Regular Grid DEM Data Compression by Using Zero-Crossings: The Automatic Breakline Detection Method

by

Udomsak Chakreyavanich

Report No. 412

Department of Geodetic Science and Surveying
The Ohio State University
Columbus, Ohio 43210-1247

November 1991

To my late father, Pengkiank Tieo, who passed away during my study,
and to my mother, my wife and my daughter.

ABSTRACT

This report presents an automatic method of extracting *Breaklines* and *Breakpoints* from a set of regular grid DEM data. The regular DEM data are considered as an array of signal which can be convolved by the Laplacian of Gaussian (LoG) filter. It is found that points having positive maximum or negative minimum convoluted values *(extremas)* between zero-crossings of LoG are located at places critical for terrain description. The extremas obtained by the proposed method correspond to breaklines and breakpoints of the terrain. These breakpoints and breaklines contain explicit information and have more meaning than regular grid points.

Based on the concept of using breaklines and breakpoints as the terrain relief representation, the proposed technique can be considered as a DEM data compression process. This work shows that contour maps constructed from the extracted breaklines and breakpoints exhibit similar terrain roughness as the ones constructed from the full DEM data. The results justify the possibility of implementing this concept. Furthermore, breaklines and breakpoints obtained by this technique are almost identical to the ones manually collected by human operators.

ACKNOWLEDGEMENTS

This report was submitted in Autumn 1989 to the Graduate School of The Ohio State University in partial fulfillment of the requirements for the Ph.D. degree.

First of all, I would like to thank Chulalongkorn University, Bangkok, Thailand, which gave me the scholarships and the opportunity to pursue my Doctoral studies. Thanks should also go to every faculty member at The Department of Surveying Engineering, Chulalongkorn University, especially Dr. Wicha Jiwalai and Dr. Chugiat Wichiencharoen, who always realize that higher education is an important asset for their staff.

I would like to express my sincere thanks to my advisor, Professor Anton F. Schenk, who gave me a lot of constructive suggestions during this research, and Professor Dean C. Merchant, the best teacher I have ever met, for his sincere moral support. I also would like to thank Professor Joseph C. Loon and Professor Kim L. Boyer for their thoughtful comments while they served on my reading committee.

I extend my thanks to all faculty members at the Department of Geodetic Science and Surveying, especially Professor Ivan I. Mueller, the departmental chairman, who exposed me to the real world of Geodetic Science and Surveying.

Finally, I would like to express my gratitude to my wife, who is always beside me, and my daughter, who has not seen her parents since March 1985. Without them, I would not have been able to succeed in this study.

Contents

DEDICATION ii

ABSTRACT iii

ACKNOWLEDGEMENTS iv

LIST OF TABLES vii

LIST OF FIGURES viii

1 INTRODUCTION 1
- 1.1 Scope and Objectives of the Research 2
- 1.2 Organization of the Report 3

2 DEM DATA COMPRESSION 5
- 2.1 Introduction 5
- 2.2 DEM Data Sampling 6
  - 2.2. 1 Progressive Sampling 6
  - 2.2. 2 Composite Sampling 7
  - 2.2. 3 Optimum Sampling 8
- 2.3 Triangulated Irregular Network (TIN) 9
- 2.4 DEM Representation by Mathematical Models 12
- 2.5 DEM Data Compression 13

3 EDGE DETECTION OPERATORS 17
- 3.1 Introduction 17
- 3.2 Edge Detection Operations 18
  - 3.2. 1 Smoothing Operation 19
  - 3.2. 2 Differentiation Operation 20
  - 3.2. 3 Detecting Operation 21
- 3.3 Edge Detection Operators 22
  - 3.3. 1 Directional Edge Detection Operators 22
  - 3.3. 2 Non-directional Edge Detection Operators 23
    - 3.3. 2.1 Laplacian of Gaussian (LoG) 24
    - 3.3. 2.2 Drawbacks of LoG 27

4 DEM DATA COMPRESSION BY BREAKLINE DETECTION 31

4.1 General 31

4.2 Terrain Representation by Breakline Map 32

4.3 Breakline Detection by Edge Operators 34

4.3. 1 Detecting Breaklines by LoG 35

4.3. 2 Computational Consideration of LoG 40

4.4 Proposed DEM Data Compression Strategy 42

5 EXPERIMENTS AND RESULTS 45

5.1 General 45

5.2 Testing Data 45

5.3 Testing Procedure 48

5.4 Results 49

5.5 Analysis of the Results 71

6 CONCLUSIONS AND RECOMMENDATIONS 77

6.1 Conclusions 77

6.2 Recommendations 78

APPENDIX A. RELATIONSHIP BETWEEN σ AND f OF GAUSSIAN FUNCTION 79

List of Tables

1 UTM Coordinates of the cross-sections 47

2 UTM Coordinates of the windows 47

3 Frequency Response and Energy Level of DEM surface 50

4 Number of zero-crossings and compression rate of the windows at different σ 51

5 Number of extremas and compression rate of the windows at different σ 51

List of Figures

1 TIN method. (a). The Thiessen Polygons, (b). The Delaunauy Triangulations [Davis 86] 11

2 The principle of Makarovic's data compression [Makarovic 77b] 14

3 The LoG Function. (a). The profile of LoG, (b). LoG viewed from the top, (c). 3-D view of LoG [Greenfeld 87]. 25

4 False zero-crossing of staircase edge created by LoG [Ulupinar and Medioni 87]. 28

5 (a) Ideal Step Edge, (b) Response of LoG to the edge 37

6 (a) Ramp edge and LoG, (b) Response of LoG to the edge when $d < s$ [Huertas and Medioni 86] 38

7 (a) Ramp edge and LoG, (b) Response of LoG to the edge when $d > s$ [Huertas and Medioni 86] 39

8 (a) Roof edge and LoG, (b) Response of LoG to the edge [Huertas and Medioni 86] 39

9 Approximation LoG by DoG [Huertas and Medioni 86] 41

10 USGS DEM surface 46

11 Zero-crossings at $\sigma = 8$ 52

12 Zero-crossings at $\sigma = 4$ 52

13 Zero-crossings at $\sigma = 2$ 53

14 Extremas at $\sigma = 8$ 53

15 Extremas at $\sigma = 4$ 54

16 Zero-crossings of Section-1 54

17 Zero-crossings of Section-2 55

18 Zero-crossings of Section-3 55

19 Extremas of Section-1 56

20 Extremas of Section-2 56

21 Extremas of Section-3 57

22 Scale-space Plot of Zero-crossings of Section-1 57

23 Scale-space Plot of Zero-crossings of Section-2 58

24 Scale-space Plot of Extremas of Section-1 58

25 Scale-space Plot of Extremas of Section-2 59

26 Original contour map of window 1 59

27 Original contour map of window 2 60

28 Original contour map of window 3 60

29 Zero-crossings of window 1, $\sigma = 4$ 61

30 Zero-crossings of window 1, $\sigma = 2$ 61
31 Zero-crossings of window 2, $\sigma = 4$ 62
32 Zero-crossings of window 2, $\sigma = 2$ 62
33 Zero-crossings of window 3, $\sigma = 4$ 63
34 Zero-crossings of window 3, $\sigma = 2$ 63
35 Extremas of window 1, $\sigma = 4$ 64
36 Extremas of window 1, $\sigma = 2$ 64
37 Extremas of window 2, $\sigma = 4$ 65
38 Extremas of window 2, $\sigma = 2$ 65
39 Extremas of window 3, $\sigma = 4$ 66
40 Extremas of window 3, $\sigma = 2$ 66
41 Reconstructed contour map of window 1 from zero-crossings with $\sigma = 2$ 67
42 Reconstructed contour map of window 2 from zero-crossings with $\sigma = 2$ 67
43 Reconstructed contour map of window 3 from zero-crossings with $\sigma = 2$ 68
44 Reconstructed contour map of window 1 from extremas with $\sigma = 4$ 68
45 Reconstructed contour map of window 1 from extremas with $\sigma = 2$ 69
46 Reconstructed contour map of window 2 from extremas with $\sigma = 4$ 69
47 Reconstructed contour map of window 2 from extremas with $\sigma = 2$ 70
48 Reconstructed contour map of window 3 from extremas with $\sigma = 4$ 70
49 Reconstructed contour map of window 3 from extremas with $\sigma = 2$ 72
50 Manual breaklines map of window 1 73
51 Manual breaklines map of window 2 74
52 Manual breaklines map of window 3 75

CHAPTER 1
INTRODUCTION

Spatial information is important in describing the geometrical shape of the landform. Spatial information can be represented in many ways such as contour lines or hill shading on a map. Such hard copy representations also serve as the means for storing spatial data. As a result of improvements in computer technology, the methods of storing and using spatial information are changing. Computer maps are becoming prevalent, often replacing the conventional paper or hard copy map. Digital data are no longer new. Spatial information is now stored in digital form, so that computer technology can be utilized. The general term used for digital characterization of terrain relief is Digital Elevation Model (DEM). Since their inception in the 1950's, DEMs have been widely developed and implemented. Many techniques have been designed to generate and capture DEM data, varieties of sampling patterns have been presented and many new problems have been addressed.

In order to describe terrain relief, a vast amount of data must be used, especially in the case of rough terrain. The effort to cut down the amount of DEM data has been realized for a long time, and several techniques have been proposed for its solution. Representing terrain relief by a mathematical model is one way to solve this problem. The amount of DEM data can be dramatically reduced by this continuous form of surface description. For the discrete form of DEM, varieties of sampling patterns as well as Triangulated Irregular Network (TIN) are suggested. The TIN requires less data than a regular grid sampling method. However, all of the proposed techniques suffer from a common drawback. It is the problem of extracting significant phenomena from the surface or raw data. These phenomena are generally known as the points on the surface which have relatively large elevation changes. In the geomorphological community, many terms such as pit, peak, ridge, valley, etc. have been used to represent the phenomena. However, the general terms of *breakline* and *breakpoint* are widely accepted, in the surveying and mapping fields. It has long been recognized that they are the essential entities to describe the surface's information. These breakpoints and breaklines must be extracted as the boundary lines of the region defined by mathematical models or sampled by any sampling method. It is also required that these phenomena must be detected and exploited as a node in the TIN technique in order to optimize the technique. The process of breakpoint and breakline detection is generally required as a part of the sampling techniques.

The amount of DEM data has become an important issue when automatic DEM data generation and a highly accurate DEM data base is proposed. The National Cartographic Data Base idea presented by the United States Geological Survey (USGS) [Light 86] will be unattainable if this problem cannot be solved.

Representing a surface by a set of breakpoints and breaklines is a new way recently suggested in [Douglas 87]. It is obvious that breakpoints and breaklines of the surface are the minimum number of entities required for surface description. However, extracting breakpoints and breaklines from the surface is not a trivial task. It was estimated that more than half of DEM work cost was breakline related [Kubik and Roy 86]. At the present time, manual detection is still employed. Another way of viewing the extraction of breakpoints and breaklines is as a procedure for DEM data compression. The concept of DEM data compression and breakline detection should be reconsidered and an automatic technique needs to be developed.

Recently, the results of an edge detection operator in computer vision have been applied to create a new dimension of photogrammetric and cartographic research [Greenfeld 87], [Thapa 87]. [Schenk 87] proposed to utilize the Laplacian of Gaussian (LoG) edge operator to automatically extract breakpoints and breaklines from the raw data.

1.1 Scope and Objectives of the Research

This research addresses the problem mentioned above and has the following objectives:

- to study and apply the Laplacian of Gaussian (LoG) edge operator for automatic regular grid DEM data compression,
- to evaluate and test a newly proposed technique on the regular grid DEM data of different surface types signifying whether the proposed

technique can automatically compress, without significant loss of information, the data for any surface type.

DEM regular grid data should be regarded as the gray levels of an image pixel. From this point of view, breaklines on a surface should correspond to edges on an image. By means of the LoG operator, the automatic regular grid DEM data compression or automatic breakline detection can be achieved. Breaklines can be detected by considering the zero-crossings and the points having the minimum and maximum of the convolution of the second derivative of the Gaussian function (Laplacian of Gaussian, LoG) with the signal derived from the terrain height at every grid point.

In order to conduct this research, a few constraints must be imposed on DEM data. In this research, the following assumptions are made:

- The regular grid DEM data is available and dense enough to contain all the surface information such as breaklines and breakpoints.
- The application of DEM and the Nyquist sampling theorem govern the grid sampling density. This is not considered in this research. Therefore, all breakpoints and breaklines extracted from the data are restricted to the DEM grid density.

1.2 Organization of the Report

In Chapter 2, different techniques to reduce the amount of DEM data are presented. Particularly, DEM data sampling and TIN techniques are described. The idea of data compaction is also included. Both strengths and weaknesses of each method are presented. Chapter 3 is devoted to the edge detection operator. Three basic operations, smoothing, differentiation and detection are emphasized so that the edge detecting operator can be appropriately applied. In addition, two types of edge operator, directional and non-directional, are also discussed. The LoG is presented in detail, including its drawbacks. In Chapter 4, a new idea of DEM data compression by breakline detection is presented. The proposed DEM data compression strategy is also included in Chapter 4. Chapter 5 is dedicated to the experiments and results. Test data sets, testing procedure, results and analysis of the results are presented. The last chapter presents conclusions and recommendations. An Appendix presenting the derivation of the relationship between spatial and frequency parameters of the Gaussian function is also enclosed.

CHAPTER 2
DEM DATA COMPRESSION

2.1 Introduction

Digital Elevation Model (DEM) was created initially for route surveying and design by Prof. Charles L. Miller at the Massachusetts Institute of Technology (MIT) in the 1950's [Doyle 78]. Originally, the term Digital Terrain Model (DTM) was used. Later the term DEM became widely accepted for general surface description by many organizations in the surveying and mapping community, such as USGS [Allder et al. 82]. The DEM, which can be defined as the digital representation of the surface's relief, can be represented by a discrete or a continuous form [Burrough 86], [Mark 78].

In its discrete form, DEM can be represented by the set of X,Y,Z coordinates of limited discrete points. These can be obtained from map digitizing, ground surveying or photogrammetric systems [Loon 85]. Accuracy of DEM is governed by the location of sampled points and method of interpolation. Ideally, the density and location of the discrete points should be such that the terrain can be reconstructed from the stored DEM data.

In addition to the discrete form, DEM data can also be portrayed continuously using mathematical models. Several models have been developed and utilized in many applications, such as covariance, variogram, spectrum and fractals [Kubik and Roy 86].

During the DEM development, many problems have been addressed, and a variety of solutions have been proposed. One widely discussed problem is the amount of DEM data. It has long been recognized that the huge amount of DEM data is a major shortcoming in the process of data storage and data processing. This problem has been seriously considered in the formation of the National Digital Cartographic Database, and in the proposed creation of high accuracy DEM database [Allder et al. 82], [Kubik and Roy 86]. It is estimated that more than half of the 10^{14} bits of storage space required for a digital mapping system containing the 54,000 quadrangles covering the

US at 1:24,000 scale is DEM data [Light 86]. In addition, the invention and application of automatic and semi-automatic DEM data acquisition techniques such as stereo correlation [Claus 84], image matching [Makarovic 84a], [Makarovic 84b] and Gestalt Photo Mapper II [Elassal 78] has increased the significance of this problem.

In order to solve this problem, a variety of approaches have been proposed. These include different techniques of DEM data sampling, representation and data compression. The next four sections of this chapter will be devoted to a review of these approaches.

2.2 DEM Data Sampling

As mentioned above, DEM data can be obtained by many techniques such as digitizing contour lines, ground surveying or photogrammetric system. In this research, a variety of sampling techniques in photogrammetric system will be covered due to the fact that the photogrammetric method is more efficient than any other method.

Generally, DEM data are sampled in the grid pattern fashion, with density matched to the local roughness of the terrain relief. The sampling process is usually designed to work automatically or semi-automatically. To understand the merits and drawbacks of DEM data sampling, three well-known sampling techniques, Progressive Sampling [Makarovic 73], Composite Sampling [Makarovic 77a] and Optimum Sampling [Ayeni 82], will be reviewed.

2.2.1 Progressive Sampling

Progressive Sampling was created by Makarovic [Makarovic 73] to acquire DEM data in a regular grid form of varying density. In order to proceed with this sampling technique, the terrain classification as well as sampling density for each terrain class must first be determined. Then, the process initially samples (zero run) at low density, and the sampled data are used to compute second differences between grid points. After that, the second differences are compared to a predetermined threshold value. If the differences exceed the threshold value, the sampling density in that area must be increased for the next run. In other words, curvature analysis is used to determine whether the grid needs to be locally densified. This procedure is repeated in successive sampling runs until the local density of the grid matches the terrain roughness.

The Progressive Sampling technique has some advantages which should be addressed. First, this method was one of the first to be used in an automatic system [Hahn and Förstner 88], because the sampling is not difficult to automate. Second, the storage requirements of this sampling method are much less than full grid measurement, if the terrain has only a few breaklines and is quite smooth. However, if the terrain has a lot of breaklines or features, the method will not give satisfactory results.

To provide a complete analysis of the method, its shortcomings should be considered. First, in order to implement this technique, terrain classification of the surface must be accomplished. Unfortunately, there does not exist an acceptable terrain classification standard [Roy 87]. Different applications use the different criteria for terrain classification. Terrain classification is the application oriented task [Kubik and Roy 86].

The second drawback of this method is the lack of capability to capture significant features lying within a grid cell. Since the differences used to control the grid density are computed from the grid intersections, the relevant features lying inside a grid cell will be overlooked. To avoid this problem, high density grid can be applied. This solution, however, will produce more redundant data.

Finally, the grid sampled points of this technique are not the minimum set of points representing the terrain information. As several authors have indicated [Hahn and Förstner 88], [Peucker et al. 78], terrain relief should be described by points sampled along the morphological features such as valley lines and ridge lines rather than in a grid pattern. Therefore, if this technique is implemented with low density grid, the characteristic of terrain cannot be extracted. However, if high grid density is employed, the sampled points will contain redundant data.

2.2.2 Composite Sampling

This sampling method was presented by [Makarovic 77a] to add more terrain information and reduce sampled points of the Progressive Sampling technique. In this method, information on relevant morphological features such as breaklines and breakpoints is included. These features are manually extracted from the terrain to form the boundary lines of homogeneous regions so that the Progressive Sampling can be applied within. We can see that more terrain information is extracted in this method than in the Progressive Sampling. The manual breaklines and breakpoints extraction are the only difference between these two methods. At the present time, the Composite Sampling strategy is the technique widely used in photogrammetric practice to obtain DEM data [Hahn and Förstner 88].

Even though the Composite Sampling method was developed to add more terrain information into the Progressive Sampling, it suffers from the same drawbacks held by Progressive Sampling. Terrain classification is still required, and the sampling pattern is still in the form of a grid pattern. Furthermore, this method produces a new drawback which can be considered a critical one. Extracting significant features such as breakpoints and breaklines is a costly procedure and is difficult to automate. Michael A. G. Toomey estimated that 60% of the cost of DEM was breakline associated [Kubik and Roy 86]. Breakline extraction is a subjective process which is difficult to standardize and automate.

2.2.3 Optimum Sampling

[Ayeni 82] proposed a new method of DEM data sampling. This method uses *Harmonic Vector Magnitude (HVM)* as a parameter to classify the terrain roughness and control the sampling grid density.

The optimum sampling method utilizes the least squares technique to fit a combination of two mathematical functions, i. e. a Linear Filter and Double Fourier Function, to a surface. The resulting coefficients from the Fourier Function are used to measure the deviation of the surface from the linear plane. Then the value of HVM, which Ayeni defined as:

> ...*the square root of the sum of squares of coefficients for terms of specified m and n harmonics,*

is used as the parameter for terrain classification and for determining the optimum sample size. Technically, HVM is nothing but the value of square root of energy at certain harmonics. Ayeni exploited this quantity to measure the roughness of terrain deviating from the fitting plane. HVM is combined with the concept of Progressive Sampling to choose an optimum sampling grid size for different terrain types. The empirical linear regression was also presented by Ayeni to determine the optimum sample size instead of actually computing HVM value.

This method has one merit which is worth noting. Terrain classification and optimum sample size of the optimum sampling technique can be determined automatically by considering the value of HVM. Recently, [Balce 86] has shown the implementation of this method as well as the Progressive Sampling on real data. Guidelines and recommended strategy were also given by Balce for DEM data sampling in practice.

It is very important, however, to note that the correlation function which is used to determine the optimum sample size of this sampling technique is a *stationary function*. [Hardy 77] indicated that the variations of terrain are not necessarily stationary random functions. Therefore, the sample size obtained from correlation function may not be the correct optimum sample size. In addition, this method still utilizes a grid sampling pattern, which as stated earlier, cannot give the minimum number of points to describe the surface information.

2.3 Triangulated Irregular Network (TIN)

As mentioned above, sampling data by grid pattern cannot completely extract terrain information. Accordingly, many sampling patterns have been proposed to overcome this problem. [Morrison 71] presented six standard sampling patterns which are well known in the cartographic community, namely:

- Unaligned Random
- Aligned Random
- Unaligned Stratified
- Aligned Stratified

- Unaligned Systematic
- Aligned Systematic

The purpose of these sampling patterns is to extract more morphological features from terrain. The sampled points can be located randomly. They are not necessarily in the grid pattern. These sampling patterns, however, are difficult to implement in an automated way.

In 1977 [Peucker et al. 77] presented a new method of terrain representation by using a set of Triangulated Irregular Networks (TIN). TIN was developed by considering two important factors, the non-stationary property of terrain roughness and the real representation of terrain [Peucker et al. 78]. TIN is the irregular discrete method of DEM representation. Data points are captured in the form of Triangulated Irregular Networks (TIN) in which every node of TIN is sampled at the point so that the triangle forms the terrain facet. These triangulations are usually known as ***Delaunauy Triangulations.*** The triangulations are formed by the following criteria:

- all three angles are as nearly equal as possible
- the longest sides of the triangles are as short as possible

The first step in the TIN method is to determine the ***Thiessen Polygons.*** In a field of scattered points of Figure 1(a), there is an irregular polygon surrounding each point such that every location within the polygon is closer to the enclosed point than it is to any other. This polygon is called the Thiessen Polygon. Surrounding the Thiessen Polygon enclosing point A are other Thiessen Polygons which enclose the Thiessen Neighbors of A. The triangle network connecting point A with its Thiessen Neighbors is the Delaunay Triangle as shown in Figure 1(b). For an arrangement of points, both Thiessen Polygons and Delaunay Triangles are unique. In order to obtain a good terrain description, the discrete data points used in TIN must be collected at the points where significant changes occur, such as along ridge line. These points are rich in information and can be viewed as breakpoints or breaklines, which will be defined later in Chapter 4. This promising method has created a lot of interest in the surveying and mapping community. Many

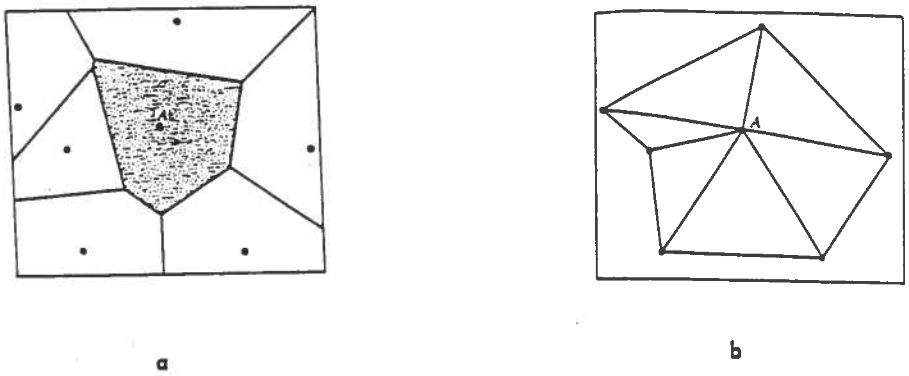

Figure 1: TIN method. (a). The Thiessen Polygons, (b). The Delaunauy Triangulations [Davis 86]

contouring packages which are available on the commercial market at the present time utilize the TIN technique.

There are two merits of this DEM representation which should be pointed out:

- This technique is able to represent more terrain information than the regular grid pattern presented in the previous sections. This is due to the formation of irregular triangles which can be viewed as the network of homogeneous terrain tiles.
- This method requires less storage space than the grid sampling pattern. As stated earlier, all sampled points in TIN should be the points where terrain discontinuities occur instead of at grid intersection. As a result, redundant sampled points from a grid pattern can be eliminated.

It seems as if TIN is the most appropriate approach to solve the problem of the huge amount of DEM data. However, this method has some deficiencies which are worth noting.

- To obtain a good terrain description, the sampled points must be collected so that they consist of all significant terrain discontinuities. It is obvious that if sampled points are not located at the discontinuity points, TIN will give the wrong surface description. Unfortunately, collecting these phenomena are costly and difficult to automate; therefore, one must be aware of exploiting this method.

- Data structure in the data-base is much more complicated than regular DEM data. In the TIN technique, pointers must be used to keep track of the connectivity of triangles of the whole surface.

- The TIN has limits in adaptability. [Douglas 87] has shown that TIN needs the transition density at the region where the terrain changes from one type to another type. This means that TIN creates some redundant triangles in the transition area.

- The graphic produced from TIN, such as a contour line, can be distorted by the effect of construction method [Douglas 87].

2.4 DEM Representation by Mathematical Models

In addition to the discrete form, DEM can also be represented in a continuous form. Many mathematical models have been used to describe the terrain relief. The quantities used in this technique are the sets of mathematical coefficients or parameters derived from the raw data.

Under this type of DEM representation raw data must be obtained first. Then the terrain roughness is modelled by means of mathematical models. Different mathematical models have different advantages and are applicable in different fields. While a geologist prefers to use the spectrum to model the terrain, a geomorphologist prefers to use the covariance. After the terrain is modelled, the only quantity needed to describe a DEM surface is a set of mathematical parameters or coefficients. Even though this technique requires much less data to store and represent the DEM surface, its drawbacks should not be overlooked. These drawbacks are :

- Due to the limitation of interpolation techniques, mathematical models cannot be defined across breaklines. This means that breaklines must be collected as the boundaries of the homogeneous regions defined by mathematical models. However, as mentioned above, breakline extraction is difficult to standardize and automate because of the subjective and scale dependent characteristic of the breakline.

- Information loss can occur during the process of modelling the surface due to the truncation of the number of coefficients or due to the non-stationary property of terrain variations. Practically, a limited number of coefficients is used. Therefore, the terrain representation by mathematical models may be slightly distorted from the real surface by the truncated parameters. In addition, representing terrain variations, which are not necessarily stationary random functions, by periodic functions can cause information loss in this technique.
- The models are application oriented. As one model may be suitable for a specific application it may not be suitable for another applications. There does not exist any one mathematical model satisfying all applications. So, many mathematical models and standards have been proposed.

2.5 DEM Data Compression

Other than the three approaches presented above, namely,

- Different sampling techniques
- TIN representation of DEM surface
- Mathematical models representing the DEM surface,

the amount of DEM data can also be reduced by techniques of data compression. As [Hahn and Förstner 88] said,

> *This last strategy is the most promising, as it in principle uses all available information and conceptually is closest to morphological sampling ...*

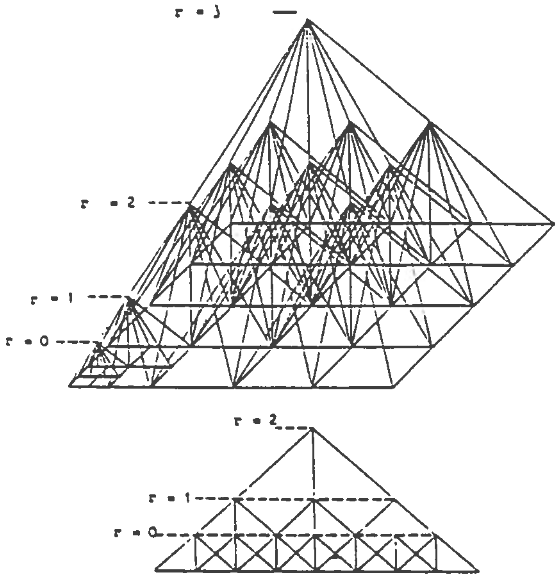

Figure 2: The principle of Makarovic's data compression [Makarovic 77b]

The data compression approach should be considered as the most appropriate method for automatic DEM data acquisition, because dense DEM data can be automatically obtained by many newly developed techniques such as image matching [Makarovic 84a], [Makarovic 84b] and stereo correlation [Claus 84]. Actually, data compression or data compaction has long been widely used in the non-spatial data community such as data communication [Clarke 85], but a few DEM data compression techniques have been exercised. In general, there are two techniques being considered to be under DEM data compression. The first one is the technique of extracting relevant surface information from dense DEM data. This information is explicit information which has more meaning than grid DEM data which is implicit. The second one is the technique of physical compaction of DEM data in the short form, such as binary or integer format, or by using techniques from data communication theories [Roy 87].

To extract the surface information from dense DEM data, [Makarovic 77b] and [Makarovic 83] used the inverse procedure of Progressive Sampling to compress the regular grid DEM data. Figure 2 shows the concept of Makarovic's data compression. The second differences between the grid intersections are computed and compared with the predetermined threshold value. These differences are used to control the grid density for data sampling. If their values are smaller than the threshold value, the grid density will be reduced. The procedure is executed in a successive manner until the differences are larger than the threshold value. [Makarovic 83] pointed out that this method is influenced by the terrain roughness, grid density and threshold values. It is important to note that the shortcoming of the grid sampling pattern is still not eliminated in this technique.

Recently, [Hahn and Förstner 88] have utilized techniques like Feature Base Matching (FBM) and Least Squares Matching (LSM) to form a dense grid of the terrain. Then, the number of DEM data is reduced and extracted based on a grid laid over the terrain. The points within each groundel (non-overlapping ground element) are reduced to one point. This point is considered as a representative of that groundel. The reduction is done by the technique of Median Filtering of the height. The results of this technique are coordinates of every groundel. The data obtained from this technique are still in grid pattern and no relevant terrain characteristics, such as breaklines and breakpoints, are extracted. Moreover, this technique requires the epipolar condition to form the terrain model and yields good results in smooth terrain and well textured images only. This is one of the newest approaches presented in the photogrammetric community.

The other technique, which can be categorized under DEM data compression, is a physical compression method of DEM data in computer media. This technique sometimes is called DEM data compaction. In order to reduce the storage space, DEM data can be stored in the form of integer or binary values instead of real values which occupy more space. In general, this technique extracts the surface information from the raw data first, then stores the extracted data in the short form. The extracted data are obtained by some criteria from other fields such as data communication, which data characteristics have been well known. However, these characteristics may not necessarily be valid for the terrain. [Roy 87] utilized the data communication theories to compress DEM data. This technique gives a satisfactory result, but it still needs to be further investigated in terms of noise problems and compression on different terrain types.

At this time, there does not exist a satisfactory DEM data compression technique which can give a good surface description and work automatically on every terrain type. The existing DEM data compression techniques do not extract and save the important characteristics of the surface (this will be discussed later in Chapter 4) from the raw data. Instead, they compress DEM data by using some arbitrary criteria and terrain classification, for which there does not exist an acceptable standard yet. Automatic DEM data compression on every terrain type cannot be achieved, as long as human intervention is needed for terrain classification and the matching techniques do not yield a good result on every terrain type.

CHAPTER 3

EDGE DETECTION OPERATORS

3.1 Introduction

It has long been recognized that physical object boundaries are important phenomena in the human visual and recognition systems [Attneave 54]. Psychological studies about the human visual and recognition process have a lot of impact on the research of image analysis. In order to develop an intelligent system for image understanding, fundamental phenomena in human vision such as object boundaries must be obtained. "Edge" is the word used to describe object boundaries on an image, which is considered as an essential phenomenon for human vision. Due to the importance of edges, extensive work has been done to extract edges from an image.

In general, boundaries of an object in a scene correspond to intensity changes on the image. To extract boundaries of an object, the intensity changes must be extracted. However, not only do object boundaries cause intensity changes, the reflection of object surfaces, illumination of light sources and image's view point are also factors which influence the occurrence of intensity changes on the image [Marr and Hildreth 80]. Therefore, the intensity changes on the image may not correspond to the boundaries of objects.

To extract edges from the intensity changes, a variety of techniques have been proposed based on a combination of studies from many disciplines, for instance, psychological studies, image processing and artificial intelligence research. In addition to the software development, new hardware was also designed to increase the capability of new edge detection techniques, e. g. convolver [Nishihara and Larson 81]. After edges have been detected, they must be grouped to form a meaningful description of the scene on the image.

In the next sections, the most common operations in edge detection will be described. Then two types of edge detection operators will be presented later in this chapter.

3.2 Edge Detection Operations

In the early days, edges were detected by using a simple technique, such as an arbitrary threshold of grey level, or simple implementation of gradient function [Roberts 65] to locate the intensity changes in an image. These types of edge detection techniques are not applicable to an image of the physical real world where intensity changes are complicated and do not correspond to the edges of objects. Therefore, a sophisticated edge detection technique must be employed.

In order to design an edge detection operator, one needs to understand the characteristics of edges appearing on an image. [Hildreth 83] stated that

> *Changes in intensity will occur in the image at a range of different scales. If we look at individual image elements(pixels), we find intensity changing from pixel to pixel. Often, there will be uniform changes over some distance. Most edges in the real world are sharp edges: the intensity function will be composed of a few steep changes over a small number of pixels.*

This means that an operator which detects intensity changes at one specific scale is not suitable to detect changes in the real world. A multi-scale operator is needed for detecting intensity changes occurring at finer and coarser resolutions. This multi-scale property is the essential characteristic which the edge operator must hold [Marr and Hildreth 80].

Generally, edge detection operators consist of three basic operations:

- Smoothing Operation
- Differentiation Operation
- Detecting Operation

The following three sections will cover all these operations in detail so that the operations can be further implemented in another field such as photogrammetry or cartography.

3.2.1 Smoothing Operation

The intensity changes appearing on an image create the stimuli for object recognition by a viewer. These intensity changes do not necessarily correspond to the edges of objects. For example, intensity changes may be caused by the shadows or orientation of the object. Fortunately, intensity changes due to edges of objects are usually steep. In order to reduce the effect of noise in the intensity changes, a smoothing function must be applied to the image intensities. This operation may be considered as a digital filtering process.

The reason why the image must be smoothed can be viewed from the property of a differentiation operator which is usually used for detecting the intensity changes. As described in [Rosenfeld and Kak 82], differentiation is a noise enhancing operator, forcing an image to be smoothed prior to differentiation. As a result of the smoothing operation, all minor intensity fluctuations will be filtered out. This is the first objective of a smoothing operation in an edge detection procedure.

Secondly, the smoothing operation is used to control the resolution at which edges will be detected. This means that if an image is smoothed at a coarse resolution, the detected edges will be the coarse edges appearing on the image. In other words, the image will be blurred by the coarse scale smoothing function and coarse edges will be extracted. On the other hand, if the image is smoothed by a fine scale smoothing function, the detailed edges can be extracted.

In order to smooth the image, an arbitrary smoothing function cannot be used. The shape, size and properties of a smoothing function or filter must be taken into consideration because of two reasons:

- Since the changes in the real world are generally localized in space [Hildreth 83], a smoothing function must be spatially limited.
- If band limiting in frequency is used to restrict the scale at which edges will be detected, the band limiting generally will conflict with the spatially localized property above.

Therefore, the smoothing function which will be used in the edge detection operator must optimize these two constraints. It is interesting to note at this point that Gaussian Density Function, which is defined as:

$$G(r) = \frac{1}{\sigma} \exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{1}$$

where :

r = the distance from the center of the operator
σ = the space constant of the Gaussian,

is the only function minimizing the product of bandwidth in both spatial and frequency domains [Marr and Hildreth 80]. This can be seen by the fact that the Fourier Transform of the Gaussian Function is also the Gaussian Function.

3.2.2 Differentiation Operation

After an image is smoothed, the next step in edge detection operation is to determine what pixels correspond to the intensity changes on the image. A differentiation technique is utilized in this operation due to the fact that intensity changes correspond to a peak (maxima or minima) in the first derivative or a zero-crossing in the second derivative. The choice of using the first or second derivative to detect intensity change is still arguable.

As stated in [Ulupinar and Medioni 87],

> *...all detectors using first derivative require a thinning step, since an edge responds with a broad peak.*

In order to avoid the above problem, the second derivative is widely used.

In addition to first and second order, differentiation can also be classified by its direction. Both directional and non-directional derivatives are used in several edge operators. The dispute between these two derivatives about which one is optimal is far from settled.

In this research, directional and non-directional derivatives will be utilized to classify edge detection operators.

In the directional technique, a derivative can be taken in some specific directions such as the direction of the gradient, the perpendicular direction to the edge orientation. It is not necessary that two perpendicular directions must be taken. Derivatives may be taken in more than two arbitrary directions.

In contrast to the directional derivative, a non-directional derivative can be achieved regardless of what the orientations of the edges are. The details, advantages and drawbacks of these two techniques are worth studying and will be presented later in Sections 3.3.1 and 3.3.2.

3.2.3 Detecting Operation

As a result of the differentiation operation, pixels corresponding to intensity changes are determined. These pixels represent information of the image in which the viewer is interested. Nevertheless, the scattered pixels cannot give a meaningful description. An operation to group and draw lines connecting the pixels to form the boundaries of objects in the scene is required. This process is called *Line Segmentation*. Line segmentation is another field of research in which many topics are still open. The result of this process is the image information described by sets of points and lines. This sketch was called *Primal Sketch* of the image, initially created by [Marr 82]. The primal sketch of an image is an important entity for further image analysis, such as motion and stereo disparity.

Generally, line segmentation consists of two basic operations. First is the process of grouping and connecting lines between the edge pixels. Secondly, after jagged lines are obtained a smoothing process must be applied. The second operation can be considered analogously as cartographic generalization.

In summary, an edge detection operator consists of three fundamental operations, namely: smoothing, differentiation and detection. Each of these operations can be done by several techniques, but the characteristic of edges appearing on the image and the knowledge of scene are the constraints which we must take into consideration to choose what techniques should be applied.

3.3 Edge Detection Operators

In order to present edge detection operators, some kind of classification must be used to categorize or group the operators. There exists several methods to classify edge detection operators. Generally, the derivative technique is utilized as the primary entity to categorize edge operators. Under this research two types of differentiation, namely directional and non-directional differentiation, will be used.

3.3.1 Directional Edge Detection Operators

In the early days of edge detection, edge operators used directional first and second derivatives [Davis 75], [Marr and Poggio 79]. For these types of operators, derivatives can be taken in as many directions as we want. According to [Hildreth 85], even derivatives in two directions of vertical and horizontal are theoretically sufficient to detect edges at all directions. Several edge operators utilize directional differentiation in many orientations, such as Persoon's operator and Canny's operator [Persoon 76], [Canny 83]. The directions which edge detection operators use most are:

- the perpendicular direction to edge orientation [Marr and Poggio 79]
- parallel to edge orientation [Marr and Poggio 79]
- the direction of gradient [Canny 83], [Haralick 84].

The advantage of this type of edge detection operator is that more precise edge location can be obtained by this method. As indicated by several authors, directional operators give more precise edge locations than non-directional operators, especially when edge orientation is not linear; see for example [Marr and Hildreth 80], [Ulupinar and Medioni 87], [Hildreth 83]. The reason behind this advantage will be explained later in section 3.3.2. This is the most advantage a directional operator has over a non-directional operator.

As [Torre and Poggio 86] said,

> *...the choice between rotationally invariant operators (rotational filters and rotational invariant differential* RID *operators, as the Laplacian or the second derivative in the direction of gradient) or directional operators (directional filters and directional differential* DD *operators such as directional derivatives) depends on the subsequent information processing task.* RID *operators ensure closed edge contours, that are not provided in general by* DD *operators.*

While the directional operator is preferred over the isotropic operator in terms of detecting edge locations, it holds some drawbacks which should not be overlooked. First, the derivative direction, which is required in the differentiation operation, must be determined. This adds another computation step. In addition, when the cross-sections of the operator are different, more computation time must be required. Finally, the smoothing operation and differentiation operation cannot be combined into one step, as is feasible in the non-directional operator. This can be simply explained by the fact that the smoothing operator is a scalar quantity while the gradient, defined as follows, is a vector operator.

$$\vec{\nabla} = \left(\frac{\partial}{\partial X}\vec{X}, \frac{\partial}{\partial Y}\vec{Y}\right) \tag{2}$$

Generally speaking, this method is computationally more expensive than rotational invariant operators.

3.3.2 Non-directional Edge Detection Operators

To avoid the need to calculate directional derivatives in a variety of directions at each point, non-directional differentiation is utilized. The *Laplacian Operator* ∇^2 which is defined as follows :

$$\nabla^2 = \frac{\partial^2}{\partial X^2} + \frac{\partial^2}{\partial Y^2}, \tag{3}$$

is chosen to approximate the second directional derivative because its value is independent of the orientation of the coordinate system. Actually, Laplacian is the *divergence of gradient* or dot product of two gradients, i.e.,

$$\nabla^2 = div(\vec{\nabla}). \tag{4}$$

Therefore, the Laplacian is a scalar operator and it is easy to manipulate or combine with a smoothing operator, which we will see later. The *Laplacian of Gaussian (LoG)* is a well known non-directional operator utilizing the advantages of Gaussian function. It will be discussed in the next three sections.

3.3.2.1 Laplacian of Gaussian (LoG)

The Laplacian of Gaussian (LoG) operator is the result of a combination of a Laplacian Operator (∇^2) and a Gaussian function. LoG utilizes the essential property of the Gaussian function, localization in both the spatial and frequency domain. LoG can be derived by applying the Laplacian operator to the Gaussian function (see Equation 1). For the two-dimensional case,

$$LoG = \nabla^2 G(r) = \frac{1}{\sigma^4}\left(\frac{r^2}{\sigma^2} - 2\right)\exp\left(-\frac{r^2}{2\sigma^2}\right), \tag{5}$$

where G(r), r and σ are the quantities defined earlier in Equation 1. Theoretically, LoG operator is infinite in size. However, it is computationally impossible to use an infinite size operator. Therefore, the certain support or size of operator must be implemented. Figure 3 shows the shape of LoG with finite 3W support in two-dimensional space. W in the figure is the diameter of the inner circle of LoG or excitatory region and is related to σ by:

$$W = 2\sqrt{2}\sigma. \tag{6}$$

The LoG operator simplifies the computation of the smoothing and differentiation operation, due to the fact that the Laplacian is a scalar operator. This can be shown as follows:

Given:

I is the intensity function of the image.

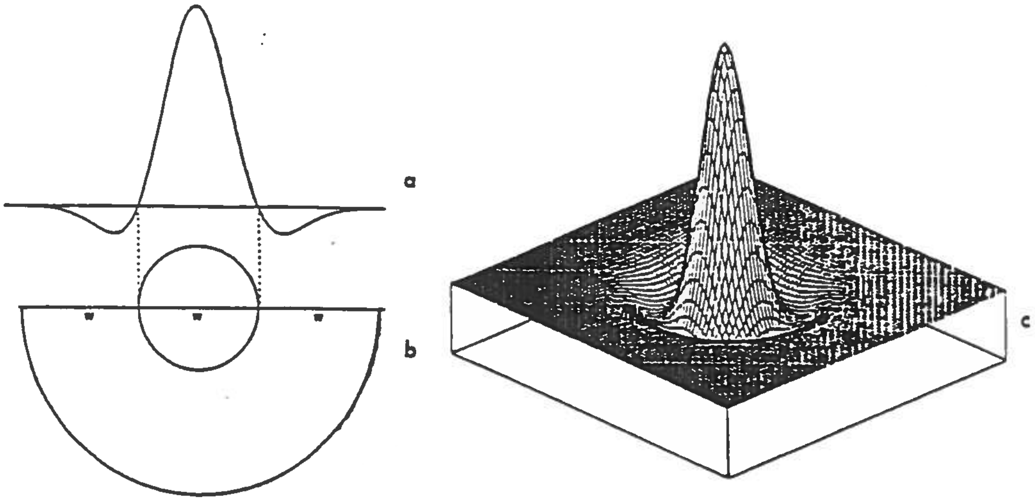

Figure 3: The LoG Function. (a). The profile of LoG, (b). LoG viewed from the top, (c). 3-D view of LoG [Greenfeld 87].

G is the Gaussian function.
$*$ is the convolution operator.
∇^2 is the Laplacian operator.

Therefore:

$G * I$ is the smoothed image.
$\nabla^2(G * I)$ is the Laplacian of the smoothed image.

But, we can write:

$$\nabla^2 (G * I) = \left(\nabla^2 G\right) * I, \tag{7}$$

due to the property of convolution.

This property implies that the $\nabla^2 G$ operator can be convolved directly to the image instead of applying a smoothing function to the image and then differentiating the smoothed image. This is the first advantage which LoG has over directional operators.

Secondly, all convoluted values from LoG are scalar quantities. It is easier to manipulate scalar quantities of non-directional operators rather than vector quantities of directional operators.

Thirdly, zero-crossings from LoG are rich in information. Zero-crossings consist of important attributes such as slopes and directions, in addition to their locations. These attributes from the different filter sizes are the information which can be utilized to describe the original signal.[Yuille and Poggio 83] showed that the original signal can be completely reconstructed from the zero-crossings of two different filter sizes, by finger print theorem. As they said,

> *Our proof is constructive. It shows how the original signal can be reconstructed by information from the zero-crossings across the scales.*

This is the important advantage of LoG over other edge operators, especially when reconstruction of the signal is required.

Fourthly, multi-band filtering of LoG can be used to refine the position of edges. Due to the fact that LoG is a band-pass filter, LoG with large filter size will smooth out all detailed signals, and zero-crossings will be displaced from the true edge position. On the other hand, LoG with small filter sizes gives good localization as well as creates false zero-crossings because of noise or high frequency phenomena. Therefore, zero-crossings at the coarse scale should be used to determine the occurrence of edges, while precise location of edges can be obtained at the finer scale.

However, [Clark 88] showed that zero-crossings of LoG can produce both *phantom* and *authentic* edges if the shape of the signal is similar to double steps. Phantom edges are false edges which must be eliminated before further analysis can be done. This means that only authentic edges should be used in the image's primal sketch. In order to distinguish authentic edges from phanton edges, he suggested that the sign of the product of first and third derivatives should be used:

$$\chi = \nabla(\nabla^2 f) \cdot \nabla f \tag{8}$$

In this equation, f is the signal function. Clark stated that authentic edges correspond to the points having the negative value of χ. The positive value of χ reflects the positive or negative value of both first and third derivatives. Therefore, points having positive χ are phantom edges. When χ is zero, there is no edge. To precisely locate the real edges, one needs to trace the authentic zero-crossings through the scale. This is the spatial coincidence assumption suggested in [Marr and Hildreth 80].

In addition to the above advantages, LoG also has many properties and advantages which should be addressed.

1. The accuracy of zero-crossing locations can reach the sub-pixel level by interpolation between two neighboring edge pixels of the opposite sign [Huertas and Medioni 86].
2. The scale or resolution at which edges will be detected can be controlled by the value of σ or W.
3. The shape of LoG is similar to the shape of a circular symmetric filter of the ganglion cell of the receptive field at the retina, which human visual systems use to detect edges [Marr and Hildreth 80].

3.3.2.2 Drawbacks of LoG

Even though LoG holds several advantages, the drawbacks of LoG should not be neglected. As stated in [Ulupinar and Medioni 87]

- *Zero-crossings do* NOT *always correspond to actual edges, but to inflection points instead.*
- *Zero-crossings do not always indicate the true position of an edge.*

The inaccurately located edges of LoG operators are caused by two sources [Chen and Medioni 87], namely,

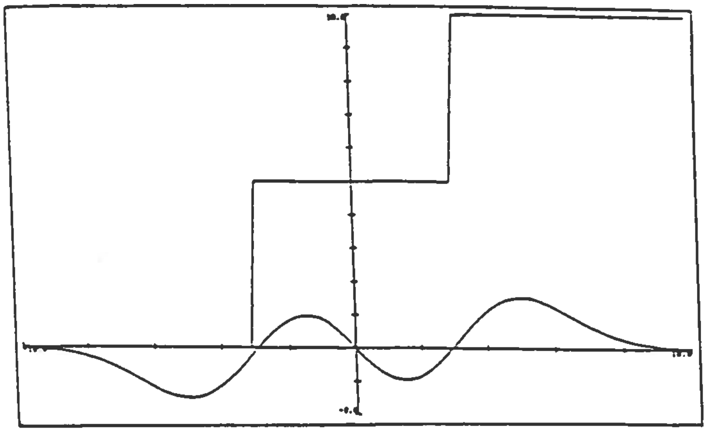

Figure 4: False zero-crossing of staircase edge created by LoG [Ulupinar and Medioni 87].

- the interaction between edges,
- the lack of symmetry of the edge profile.

This creates a false response of intensity changes as shown in Figure 4 by [Ulupinar and Medioni 87]. In addition to the false zero-crossings, LoG does not give a good edge position either. As shown in [Berzins 84], zero-crossings of LoG at a sharp edge corner can be displaced due to the errors introduced by approximating the directional derivative by the Laplacian. The error becomes larger for an acute corner. Approximating directional derivatives by a Laplacian can also cause the displacement of edges due to the non-linearity of the intensity change. As [Marr and Hildreth 80] say,

> ... *Zero Crossing of the second directional derivative, in the direction of the gradient, will coincide with zero crossings of the Laplacian of the image if the intensity variation of the image is linear along the line of the zero crossings.*

The size of the error depends on the magnitude of the second-directional derivative in the direction parallel to the edge. Spurious contours caused by the non-linearity of the intensity change actually are the phantom edges named in [Clark 89].

The next shortcoming of LoG which should be pointed out is the detected edges from LoG always form closed contours [Marr and Hildreth 80]. Therefore, at the place where there are an odd number of edges, LoG is unable to give the correct answer. The false edges will be created due to this deficiency.

To keep everything in perspective, the above problems should be resolved. [Berzins 84] has showed that displacements of edges by LoG is much less than σ provided that certain situations are fulfilled such as:

- the size of considered region is large compared to σ,
- the radius of curvature of the edge is large compared to σ,
- the distance to the nearest sharp corner is large compared to Θ/σ, when Θ is the angle of the corner in radians.

Nevertheless, the closed contour character of LoG is still there and it may create false edges. There is no way to solve this drawback of LoG. This shortcoming must be realized, if LoG is utilized. It is very important to state that the properties and drawbacks of LoG presented above are the constraints establishing the conditions under which it is valid to implement the LoG for edge detection or other applications. To implement LoG in any application, the properties as well as shortcomings must be kept in mind.

CHAPTER 4

DEM DATA COMPRESSION BY BREAKLINE DETECTION

4.1 General

In Chapter 2, a variety of solutions proposed by many authors has been presented for the problem of reducing the amount of DEM data. As mentioned in section 2.5, DEM data compression is the most promising strategy for automated DEM data acquisition. As a result, DEM data compression should be deeply investigated. Generally, DEM data compression can be defined as the process of reducing the amount of data by not losing the information. In order to accomplish DEM data compression, one must be clear about what information is important and should be extracted from the surface as surface representations. The information should give a completely meaningful geometric description of the surface. The process of extracting the explicit information from regular grid DEM data which are implicit information should be considered as a DEM data compression process. The explicit information obtained from data compression is the only entity needed to be saved as the DEM surface representation in the DEM data-base.

From this point of view, TIN and the method of representing the DEM surface by mathematical models can be considered as DEM data compression techniques. Modelling and representing the surface geometry by a set of mathematical models and parameters is one way to describe the geometrical shape of the continuous surface. Unfortunately, this method suffers from several drawbacks, as presented earlier in Chapter 2.

On the other hand, the TIN technique seems to be a promising candidate method for DEM data compression because it can extract and represent geometrical descriptions well. However, it also has some shortcomings of which one must be aware.

In this chapter, a new way of describing the surface information and a new DEM data compression technique will be presented.

4.2 Terrain Representation by Breakline Map

It has long been recognized that breaklines and breakpoints are the important entities in describing shape of terrain. As Gruen said [Kubik and Roy 86],

> *When it comes down to high precision DTM, ..., and the terrain will consist solely of breaklines and single points.*

The importance of breaklines and breakpoints can also be supported by the incorporation of manual breakline acquisition in the Composite Sampling technique [Makarovic 77a]. The TIN method described in Chapter 2, also exploits these features to optimize its performance.

Generally, breakpoints and breaklines correspond to significant phenomena on the terrain such as pits, peaks, ridges and channels. These features are important characteristics for geometrical surface description, especially in geomorphology and hydrology [O'Callaghan and Mark 84]. These phenomena are the explicit information which contain more information about the surface than other features.

Recently, [Douglas 87] presented a new type of DEM by locating ridges and channels. Under his approach, Douglas has proposed to represent the DEM surface by sets of pits, peaks, ridges and channels. These features can be obtained by the combination of the elaborate geomorphological definitions, such as pits, peaks, summits, etc. and several computation algorithms, e.g., the Susan Jensen algorithm and Douglas line reduction algorithms [Douglas 87]. The features are called *RICHLINES* which stands for *RIdge and CHannel and other information-rich LINES*. First, this approach needs to locate basins or hills by sorting all elevations of DEM data. Then the window size is selected to search for slope lines by considering the elevation of every point within the window. In order to accomplish this computation scheme, many logical algorithms must be programmed based on the definition of geomorphological terms mentioned above. This technique is in the experimental stage where many algorithms have been combined and examined. Several methods give the wrong result in isolating basins and hills from ridges and channels. To overcome this problem, Douglas devised a new algorithm requiring a series of tests which was determined to be too expensive [Douglas 87]. The concept of using RICHLINES as a DEM representation is very interesting. However, the procedures proposed by Douglas are very complicated and costly.

Due to the essence of RICHLINES presented above, the idea of DEM representation by a set of RICHLINES should be adopted. The generally accepted term *BREAKLINES* should be employed instead of RICHLINES, because both breaklines and RICHLINES represent the same phenomena on the surface. Therefore, under this research, *BREAKLINE MAP* is accepted as the real geometrical description of the surface.

A breakline map is a set of points and lines with their data (X,Y,Z coordinates), which correspond to breakpoints and breaklines on the original DEM surface. The definitions of both breakline and breakpoint will be presented later in section 4.3. A breakline map may be viewed analogously as a contour map. In a topographic map contour lines are very important features in describing terrain roughness. As a result, the DEM can be obtained by digitizing contour lines. However, contour lines are the derived quantities obtained from an interpolation technique. Contour lines are not actually seen by the human operator like planimetric features such as houses or roads. In a manual method, contour lines are drawn by the process of interpretation where personal preference is introduced. So, two sets of terrain descriptions may be achieved if two different operators who have different knowledge of the area are used to compile contour maps.

On the other hand, breakpoints and breaklines are not the derived features. They actually exist on the surface. Representing terrain roughness by the set of breakline is the real geometrical description of the surface. Nevertheless, the process of breaklines extraction suffers two majors drawbacks due to the natural characteristic of breaklines. First, breaklines possess a scale-dependent property. As we know, one line may be considered as a breakline at a high precision DEM, but it may not be the breakline at a low precision scale. This means that two different scale DEM's must be described by two different breakline maps. This problem has long been known as DEM surface generalization in the cartography community. It is very important to understand that DEM surface generalization and breakline detection are closely related but different. DEM surface generalization is the process of determining a breakline map at small scale from a breakline map at large scale. On the other hand, breakline detection is the process of extracting breaklines from the given raw data whose density is controlled by the final map scale.

Second, the breakline extraction process is a subjective task. According to the scale dependent character of breaklines, it is very difficult for an operator to decide which lines or points should be considered as breaklines or breakpoints. This subjective task is enormous if it is carried out manually.

To keep everything in perspective, two drawbacks of the breakline extraction process must be solved. Solid definitions of breaklines must be defined and an automatic procedure should be developed. The concept of using a breakline map as a DEM representation should be considered a DEM data compression process. Compressing DEM data by breakline detection and representing the DEM surface by a breakline map is the approach proposed under this research.

4.3 Breakline Detection by Edge Operators

In the last few years edge operators have played an important role in many automatic photogrammetric and cartographic applications. [Greenfeld 87] and [Thapa 87] utilized LoG for stereo matching in photogrammetry and for detecting critical points in line generalization, respectively. [Schenk 87] proposed the new application of LoG for DEM data compression. An edge operator, being used to detect the intensity changes on an image, can be implemented to detect elevation changes on DEM surface as well.

To implement an edge operator in DEM breakline detection, one should view regular grid DEM data as intensity values of every pixel on the image. Accordingly, the elevation value of each grid point on the DEM surface corresponds to the grey level of each pixel on the image. From this proposed approach, points having significant elevation changes can be automatically detected by means of an edge operator.

Before pursuing the details of breakline detection by edge operators, the existing definition of breaklines should be reviewed. There have been some definitions of breaklines and breakpoints defined by different authors such as:

> *...a characteristic point is a singular point or one where there is a significant change of slope.* [Collins and Moon 81]

> *These breaklines represent discontinuities in the terrain.* [Kubik and Roy 86]

> *A terrain breakline has been defined as a line where there is a sudden or abrupt change in slope.* [Ayeni 76]

There does not exist the solid definition of breaklines which is suitable for this research. Therefore, in this research, breaklines and breakpoints are defined as follows:

A BREAKLINE is the loci of breakpoints.

A BREAKPOINT is a point having local maximum curvature of the surface, where curvature is defined as:

$$\kappa = \frac{1}{\rho} = \left\| \frac{d^2y/dx^2}{\{1 + (dy/dx)^2\}^{\frac{3}{2}}} \right\| \tag{9}$$

and ρ is radius of curvature. When dy/dx is small, we can write

$$\kappa = \frac{1}{\rho} \approx \left\| d^2y/dx^2 \right\| \tag{10}$$

From the above definition, breaklines can be automatically detected by using elevation data as the input signal to edge operators.

4.3.1 Detecting Breaklines by LoG

As presented in Chapter 3, there exist several edge operators in the computer vision community. However, LoG is a promising operator and widely accepted [Chen and Medioni 87],

> *...the edges from this detection scheme have become a standard gauge against which all other methods are compared.*

In order to utilize LoG to detect breaklines from DEM raw data, it is important to remember that properties, advantages and drawbacks of LoG establish the conditions under which LoG can be applied. Points having significant elevation change can be detected from the zero-crossings of the convoluted value of the second derivative of the Gaussian Function (LoG) with the signal derived from terrain height at every grid point.

Detecting breaklines by using LoG has more advantages than using any other edge operator, because of LoG's properties. To visualize these merits, it is interesting to relate the properties of LoG to the process of breakline detection.

- Breaklines, which correspond to localized discontinuities of elevation changes on the DEM surface, can be detected by using the localized property of LoG in both the spatial and frequency domain.
- Breaklines acquired from LoG can give us more descriptions about the terrain such as ridges or valleys. As will be presented later, points having a negative minimum second derivative value are the points that have maximum curvature on a concave-down curve. Therefore, these points are located along the ridge or at the hill-top. This is very important because LoG can extract information from the data and make this information more explicit.
- Even though the Gaussian function is a low-pass function, LoG will not give a better localization of breakline positions, particularly in the area where the orientation of breaklines is changing rapidly. However, if the orientation of breaklines is nearly linear, zero-crossings of LoG will give the position of breaklines as accurate as a directional operator. Also, if the size of LoG is small, [Berzins 84] has shown that the displacement between the true position and the position located by LoG is not more than the value of σ.

However, there exist some problems in applying LoG to detect breakpoints and breaklines. Generally, all edge operators are interested in detecting the

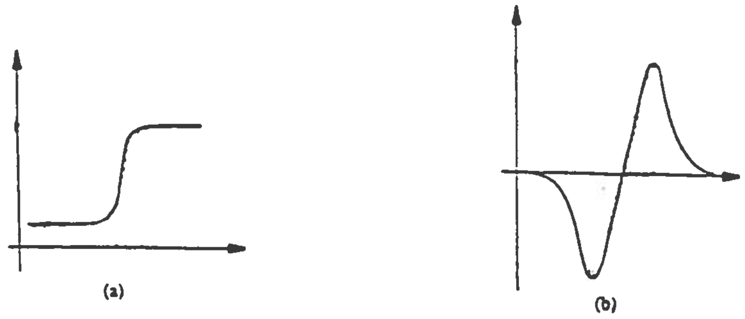

Figure 5: (a) Ideal Step Edge, (b) Response of LoG to the edge

ideal step edges as shown in Figure 5. Unfortunately, the occurrence of ideal step edges in the real world is rare. Ramp and roof edges are common (see Figure 6, 7 and 8).

The roof edge may be viewed as the cross-section of a hill or a valley which a ridge and valley line passes through the top of the roof. In addition to the hill-top point, both points at the bottom of hill are also important for the description of the shape of the terrain relief. These three points are breakpoints which we must extract from the raw data. While roof edges represent a cross-section of a hill or a valley, a ramp edge should be considered as a cross-section of a localized change of terrain relief. We can see that only the top and the bottom points of the ramp's slope sides are critical for describing terrain shape.

Theoretically, zero-crossings of LoG or zero values of the second derivative of the convoluted signal correspond to the points having minimum or maximum slope in the spatial domain. These points are well known as ***inflection points.*** If our task is to detect edge points for an ideal step edge (Figure 5), zero-crossings should be used because the inflection points correspond to the edge. Unfortunately, zero-crossings of LoG are not located at the breakpoints for both roof and ramp edges.

From Figure 5 and Figure 7, we can see that LoG gives the extreme value at the breakpoint. This is due to the fact that the radius of curvature (ρ), as defined in equation 10 (page 35), at breakpoints is minimum between

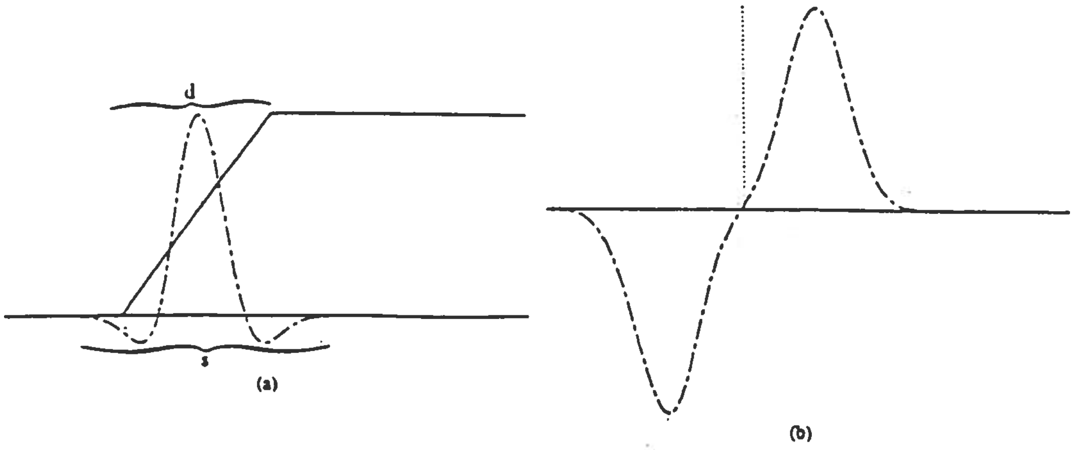

Figure 6: (a) Ramp edge and LoG, (b) Response of LoG to the edge when $d < s$ [Huertas and Medioni 86]

two inflection points. When ρ is minimum, the value of $\|d^2y/dx^2\|$ must be maximum. This implies that the second derivative has to be positive maximum or negative minimum.

In order to extract breakpoints from the given DEM data, points having negative minimum and positive maximum convoluted values (extremas) between zero-crossings must be utilized. This also can be confirmed by the statement [Douglas 87],

> *It can be easily shown that maxima and minima of the second derivative of a curve in two-dimensional space will project onto the curve to locate points of maximum change in curvature. In three-dimensional space, maxima of the surface representing the absolute value of the second derivative, (or the slope of the slope surface) also represent locations of maximum curvature in the original surface. These maxima are, of course, ridge lines, and their projection onto the original surface are the lines of break in slope.*

Extremas not only give the location of breakpoints, they also give more information about breakpoints. For example, breakpoints with positive maximum values are points lying along the bottom of valleys or at the bottom of basins

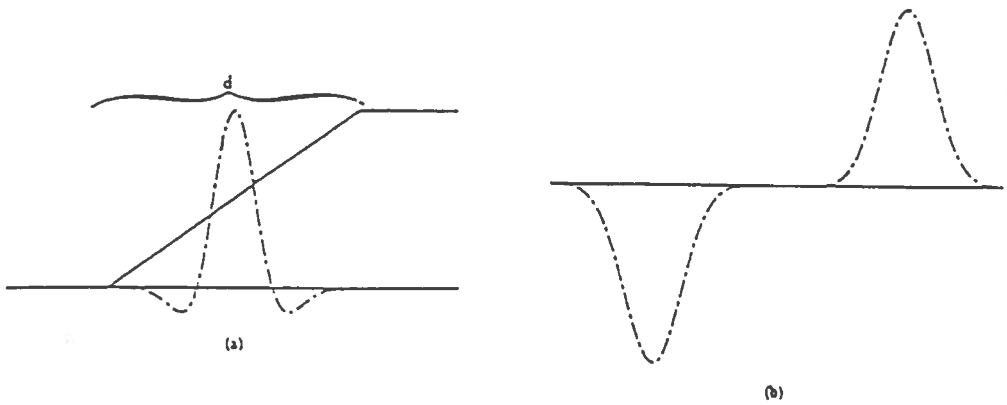

Figure 7: (a) Ramp edge and LoG, (b) Response of LoG to the edge when $d > s$ [Huertas and Medioni 86]

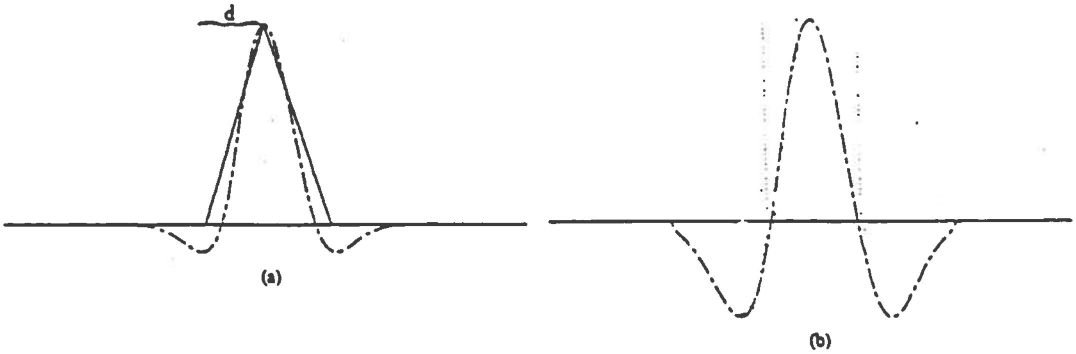

Figure 8: (a) Roof edge and LoG, (b) Response of LoG to the edge [Huertas and Medioni 86]

because the positive value of d^2y/dx^2 represents the concave-up shape of terrain and is located at the point whose radius of curvature is minimum.

With these properties, extremas must be used to detect breakpoints. The first step to find extremas is searching for zero-crossings. Zero-crossings must be extracted to confine the inflection region such that extremas can be determined. The detail and strategy of this new method will be elaborated on in section 4.4.

4.3.2 Computational Consideration of LoG

In order to use LoG to detect breaklines on the DEM surface, all signals derived from DEM grid data must be convolved by $\nabla^2 G$ (see section 3.3.1). Even though the convolution process is straightforward, the computations required to convolve all the signals is enormous due to the size of LoG. This has been shown by several authors, viz., [Chen et al. 86], [Marr and Hildreth 80], and of course [Nishihara and Larson 81]. To solve this problem, several approaches have been proposed.

First, LoG can be approximated by the Difference of two Gaussians (DoG). This method was presented in [Marr and Hildreth 80] who developed LoG. DoG in two dimensions is defined as :

$$DoG(\sigma_1, \sigma_2) = \frac{1}{\sqrt{2}\pi\sigma_1} \exp\left(-\frac{R^2}{2\sigma_1^2}\right) - \frac{1}{\sqrt{2}\pi\sigma_2} \exp\left(-\frac{R^2}{2\sigma_2^2}\right), \qquad (11)$$

where,

$$R^2 = X^2 + Y^2. \qquad (12)$$

They showed that the shape of DoG is nearly the same as the shape of LoG, when the ratio between the values of σ_1 and σ_2 is about 1.6. By this technique, LoG in two dimensions is decomposed into two passes of one dimension Gaussian, one for each direction, i. e. X and Y directions.

Figure 9 shows that DoG needs a slightly larger support than the LoG. [Hildreth 80] has proposed to use a support of 4W for DoG while others, like

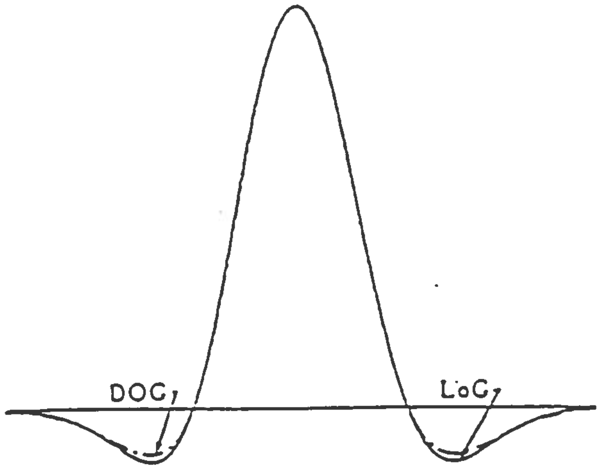

Figure 9: Approximation LoG by DoG [Huertas and Medioni 86]

[Huertas and Medioni 86] have found that 3W is sufficient for LoG (W is the excitatory region of LoG).

The second method to reduce the computation burden of LoG can be done by decomposing LoG into a sum of two separable filters. This technique was initially developed at the University of Southern California [Chen et al. 86]. The Laplacian of Gaussian ($\nabla^2 G$) in two dimensions can be rewritten as the sum of two separable filters as follows:

$$\nabla^2 G(X,Y) = h_{12}(X,Y) + h_{21}(X,Y) \tag{13}$$

where,

$$h_{12}(X,Y) = h_1(X) * h_2(Y) \tag{14}$$

$$h_{21}(X,Y) = h_2(X) * h_1(Y) \tag{15}$$

$$h_1(\xi) = \left(1 - \frac{\xi^2}{\sigma^2}\right) \exp\left(-\frac{\xi^2}{2\sigma^2}\right) \tag{16}$$

$$h_2(\xi) = \exp\left(-\frac{\xi^2}{2\sigma^2}\right) \tag{17}$$

This decomposition method uses only a single value of σ rather than two values as used in the DoG method.

In addition to the above two methods, LoG can be achieved by means of convolution through hardware. Specifically, convolution can be done by either parallel processing or a convolver [Nishihara and Larson 81]. By these approaches, the convoluted results may be obtained in real time or nearly real time. Nevertheless, the breakline detection process usually is not a real time task.

For this research, LoG by the method of USC is chosen because of the following reasons:

- It is a real LoG, not an approximation of LoG by DoG.
- LoG requires smaller support than DoG.
- It requires a single value of σ, instead of two values of σ if DoG is chosen.

4.4 Proposed DEM Data Compression Strategy

From the definition of breaklines given above, breaklines can be detected from the surface by detecting extremas within the inflection region confined by zero-crossings of LoG. In order to utilize LoG to detect breaklines or breakpoints, one must realize that a smoothing function, namely the Gaussian function, is a low-pass filter. Therefore, if a large filter size is utilized, some breaklines or breakpoints on the given surface may be filtered out. The DEM surface needs to be convolved with the appropriate filter size. The size of filter can be determined from the relationship between σ and f, the spatial and frequency parameters of the Gaussian function. Equation 18 shows the relationship of these two quantities. The derivation of the equation can be found in Appendix A.

$$f = \frac{3}{\sigma\pi\sqrt{2}} \tag{18}$$

From a given DEM surface, the total power spectrum energy of the surface can be approximated from its Fourier Transform. Then, the level of the energy needed to be preserved should be selected. In the frequency domain, the cut-off frequency (which is defined in equation 20 below) is computed from the preserved energy level. With the value of the cut-off frequency, the σ or size of filter can be determined from equation 18.

It is important to note that as the density of the DEM data is controlled by the final map scale, the highest frequency which we can acquire from the data is the *Nyquist Frequency* whose wavelength is exactly $2\triangle$, where $\triangle$ is the distance between successive grid points. Therefore, grid density of DEM data should be determined by considering the highest frequency phenomenon which will be recovered in the map.

Since LoG is the band-pass filter with bandwidth of 1.2 octaves, the size of two successive filters must not be separated by more than 1.2 octaves. However, to account for noise and ease of implementation, a one octave spacing is preferred. Therefore, the designed σ value should be a full octave value so that the next smaller filter size will be half of this value. The following are the steps proposed for DEM data compression:

1. Create the signal and design the filter size in the following manner :

    - From the given regular grid DEM data of a specific density controlled by the application of DEM and Nyquist sampling theorem, the signal $(h(f))$ will be created at every point.
    - Compute a Fast Fourier Transform (FFT) of the signal, $(H(f))$.
    - Estimate the total power spectrum (P_S) of the surface from the computed FFT. Since the total power spectrum is defined as the Fourier Transform of autocorrelation function not of the raw signal, the power spectrum obtained from this FFT is not the true value. However, we are interested in computing the relative energy content and use this value to approximate the size of LoG. Therefore, the total power spectrum of the surface may be approximated by P_S from FFT of the raw signal.

$$P_S = \sum_{f=0}^{f_q} \|H(f)\|^2 \tag{19}$$

where f_q is Nyquist frequency.

- Design the preserved power spectrum level (P_c) with respect to the estimated total power spectrum. For example, P_c may be chosen to be equal to $0.95P_S$.
- From the value of P_c, the cut-off frequency (f_m) is determined. The cut-off frequency is the highest frequency contributing to P_c such that
$$\sum_{f=0}^{f_m} \|H(f)\|^2 \leq P_c \tag{20}$$
- With the cut-off frequency (f_m), the estimated value of σ is determined by equation 18.
- Choose the filter size by rounding off the σ value to the nearest full octave, i. e., 8,16,32.

2. Apply LoG with size designed above to the signal.

3. Detect zero-crossing points by the method based on a four connectivity search such that each point will be analyzed with respect to its surrounding neighbor points [Greenfeld 87].

4. Find extremas by detecting points having negative minimum and positive maximum convoluted values in both the horizontal and vertical directions between zero-crossings.

5. Repeat steps 2 to 4 with the filter whose size is equal to half of the previous size until the desired level of precision is attained.

6. Detect the common zero-crossings and extremas points obtained from every channel. These extremas are the terrain breaklines and breakpoints which can be viewed as the primal sketch of DEM surface.

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 General

This chapter presents the experiments of the proposed DEM data compression method. It includes the testing data set, testing procedure and the results. Section 5.2 explains in detail the testing data set used in this research. Then the testing procedure is summarized in section 5.3. The results are presented in section 5.4, and analysis of the results are described in the last section.

5.2 Testing Data

Data used in the research are DEM data obtained from United State Geological Survey (USGS). The data are generated from the combination of [Elassal and Caruso 83]:

- the Gestalt Photo Mapper II,
- manual profile from stereomodels,
- the Digital Cartographic Software System (DCASS).

It covers one 7.5 minute quadrangle map. The data are stored as profiles from west to east and from south to north in UTM coordinate system. The spacing of points along and between each profile is 30 meters. Accordingly, every profile may not consist of the same number of points.

The DEM data covering the area of Grove Oak, Alabama, is obtained from magnetic tape serial No. RJN785. Originally, there were 387 profiles; however, only 378 profiles have been used due to the fact that a few profiles at the beginning and end of the area do not have the same number of points as the ones between them.

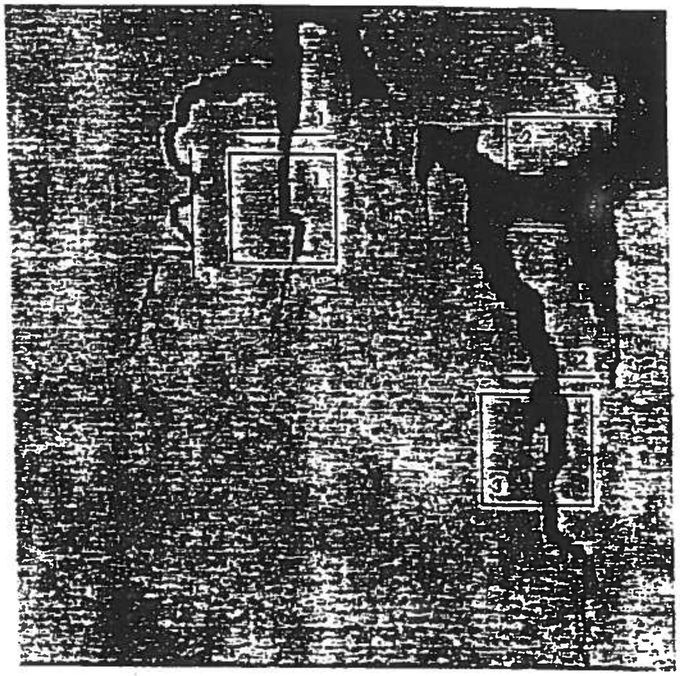

Figure 10: USGS DEM surface

As a result, only the DEM data of UTM zone 16 from Northings 3,804,000 to 3,817,740 and Eastings 580,470 to 591,780 is used. The data consists of 378 × 459 grid points. The highest and lowest elevation of the data are 394 and 176 meters, respectively. For convenience, data is densified to 512 × 512 grid points by the method of bilinear interpolation. Therefore, spacing of the grid in easting and northing directions are 22.13 meters and 26.89 meters, respectively. Figure 10 shows the USGS DEM area. The lighter areas depict higher elevations.

In order to evaluate the proposed breakline detection method, three cross-sections and three windows featuring different types of terrain are selected for the experiment. The location of the cross-sections and windows are also included in Figure 10. The first and second cross-sections consist of 70 points and lie in a north-south direction. While the last cross-section consists of 100 points in an east-west direction. Every window covers 85 × 85 points. The details of the size and location of the cross-sections and the windows are presented in Tables 1 and 2.

Table 1: UTM Coordinates of the cross-sections

Section	Starting Point	Ending Point

1	Easting = 582,683.307
	Northing = 3,808,839.922
		Easting = 582,683.307
		Northing = 3,810,722.114

2	Easting = 586,777.926
	Northing = 3,814,217.613
		Easting = 586,777.926
		Northing = 3,816,099.805

3	Easting = 582,993.170
	Northing = 3,807,791.272
		Easting = 585,206.478
		Northing = 3,807,791.272

Table 2: UTM Coordinates of the windows

Window	Lower-left corner	Upper-right corner

1	Easting = 583,015.303
	Northing = 3,808,571.037
		Easting = 584,896.614
		Northing = 3,810,856.556

2	Easting = 580,577.133
	Northing = 3,814,352.055
		Easting = 582,458.444
		Northing = 3,816,637.574

3	Easting = 587,109.922
	Northing = 3,813,679.843
		Easting = 588,991.233
		Northing = 3,815,965.362

5.3 Testing Procedure

In the experiment, the DEM data is convolved by LoG as described in section 4.4 with different σ's. Then both visualized and quantitative evaluations are performed. For visualized evaluations, two aspects are examined. First, contour maps of all three windows, constructed by using breakline maps obtained from the proposed method, are compared with the corresponding contour maps generated from the full grid data points. The experiments are carried out using different σ. The results are the tools used to evaluate how well the breakline maps can portray the terrain description and be used as a DEM data-base. In addition, the analysis of scale space delocalization of zero-crossings can be done within the windows and cross-sections.

Second, the detected breakline maps are compared with the manual ones. Manual breakline maps were compiled by fifteen people consisting of faculty, staff, graduate students of the Department of Geodetic Science and Surveying, and five cartographers from Defense Mapping Agency (DMA). They were asked to draw the minimum numbers of breaklines and breakpoints on contour maps generated from the full regular grid data. The manual results were used as the standard of comparison for both visualized and quantitative evaluation.

The following steps are performed to examine the proposed method as described above.

- Original DEM elevations are expanded to cover the whole gray level range, i. e., from 0 to 255 and used as the input signals for the next step.
- Convolve the signal derived from DEM data with LoG designed earlier.
- Detect zero-crossings and points having negative minimum and positive maximum convoluted values (extremas) between zero-crossings as presented in section 4.4.
- Overlay zero-crossings and extremas on the image of the surface with different σ's so that one can see the effect of LoG on different channels.

- Locations of zero-crossings and extremas are plotted on the cross-sections so that the breakpoints from the multi-values of σ can be investigated. In addition, the goodness of the detected breakpoints can be examined by considering their locations with respect to the shape of the terrain.

- For the three windows, the following are performed:

  1. Generate a contour map of all three windows by a contouring package named *QUICK-SURF* [Schreiber 88] available at the Department of Geodetic Science and Surveying using every original grid data point as control points.
  2. Plot the extremas obtained from the proposed method for all three windows at the same scale as the contour maps acquired above.
  3. Evaluate the positions of the detected breaklines from (2) by overlaying them on the contour maps. The evaluation is done with the different values of σ so that the displacement of zero-crossings across the scales can be investigated.
  4. Compare breakline maps with the manual breakline maps described earlier in both visualized and quantitative aspects.
  5. Reconstruct the contour maps of all three windows by using the same contouring package used in step (1). However, this time, only the breakpoints obtained in step (2) and points lying along the window boundary are used as the control points, not all grid points. The maps are generated from extremas with different σ.
  6. The reconstructed contour maps obtained in (5) now are able to be visually compared with the original contour maps generated from the DEM grid data in step (1).

5.4 Results

As mentioned in the last chapter, the frequency response of the surface is very important to design size of the LoG. Therefore, before applying LoG to the signal, the estimated total power spectrum level (P_c) of the surface must be determined. Table 3 shows the values of σ, cut-off frequency (f_m)

Table 3: Frequency Response and Energy Level of DEM surface

σ	f_m	Energy Level
8	$22 \div 256$	96.3 %
4	$43 \div 256$	98.4 %
2	$86 \div 256$	99.3 %
1	$173 \div 256$	99.8 %

and power spectrum level (P_c) of the surface. In the table f_m represents the frequency obtained from Discrete Fast Fourier Transform of the whole surfaces. Their values are rounded to the nearest integer.

Figure 11 to Figure 13 are the results of zero-crossings of LoG convolved at $\sigma = 8$ to $\sigma = 2$ overlaid on the original surface. Figure 14 and Figure 15 show the extremas between zero-crossings at σ equal to 8 and 4, respectively.

Three cross-sections showing the shape of terrain and locations of zero-crossings and extremas at different σ are presented in Figure 16 to Figure 21. Figure 22 to Figure 25 are the scale space of zero-crossings and extremas. The contour maps generated from the full grid data points of all three windows are shown in Figure 26 to Figure 28. Figure 29 to Figure 34 are the zero-crossing maps of all three windows at different σ at the same scale as the contour maps. Figure 35 to Figure 40 show the extrema maps. The contour maps reconstructed from the zero-crossings at different σ are presented in Figure 41 to Figure 43, whereas Figure 44 to Figure 49 are the ones reconstructed from the extremas. Finally, in Figure 50 to Figure 52 the results of manual breakline maps are presented.

Table 4 and Table 5 represent the number of zero-crossings and extremas at different σ as well as the compression rate (C_r) of all three windows which is defined as:

$$C_r = \frac{n}{n_0} \tag{21}$$

Table 4: Number of zero-crossings and compression rate of the windows at different σ

Window	n_0			C_r		
	8	4	2	8	4	2
1	219	503	1202	32.99	14.36	6.01
2	218	412	1047	33.14	17.54	6.90
3	320	566	1157	22.58	12.76	6.24

Table 5: Number of extremas and compression rate of the windows at different σ

Window	n_0			C_r		
	8	4	2	8	4	2
1	310	695	1517	23.30	10.39	4.76
2	256	531	1346	28.22	13.61	5.37
3	455	760	1420	15.88	9.51	5.09

where,

n = number of grid points within the window
n_0 = number of breakpoints within the window.

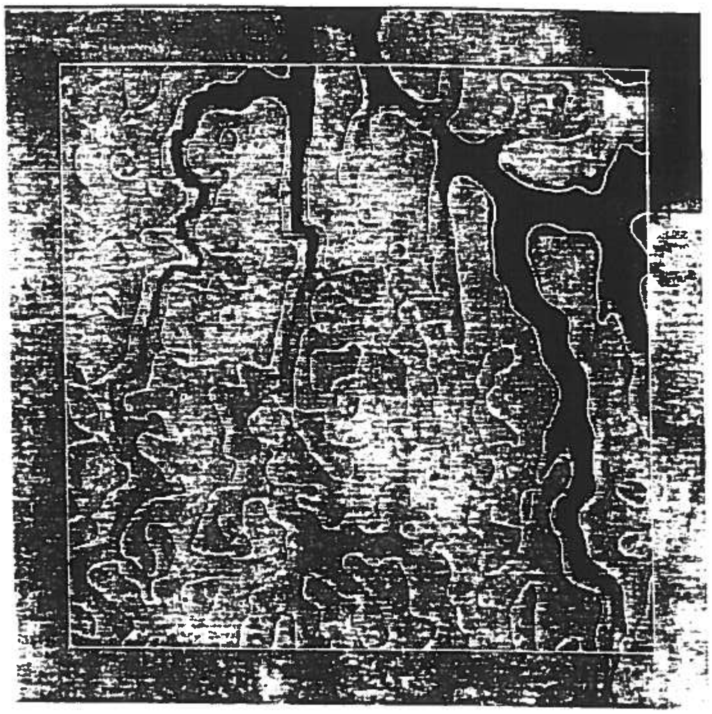

Figure 11: Zero-crossings at $\sigma = 8$

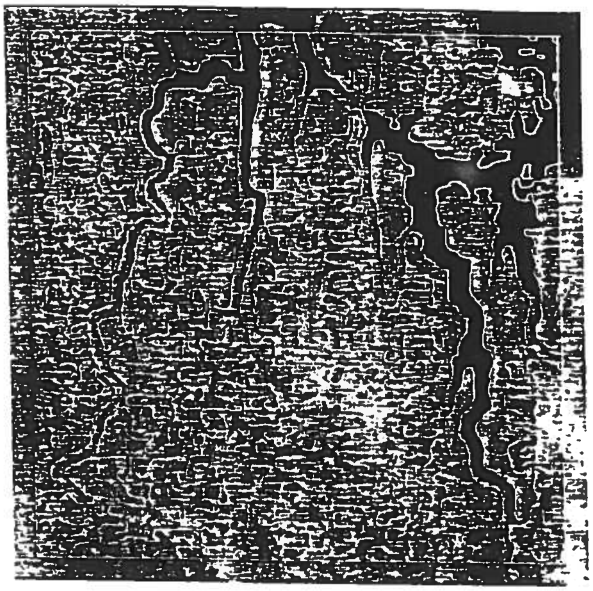

Figure 12: Zero-crossings at $\sigma = 4$

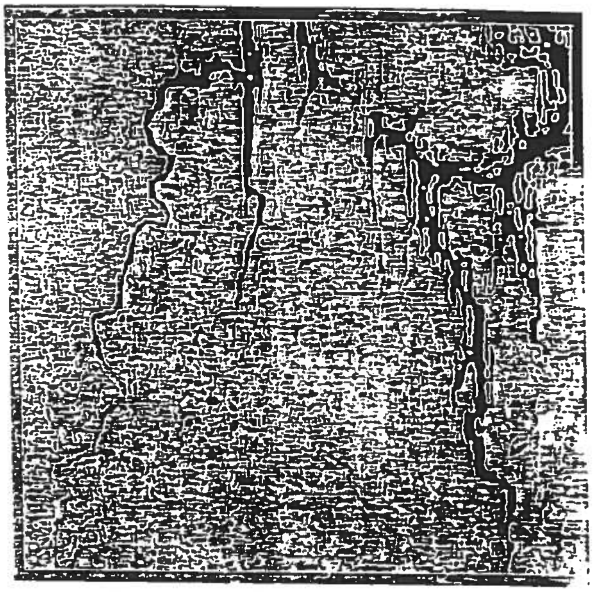

Figure 13: Zero-crossings at $\sigma = 2$

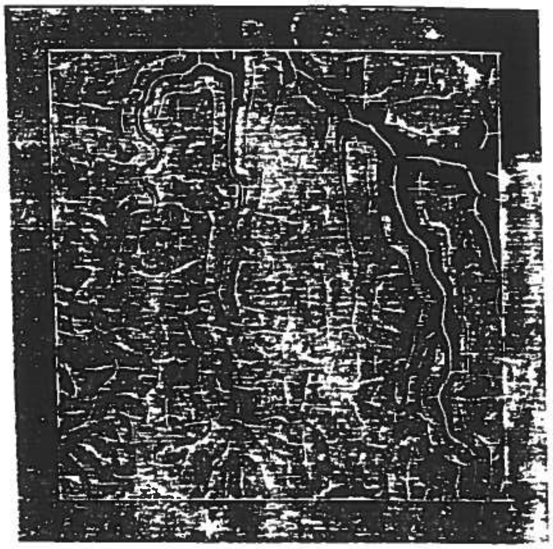

Figure 14: Extremas at $\sigma = 8$

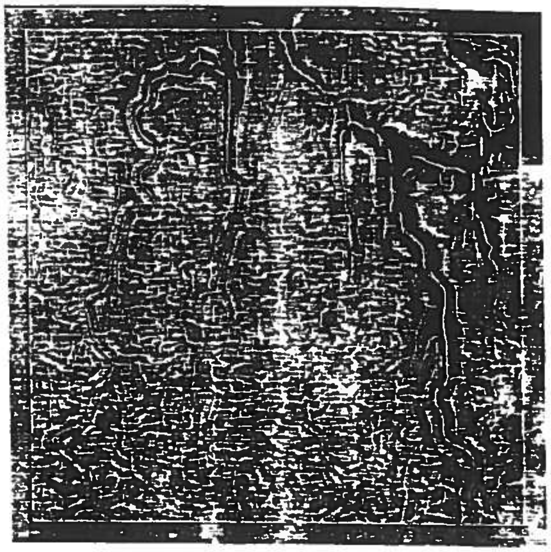

Figure 15: Extremas at $\sigma = 4$

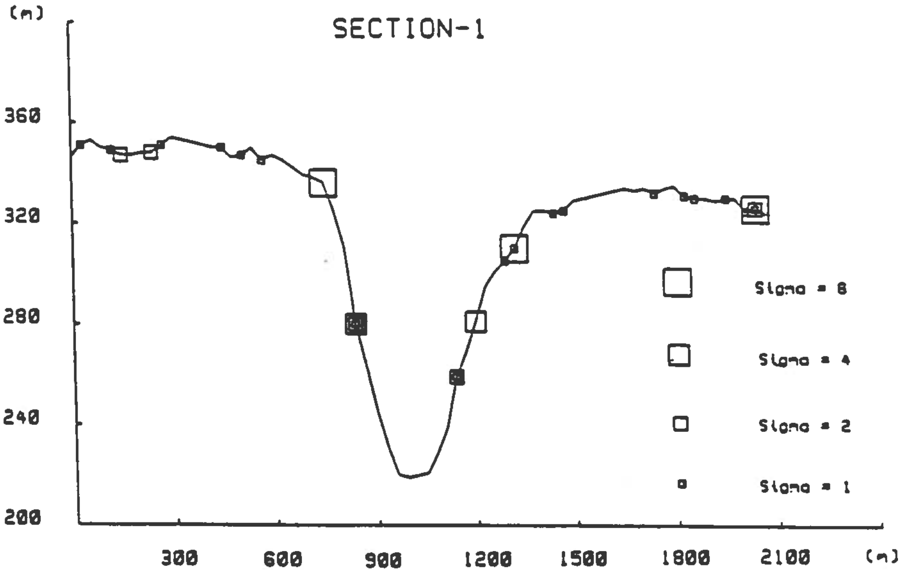

Figure 16: Zero-crossings of Section-1

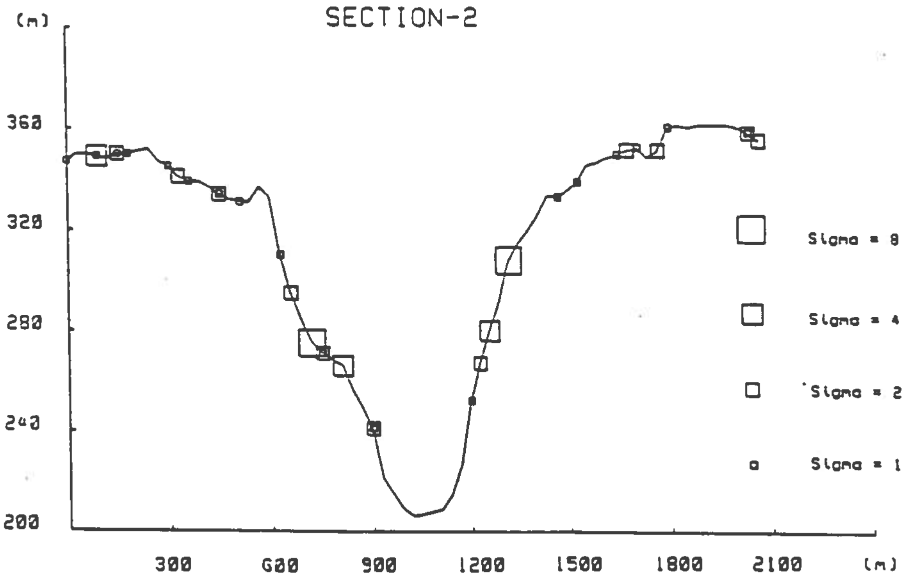

Figure 17: Zero-crossings of Section-2

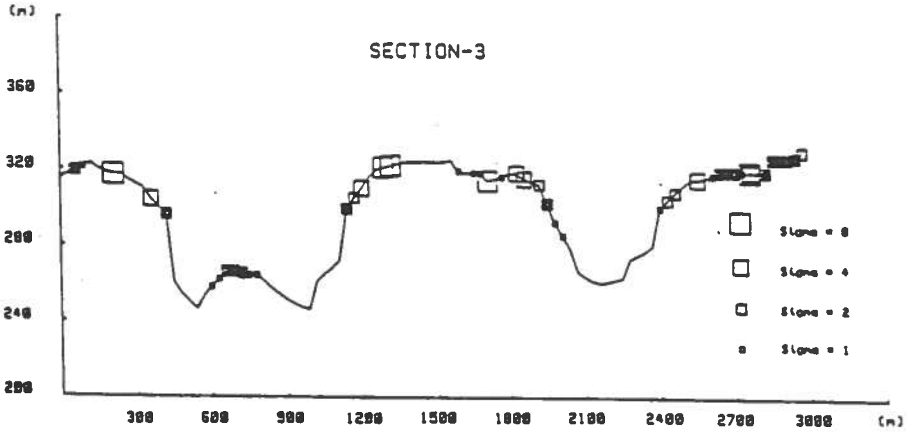

Figure 18: Zero-crossings of Section-3

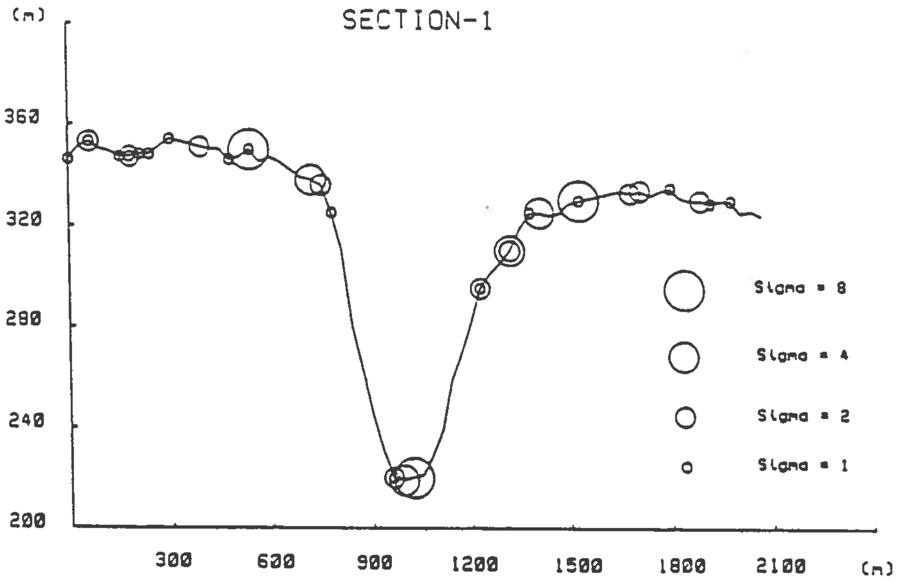

Figure 19: Extremas of Section-1

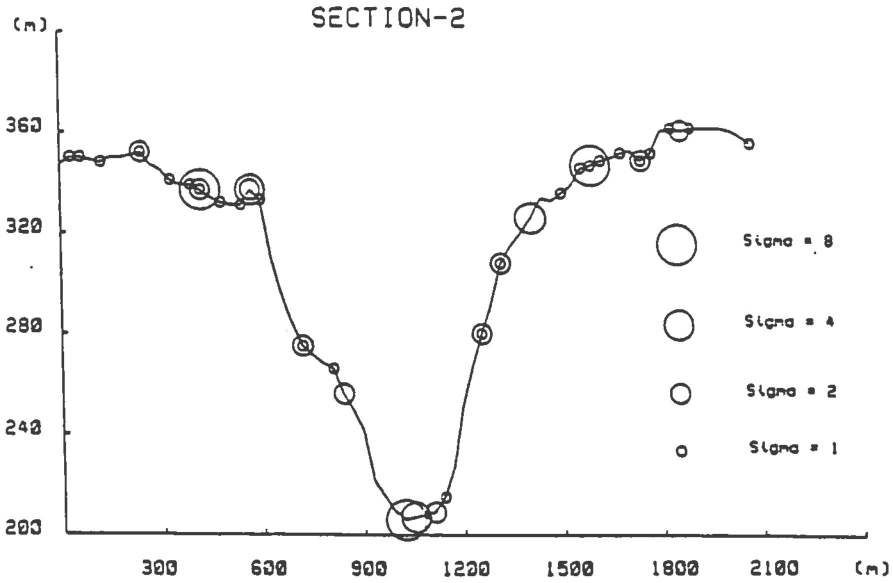

Figure 20: Extremas of Section-2

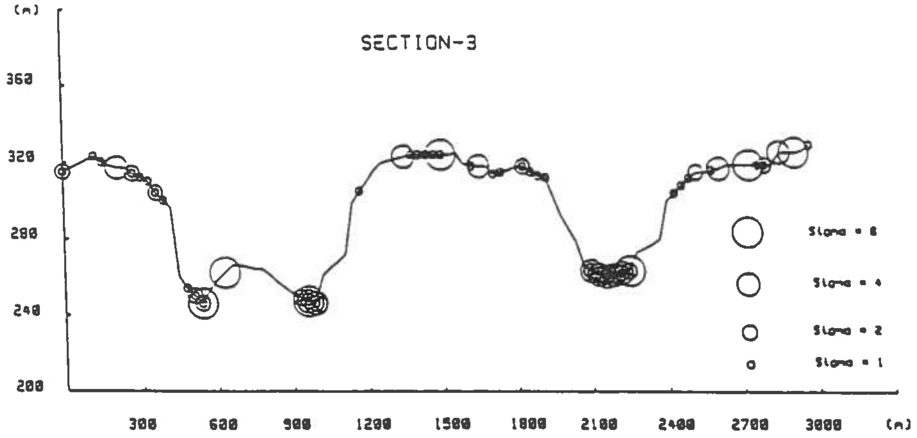

Figure 21: Extremas of Section-3

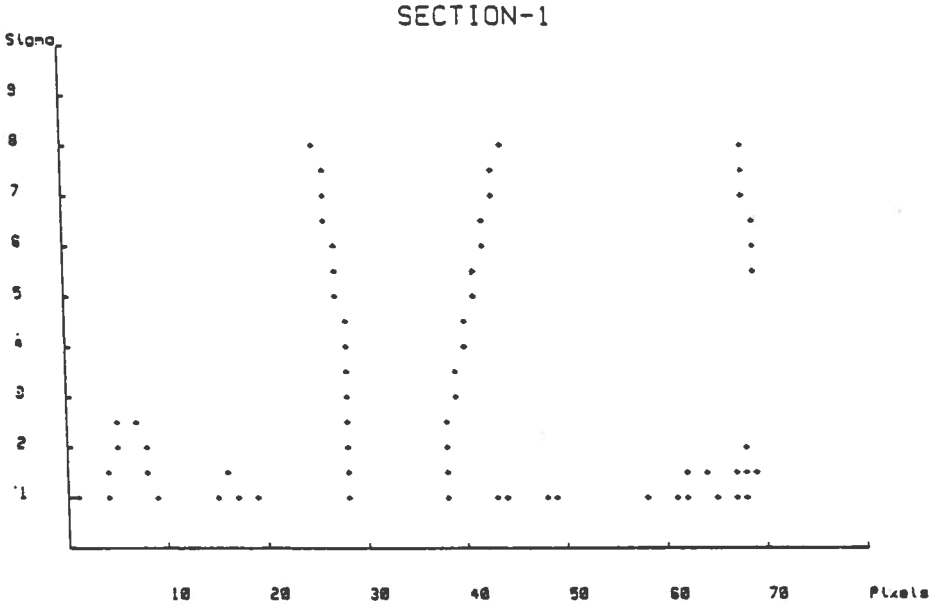

Figure 22: Scale-space Plot of Zero-crossings of Section-1

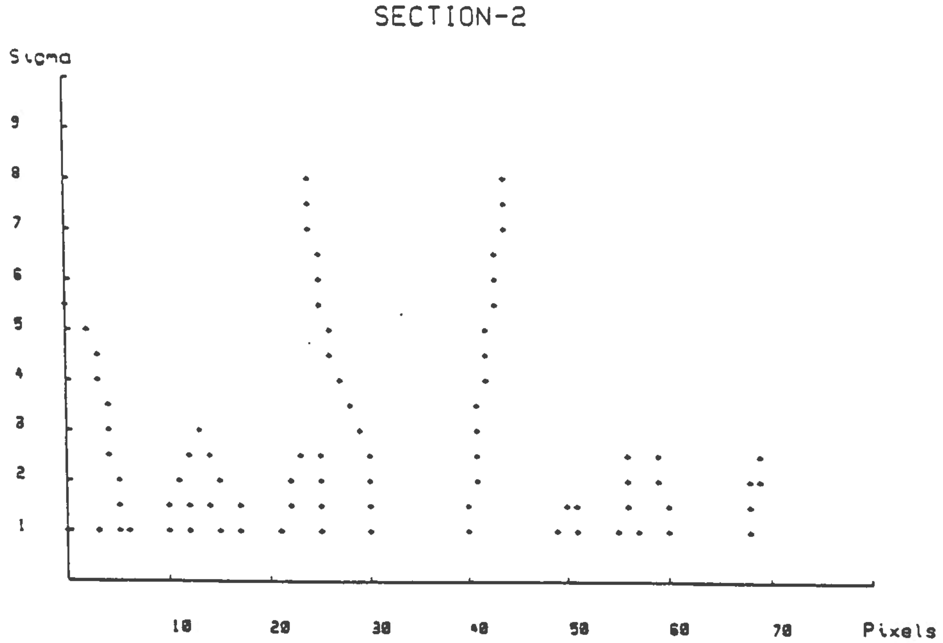

Figure 23: Scale-space Plot of Zero-crossings of Section-2

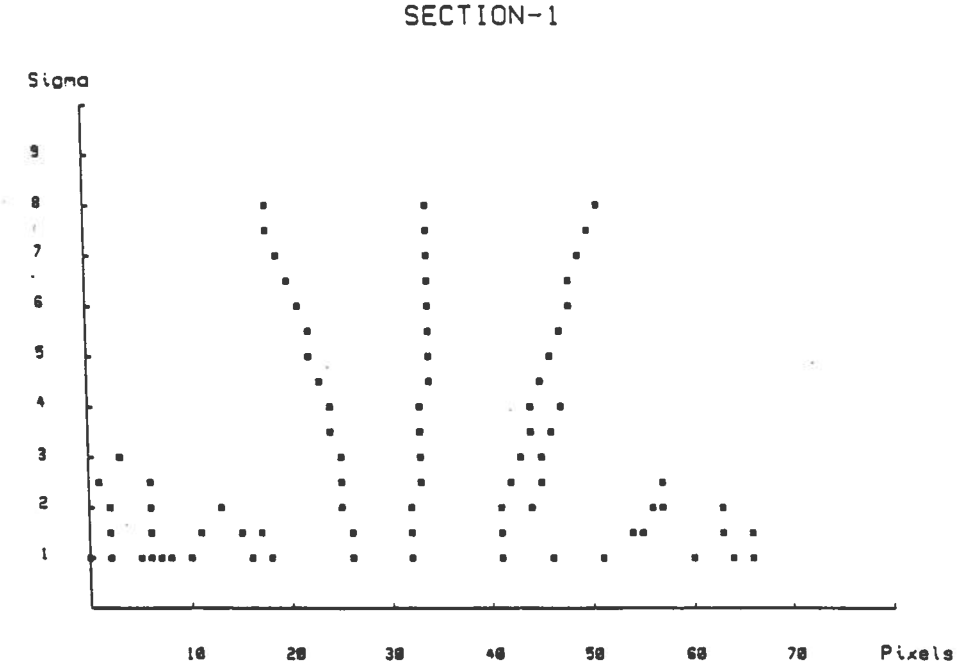

Figure 24: Scale-space Plot of Extremas of Section-1

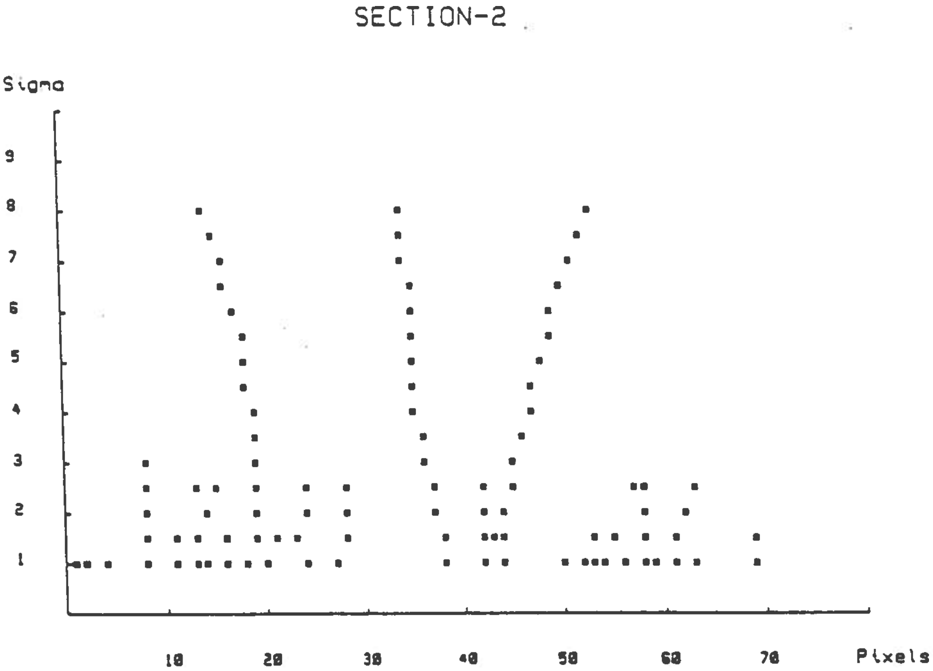

Figure 25: Scale-space Plot of Extremas of Section-2

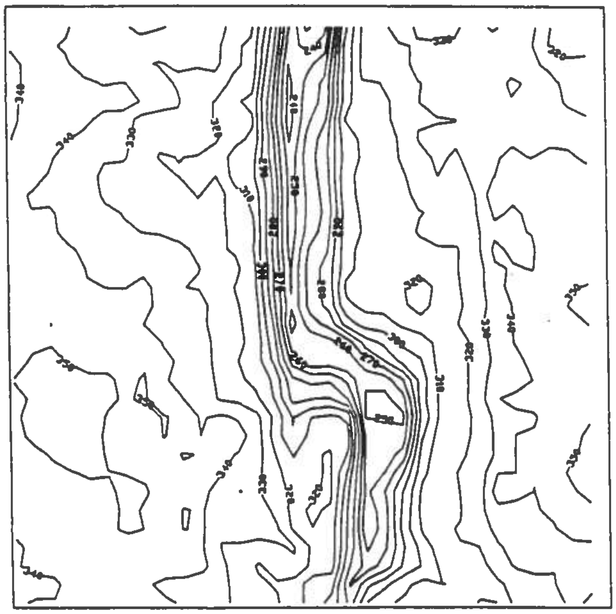

Figure 26: Original contour map of window 1

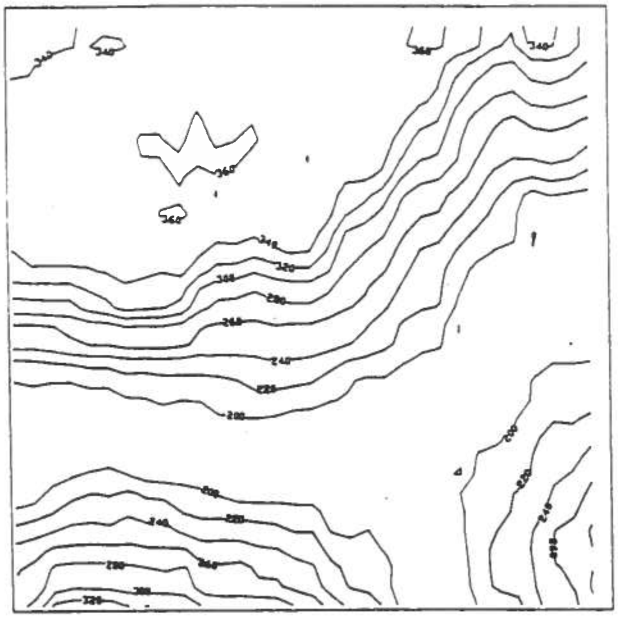

Figure 27: Original contour map of window 2

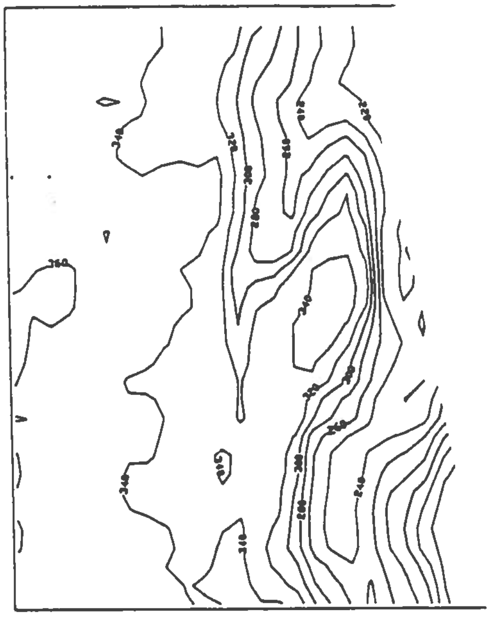

Figure 28: Original contour map of window 3

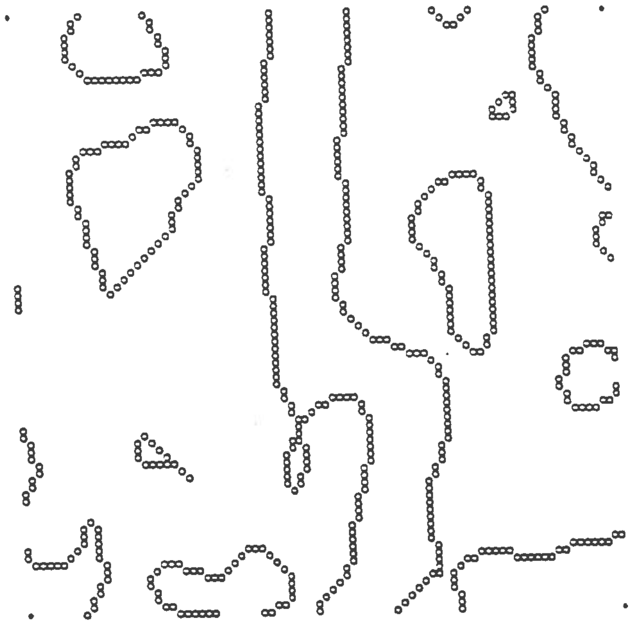

Figure 29: Zero-crossings of window 1, $\sigma = 4$

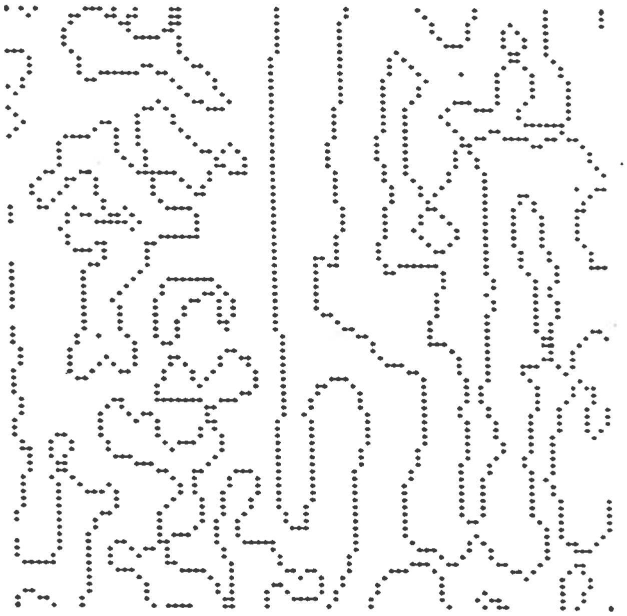

Figure 30: Zero-crossings of window 1, $\sigma = 2$

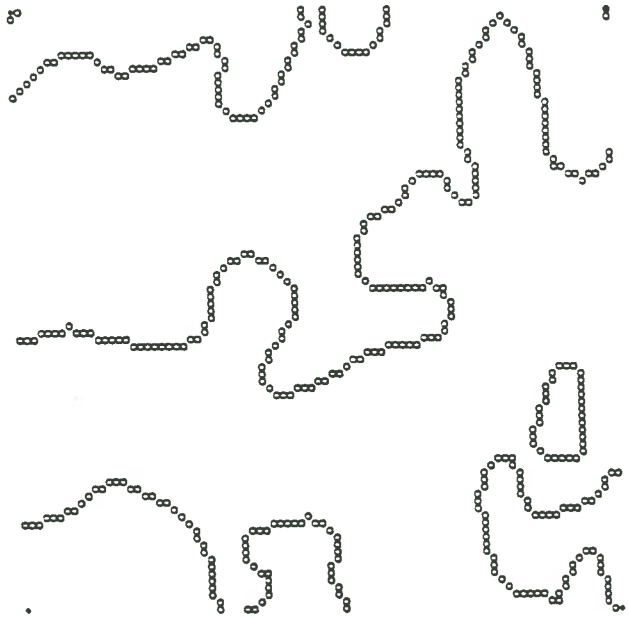

Figure 31: Zero-crossings of window 2, $\sigma = 4$

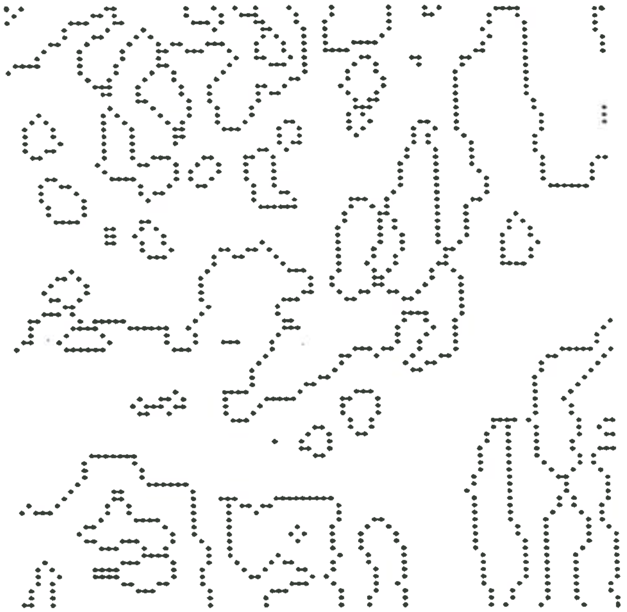

Figure 32: Zero-crossings of window 2, $\sigma = 2$

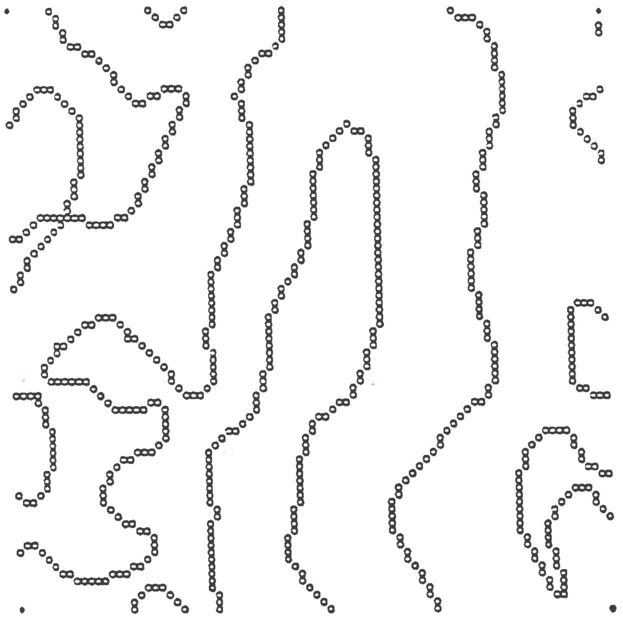

Figure 33: Zero-crossings of window 3, $\sigma = 4$

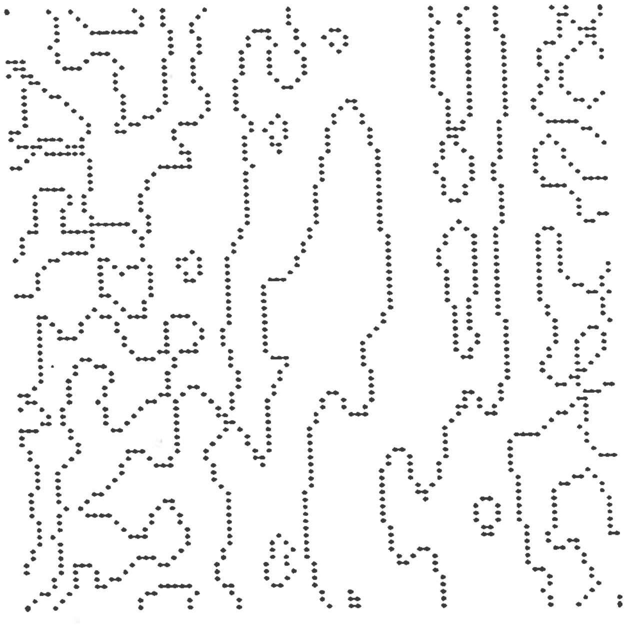

Figure 34: Zero-crossings of window 3, $\sigma = 2$

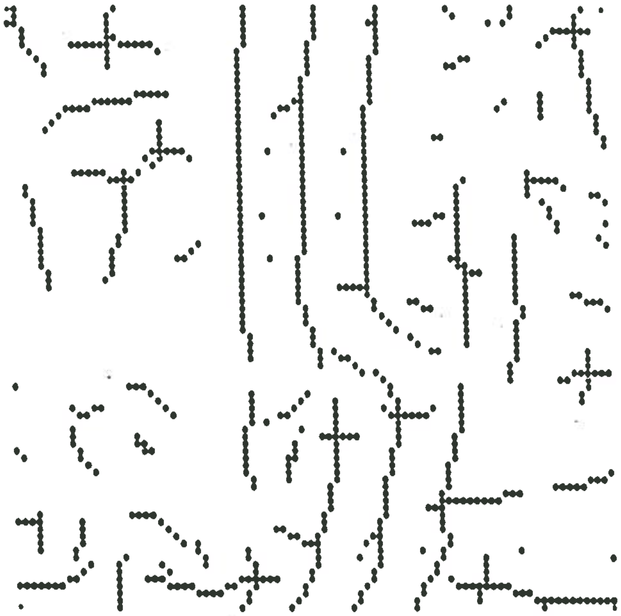

Figure 35: Extremas of window 1, $\sigma = 4$

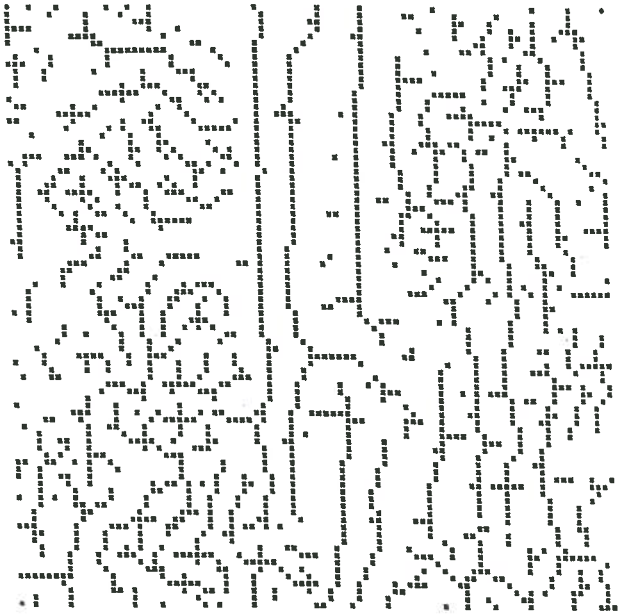

Figure 36: Extremas of window 1, $\sigma = 2$

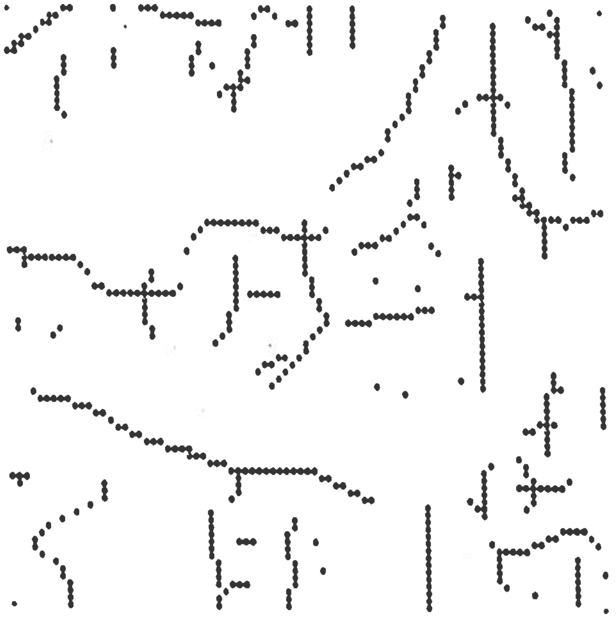

Figure 37: Extremas of window 2, $\sigma = 4$

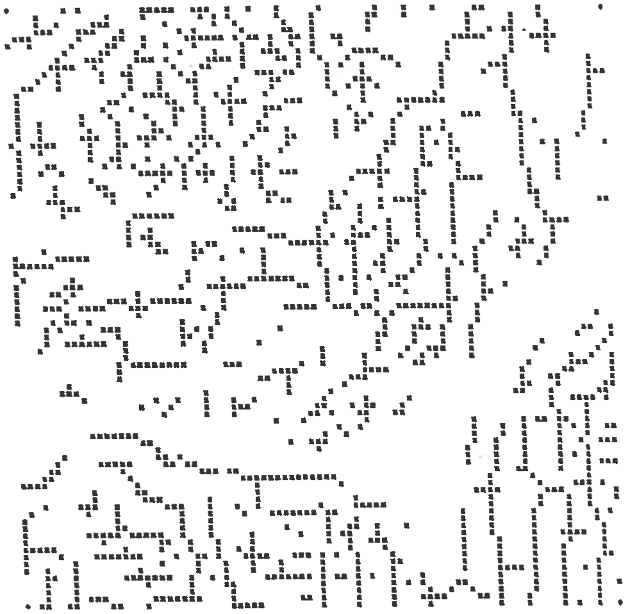

Figure 38: Extremas of window 2, $\sigma = 2$

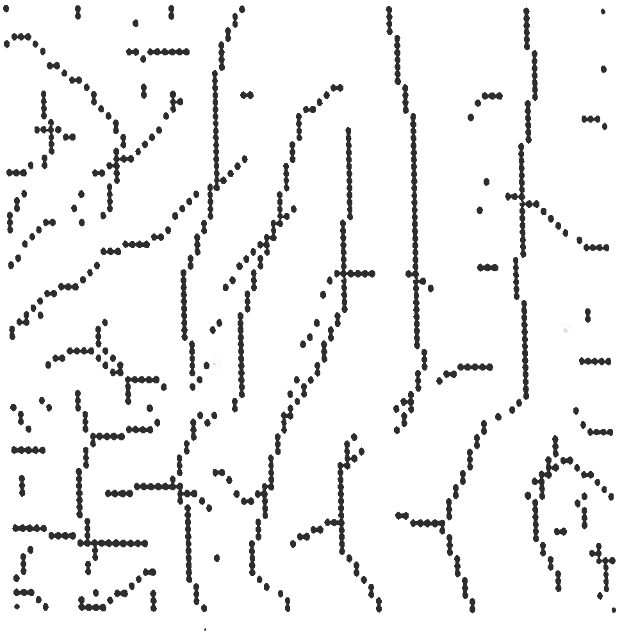

Figure 39: Extremas of window 3, $\sigma = 4$

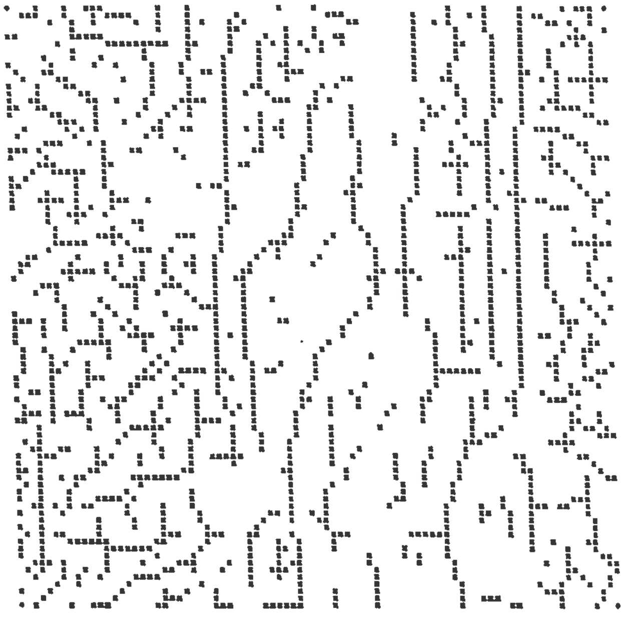

Figure 40: Extremas of window 3, $\sigma = 2$

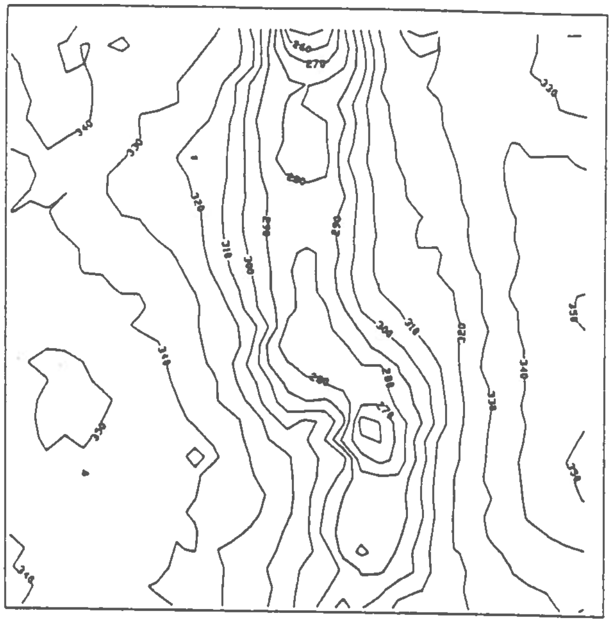

Figure 41: Reconstructed contour map of window 1 from zero-crossings with $\sigma = 2$

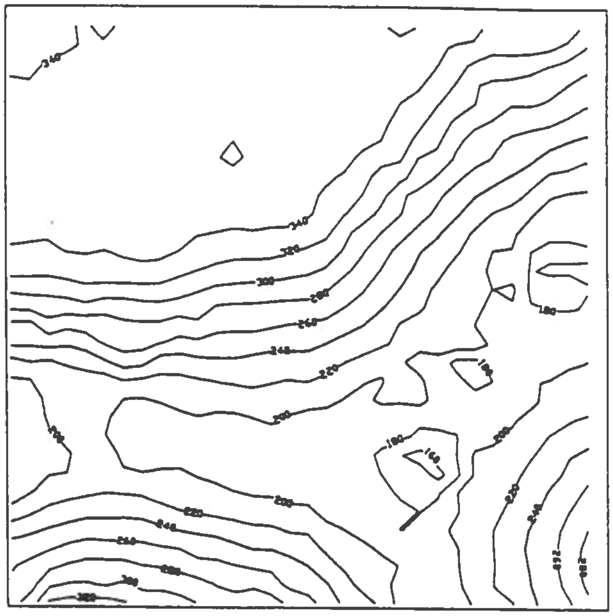

Figure 42: Reconstructed contour map of window 2 from zero-crossings with $\sigma = 2$

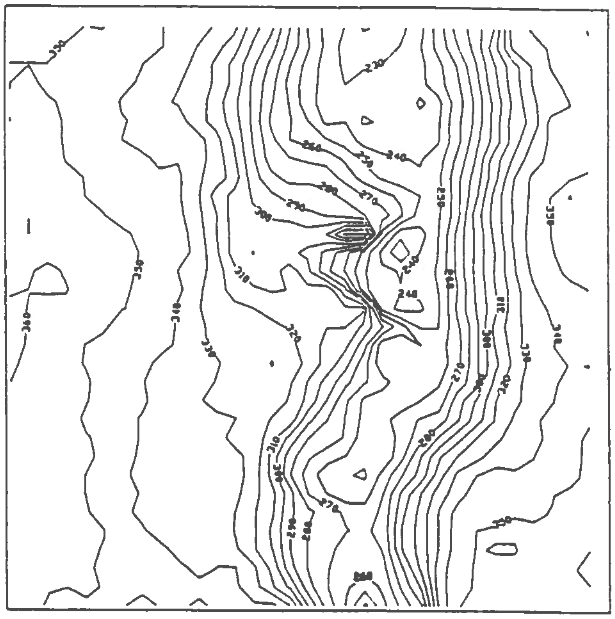

Figure 43: Reconstructed contour map of window 3 from zero-crossings with $\sigma = 2$

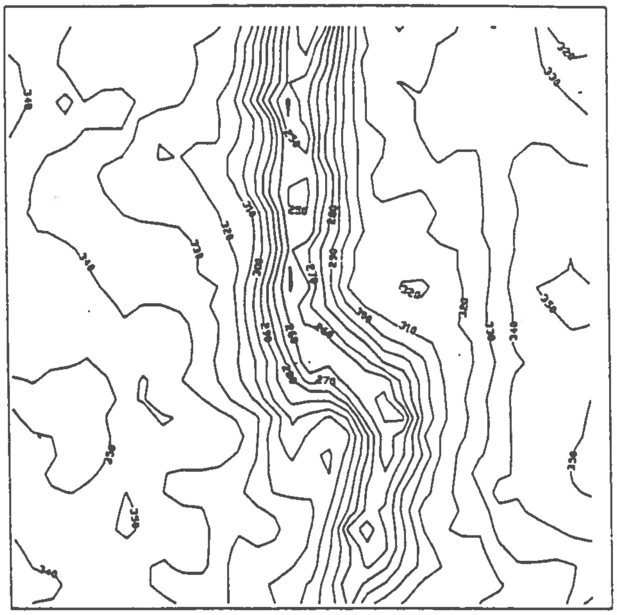

Figure 44: Reconstructed contour map of window 1 from extremas with $\sigma = 4$

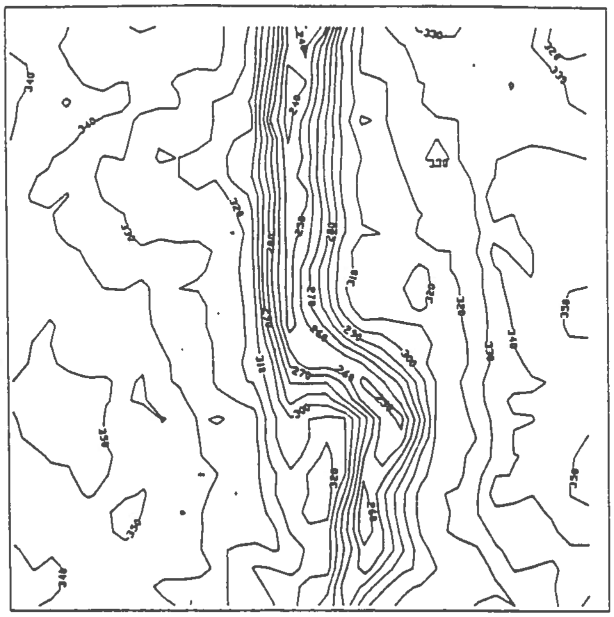

Figure 45: Reconstructed contour map of window 1 from extremas with $\sigma = 2$

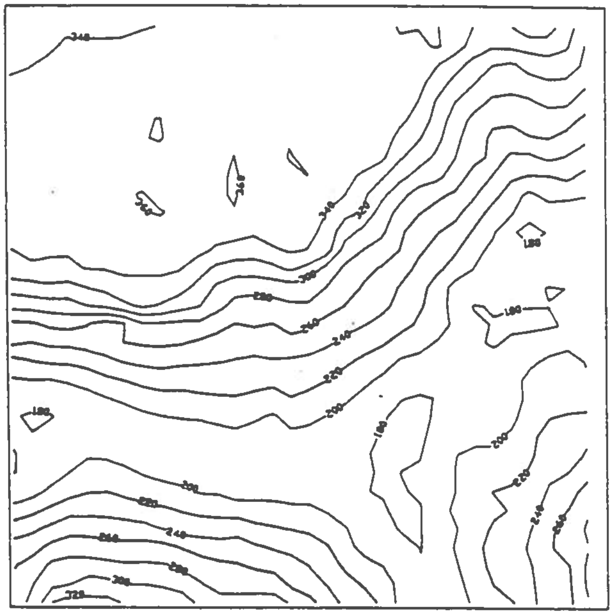

Figure 46: Reconstructed contour map of window 2 from extremas with $\sigma = 4$

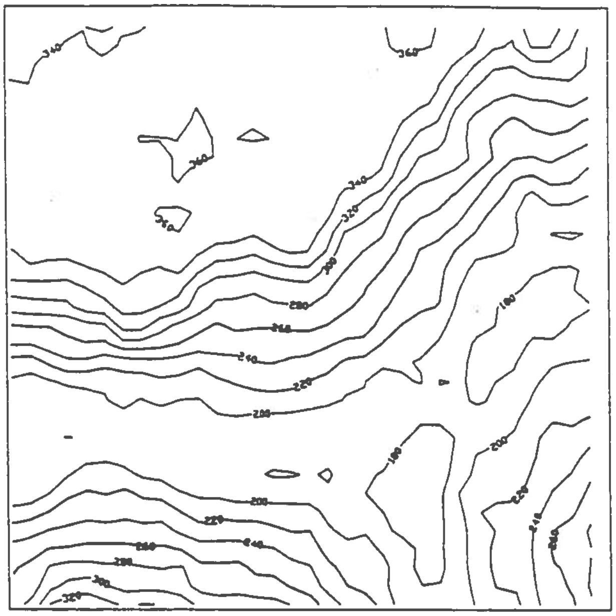

Figure 47: Reconstructed contour map of window 2 from extremas with $\sigma = 2$

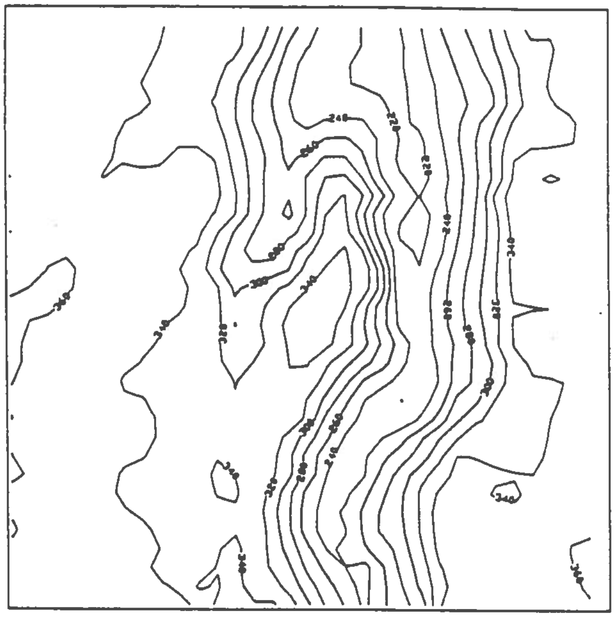

Figure 48: Reconstructed contour map of window 3 from extremas with $\sigma = 4$

5.5 Analysis of the Results

From Figure 11 to Figure 13, one can see that as the value of σ decreases more zero-crossings show up. This can be explained simply that the LoG with large σ has a larger spatial extent than the one having a smaller σ. As a result, the surface smoothed with larger filter is smoother than the one obtained from the smaller filter. Zero-crossings detected from LoG with large σ show a rough surface description. However, as we will see later, the location of zero-crossings at a large σ are displaced by the effect of smoothing and other reasons presented earlier in the last chapter. As the filter size gets smaller, one can expect a better localization from zero-crossings. At the smaller σ, small variations or high frequency phenomena are not rubbed out. Therefore, they create noisy zero-crossings as shown in Figure 13.

Spatial coincidence across the scale proposed by [Marr and Hildreth 80] can be examined from Figure 29 to Figure 34. One can see that zero-crossings occurring across the scale exhibit significant changes in the input signal. For this case, the zero-crossings due to the slope of the valleys show up in every channel, while some zero-crossings in the smallest σ (Figure 34) are created by noisy phenomena. From this result, only strong zero-crossings should be saved and noisy zero-crossings must be eliminated. This can be done by locating the common zero-crossings which occur across the scale.

In order to analyze the experiment in depth, three cross-sections presented in Figure 16 to Figure 18 should be examined. The occurrence of zero-crossings confirms our expectation that they correspond to the inflection points. Figure 22 to Figure 23 are the scale space plots showing the position of zero-crossings against the value of σ. The figures show that zero-crossings at the smaller σ values give better localization than at the large scales. Figure 19 to Figure 21 present the extremas detected from the different σ. It can be seen that extremas occur at the points which are critical for terrain description, i.e. bottom point of the valley and at the edge of the valley. These points are breakpoints which must be preserved as the surface description. In order to further examine the property of extremas, Figure 35 to Figure 40 should be investigated. These represent extremas plotted at the same scale as the original maps shown in Figure 26 to Figure 28. By overlaying the plot of extremas on the maps, one can see that extremas give the position of points

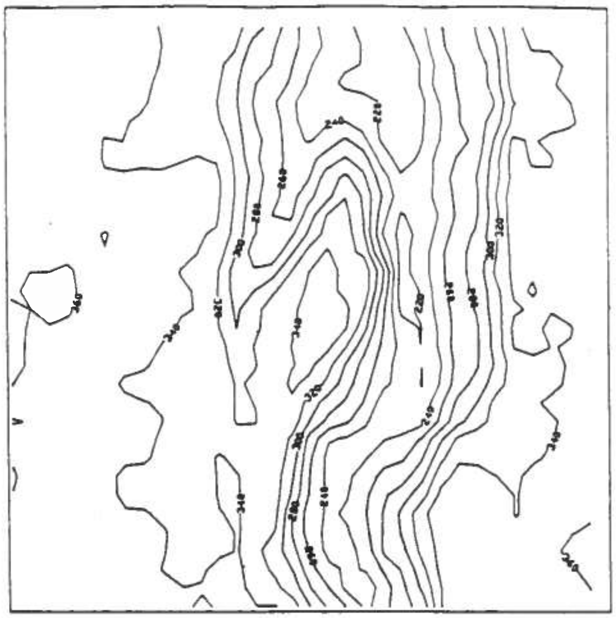

Figure 49: Reconstructed contour map of window 3 from extremas with $\sigma = 2$

which are essential in surface description. On the other hand, the plot of zero-crossings in Figure 29 to Figure 34 do not give as satisfying results as we acquire from the extremas plots. It is interesting to note that breaklines along the bottom of valleys have positive maximum convoluted values while the line along the rim of valleys have negative minimum convoluted values. This information can be exploited for further interpretation of terrain. Scale space plots of extremas are presented in Figure 24 to Figure 25. Clearly, extremas across the scale do not exhibit the same properties as zero-crossings.

In order to show that extremas maps should be considered as a compression set of data of the terrain, contour maps of all three windows are reconstructed with the same contouring package used in generated the original maps. Figure 41 to Figure 43 present the contour maps constructed from zero-crossings with $\sigma = 4$ to $\sigma = 2$. Figure 44 to Figure 49 show the maps constructed from extremas. The results show that maps reconstructed from extremas give much better results than the ones acquired from zero-crossings. Considering maps reconstructed from extremas with $\sigma = 4$ and $\sigma = 2$, one can see that they give terrain description similar to the original ones.

There is one important thing that should be said at this point that extremas

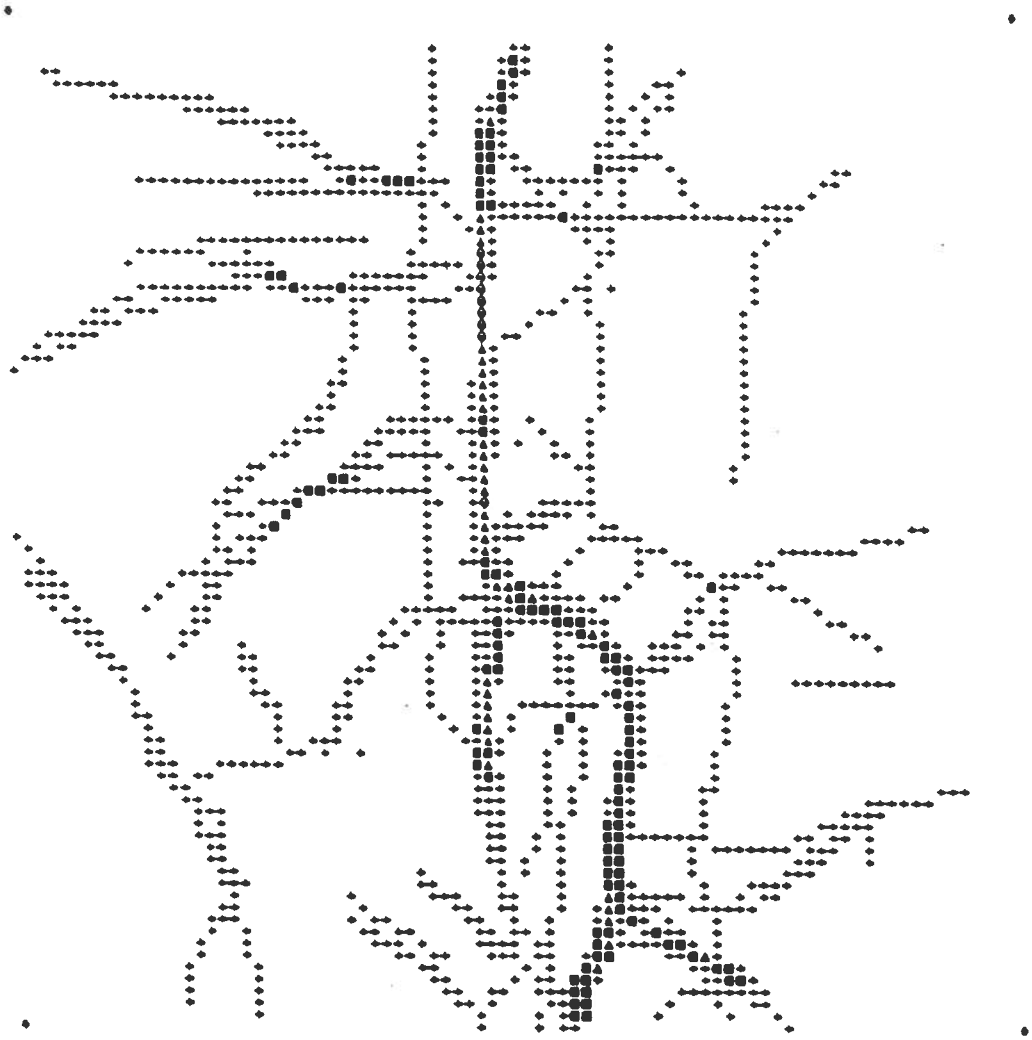

Figure 50: Manual breaklines map of window 1

◆ : $n \leq 3$, ■ : $3 < n \leq 6$, ▲ : $6 < n \leq 9$, ● : $n > 9$

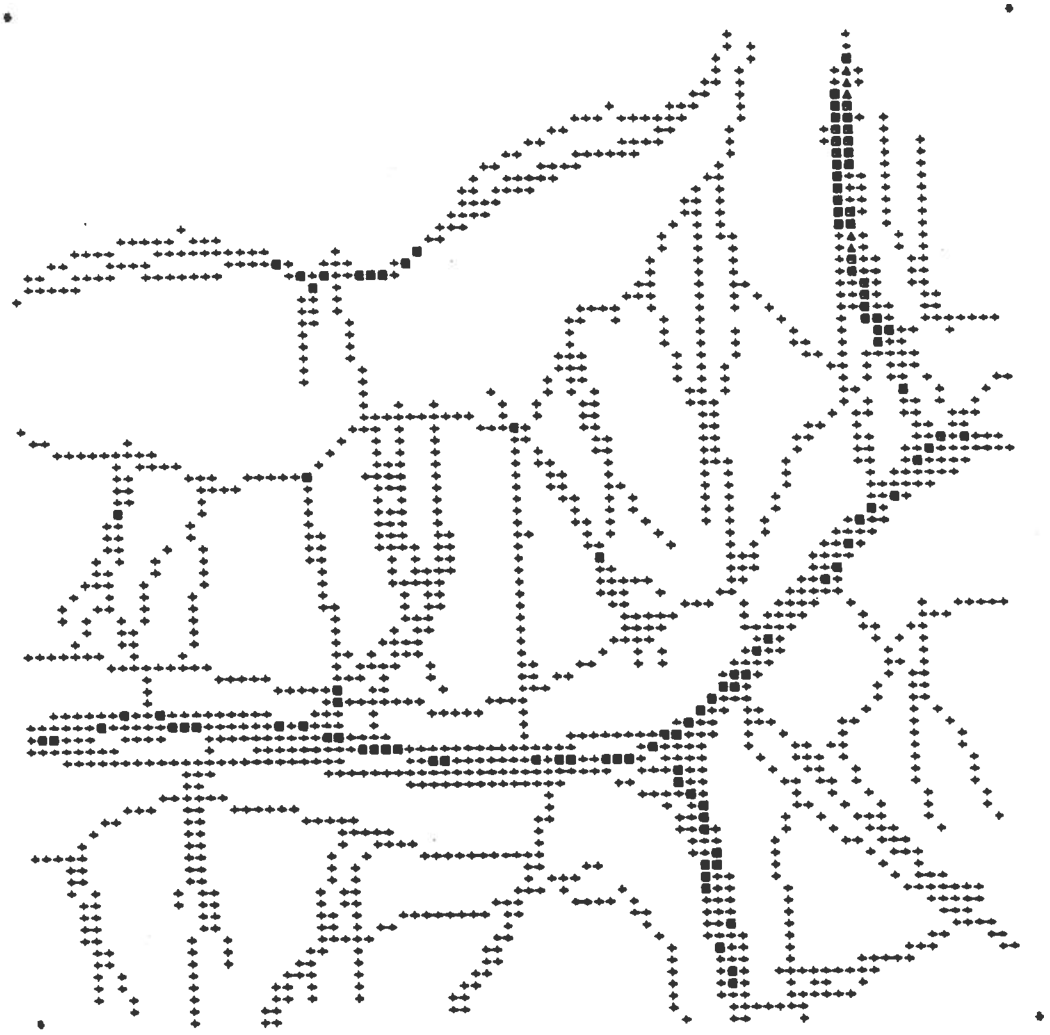

Figure 51: Manual breaklines map of window 2

+ : $n \leq 3$, ■ : $3 < n \leq 6$, ▲ : $6 < n \leq 9$, ● : $n > 9$

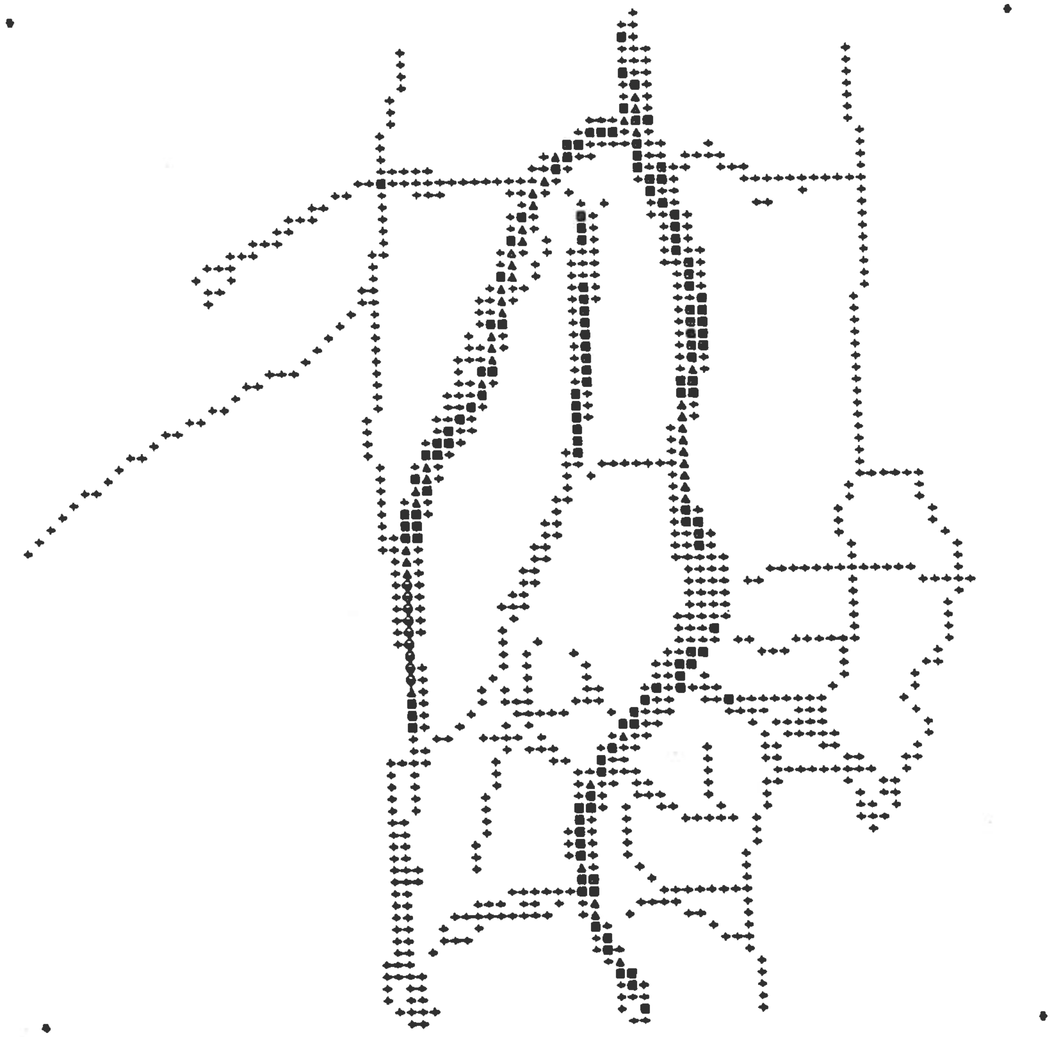

Figure 52: Manual breaklines map of window 3

+ : $n \leq 3$, ■ : $3 < n \leq 6$, ▲ : $6 < n \leq 9$, ● : $n > 9$

are the points obtained by finding points having negative minimum or positive maximum convoluted values between zero-crossings in horizontal and vertical directions. Therefore, if zero-crossings due to noise are not removed, the extremas will be false breakpoints. As a result, more redundant points will be obtained. To overcome this problem, authentic zero-crossings occurring across the scales must be detected and noisy ones must be eliminated. This task needs more exploration in terms of finding the correspondence zero-crossings across the scales; and it is still open for further research.

Considering Table 5, it is appropriate to say that the compression rate seems to be terrain-type dependent. This means that for the smooth terrain, such as window 2, this technique can give a high compression rate. It can also be simply explained by the rule of thumb that we need more data to describe rough terrain.

Finally, the manual breakline maps, shown in figure 50 to figure 52 are compared with the ones obtained by the proposed method. As one can expect, the results from different persons reflect his or her background, knowledge and experience. The results presented in the figures show all pixels chosen by the sampler as breaklines and breakpoints with the number of people who choose those pixels. It is clear that the results from the proposed technique shown on, Figure 29, 31 and 33 give the nearly complete descriptions. The detected breakpoints are located accurately about 1-2 pixels away from the ones most selected by human. Nevertheless, for window-2, some breakpoints are missing at the three way breakline junction, even at very small filter size ($\sigma = 2$). This is the effect created by the drawback of zero-crossings, as presented in Chapter 3, and contributes to the occurrence of extremas.

CHAPTER 6
CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

This work presents a new approach for solving the problem of the huge amount of data of the discrete form of DEM representation. Extracting breakpoints from the dense regular grid DEM data is considered as a process of DEM data compression. All breakpoint information is explicit information which should be kept as DEM data in the data-base instead of storing data in a regular grid pattern which are implicit and occupy much computer storage space.

It has been known for a long time that breakpoints and breaklines are important entities in surface description. Unfortunately, there does not exist a satisfactory technique to extract breakpoints from a given set of DEM data, due to the fact that detecting breakpoints is a subjective and scale-dependent task.

This research presents a new method of extracting breakpoints and breaklines from the regular grid DEM data by using the LoG (Laplacian of Gaussian) edge detection operator. By regarding DEM data as the input signal, points having local maximum curvature can be detected. These correspond to the points having positive maximum and negative minimum convoluted value of the signal between zero-crossings of LoG.

Experiments show that both drawbacks of breakpoint, which are subjective and scale-dependent properties, can be overcome. In addition, it may be possible to achieve an automatic technique. The detected breakpoints are the explicit information which has more meaning than just regular grid points. The new method described above is able to detect breakpoints accurately to 1-2 grid size for this particular case, compared with the points chosen by humans. Contour maps reconstructed from detected breakpoints confirm the possibility of using the concept of breakpoints as a surface description. Moreover, information from detected breakpoints can also be utilized for further interpretation such as direction of slope, pit, peak, ridge lines and valley lines.

6.2 Recommendations

This research is only a first step in the proposed conceptional approach, using breakpoints and breaklines as a surface description. However, more work needs to be done to reach a workable operational stage. Specific recommendations for further work are given below.

First, other edge operators can be applied and compared with the method presented here. However, they must be carefully implemented. As stated earlier, the ramp edge and roof edge shape of terrain are common while the step edge shape is rare. Consequently, edge operators, which normally are concerned with step edges must be carefully studied and implemented. Since edge operators have different drawbacks, some of them might give a better spatial location than the others. Nevertheless, every aspect must be evaluated and compared, for example, computation time, compression rate, accuracy, the possibility of further terrain interpretation.

Second, the improvement of the method presented here can be achieved by eliminating false zero-crossings due to noise. This can be done by finding correspondence across scales of authentic zero-crossings. The problem of finding the corresponding zero-crossings is a problem by itself which needs deeper investigation. This topic is still open for further research, and its outcome will contribute extensively to the implementation of LoG. Compression rate of the method can be increased if only extremas between real zero-crossings are used instead of all extremas from every zero-crossing.

Finally, the possibility of using LoG for DEM generalization should be explored. This is due to the fact that the smoothness of surface can be controlled by the size of filter or σ of LoG. Breakline maps at the various scales which are the DEM surface description can be obtained by convolving the surface with different values of σ. The concept of using LoG to smooth the signal and locate the critical points in one dimension has been successfully implemented in line generalization in [Thapa 87].

APPENDIX A

RELATIONSHIP BETWEEN σ AND f OF GAUSSIAN FUNCTION

In this appendix, the relationship between σ and f, i.e., parameters in the spatial and frequency domains of the Gaussian function, will be derived. This relationship is very important for determining how large an LoG should be used in the DEM data compression procedure. The derivation is conducted in one dimension starting from the Gaussian function in spatial domain.

The Gaussian function in one dimension can be written as:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{22}$$

Therefore, the Fourier Transform of the Gaussian function can be obtained as follows:

$$\overline{G}(f) = \int_{-\infty}^{+\infty} G(x) \exp\left(-\imath\, 2\pi f x\right) dx$$

where, f = frequency
$\imath = \sqrt{-1}$

Now, we can write the above equation in the form of:

$$\overline{G}(f) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left(-\frac{x^2 + \imath\, 4\pi f x \sigma^2}{2\sigma^2}\right) dx \tag{23}$$

We can write:

$$x^2 + \imath\, 4\pi f x \sigma^2 = (2\pi f \sigma^2)^2 + (x + \imath\, 2\pi f \sigma^2)^2$$

therefore, equation 23 can be rewritten as follows:

$$\overline{G}(f) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left(-\frac{(2\pi f\sigma^2)^2}{2\sigma^2}\right).\exp\left(-\frac{(x + \imath\, 2\pi f\sigma^2)^2}{2\sigma^2}\right) dx$$

$$\overline{G}(f) = \frac{\exp\left(-\frac{(2\pi f\sigma^2)^2}{2\sigma^2}\right)}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left(-\frac{(x + \imath\, 2\pi f\sigma^2)^2}{2\sigma^2}\right) dx \qquad (24)$$

Let's define $S = x + \imath\, 2\pi f\sigma^2$. Therefore, we can write:

$$dS = dx$$

Then equation 24 can be written:

$$\overline{G}(f) = \frac{\exp\left(-\frac{(2\pi f\sigma^2)^2}{2\sigma^2}\right)}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left(-\frac{S^2}{2\sigma^2}\right) dS$$

$$\overline{G}(f) = \frac{\exp\left(-\frac{(2\pi f\sigma^2)^2}{2\sigma^2}\right)}{\sigma\sqrt{2\pi}}.\sqrt{2\pi\sigma^2}$$

$$\overline{G}(f) = \exp\left(-2\pi^2 f^2\sigma^2\right) \qquad (25)$$

Equation 25 describes the relationship between σ and f. One can see that the relationship is also dependent on the value of $\overline{G}(f)$. If we continue expanding equation 25, we will get:

$$\overline{G}(f).\exp\left(2\pi^2 f^2\sigma^2\right) = 1$$

$$\ln\overline{G}(f) + 2\pi^2 f^2\sigma^2 = 0$$

$$\sigma^2 = \frac{\ln\left(\frac{1}{\overline{G}(f)}\right)}{2\pi^2 f^2}$$

$$\sigma = \frac{\sqrt{\ln\left(\frac{1}{\overline{G}(f)}\right)}}{\pi f\sqrt{2}} \tag{26}$$

From equation 26, if the value of $\overline{G}(f)$ is chosen the σ can be computed from the cut-off frequency f.

In order to choose the value of $\overline{G}(f)$, one may select a small value, such as 0.0001, or by considering the confidence level percentage. However, as stated by [Huertas and Medioni 86], the maximum frequency passing through the LoG filter is equal to three times the frequency at the maximum response, which happens at:

$$f = \frac{1}{\sigma\pi\sqrt{2}}$$

Therefore, we can write:

$$f_{max} = \frac{3}{\sigma\pi\sqrt{2}}$$

Or,

$$\sigma = \frac{3}{\pi f_{max}\sqrt{2}} \tag{27}$$

Now, if we compare equation 26 with equation 27, we will get:

$$\sqrt{\ln\left(\frac{1}{\overline{G}(f)}\right)} = 3$$

$$\ln\left(\frac{1}{\overline{G}(f)}\right) = 9$$

$$\overline{G}(f) = \frac{1}{\exp(9)}$$

$$\overline{G}(f) \approx 0.00012341$$

One may use the value of $\overline{G}(f)$ above in the process of designing σ from f. This value of $\overline{G}(f)$ is equal to 99% of the confidence level of the normal distribution which one can also determine by the usual statistical methods.

References

Allder, W. R. , Caruso, V. M. , Pearsall, R. A. and Troup, M. I. *An Overview of Digital Elevation Model Production at the United States Geological Survey.* Proc. Auto Carto V. Environmental Assessment and Resource Management, ASP-ACSM, August 1982.

Attneave, F. *Some Information Aspects of Visual Perception.* Psychological Review, 61, pp. 183-193, 1954.

Ayeni, O. O. *Objective Terrain Description and Classification for Digital Terrain Models.* XIII congress of ISP, Comm. III, Helsinki, Finland, 1976.

Ayeni, O. O. *Optimum Sampling for Digital Terrain Models: A trend Toward Automation.* PERS Vol. 48, No. 11, pp. 1687-1694, November 1982.

Babaud, J., Witkin, A.P., Baudin, M. and Duda, R.O. *Uniqueness of Gaussian Kernel for Scale Space Filtering.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 1, pp. 26-33, January 1986.

Balce, A. *Determination of Optimum Sampling in Grid Digital Elevation Models Data Acquisition.* Proc. ISPRS Comm. III Symposium, Finland, Int. Archives of Photogrammetry and Remote Sensing. Vol. 26, Part 3.1, pp. 40-55, 1986.

Ballard, D. H. and Brown, C. M. *Computer Vision.* Prentice Hall Inc., New Jersey, 1982.

Bergholm, Fredrik. *Edge Focusing.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 6, November 1987.

Berzins, V. *Accuracy of Laplacian Edge Detectors.* Computer Vision Graphics and Image Processing 27(2), pp. 195-210, 1984.

Brady, Michael. *Computational Approaches to Image Understanding.* Computing Surveys, Vol. 14, No. 1, pp. 3-71, March 1982.

Burrough, P. A. *Principle of Geographical Information Systems for Land Resources Assessment.* Clarendon Press, London, 1986.

O'Callaghan, John. F. and Mark, David. M. *The Extraction of Drainage Networks from Digital Elevation Data.* Computer Vision, Graphic and Image Processing 28, pp. 323-344, 1984.

Canny, J. F. *Finding Edge and Lines in Image.* MIT. Artificial Intelligence Laboratory Technical Report No. 720, 1983.

Canny, J. F. *A Computational Approach to Edge Detection.* IEEE Transactions on Pattern and Machine Intelligence. Vol. 8, No. 6, November 1986.

Charif, M. and Makarovic, B. *Optimum Progressive and Composite Sampling for Digital Terrain Model.* Proc. ISPRS Comm. III Symposium, Kyoto, Int. Archives of Photogrammetry and Remote Sensing, Vol. 27, Part. B10, pp. 264-280, 1988.

Chen, J.S., Huertas, A. and Medioni, G. *Very Fast Convolution with Laplacian-of-Gaussian Masks.* IEEE Proc. Computer Vision and Pattern Recognition, pp. 293-298, 1986.

Chen, J.S. and Medioni, G. *Detection, Location and Estimation of Edges.* IEEE Proc. of Image Understanding Workshop, Vol. 2, DARPA, pp. 988-1000, Febuary 1987.

Clark, James. J. *Singularity Theory and Phantom Edges in Scale Space.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 5, pp. 720-727, September 1988.

Clark, James. J. *Authenticating Edges Produced by Zero-Crossing Algorithms.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 1, pp. 43-57, January 1989.

Clarke, Keith. C. *Strategies for Spatial Data Compression.* Proc. Digital Representations of Spatial Knowledge, Auto. Carto. VII, pp. 97-107, 1985.

Clauss, Michael. *Digital Terrain Model Through Digital Correlation.* Photogrammetria, No. 39, pp. 183-192, 1984.

Collins, S. H. and Moon, G. C. *Algorithms for Dense Digital Terrain Models.* PERS Vol. 47, No. 1, pp. 71-76, January 1981.

Davis, John. C. *Statistics and Data Analysis in Geology, 2nd Edition.* John Wiley & Sons, New York, 1986.

Davis, L.S. *A Survey of Edge Detection Techniques.* Computer Graphics and Image Processing, No. 4, pp. 248-270, 1975.

Douglas, David. H. *Experiments to Locate Ridges and Channels to Create a New Type of Digital Elevation Model.* Canadian Surveyor, Vol. 41, No. 3, pp. 373-406, 1987.

Doyle, Frederick J. *Digital Terrain Models: An Overview.* PERS Vol. 44, No. 12, pp. 1481-1485, December 1978.

Elassal, Atef. A. *USGS Digital Cartographic File Management System.* Proc. Digital Terrain Model Symposium, St. Louis, MO. May 1978.

Elassal, Atef. A. and Caruso, Vincent M. *U.S.G.S. Digital Cartographic Standards. Digital Elevation Models.* U.S. Geological Survey Circular 895-B, 1983.

Frederiksen, P., Jacobi, O., and Kubik, K. *Measuring Terrain Roughness by Topological Dimension.* ISPRS Working Group III/3 Colloquium, Stockholm, Sweden, April 18-20, 1983.

Frederiksen, P., Jacobi, O., and Kubik, K. *Modelling and Classifying Terrain.* ISPRS Congress, Commission III, Rio de Janeiro, Brazil, July 1984.

Greenfeld, J.S. *A Stereo Vision Approach to Automatic Stereo Matching in Photogrammetry.* Ph.D. Dissertation, Dept. of Geodetic Science and Surveying, The Ohio State University, 1987.

Hahn, M. and Förstner, W. *The Applicability of a Feature Based and a Least Squares Matching Algorithm for DEM-Acquisition.* Proc. ISPRS Comm. III Symposium, Kyoto, Int. Archives of Photogrammetry and Remote Sensing, Vol. 27, Part B9, pp. 137-150, 1988.

Haralick, R.M. *Digital Step Edges from Zero-crossing of Second Directional Derivatives.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, No. 1, January 1984.

Hardy, R.L. *Least Squares Prediction.* PERS Vol. 43, No. 4, pp. 475-495, April 1977.

Hildreth, Ellen C. *Implementation of a Theory of Edge Detection.* MIT Report, AI-TR-579, April 1980.

Hildreth, Ellen C. *Edge Detection in Man and Machine.* Robotic Age, Vol. 3, No. 3, Sept-Oct 1981.

Hildreth, Ellen C. *The Detection of Intensity Change by Computer and Biological Vision System.* Computer Vision, Graphic & Image Processing No. 22, pp. 1-27, 1983.

Hildreth, Ellen C. *Edge Detection.* MIT. Artificial Intelligence Laboratory Memo. No. 858, September 1985.

Huertas, A. and Medioni, G. *Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks.* IEEE Transaction on Pattern Analysis and Machine Intelligence. Vol. 8, No. 5, September 1986.

Kubik, K. and Roy, B.C. *Digital Terrain Model Workshop Proceedings, ISPRS Working Group III/3.* Dept. of Geodetic Science and Surveying, The Ohio State University, 1986.

Kubik, K. and Fredericsen, P. *Digital Elevation Models: Review and Outlook.* ASPRS/ACSM Spring Convention Technical Paper, 1987.

Levine, M.D. *Vision in Man and Machine.* McGraw Hill Inc., 1985.

Light, Donald L. *Mass Storage Estimates for the Digital Mapping Era.* PERS, Vol. 52, No. 3, March 1986.

Loon, J.C. *Class-Notes, Geodetic Science 631* Dept. of Geodetic Science and Surveying, The Ohio State University, 1985.

Lu, Yi and Jain, Ramesh. C. *Behavior of Edges in Scale Space.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 4, pp. 337-356, April 1989.

Lunscher, W. H. H. and Beddoes, M. P. *Optimal edge Detector Design I: Parameter Selection and Noise Effects.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 2, pp. 164-177, March 1986.

Lunscher, W. H. H. and Beddoes, M. P. *Optimal Edge Detector Design II: Coefficient Quantization.* IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 8, No. 2, pp. 178-187, March 1986.

Makarovic, B. *Progressive Sampling for Digital Terrain Models.* ITC Journal 1973-3.

Makarovic, B. *Composite Sampling for Digital Terrain Models.* ITC Journal 1977-3.

Makarovic, B. *Regressive Rejection–A Digital Data Compression Technique.* ASP/ACSM Fall Technical Meeting, Little Rock, U.S.A. 1977.

Makarovic, B. *A Test on Compression of Digital Terrain Model Data.* ITC Journal 1983-2.

Makarovic, B. *Automatic Production of DTM Data Using Digital Off-line Technique.* Proceeding ISPRS Congress Rio de Janeiro, 1984.

Makarovic, B. *Performance of Digital Off-line Systems for Automatic Production of DTM Data.* ITC Journal 1984-2.

Mark, David M. *Concepts of DATA STRUCTURE for Digital Terrain Model.* Proc. Digital Terrain Model Symposium, ASPRS-ACSM, St. Louis, MO, May 1978.

Marr, David and Hildreth, Ellen C. *Theory of Edge Detection.* Proc. of the Royal Society of London. B 207, pp. 187-217, 1980.

Marr, David. *Vision.* W.H. Freeman Co., San Francisco. 1982.

Marr, David and Poggio, T. *A Computational Theory of Human Stereo Vision.* Proc. of Royal Society of London, B 204, pp. 301-328, 1979.

Morrisón, J.L. *A Link Between Cartographic Theory and Mapping Practice - The Nearest Neighbor Statistic.* Geographical Review, Vol. 60, 1970.

Morrison, Joel L. *Method-Produced Error in Isarithmic Mapping.* American Congress on Surveying and Mapping, Technical Monograph No. CA-5, Washington, D.C., 1971.

Nalwa, V. S. and Binford, T.O. *On Detecting Edges.* IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 8, No. 6, November 1986.

Nishihara, H.K. and Larson, N.G. *Towards a Real-Time Implementation of the Marr-Poggio Stereo Matcher.* Proc. DARPA Image Understanding Workshop, Washington, D.C., April 1981.

Persoon, E. *A New Edge Detection Algorithm and Its Applications.* Computer Graphics and Image Processing, No. 5, pp. 425-446, 1976.

Peucker, Thomas and Chrisman, Nicholas. *Cartographic Data Structure.* American Cartographer, Vol. 2, No. 1, pp. 55-69, 1975.

Peucker, Thomas K., Fowler, Robert J., Little, Jame J. and Mark, David M. *Digital Representation of Three Dimensional Surfaces by Triangulated Irregular Network(TIN).* Tech. Report. 10, ONR Contract # N00014-75-C-0886, Dept. of Geography, Simon Fraser University, Burnady, B.C., Canada, 1977.

Peucker, Thomas K., Fowler, Robert J., Little, Jame J. and Mark, David M. *The Triangulated Irregular Network.* Proc. Digital Terrain Models (DTM) Symposium, ASP-ACSM, pp. 516-540, 1978.

Press, William H., Flannery, Brian P., Teukolsky, Saul A. and Vetterling, William T. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, New York, 1988.

Roberts, L.G. *Machine Perception of Three Dimensional Solids.* Optical and Electro-Optical Information Processing, MIT Press, pp. 159-197, 1965.

Rosenfeld, A. and Kak, A.C. *Digital Picture Processing.* 2nd Edition, New York, Academic, 1982.

Roy, B.C. *Compression of Equally Spaced Digital Elevation Model (DEM).* Ph.D. Dissertation, Dept. of Geodetic Science and Surveying, The Ohio State University, 1987.

Schenk, A. *Eine Methode Zur Bestimmung von Signaländerungen und Anwendungsbeispiele.* Zeitschrift für Vermessungswesen, Heft 4, April 1987.

Schreiber Instruments, Inc. *QuickSurf Version 2.* Denver, Colorado, 1988.

Thapa, K. *Detection of Critical Points: The First Step to Automatic Line Generalization.* Dept. of Geodetic Science and Surveying, The Ohio State University, Report No. 379, 1987.

Torre, Vincent and Poggio, Tomaso A. *On Edge Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 2, pp. 147-163, March 1986.

Ulupinar, Fatih and Medioni, Gérard. *Estimation and Accurate Localization of Edges.* IEEE, Proc. Image Understanding Workshop, Vol. 2, DARPA, pp. 968-980, February 1987.

Ulupinar, Fatih and Medioni, Gérard. *Refining Edges Detected by* LoG *Operator.* IEEE, Proc. of CVPR, pp. 202-207, 1988.

Weaver, H. Joseph. *Applications of Discrete and Continuous Fourier Analysis.* John Wiley & Sons, Inc., 1983.

Witkin, A.P. *Scale-Space Filtering.* Proc. of the International Joint Conference on Artificial Intelligence, Karlsruhe, pp. 1019-1022, 1983.

Yuille, A.L. and Poggio, T. *Fingerprints Theorems for Zero-Crossings.* A.I. Memo No. 730, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, October 1983.

Yuille, A.L. and Poggio, T. *Scaling Theorems for Zero-crossings.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 1, pp. 15-25, January 1986.