \( x \) is unity, therefore

\[ x_i - x_{i-1} = 1 \]

Rewriting (3.6) one gets

\[ d_{i+1} = d_i + 2*dy - 2*dx(y_i - y_{i-1}) \]

If \( d_i < 0 \), then \( S_i \) is selected. Hence

\[ y_i = y_{i-1} \] and
\[ d_{i+1} = d_i + 2*dy \]  \hspace{1cm} (3.7)

If \( d_i > 0 \) then \( T_i \) is selected. Hence

\[ y_i = y_{i-1} + 1 \] and
\[ d_{i+1} = d_i + 2*(dy-dx) \]  \hspace{1cm} (3.8)

Equations (3.7) and (3.8) provide one with an iterative way of computing \( d_{i+1} \) from \( d_i \) and to choose between \( S_i \) and \( T_i \).

The initial coordinates are \((0,0)\), therefore when \( i=1 \) from (3.4) one gets;

\[ d_1 = 2*dy - dx \]  \hspace{1cm} (3.9)
Note that the arithmetic required to evaluate (3.4), (3.7) and (3.8) uses integers only and thus is minimal and fast.

The following is the general implementation of the Bresenham's Algorithm (CIS 781, 1985):

```plaintext
xinc=1
dx=x2-x1
if (dx<0)
dx= (-dx)
xinc=(-1)
yinc=1
dy=y2-y1
if (dy<0)
dy= (-dy)
yinc= (-1)
x=x1
y=y1
plot (x,y)
if (dx>=dy)
acc=2*dy-dx
while (x.ne.xl)
x=x+xinc
if (acc<0)
y=y+yinc
acc=acc+2*(dy-dx)
else
acc=acc+2*dy
endif
plot (x,y)
endwhile
else
acc=2*dx-dy
while (y.ne.y2)
y=y+yinc
if( acc<0 )
x=x+xinc
acc=acc+2*(dx-dy)
else
acc=acc+2*dx
endif
plot (x,y)
endwhile
endif

A similar general algorithm is also given in Rogers (1985).
```
3.2.2 Chain Encoding

Chain coding is a data structure for storing digital curve data in computer memory. Chain codes were devised by Freeman (1974) and hence it is also called Freeman chain code. Usually the coordinates of the first point are given and the subsequent entries are numbers giving the direction and magnitude between each point in the outline and its neighbour. There are eight possible directions between a pixel and its neighbour. As shown in figure 8, these numbers are from 0 to 7 counter clockwise.

![Diagram of Freeman Chain-Code and Pixel Neighbourhood](image)

Figure 8: Simple Freeman Chain-Code and Pixel Neighbourhood. The dark square at the center shows the pixel one is interested in. The pixel has 8 neighbours numbered from 0 to 7.
Chain codes have the following advantages (Wilf, 1981):
1. It is easy to construct
2. It is compact
3. It is easy to understand
4. It is convenient for certain mathematical operations
5. Useful for certain types of statistical object recognition.

However, it has the following (Wilf, 1981) disadvantages:
1. Certain simple operations such as simple rotations and scale change are awkward to perform. For these operations we need to transform them back to coordinates.
2. The chain code has discontinuities even though the line it is supposed to represent is continuous.
3. The chain code assumes a unit distance between pixels. But if one lets the horizontal and vertical distance between pixels to be one unit, the diagonal distance will be 1.41 units. This means that for odd chain code the distance between pixels will be 1.41 units otherwise unity.
4. Chain codes are susceptible to noise.

The disadvantages 2 and 3 of the chain codes may be alleviated by introducing the following modifications:

The jumps in the chain code may be alleviated by introducing the so called Extended Freeman Chain code (EFC) (McKee and Aggarwal, 1977).
Let \((e(i), i=1,N)\) represent the EFC and \((f(i), i=1,N)\) be the Simple Freeman chain code (SFC). Then define

\[ e(1) = f(1) \]

Now for \(2 < i < N\), one chooses \(k_i\) such that

\[ ic = \text{abs}(e(i-1)-(f(i)+8*k_i) \quad \text{and} \quad ic < 4 \]

then the EFC is given by:

\[ e(i) = f(i) + 8*k_i \]

The non-uniform scaling of the chain code may be overcome by introducing the so called Compensated Extended Freeman Chain code (CEFC). One can form the CEFC by repeating the chain code three times if it is odd and repeating it twice if it is even.

It is not necessary to worry about the mathematical operations on the chain code since these are not required for this dissertation. McKee and Aggarwal (1977) have suggested that one can smooth the data by adding nine CEFCs consecutively and thus obtaining the so called Smoothed Compensated Extended Freeman Chain-code (SCEFC). In fact one can add any number of CEFCs to get SCEFC. Naturally, one can achieve greater smoothing by adding more number of CEFCs e.g. see section 4.7.
The jumps in the chain code may also be alleviated by using the following equation (Pavlidis, 1980). Eccles et al. (1977):

\[ s(i) = \frac{e(i) - e(i-1) + 11}{\text{mod } 8 - 3}. \]

Where \( e(i) \ i = 1,2,\ldots,N \) are the SFC chain codes and \( s(i) \ i = 1,2,\ldots,N \) are called curvature elements.

Eccles et al. (1977) found that \( s(i) \) closely approximates the curvature of a digitized line.

3.2.3 Second Derivative of The Gaussian

The Gaussian function is given by:

\[ f(t, \sigma) = \frac{1}{\sqrt{2 \pi} \sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right). \]

The first derivative of the Gaussian is given by:

\[ f'(t, \sigma) = \frac{df(t, \sigma)}{dt} = -\frac{t}{\sqrt{2 \pi} \sigma^3} \exp\left(-\frac{t^2}{2\sigma^2}\right) \]

and the second derivative is equal to

\[ f''(t, \sigma) = \frac{df'(t, \sigma)}{dt} \]

\[ = -\frac{1}{\sqrt{2 \pi} \sigma^3} \exp\left(-\frac{t^2}{2\sigma^2}\right) - \frac{t^2}{\sqrt{2 \pi} \sigma^5} \exp\left(-\frac{t^2}{2\sigma^2}\right) \]

\[ = -\frac{1}{\sqrt{2 \pi} \sigma^3} \left( \sigma^2 - t^2 \right) \exp\left(-\frac{t^2}{2\sigma^2}\right) \]

\[ = C \left( \sigma^2 - t^2 \right) \exp\left(-\frac{t^2}{2\sigma^2}\right) \]
The value of $C$ is equated to unity because if one takes the actual value of $C$, there will be no zero-crossings at higher values of sigma.

Figure 9 shows the mask obtained from the second derivative of the Gaussian for $\sigma = 12$.

![Diagram of second derivative of the Gaussian mask for $\sigma = 12$.]
3.2.4 Convolution

The German word for convolution is 'Faltung' which means folding. Convolution is used a lot in time series analysis (Robinson and Silvia, 1979), (Robinson, 1983). As applied to time series analysis, convolution involves the following steps (Robinson and Silvia, 1979):

1. Fold the time series $h_k$ (i.e. given the time series $h_k$, its folded version is $h_{-k}$).
2. Shift the time series $h_{-k}$ for a particular value of $n$ resulting in $h_{n-k}$.
3. Given the folded-shifted time series $h_{n-k}$, the remaining steps in convolution are to multiply the time series $s_k$ and $h_{n-k}$ term by term for a particular shift $n$, then sum the product $s_k \cdot h_{n-k}$ over all $k$ resulting in the output time series $x_n$ at $n$.

Mathematically it may be expressed as:

$$x_n = \sum s_k \cdot h_{n-k}$$

The above formula applies only to the discrete case. For the continuous case, one has to use the integral form of the convolution. For example the Gaussian kernel $g(x, \sigma)$ given by:

$$g(x, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

may be convolved with the function $f(x)$ as follows:
\[ F(x, \sigma) = f(x) \ast g(x, \sigma) \]

\[ F(x, \sigma) = \frac{1}{(\sqrt{2\pi}\sigma)} \int f(u) \ast \exp\left(-\frac{(x-u)^2}{2\sigma^2}\right) \, du. \]

where \( \ast \) is the convolution operator.

In this research the signal used is in the form of chain code which is a discrete representation of the curve. Therefore, the given chain code \( f(i); i = 1, 2, \ldots, N \): and its derivatives may be convolved with the mask of the second derivative of the Gaussian \( M_S(i); i = 1, 2, \ldots, M \) (where \( M \) is the mask size) to give the convoluted values \( C(i); i = M, M+1, \ldots, N-M \).

Mathematically it may be written as:

\[ C(k) = \sum_{i=-M}^{M} M_S(i) \ast f(i+k), \quad k = M, M+1, \ldots, N-M. \]

From the above formula, it is clear that one cannot get any zero-crossings in the first and the last \( M \) pixels part of the curve. This should not be a problem if the first and last points are fixed and a sufficiently small pixel size is selected. However, this problem can be easily solved either by repeating the first and last pixels \( M \) number of times or by extending the curve on either side.
It should be noted that unlike in the time series analysis, the convolution process in this research does not involve folding of one series.

3.2.5 Post Processing Using a Dynamic Mask

The zero-crossings of the convoluted values of the signal (derived from the chain codes of the digitized curve) with the second derivative of the Gaussian include some points which are not critical. One needs to remove these points if one is interested in only the minimum number of critical points which are sufficient to represent the curve. Some researchers e.g. Marimont (1984) has suggested that dynamic programming should be used to remove these points. But dynamic programming involves many iterations to achieve an optimum solution. Consequently, it is very expensive. Therefore, in this dissertation a dynamic mask is passed through the zero-crossings so that only the critical points are retained from among the zero-crossings.

Note that the dynamic mask used in this section is not the same as the one used in the Gaussian convolution. In this case, the dynamic mask is nothing but a rectangle of width equal to twice the specified tolerance. The length of the mask, as explained below, depends on the complexity of the line.
The concept of passing a dynamic mask in order to throw away the unwanted data was discussed in Lozover and Preiss (1983) and Imai and Iri (1986).

The process of passing the dynamic mask may be explained in the following steps (Lozover and Preiss, 1983):

1. Select the first three points from the zero-crossings.
2. Place the mask over these points so that the mask centerline coincides with the two end points.
3. Check if the intermediate point lies outside the mask.
4. If the intermediate point lies outside the mask, this point is taken to be the critical point and the process is repeated from step 1 with the last selected critical point as the first point over which the mask will be placed.
5. If the intermediate point falls inside the mask, the mask length is increased by one more point, and the process is repeated from step 2. This process is continued until one intermediate point falls outside the mask. Once one point is found which lies outside the mask, the process is repeated from step one with the last selected critical point as the first point of the mask.

From the above explanation it is obvious that the length of the mask is dynamic and it increases and decreases depend-
ing on the shape of the curve described by the zero-crossings.

It should be noted that this method is much cheaper than dynamic programming.

Fischler and Bolles (1986) describe a similar method as mentioned in section 2.2. But they use sticks (masks) of different lengths at different times. Obviously, it is better and more economic to use a mask of dynamic size rather than passing masks of different sizes at different times through the same data.

3.2.6 Step by Step Procedure for Critical Points Detection Using Zero-Crossings

The process of finding zero-crossings described above may be outlined as follows:

1. If the data are in vector format transform them to raster form by Bresenham's algorithm

2. Compute the chain code and then the extended chain code and its other derivatives from the rasterized data.

3. Compute the mask (masks) from the second derivative of the Gaussian for a particular (different values) of sigma.
4. Convolve the extended chain code (the signal) with the mask.

5. Look for the zero-crossings in the convoluted values and save them.

6. Pass the dynamic mask through the zero-crossings to select the critical points. This step may not be required for larger values of sigma and if the presence of a few noncritical points is not harmful.

3.3 CRITICAL POINTS DETECTION USING NORMALIZED SYMMETRIC MATRIX

In this section an algorithm to detect critical points in a digitized curve is described which uses the normalized symmetric scattered matrix. First the theory behind the technique is explained and then the step by step procedure to detect critical points is given.

3.3.1 The Nature of Scatter Matrices and Their Eigenvalues

Consider the geometry of the quadratic form associated with a sample covariance matrix. Suppose \( P = (p_1, p_2, \ldots, p_n) \) be a finite data set in \( \mathbb{R}^2 \) and \( P \) is a sample of \( n \) independently and identically distributed observations drawn from real two dimensional population.
Let \((\mu, \Sigma)\) denote the population mean vector and variance matrix and let \((\mathbf{v}_p, \mathbf{V}_p)\) be the corresponding sample mean vector and sample covariance matrix these are then given by (Uotila, 1986)

\[
\mathbf{v}_p = \frac{1}{n} \sum \mathbf{p}_i; \quad \mathbf{V}_p = \sum (\mathbf{p}_i - \mathbf{v}_p)(\mathbf{p}_i - \mathbf{v}_p)\]

Multiply both sides of the equation for \(\mathbf{V}_p\) by \((n-1)\) and denote the RHS by \(\mathbf{S}_p\) viz:

\[(n-1) \mathbf{V}_p = \mathbf{S}_p \quad \text{or} \quad \mathbf{S}_p = (n-1) \mathbf{V}_p\]

The matrices \(\mathbf{S}_p\) and \(\mathbf{V}_p\) are both 2X2 symmetric and positive semi-definite. Since these matrices are multiples of each other they share identical eigen-spaces.

According to Anderson and Bezdek (1983) one can use the eigenvalue and eigenvector structure of \(\mathbf{S}_p\) to extract the shape information of the data set it represents. This is because the shape of the data set is supposed to mimic the level shape of the probability density function \(f(x)\) of \(x\). For example, if the data set is bivariate normal, \(\mathbf{S}_p\) has two real, non-negative eigenvalues. Let these eigenvalues be \(\lambda_1\) and \(\lambda_2\). Then the following possibilities exist (Anderson and Bezdek, 1983):

1. If both \(\lambda_1\) and \(\lambda_2 > 0\), then the data set \(\mathbf{P}\) is degenerate, and \(\mathbf{S}_p\) is invertible and there exist with probability 1,
constants $a$, $b$, and $c$ such that $ax+by+c = 0$. In this case the sample data in $P$ lie on a straight line.

2. If $\lambda_1 > \lambda_2 > 0$, then the data set represent an elliptical shape.

3. If $\lambda_1 = \lambda_2 > 0$, then the sample data set in $P$ represent a circle.

3.3.2 **Eigenvalues of the Normalized Symmetric Scatter Matrix (NSS)**

Supposing that one has the following data:

$$P = (P_1, P_2, \ldots , P_n)$$

where $P_i = (x_i, y_i)$

Then the normalized scattered matrix $A$ is defined as

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = S_p/\text{trace}(S_p)$$

For the above data set $A$ is given by:

$$\text{Deno} = \sum ((x_i - x_m)^2 + (y_i - y_m)^2)$$

$$a_{11} = 1/\text{Deno} \sum (x_i - x_m)^2$$

$$a_{12} = 1/\text{Deno} \sum (x_i - x_m)(y_i - y_m) \quad (3.10)$$

$$a_{21} = 1/\text{Deno} \sum (x_i - x_m)(y_i - y_m)$$

$$a_{22} = 1/\text{Deno} \sum (y_i - y_m)^2$$

where $\mu_x = (x_m, y_m)$ is the mean vector defined as
\[ x_m = \frac{\sum x_i}{n}, \quad \text{and} \quad y_m = \frac{\sum y_i}{n} \]

Note that the denominator in (3.10) will vanish only when all the points under consideration are identical.

The characteristic equation of \( A \) is given by:

\[ |A - \lambda I| = 0 \]

which may be written as (for 2x2 matrix)

\[ \lambda^2 - \text{trace}(A) \lambda + \text{Det}(A) = 0 \]

where \( \text{Det}(A) = \text{Determinant of } A \).

By design the trace of \( A \) is equal to 1. Hence the characteristic equation of \( A \) reduces to

\[ \lambda^2 - \lambda + \text{Det}(A) = 0 \]

The roots of this equation are the eigenvalues and are given by:

\[ \lambda_1 = \frac{1 + \sqrt{1 - 4 \times \text{Det}(A)}}{2} \quad \text{and} \quad \lambda_2 = \frac{1 - \sqrt{1 - 4 \times \text{Det}(A)}}{2} \]

For convenience put \( D_x = \sqrt{1 - 4 \times \text{Det}(A)} \), then

\[ \lambda_1 = \frac{1 + D_x}{2} \quad (3.11) \]
\[ \lambda_2 = \frac{1 - D_x}{2} \quad (3.12) \]

Now \( \lambda_1 \) and \( \lambda_2 \) satisfy the following two conditions:

\[ \lambda_1 + \lambda_2 = 1 \quad (3.13) \]
Since the sum of the roots of an equation of the form
\[ ax^2 + bx + c = 0 \] are \( \lambda_1 + \lambda_2 = -\frac{b}{a} \)

Subtracting (3.12) from (3.11), one obtains
\[ \lambda_1 - \lambda_2 = D_x \] (3.14)

Since the eigenvalues \( \lambda_1 \) and \( \lambda_2 \) satisfy the equations (3.13) and (3.14) the three cases discussed in section 3.3.1 reduce to the following form (Anderson and Bezdek, 1983):

1. The data set represent a straight line if and only if \( D_x = 1 \)
2. The data set represent an elliptical shape if and only if \( 0 \leq D_x \leq 1 \)
3. The data set represent a circular shape if \( D_x = 0 \).

3.3.3 Algorithm to Detect Critical Points Using NSS Matrix

The fact that the analysis of the eigenvalues of the NSS matrix can be used to extract shape of the curve represented by the data set, may be exploited to detect critical points in the digital curve.

Assuming that the data is gross error free, and devoid of excessive noise, one can outline the algorithm to detect critical points in the following steps:

1. First take three points from the data.
2. Compute the NSS matrix and hence its eigenvalues.

3. If $D_X$ is greater than a certain tolerance (e.g. 0.95) add one more point to the data and repeat from step 2.

4. If $D_X$ is less than the tolerance point, point 2 is a critical point. Retain point 2 and repeat the process from step 1 with point two as the new starting point.

5. Repeat the process until the end of the data set is reached.
4.1 INTRODUCTION

In this section, results of finding critical points using the zero-crossings of the second derivative of the Gaussian with different signals derived from the digitized data, are given. In addition, results of different tests performed to find the best signal (signals) for this research are included. Moreover, results of tests performed to find the effects of using the different channels, different thresholding on the convoluted values, rotation, and smoothing of the signal are given. Furthermore, the results of tests performed to find the best way of raster to vector conversion are included.

Comparison between manual and algorithmic critical points detection is made. Finally, the results of critical points detection in vector data using NSS matrix are given.
4.2 VECTOR TO RASTER CONVERSION AND PIXEL SIZE SELECTION

As observed in section 3.2.1, if the original data is in vector form (which is usually the case for lineal data derived from existing maps using manual methods of digitization), one needs to transform them to raster form. After experimenting with different pixel sizes it was found that the pixel size of 0.25 mm was very suitable for the purpose of this research. It is because 0.25 mm is also equal to the resolution of the digitizer used. Results of pixel size selected to be equal to the shortest digitized vector were obtained. It was found that this method of pixel size selection was not appropriate for this research. If one selects a pixel size of larger than 0.25 mm, the original line will be distorted by an unacceptable amount. However, one can use a pixel size smaller than 0.25 mm but it will be more expensive in terms of storage and computations with a very little gain in accuracy.

Figure 10 is a pixel plot of the test figure for pixel size 0.25 mm. There are 358 vectors of digitized data and 1348 pixels in this figure. It is included to indicate the fact that there is some approximation involved in the process of vector to raster conversion.
4.3 EFFECT OF USING DIFFERENT SIGNALS ON ZERO-CROSSINGS

Unlike the previous researchers in this field such as Asada and Brady (1986), and Mokhtarian and Mackworth (1986) the path length is not used as a signal in the method of finding critical points in a digitized curve. It is because if path length is used as a signal for the convolution, one has to perform the convolution process for both the first and second derivatives of the Gaussian in order to compute the curvature. But if we use the signals derived from the chain code one only needs to perform the convolution with the second derivative of the Gaussian since curvature information is imbedded in these type of signals. As noted in section 3.2, the signals used in this research are the simple Freeman chain-code and other signals derived from it. As explained in section 3.2.2, the following are the five different signals used in this research:
1. Simple Freeman Chain-code (SFC)
2. Extended Freeman Chain-code (EFC)
3. Compensated EFC (CEFC)
4. Smoothed CEFC (SCEFC)

In order to find the effect of using different signals for the convolution with the second derivative of the Gaussian the zero-crossings of the convoluted values for these different signals are computed and plotted.

Figures 11, 12, 13, 14 and 15 show the test figure with the zero-crossings for the signals SFC, EFC, CEFC, SCEFC, and curvature respectively. The zero-crossings are indicated by a '+' sign. Note that the plus sign has an offset so that it would not obscure the actual selected zero-crossings. The actual position referred is the bottom left hand corner of the '+' sign. For this test, $\sigma = 6$, and there was no thresholding used. Table 1 summarizes the results.

An examination of figures 11, 12, 13, 14, and 15 and Table 1 shows that the number of zero-crossings at a particular channel depends upon the type of signal used. Maximum number of zero-crossings ($=160$) were obtained for the CEFC signal and minimum number ($=65$) were obtained for EFC and SFC signals.
Table 1
Zero-Crossings for Different Signals

<table>
<thead>
<tr>
<th>signal</th>
<th>figure no.</th>
<th># of zero-crossings</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFC</td>
<td>11</td>
<td>65</td>
</tr>
<tr>
<td>EFC</td>
<td>12</td>
<td>65</td>
</tr>
<tr>
<td>CEFC</td>
<td>13</td>
<td>160</td>
</tr>
<tr>
<td>SCEFC</td>
<td>14</td>
<td>150</td>
</tr>
<tr>
<td>curvature</td>
<td>15</td>
<td>79</td>
</tr>
</tbody>
</table>

As explained in section 3.2.2, the difference between CEFC and SCEFC is that the latter is the smoothed version of the former. The fact that there are more zero-crossings for CEFC than for SCEFC signal, indicates that smoothing reduces the zero-crossings. The effect of smoothing the signal will be examined in section 4.7.

Another fact that may be noticed from these results is that the zero-crossings of the convolutions of CEFC and SCEFC (see figures 13 and 14) are almost able to reproduce the original curve at this particular channel. But the corresponding zero-crossings of the convolutions of the SFC and EFC signal (see figures 11 and 12) are unable to do so. Nevertheless, the latter do not leave out the critical points.
Figure 11: Zero-Crossings of the SFC signal. For $\sigma = 6$ with no thresholding.

Unlike the other signals, the zero-crossings of the convolution of the curvature with the second derivative of the Gaussian has the effect of smoothing the curve (see figures 15 and 16). This signal does not pick up the critical points nor does it reproduce the original curve at this channel. But it selects points on either side of the critical points.
Figure 12: Zero-Crossings of the EFC signal. For $\sigma = 6$ with no thresholding.

Figure 13: Zero-Crossings of the CEFC signal. For $\sigma = 6$ with no thresholding.
Figure 14: Zero-Crossings of the SCEFC signal. For sigma = 6 with no thresholding.

Figure 15: Zero-Crossings of the Curvature signal. For sigma = 6 with no thresholding.
4.4 EFFECT OF CHANGING THE SCALE OF GAUSSIAN

As the scale of the Gaussian is increased, the width of the mask obtained from the second derivative of the Gaussian also increases. For example, the mask width = 29 for $\sigma = 3$, and the corresponding mask width for $\sigma = 12$ is 113.

It should be noted that the words mask width and filter size are used interchangeably.

When one performs the convolution of the signal with the mask obtained from the second derivative of the Gaussian, it has the effect of smoothing the signal. Therefore, greater smoothing will take place for larger mask width. Consequently, there will be fewer zero-crossings for larger mask width. But the zero-crossings of the convoluted val-
ues for higher values of $\sigma^*$ (i.e. for larger mask size) tend to pick up the critical points except for a few redundant points which are introduced as a result of the sudden bends of the curve on either side of the selected uncritical point.

Figures 14, 17, and 18 show the effect of changing the scale of Gaussian on the zero-crossings for the SCEFC signal. These figures were obtained for $\sigma$ equal to 6, 3, and 12.

Table 2 shows the effect of changing the $\sigma$ on zero-crossings for different signals.

Table 2
Effect of Changing Sigma on Zero-Crossings
for different signals.

<table>
<thead>
<tr>
<th>signal</th>
<th>sigma</th>
<th>Figure No.</th>
<th># of Zero-crossings</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCEFC</td>
<td>3</td>
<td>17</td>
<td>237</td>
</tr>
<tr>
<td>SCEFC</td>
<td>6</td>
<td>14</td>
<td>150</td>
</tr>
<tr>
<td>SCEFC</td>
<td>12</td>
<td>18</td>
<td>81</td>
</tr>
<tr>
<td>Curvature</td>
<td>3</td>
<td>19</td>
<td>169</td>
</tr>
<tr>
<td>Curvature</td>
<td>6</td>
<td>15</td>
<td>79</td>
</tr>
<tr>
<td>Curvature</td>
<td>12</td>
<td>20</td>
<td>56</td>
</tr>
</tbody>
</table>
Similarly figures 19, 15, and 20 illustrate the effect of changing $\sigma$ on zero-crossings for curvature as signal.

From Table 2 and figures 14, 15, and 17-20, it is clear that for SCEFC and curvature types of signals, the number of zero-crossings increase as the value of $\sigma$ decreases. Moreover, for signals SCEFC, it is observed that as the value of $\sigma$ increases the less critical points, for example, points along noisy interval and along straight lines are dropped off.

The above results confirm that the number of zero-crossings increases as the scale of the Gaussian decreases. This point is supported by previous findings by other researchers such as Witkin (1983), Asada and Brady (1986), and Mokhtarian and MacKworth (1986).

Unlike the other signals that are used, curvature behaves differently. In this case, as the scale of Gaussian $\sigma$ increases the zero-crossings further smooth the curve. If one compares figure 16 with figure 21, one can see that figure 21 is much smoother than figure 16. But the basic shape of the curve has been retained. This is a very interesting result which could have a lot of application in Cartography.
The effect of changing the scale factor $\sigma$ for signals SFC and EFC was found to be similar to the ones obtained for SCEFC except for the fact that there are fewer zero-crossings in the former cases for a particular $\sigma$. This was also evident from the results of section 4.2.
Figure 18: Zero-Crossings for SCEFC Signal with sigma = 12

Figure 19: Zero-Crossings for Curvature Signal with sigma = 3
Figure 20: Zero-Crossings for Curvature Signal with sigma = 12

Figure 21: Connection of the Zero-Crossings of figure 20 by straight lines. For sigma = 12 with no thresholding.
4.5 EFFECT OF THRESHOLDING THE CONVOLUTED VALUES ON ZERO-CROSSINGS

The signals that are used are fraught with noise. Due to the presence of noise in the signal, one gets a lot of zero-crossings which do not represent the critical points or which are not needed to retain the overall character of the curve. In order to overcome this problem, the magnitudes (absolute values) of the differences of the convoluted values were examined. It was found that the magnitudes of the differences of the convoluted values are, in general, larger for critical points than for other noncritical or noisy points in cases when the signal used was neither simple Freeman code nor curvature derived from it.

In the case of SFC, the above technique fails because of the jumps present in it. The curvature signals derived from SFC also do not yield to the above thresholding.

The above technique was applied to the results of figure 12 in which $\sigma = 6$ and there were 65 zero-crossings and many of them were redundant. By applying a threshold tolerance of 0.6, it was possible to discard all the non-critical points. The results of this operation are shown in figure 22. In this figure there are 45 points selected as critical points.
From the figure it is clear that the noisy portion of the curve, in the bottom left hand part, has been smoothed. But all the major features of the curve have been retained.

There are 81 zero-crossings in figure 18. Many of them are redundant. The redundant zero-crossings (i.e. the ones in the noisy portion of the curve and along the straight line) are successfully removed by thresholding. The results of thresholding for SCEFC signal are shown in figure 23 in which there are 46 points. The thresholding value used in this case was equal to 12.0.

Figure 22: The Results of Thresholding on Zero-Crossings. Signal Used: EFC Sigma = 6 Thresholding Tolerance = 0.6, 45 points selected.

It is interesting to note that figure 22 and 23 use signals EFC and SCEFC with different values of $\sigma$ namely 6 and 12.
But it has been possible to obtain almost identical results by thresholding. Note that the threshold value used in figure 22 is equal to 0.6 whereas it is equal to 12.0 in figure 23. This difference is due to two reasons viz:

1. The SCEFC signal is compensated by writing the even signal twice and the odd signal three times as explained in section 3.2.2.
2. It has been smoothed by adding 21 elements consecutively.

Figure 23: The Results of Thresholding on Zero-Crossings. Signal Used: SCEFC Sigma = 12 Thresholding Tolerance = 12, 46 points selected.
4.6 EFFECT OF ROTATION OF THE SIGNAL ON ZERO-CROSSINGS

Chain Codes and other signals derived from e.g. EFC, SCEFC, and curvature are sensitive to rotations of the curve (McKee and Aggarwal, 1977) and (Freeman, 1974). Various tests were performed in order to find the effect of rotation of the curve on zero-crossings and consequently on the detection of the critical points.

McKee and Aggarwal (1974) state that the effect of rotations would be the worst if one goes through the opposite direction of the curve. In other words, the results would be most affected if one reverses the starting and ending points in the case of open curve and if one goes along opposite direction in the case of closed curve.

The test for reversing the direction was performed on the test figure for SCEFC signal with exactly the same parameters as for figure 23. It was found that the results obtained were identical to those given in figure 23. The number of zero-crossings were equal to 46 which is the same as in figure 23. In addition, extensive tests were performed for different thresholding and different rotations of the curve for the signals EFC and SCEFC.

Table 3 summarises the results of rotating the curves by various angles for EFC signal. The thresholding was changed so as to keep only the critical points.
Table 3

Effect of Rotating the Curve on Zero-Crossings
for EFC signal

<table>
<thead>
<tr>
<th>Rotation Angle</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Zero-crossings</td>
<td>45</td>
<td>44</td>
<td>46</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>Thresholding</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.8</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 4 summarises the results of rotating the curves by various angles for SCEFC signal. The thresholding was changed so as to keep only the critical points. The thresholding needs to be increased as the scale of the Gaussian is increased. The correlation between thresholding and the scale of the Gaussian has not been established. Therefore, it is set by trial and error.

Table 4

Effect of Rotating the Curve on Zero-Crossings
for SCEFC signal.

<table>
<thead>
<tr>
<th>Rotation Angle</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Zero-crossings</td>
<td>46</td>
<td>47</td>
<td>47</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td>Thresholding</td>
<td>12</td>
<td>9.5</td>
<td>10.5</td>
<td>10.0</td>
<td>10.</td>
</tr>
</tbody>
</table>

It is observed from Table 3 that when the curve is rotated by various angles, different numbers of zero-crossings were
obtained. However, the change in the number of zero-crossings for different rotations is less than 4 for the signal EFC except for 45° rotation. In the latter case, one had to increase the thresholding so as to get about the same number of zero-crossings as in other rotations.

There was one redundant point introduced in each of the rotations by 30°, 45°, and 60°. However, there were no redundant points introduced for the rotation of the curve by 15° and for no rotations (see figure 22). The rotations would have no effect at all on zero-crossings if pixel size is reduced further.

It is seen from Table 4 that different thresholding was required for different amounts of rotations of the curve so as to retain about the same number of critical points for SCEFC signal. It should be noted that the thresholding had to be reduced by as much as 2.5 units (for 15° rotation) in order to keep all the critical points of the curve.

There was only one redundant point for each of the rotations 15° and 30° and none for other rotations.

A comparison of Tables 3 and 4 reveals that SCEFC is more sensitive to rotations of the curve than EFC signal. However, it should be pointed that as long as one uses the lowest thresholding value (= 9.5) for SCEFC signal and
(=0.6) for EFC signal, one should be able to retain all the critical points in a curve.

That is to say that the detection of critical points using either of the signals are robust enough to give the desired results. However, in terms of computation and storage requirement, EFC is preferable since it takes less computation and less storage.

The effect of rotations on zero-crossings was also studied by rotating the curve into four different quadrants. Results comparable to those given in Tables 3 and 4 were obtained for both signals EFC and SCEFC.

### 4.7 EFFECT OF SMOOTHING THE SIGNAL ON ZERO-CROSSINGS

In order to find the effect of smoothing the signal on zero-crossings, tests were performed on SCEFC signal by varying the number of points used for smoothing. Note that smoothing of the signal is achieved by consecutively adding a fixed number of signals at a time as noted in section 3.2.2. The smoothing prior to zero-crossings computed is introduced for the following two reasons:

1. It removes the noise in the signals (Ecoles et al. 1977).
2. It enables one to obtain the critical points without thresholding.

Table 5 contains the results of smoothing the signal on zero-crossings. In this study, smoothing is performed for fixed $\sigma = 12$, and no thresholding was applied.

Table 5

<table>
<thead>
<tr>
<th>No. of Zero-Crossings for Different Levels of Smoothing the Signal; $\sigma = 12$; No Thresholding.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Points Used in Smoothing</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>No Smoothing</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>37</td>
</tr>
<tr>
<td>111</td>
</tr>
</tbody>
</table>

It is clear from Table 5 that the number of zero-crossings decreases as the smoothing is increased. This is to be expected because the smoothing removes the noise in the
signal. It was observed that the points that were removed at greater smoothing were the ones from the noisy and straight line portion of the curve. The difference in the number of zero-crossings between no smoothing and smoothing using 37 number of pixel at a time was equal to 15.

Smoothing by adding up to nine points at a time was not sufficient to introduce any difference in zero-crossings, because of the small pixel size (\(=0.25 \text{ mm}\)) used. However, nearly 38% reduction in the number of zero-crossings was achieved by increasing the number of points used in smoothing at a time to 111.

4.8 EFFECT OF TAKING COMMON ZERO-CROSSINGS FROM DIFFERENT CHANNELS

Marr and Hildreth (1980) speculated that zero-crossings coinciding spatially over at least two channels are physically significant. This speculation was made by Marr and Hildreth for the grey-level detection. Its validity has been questioned by Babuaud et al. (1986).

In order to test if the speculation by Marr and Hildreth is correct for the type of problem at hand i.e. the detection of critical points in a digitized curve, many tests were performed for different channels. The result of one of the
test set for $\sigma = 3, 6, \text{ and } 12$ with no thresholding are shown in figure 24. In this figure, zero-crossings which are common to at least two channels out of three channels 3, 6, and 12 are given.

It is clearly seen from figure 24 that the speculation made by Marr and Hildreth (1980) for the type of signal used in this research is not correct. It is observed that many of the critical points have been missed. Therefore, it is concluded that zero-crossings from different channels are not required to detect critical points.

There are many zero-crossings for smaller values of sigma e.g. see figures 13 and 14. The problem is how to select the critical points from among these large numbers of zero-crossings. The same problem also arises in edge detection. In order to solve this problem Marr and Hildreth (1980) suggested that one should use zero-crossings which are common to at least two channels. But in this research, it was found that the problem can be solved by thresholding instead of taking the common zero-crossings from a number of channels. This is demonstrated by the results of section 4.5. Therefore, zero-crossings which are common to more than one channel do not necessarily represent critical points nor are "Physically significant" points for the type signal used in this dissertation.
Figure 24: Zero-Crossings which are Common to at least two Channels. Sigma = 3, 6, and 12; No Thresholding; signal: EFC.

4.9 RASTER TO VECTOR CONVERSION

It is known that the zero-crossings are rich in information (Marr and Hildreth, 1980). But it had not been pointed out before that the zero-crossings of the second derivative of the Gaussian with the signal derived from chain code is an excellent method of raster to vector conversion.

The zero-crossings of the convoluted values of the second derivative of the Gaussian with any of the signals SFC, EFC, and SCEFC may be used for the purpose of raster to vector conversion. This may be achieved in the following steps:
1. First compute the zero-crossings of the convolution of the second derivative of the Gaussian with any of the signals SFC, EFC, and SCEFC; using $\sigma < 4$ for SFC and EFC and $\sigma < 5$ for SCEFC. The different sigma values are used, because, for a particular sigma of the Gaussian different numbers of zero-crossings are obtained for different signals as shown in section 4.4.

2. Apply a low thresholding (as explained in section 4.4) to make the algorithm work fast.

3. Pass a dynamic mask (as explained in section 3.2.5) to eliminate the unwanted detail with a tolerance equal to the pixel size.

The selected points for the test figure are given in figure 25. In this figure, SCEFC was the signal used with $\sigma = 4$. In addition, a thresholding of 0.5 and a dynamic mask of width 0.01 inch were applied.

Figure 26 was obtained by joining the zero-crossings of figure 25 by straight lines. It is observed that except for minor deviations caused by vector to raster conversion process, figure 26 reproduces the original curve.

Note that there were 1324 pixels of size 0.25 mm. in the curve which have been reduced to 78 points after the raster to vector conversion process.
Figure 25: The Results of Raster to Vector Conversion. The '+' signs represent the selected points. $\sigma = 4$ Signal Used: SCEFC; Thresholding = 0.5 Dynamic Mask Width = 0.01 in.; 78 points selected.

Figure 26: The Results of Connecting the Selected Points of Figure 25 by Straight Lines.
It should be noted that similar results may be obtained if one uses either SFC or EFC as signal. Moreover, smoother results (more points) may be obtained by reducing the mask width and by decreasing the value of sigma.

4.10 COMPARISON BETWEEN MANUAL AND ALGORITHMIC CRITICAL POINTS DETECTION

In this section, results of critical points detection in the test figure by a group of people are given. These results are then compared with the results obtained from the zero-crossings technique of critical points detection. Further, the results of critical points detection by cartographers and non-cartographers as published in Marino (1979) are used to test if zero-crossings algorithm detects the same critical points.

4.10.1 Manual Critical Points Detection: The Experiment

In order to find if the zero-crossings method of critical points detection can mimic humans or not, the test figure was given to a group of 25 people who had at least one course in Cartography. In addition, they were told about the nature of critical points.
In the first experiment, they were asked to select not more than 31 points from the test figure. The results of critical points detection by the above group are shown in figure 27.

![Figure 27: Points selected by respondents. Each dot represents five respondents](image)

In figure 27 each dot represents 5 respondents. A point was rejected if it was selected by less than four respondents.

Figure 28 shows the results of critical points detection by zero-crossings algorithm. In this figure $\sigma = 22$; thresholding=11 and SCEFC was the signal used. Note that similar results may be obtained by using EFC signal with different $\sigma$. 
A comparison of figures 27 and 28 reveals that the critical points selected by the humans are almost identical with those selected by zero-crossings computer algorithm. Only five or less people slightly disagreed in the selection of a few points with zero-crossings algorithm. Moreover, the algorithm has only one non-critical point in the upper middle part of the curve. This point was introduced due to the abrupt changes in the direction of the curve on either side of the point.

It should be pointed out that the results of critical points selection could have been different if the respondents were asked to select all the critical points in the curve. However, the zero-crossings algorithm also can be made to detect different levels of critical points as evi-
denced by the results of figure 30 either by changing the thresholding or by changing the scale of the Gaussian.

In the second part of the experiment, the respondents were asked to select at the most 50 points from the curve. The results of the experiment are shown in figure 29.

Figure 29: Results of Critical Points Selection by Humans. When asked to select less than 50 points.

The results given in figure 29 are very close to those given in figures 22, 23, 25, and 32. Note that results of figure 32 are those detected by NSS matrix. The results of critical points detection using NSS matrix are given in the next section.
4.11 COMPARISON WITH THE RESULTS OF MARINO.

The results as published in Marino (1979) are given in figure 30. This figure derived from (Marino, 1979) shows the results of critical points detection by Cartographers and non-Cartographers at three different stages. Each stage is supposed to represent different level of generalization. Note that she did not display points selected by less than 16 respondents.

Figure 31 gives the results of critical points selection by zero-crossings algorithm for the same curve as that of Marino (1979).

In the first stage of the critical points selection, \( \sigma = 2 \); thresholding = 0.2; and dynamic mask of width 0.013 in. was used. In the second \( \sigma = 8 \) thresholding = 3.0 and no dynamic mask was used. In the third stage \( \sigma = 12 \) thresholding = 4.0; and no dynamic mask was used. In all the stages EFC was the signal used for the convolution with the second derivative of the Gaussian.

In Marino's experiment, there were 48-55, 24-26, and 9-10 points selected at stages one, two, and three respectively. Note that the first number in Marino's experiment is the
Figure 30: Results of Critical Points Selection by Humans. Obtained by Marino (derived from Marino, 1979)

The first number of points selected by Cartographers and the second number is the ones selected by non-Cartographers. In figure
Figure 31: Results of Critical Points Selection. by Zero-Crossings; signal used=EFC.

30, there are 52, 25, and 13 points selected for the three stages. These three different stages (or numbers of points) of critical points were found by changing the value of $\sigma$ as well as thresholding. The results of the above experiment along with Marino's results are summarized in the following Table 6:
### Table 6

Summary of three Stages of Critical Points Detection and Comparison with Marino’s Results

<table>
<thead>
<tr>
<th></th>
<th>Stage 1</th>
<th>Stage 2</th>
<th>Stage 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel (Sigma)</td>
<td>2</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>Thresholding</td>
<td>0.2</td>
<td>3.0</td>
<td>4.0</td>
</tr>
<tr>
<td>Dynamic Mask</td>
<td>0.013</td>
<td>0.004</td>
<td>0.002</td>
</tr>
<tr>
<td>No. of Critical</td>
<td>52</td>
<td>25</td>
<td>13</td>
</tr>
<tr>
<td>points Selected</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of points selected in Marino’s Experiment</td>
<td>48-55</td>
<td>24-26</td>
<td>9-10</td>
</tr>
</tbody>
</table>

In Table 6 in the last row, the first number is the number of points selected by Cartographers and the second number is the number of points selected by non-cartographers.

An examination of the results shown in figures 30 and 31 and Table 6 shows that the critical points selected by zero-crossings algorithm are identical to those selected by Cartographers and non-cartographers with the exception of two or three points. Especially, the results of the first stage of the experiment are remarkable in the sense that zero-crossings algorithm has selected points which are identical to those selected by cartographers and non-cartographers.
Tests were also performed for other curves given in Marino (1979). It was found that by changing the channels and thresholding, it was possible to select almost the same set of points as those selected by Marino's respondents. It should be noted that higher values of thresholding are required for larger values of sigma. It is because the filter width is larger for higher values of sigma as mentioned in section 4.4. The selection of a particular sigma and thresholding depends on the complexity of the line under consideration and the amount of detail to be retained.

In this section, results of critical points selection using signal only, were shown. However, it is possible to obtain similar results using SCEPC signal as well.

The above results indicate that the method of critical points detection using the zero-crossings of the convolution of the second derivative of the Gaussian with the signal derived from the digitized data does actually mimic the humans. In other words, using this algorithm one can detect the same set of critical points in a curve as those detected by humans.
4.12 RESULTS OF CRITICAL POINTS DETECTION BY NSS MATRIX

The algorithm discussed in section 3.3 is useful in detecting the critical points in vector data. The only parameter involved in this technique is $D_X$ which was defined in section 3.3.2. By varying the value of $D_X$ between say 0.8 to 1.0 one can get a varying amount of detail in a curve. Figure 32 shows the selected critical points for the test figure.

Figure 32: Results of Critical Points Selection by NSS Matrix. There were 50 points selected.

There are 50 points selected in this figure. It is clear from the figure that this method will be very useful for compression of digitized data since it retains the overall shape of the curve without retaining the unnecessary points.
CHAPTER V
LINE GENERALIZATION USING ZERO-CROSSINGS

5.1 INTRODUCTION

In this chapter, the usefulness of zero-crossings to the problem of line generalization is demonstrated. In addition, it is shown that the most popular method of line generalization so far used in cartography, namely the Forsen's (Duda and Hart, 1973) is not very useful for this purpose. Moreover, the results of generalized lines taken from existing maps of different scales are compared with those obtained from zero-crossings algorithm. Furthermore, the advantages of generalization using zero-crossings algorithm are discussed.

5.2 SELECTION OF LINES FOR TESTING AND EVALUATION

In order to test the usefulness of line generalization using the zero-crossings of the convoluted values of the second derivative of the Gaussian with the curvature signal
derived from the chain codes, we need to use the test lines from real maps.

For this purpose, USGS maps of Maryland at scales of 1:24000, 1:100000, and 1:250000 were used. Unfortunately, 1:50000 scale map of the same area was not available. From these maps, Hunting Creek in the river Miles was selected. It is a fairly complex line which has many complex features which can be useful to test the applicability of zero-crossings algorithm for the purpose of line generalization.

The selected feature was very carefully digitized using a Bendix digitizer at all the above scales. There were 620, 253, and 124 digitized points at scales 1:24000, 1:100000, and 1:250000 respectively. Figure 33 shows the selected feature at various scales. These are the lines as manually generalized by Cartographers and then digitized for this research.

Another lineal feature called Greenwood Creek was also selected from the same map series at the same scales as above. It was also digitized at the various scales. This is a very complicated feature to generalize. In addition, it also includes sections of the line which are very smooth see figure 40.
Figure 33: Hunting Creek along River Miles at Various Scales. Selected to test the applicability of zero-crossings algorithm. Figures A, B, and C are at scales 1:24000, 1:100000, and 1:250000 respectively.
5.3 INAPPLICABILITY OF FORSEN'S ALGORITHM TO LINE GENERALIZATION

The most popular method of line generalization hitherto used in Cartography has been the one described by Douglas and Peucker (1973), or as explained in Duda and Hart (1973), or as explained in Ramer (1973).

This method of line generalization is good only for data compression and it can be used to line generalization if the change in the scale between the original line and the generalized line is not very much. For example, from a scale of 1:500 to 1:750. However, if the change in the scale between the original map and the generalized map is drastic, for example, from a scale of 1:24000 to 1:100000, the above algorithm may not be used because in such cases it will leave spikes, and the lines look cluttered and aesthetically unpleasing as is clear from figures 34(A) and 36(A).

The above fact may be demonstrated using the selected line. Figure 34 (A) shows Hunting Creek generalized to scale 1:100000 from scale 1:24000 using the Forsen's algorithm. The tolerance used for this case was 0.05 inch and there were 88 points selected. The nature of this algorithm dictates that more number number of points are retained if the
Figure 34: Comparison between lines generalized by Cartographers and by Forsen's algorithm at a scale of 1:100000. Figures A, and B were generalized by Forsen's algorithm, and cartographer respectively.

tolerance is decreased. Figure 34(B) gives the corresponding lines generalized by the cartographers and there are 253 digitized points in this figure. It is clear from the figure that the lines generalized by Forsen's algorithm looks cluttered and is not anywhere close to the one generalized by the cartographer. This difference especially around points A and B is very obvious. At these points, Forsen's algorithm has left the spikes which are not seen in any maps generalized by the cartographers. Moreover, the lines generalized by the Forsen's algorithm looks clumsy and unaesthetic.

In order to show how the lines generalized by Forsen's algorithm look when enlarged, the results of figure 34(A)
were enlarged by 4 times. The enlarged line is shown in figure 35. In this figure we can see very clearly that the lines generalized by this algorithm do not look acceptable. The existence of various unpleasant spikes has become more pronounced in this figure.

Figure 35: Result of Enlarging figure 34 (A) by four times.
Figure 36: Comparison between lines generalized by Cartographers. and by Forsen's algorithm at a scale of 1:250000. Figures A, and B were generalized by Forsen's algorithm, and cartographer respectively.

Figure 36 (A) shows the results of Hunting Creek generalized from scale 1:24000 to scale 1:250000 by Forsen's algorithm. In this case the tolerance used is equal to 0.08 and there were 66 points selected. Figure 36 (B) shows the corresponding lines generalized by the cartographers at the same scale and there are 124 digitized points in this figure. It is again noticed that the lines generalized by Forsen's algorithm are not very close to those performed by the cartographers. This example further illustrates the inapplicability of the Forsen's algorithm to the problem of line generalization. However, as noted earlier this algorithm may be used for data compaction and for line generalization if it involves only a minor reduction in scale.
5.4 COMPARISON BETWEEN ALGORITHMIC AND MANUAL LINE GENERALIZATION

As noted in section 4.4, the zero-crossings of the convoluted values of the second derivative of the Gaussian with the curvature signal derived from the chain codes do not pick up the critical points but only preserve the basic shape or the character of the curve. For this reason, this method may be used to generalize lines especially when there is a drastic reduction in scale e.g. from 1:24000 to 1:100000.

Figure 37: Comparison between lines generalized by Cartographers and by Zero-Crossings algorithm at a scale of 1:100000. Figures A. and B were generalized by Zero-Crossings algorithm, and cartographer respectively.
The results of a lines generalized from scale 1:24000 to 1:100000, are shown in figure 37 (A). In this figure curvature derived from the chain codes was the signal used and $\sigma=4$. There were 136 zero-crossings retained. Figure 37(B) is the corresponding lines generalized by the cartographers and there are 253 points in this figure.

By comparing figures 37(A) and (B), it is noticed there is a remarkable similarity between the lines generalized by the Cartographer and zero-crossings algorithm except that in some edges the cartographer has made some exaggerations of detail even when it was not required and in some instances he/she has omitted the detail even though it could have been retained. Moreover, zero-crossings algorithm used nearly 50% less number of points. Despite of this fact the results are pleasingly smooth.

Figure 38 is the enlargement of figure 37(A) in order to prove that zero-crossings algorithm does not produce a cluttered figure and also leaves no spikes.

Figure 39 (A) shows the generalization of Hunting Creek from the scale 1:24000 to 1:250000. In this figure curvature was the signal used and $\sigma=12$. 
Figure 38: Result of Enlarging figure 37 (A) by four times.

Figure 39: Comparison between lines generalized by Cartographers and by Zero-Crossings algorithm at a scale of 1:250000. Figures A and B were generalized by Zero-Crossings algorithm, and cartographer respectively.
There were 79 zero-crossings retained in this figure. Figure 39(B) is the corresponding lines generalized by a Cartographer. There were 124 points digitized in this figure. Note that these numbers are mentioned to give an idea as to how many points does the zero-crossings algorithm retain to get results similar those of cartographers. But it is known that the digitizers tend to digitize redundant points.

By comparing these two figures, it can again be claimed that it has been possible to mimic the cartographer even though the reduction in scale is drastic. One also notices that the cartographer was negligent and smoothed the line around point 'a' much more than he/she should have. Maybe the cartographer was not at ease with himself/herself at the time he/she generalized this line.

In order to prove the flexibility of zero-crossings method of line generalization, Greenwood Creek was also generalized at the above scales. Figure 40 shows Greenwood Creek at various scales.

Figure 41 (A) shows the results of Greenwood Creek generalized from 1:24000 to 1:100000. In this case, the curvature was the signal used and \( \sigma = 4 \) and there were 183 points retained. Figure 41 (B) shows the corresponding lines generalized by cartographers in which there are 226 digitized points. If one examines these two figures, one notices
Figure 40: Greenwood Creek along River Miles at Various Scales. Selected to test the applicability of zero-crossings algorithm. Figures A, B, and C are at scales 1:24000, 1:100000, and 1:250000 respectively.
especially around points E and F that the Cartographer has grossly distorted the character of the line. He/she has smoothed out the detail which could have been easily depicted at this scale. Moreover, he/she has grossly exaggerated the tip ends at points M, N and P, and Q.

Figure 41: Comparison between lines generalized by Cartographers, and by Zero-Crossings algorithm at a scale of 1:100000; Figures A, and B were generalized by Zero-Crossings algorithm, and cartographer respectively; Greenwood Creek.

Similarly, fig 42(A) is the result of Greenwood Creek generalization from scale 1:24000 to 1:250000. In this case $\sigma=10$ and curvature was the signal used. There are 83
Figure 42: Comparison between lines generalized by Cartographers and by Zero-Crossings algorithm at a scale of 1:250000; Figures A, and B were generalized by Zero-Crossings algorithm, and cartographer respectively; Grenwood Creek.

points retained in this figure. Figure 42(B) is the corresponding line generalized by a cartographer and there are 135 digitized points in this figure. Here again the results are similar to those obtained by Cartographers except for the tip ends. At points X, Y, and Z the Cartographer has shortened the tip ends and has also exaggerated the width of the tip ends. Other than that the results obtained by zero-crossings algorithm are close to those obtained by Cartographers.
5.5 EFFECT OF STEP BY STEP GENERALIZATION

During the course of investigation it was found that the scale transition from 1:24000 to 1:250000 may be too drastic a change in scale. Therefore, generalization was performed at different steps i.e. first the data from 1:24000 was generalized to 1:50000 which in turn was generalized to 1:100000. This could be further generalized to 1:250000.

Figure 43: Comparison between lines generalized by Cartographers and by Zero-Crossings algorithm from scale 1:50000 to Figures B, and A were generalized by Zero-Crossings algorithm, and cartographer respectively; Scale 1:100000 Greenwood Creek.
Figure 44: Comparison between lines generalized by Cartographers, and by Zero-Crossings algorithm from scale 1:100000 to Figures A, and B were generalized by Zero-Crossings algorithm, and cartographer respectively; scale 1:250000 Greenwood Creek.

Figure 43(B) depicts the lines generalized from scale 1:50000 to 1:100000. In this case curvature was the signal used with \( \sigma = 3 \) There are 151 points retained in this case. Figure 43(A) gives the corresponding lines generalized by cartographers.

If one compares figures 41(A), 43(A) and 43(B), one can see that figure 43(A) is closer to the lines generalized by the cartographer than figure 41(A).

Similarly, fig 44(A) is the result of line generalization achieved from a scale of 1:100000 to 1:250000. In this case, \( \sigma = 3 \) and there are 87 points retained in this figure. Figure 44(B) shows the corresponding lines general-
ized by the cartographer. A comparison of figures 42 (A), 44(A), and 44(B) indicates that figure 44(A) is closer to the lines generalized by a cartographer than figure 42(A).

This shows that it maybe better to generalize lines at different steps of scale rather than generalizing directly from a larger scale to a much smaller scale. Nevertheless, with the selection of an appropriate channels one can achieve acceptable results even when one generalizes from a larger scale to smaller scale.

5.6 LINE GENERALIZATION: CRITICAL POINTS DETECTION VS SHAPE PRESERVATION

In section 1.4, the importance of critical points detection and preservation during the process of line generalization was emphasized. Many researchers in the field of Cartography such as Boyle(1970), Marino (1979), McMaster (1983), and White(1985) implied that the critical points are important and must be retained in the process of line generalization. Some of the researchers e.g. Marino (1979) implied that the detection of critical points is one way of quantifying the subjective process of line generalization.

Now in the light of this research, it has emerged that one cannot preserve all the critical points in the process of
line generalization. Some of the critical points which are likely to cause spikes in the generalized lines must be eliminated if the generalized lines are to be smooth, uncluttered, and aesthetically pleasing. On the other hand as pointed out by many researchers such as Keates (1973), Campbell (1984), and Buttenfield (1985) we need to retain the basic shape of the line or the feature during the process of line generalization. Unfortunately, it was observed that the basic shape or the character of a line is not preserved by preserving all the critical points. This is especially true if the change in the scale is very large. This is clear from many of the examples used from the real map data e.g. see figures 37, 39, 41 etc.

I do not mean that the detection of critical points is useless and therefore should not be pursued. In fact the detection of critical points is important and very useful for the purpose of data compression in cartography and in many other disciplines. Moreover, critical points are useful in shape recognition if it does not involve a scale change. For this reason, the critical points detection are important in the fields of pattern recognition, image processing, computer vision, computer graphics etc.

Critical points detection in line generalization is important if one considers it as a two step process namely:
first detect the critical points and secondly perform some kind of smoothing of the critical points. In such situations it is perfectly all right to detect the critical points. But I do not do this because the method of line generalization using the zero-crossings of the convoluted values of the second derivative of the Gaussian with the curvature signal derived from the chain code achieves both smoothing and shape preservation in one step.
CHAPTER VI
EVALUATION OF LINE GENERALIZATION

6.1 INTRODUCTION

It is said that the test of a curry is in eating. Similarly, one can say that the test (evaluation) of automatic line generalization should be in its ability to more or less reproduce what cartographers do manually.

As discussed in section 2.5, the problem of evaluating line generalization is a very difficult one in the sense that there is no objectively defined correct answer to it. It is not possible to say whether one has achieved the correct answer. These type of problems fall into the category of fuzzy sets and it does not really matter if the answers are not exact. For example, the results of figures 37 (A) and (B) (these are the lines generalized by zero-crossings algorithm and cartographers) are not identical but either of these are equally acceptable.
McMaster (1983) came up with a number of quantitative methods of evaluating the algorithms of line generalization. These quantitative methods do not provide exact methods of evaluation. However, they do provide some information about the nature of a particular algorithm.

6.2 EVALUATION BY COMPARING WITH THE RESULTS OBTAINED MANUALLY

It was seen in section 5.4 that the results of lines generalized by finding the zero-crossings of the convoluted values of the second derivative of the Gaussian with the signal are very close to those obtained by cartographers.

The results shown in figures 37, 39, 41, and 42 have demonstrated that the zero-crossings method of line generalization can mimic the results achieved by cartographers. As a matter of fact, the results obtained by zero-crossings algorithm are better than those produced by cartographers in the sense that the lines generalized automatically by computer are objective and consistent and therefore, do not depend on the mental condition of the cartographer.
6.3 EVALUATION USING MCMASTERS MATHEMATICAL METHODS

As stated in section 2.5, McMaster (1983) suggested 30 measures for the evaluation of line simplification and generalization. Out of the 30 measures, only five are used in this investigation. The following criteria as derived by McMaster (1983) are used to see how the zero-crossings method compares with the Forsen's algorithm.

1. Relative change in the curve length (RCCL).
2. Difference in the average number of coordinates (DANC).
3. Relative change in the number of coordinates (RCNC).
4. Relative change in the deflection angles (RCDA).
5. Difference in the average deflection angular change per inch (DADAC).

6.4 COMPUTATION OF MCMASTERS MATHEMATICAL MEASURES

Note that McMaster seems to have been somewhat confused when he computed his mathematical measures. For example, he says "Ratio of change in the ..." when he only computes a simple ratio between the quantities derived from the original and generalized lines. However, in this research the relative change in a certain quantity e.g. curve length between the original and generalized curve is used.
The following formula are used to compute the various mathematical measures:

In the formula given below OC = Original Curve and GC = Generalized Curve.

\[
\text{RCCL} = \frac{\text{Length of } \text{OC} - \text{Length of } \text{GC}}{\text{Length of } \text{OC}} \times 100\%
\]

\[
\text{DANC} = \frac{\# \text{ of Coordinates in } \text{OC}}{\text{Length of } \text{OC}} - \frac{\# \text{ of Coordinates in } \text{GC}}{\text{Length of } \text{GC}}
\]

\[
\text{RCNC} = \frac{\# \text{ of Coordinates in } \text{OC} - \# \text{ of Coordinates in } \text{GC}}{\# \text{ of Coordinates in } \text{OC}} \times 100\%
\]

\[
\text{RCDA} = \frac{\text{sum of DA in } \text{OC} - \text{sum of DA in } \text{GC}}{\text{sum of DA in } \text{OC}} \times 100\%
\]

\[
T_1 = \frac{\text{Sum of the DA in } \text{OC in radians}}{\text{Total Length of } \text{OC}}
\]

\[
T_2 = \frac{\text{Sum of the DA in } \text{GC in radians}}{\text{Total length of } \text{GC}}
\]

\[
\text{DADAC} = T_1 - T_2
\]

where DA = Deflection Angle
### Table 7

Various Mathematical Measures Computed for Zero-Crossings Algorithm and Forsen's Algorithm; Data: Hunting Creek.

<table>
<thead>
<tr>
<th>scale</th>
<th>Forsen's</th>
<th>Zero-Crossings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:100k</td>
<td>4.9%</td>
<td>9.2%</td>
</tr>
<tr>
<td>1:250k</td>
<td>7.6%</td>
<td>21.0%</td>
</tr>
<tr>
<td>1:100k</td>
<td>20.257</td>
<td>18.063</td>
</tr>
<tr>
<td>1:250k</td>
<td>21.070</td>
<td>19.975</td>
</tr>
<tr>
<td>1:100k</td>
<td>85.8%</td>
<td>78.0%</td>
</tr>
<tr>
<td>1:250k</td>
<td>89.4%</td>
<td>87.3%</td>
</tr>
<tr>
<td>1:100k</td>
<td>83.8%</td>
<td>80.5%</td>
</tr>
<tr>
<td>1:250k</td>
<td>88.9%</td>
<td>86.0%</td>
</tr>
<tr>
<td>1:100k</td>
<td>27.900</td>
<td>26.403</td>
</tr>
<tr>
<td>1:250k</td>
<td>29.576</td>
<td>27.672</td>
</tr>
</tbody>
</table>
### Table 8

Various Mathematical Measures Computed for Zero-Crossings Algorithm and Forsen's Algorithm; Data: Greenwood Creek.

<table>
<thead>
<tr>
<th>scale</th>
<th>Forsen's</th>
<th>Zero-Crossings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:100k</td>
<td>6.3%</td>
<td>8.6%</td>
</tr>
<tr>
<td>1:250k</td>
<td>8.1%</td>
<td>17.4%</td>
</tr>
<tr>
<td>1:100k</td>
<td>15.060</td>
<td>13.167</td>
</tr>
<tr>
<td>1:250k</td>
<td>15.830</td>
<td>14.530</td>
</tr>
<tr>
<td>1:100k</td>
<td>84.2%</td>
<td>75.0%</td>
</tr>
<tr>
<td>1:250k</td>
<td>88.4%</td>
<td>83.6%</td>
</tr>
<tr>
<td>1:100k</td>
<td>83.2%</td>
<td>74.0%</td>
</tr>
<tr>
<td>1:250k</td>
<td>90.5%</td>
<td>81.2%</td>
</tr>
<tr>
<td>1:100k</td>
<td>25.140</td>
<td>21.899</td>
</tr>
<tr>
<td>1:250k</td>
<td>28.326</td>
<td>23.656</td>
</tr>
</tbody>
</table>

6.5 **ANALYSIS**

Table 7 gives the results of the various mathematical measures computed for Forsen's algorithm and zero-crossings algorithm for Hunting Creek data. The results are computed for the two different scales 1:100000 and 1:250000. For a particular scale the relative change in the curve length (RCCL) is higher for the zero-crossings algorithm than for Forsen's algorithm. It is because unlike Forsen's algorithm the zero-crossings method does not leave spikes nor does it
make the line cluttered as the scale is reduced. In addition, note that even for the same algorithm the relative change in length increases as the scale is reduced.

The difference in the average number of coordinates per inch (DANC) for Forsen's algorithm is higher than for zero-crossings algorithm. This measure also decreases as the scale is reduced. This measure indicates that zero-crossings algorithm has better distribution of coordinates than Forsen's algorithm.

The relative change in the number of coordinates (RCNC) is lower for zero-crossings algorithm than for Forsen's algorithm. Moreover, this measure increases with the decrease in scale. This means that zero-crossings technique retained larger number of coordinates than Forsen's algorithm. This is because zero-crossings algorithm gives smoother results than Forsen's algorithm e.g. see figures 37, 39 and 41.

The relative change in the sum of the deflection angles (RCDA) shows the amount of smoothing and simplification that has taken place in the line. This factor is higher if smoothing is lower i.e. if there are less number of segments in the line. For both scales this factor is less for zero-crossings algorithm than for Forsen's algorithm. It is because zero-crossings algorithm gives smoother results.
The difference in the average deflection angular change per inch (DADAC) is closely linked with the angularity of the curve and its length. Again this factor is higher if there is a drastic change in the generalized line. This change is higher for Forsen's algorithm than for zero-crossings algorithm because the former algorithm produces results which are very different from the original curve as compared to the latter algorithm which tends to keep intact the basic character of the curve.

6.6 CHECKING THE ALGORITHMIC AND CARTOGRAPHIC CRITERIA OF ZERO-CROSSINGS

It would be very interesting to check whether the zero-crossings algorithm for line generalization satisfies the algorithmic and cartographic criteria proposed by Zoraster et al. (1984). These criteria are mentioned in section 2.5.

6.6.1 Cartographic Criteria

1. The zero-crossings algorithm preserves the map character and accuracy because in this algorithm one does not compute any new coordinates unlike in smoothing algorithms.
2. The zero-crossings algorithm tends to preserve the global features as may be noticed from previous examples such as figure 37.

3. The amount of generalization may be varied by using different thresholding or by using different channels of the Gaussian as illustrated by examples given in figures 37 and 39.

From the above discussions, it is clear that zero-crossings algorithm satisfies all the Cartographic criteria proposed by Zoraster et al. (1984).

6.6.2 Algorithmic Criteria

1. By using different channels of the Gaussian and by thresholding, the zero-crossings algorithm produces predictable reduction in data. This has been demonstrated by the results given for example in figures 22 and 23.

2. The Gaussian convolutions are invariant with respect to mathematical operations.

3. The zero-crossings algorithm is predictably controlled by the scale of the Gaussian and by thresholding.

4. The zero-crossings algorithm may be made modular to meet different map specifications by using different channels.
5. The zero-crossings algorithm is computationally fast since it does not involve computation of any square roots nor does it involve matrix manipulation.

From the above discussions it is clear that the zero-crossings algorithm also satisfies the algorithmic criteria as proposed by Zoraster et al. (1984).
CHAPTER VII
GENERAL CONCLUSIONS AND RECOMMENDATIONS

1. Using the zero-crossings of the convoluted values of the second derivative of the Gaussian with the curvature signal derived from the digitized data, it was possible to automatically generalize lines even when there was a drastic change in scale.

2. The zero-crossings of the convoluted values of the second derivative of the Gaussian with the signals derived from the chain code provide an efficient and reliable means of raster to vector conversion.

3. By thresholding the zero-crossings or by using higher channels, one can detect the critical points in a digitized curve. Such a process of critical points detection is immensely important in data compaction, image processing, image matching, industrial parts recognition, and pattern recognition.

4. For the type of problem at hand and for the type of signal used (mainly the signals derived from the chain code data) there is no need to use zero-crossings from different channels in order to capture the basic character of a digitized line.

- 163 -
5. By thresholding the convoluted values of the zero-crossings we were able to filter the noise in the zero-crossings. To my knowledge, this had not been realized before. Therefore, the previous researchers resorted to the common zero-crossings from a number of different channels.

6. The number of zero-crossings decreases as the scale of the Gaussian is increased.

7. The zero-crossings are independent of rotation of the curve.

8. The number of zero-crossings decreases as the signal (SCEFC) derived from the chain code is smoothed.

9. Different levels of the critical points may be detected by changing the scale of the second derivative of the Gaussian and by using different thresholding. Such a process of critical points detection can mimic the humans both - cartographers and non-cartographers.

10. Different numbers of zero-crossings are obtained when different signals derived from chain code are used.

11. Simple chain code may not be used as a signal for the Gaussian convolution to find the critical points because of the inherent discontinuity of the code.

12. Equivalent results may be obtained by using either EFC or SCEFC signals.
13. EFC and SCEFC may not be used as signals for the generalization of lines involving a great change in the scales between the original and the generalized maps.

14. When the process of line generalization involves a drastic change in scale e.g. 1:24000 to 1:100000 or smaller, what one needs to preserve are the basic shape and general character of a line and not the critical points. The latter conclusion is contrary to the findings and beliefs of some of the previous researchers.

15. The analysis of the eigenvalues of the normalized symmetric scattered matrix provides an useful way of detecting critical points in digitized curves.

16. The superiority of zero-crossings algorithm for finding critical points and line generalization lies in that this process is faster because it does not require the computation of any square roots nor does it involve any inversion or manipulation of matrices. Moreover, zero-crossings algorithm for line generalization is better and different from all the existing algorithms because zero-crossings algorithm performs line smoothing and data compression at the same time. In addition, unlike the existing methods of line smoothing such as weighting, the line generalized by zero-crossings algorithm passes through the existing
points and does not compute new coordinates. This is exactly the reason why zero-crossings algorithm gives results which are very close to those obtained by cartographers.

17. The limitation of zero-crossings algorithm for the purpose of line generalization is that it cannot achieve any exaggeration of detail, nor does it perform any feature displacement.

18. The same idea may be extended to the Digital Elevation Models for breakline detection (Schenk, 1986). However, for this case we need to use the two dimensional Laplacian of the Gaussian as the mask. The heights of the points may be used as the signal. This would be a very interesting area of investigation.

19. The zero-crossings method of critical points detection in raster data may be linked with an edge detection algorithm such as zero-crossings of the convoluted values of the Laplacian of the Gaussian with gray level as the signal to build a complete image processing system.
REFERENCES


Duda, R.O. and P.E. Hart (1973), Pattern Classification and Scene Analysis, Willey Interscience.


Pavlidis, T. and S.L. Horowitz (1973), "Piecewise Appro-


Topfer, F. and W. Pillewizer (1966), "The Principles of


