1989

Coverage And Visibility Problems On Topographic Surfaces

Jay Lee

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

COVERAGE AND VISIBILITY PROBLEMS ON TOPOGRAPHIC SURFACES

by

Jay Lee

Department of Geography

Submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
June 1989

ISBN 0-315-51717-4

ABSTRACT

This thesis discusses and defines visibility coverage problems by using visibility information derived from topographic surfaces. Algorithms for extracting visibility information are developed and tested on both randomly generated and real topographic surfaces to solve visibility coverage problems, such as the determination of visible regions for specific viewpoints, the determination of a minimal set of viewpoints to see the entire surface. Problems are also extended to the cases of viewpoints whose heights are allowed to vary.

A triangulated irregular network (TIN) model was selected to represent topographic surfaces in this study. A drop heuristic algorithm for extracting TIN models from grid digital elevation models (DEM) was also developed and presented. Heuristic algorithms are used to solve visibility coverage problems and are compared for their performances. Solutions to visibility coverage problems suggested that peaks and vertices on ridge lines are found to be most likely good candidates as solution viewpoints. Relief variations on topographic surfaces are found to have a positive relationship with the number of viewpoints as well as the cost of building watchtowers required to see the entire surface. Parameters describing topographic surfaces, such as statistics of

elevations and slopes of triangles, proportions of peak and ridge points in a set of vertices, the spatial autocorrelation coefficient, and the parameters of variograms are found to have similar relationship.

In loving memory of my mother, Fung Lan Hwang Lee.

ACKNOWLEDGEMENTS

I wish to thank my chief advisor, Dr. Michael Goodchild, for his support, intelligent guidance, and patience during my years at Western. I am also indebted to members of my advisory committee, Dr. D. Janelle and Dr. C. Smart, for their stimulation, encouragement and criticism on earlier versions of this thesis. I wish also to acknowledge the kind assistances I received from Dr. B. Klinkenberg for providing me DEM data.

Faculty members, staff, and my fellow graduate students in geography at Western never failed to provide warmth and friendship, sound advice and help whenever possible. In particular, I would like to extend my gratitude to Ms. Diane Shillington.

Special thanks to my wife, Jenny Lin. Without her support and care, my existence and the completion of this thesis would not have been possible.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER I INTRODUCTION

Terrain analyses have been a primary interest of many geographers since the introduction of digital computers and the concept of Digital Elevation Model in the 1960s. A wide variety of techniques is now available for applications in geography and other related fields. Most terrain analyses are based on various approximation models of topographic surfaces and information derived directly or indirectly from these models. Among many types of information which can be derived from these models, visibility information is particularly of interest in this thesis as it is useful for a variety of applications in such areas as navigation, geomorphology, terrain exploration, or visibility coverage problems.

This thesis discusses and defines visibility coverage problems by using visibility information derived from terrain surfaces. Algorithms are developed and tested on real topographic surfaces to solve visibility coverage problems, such as the determination of visible regions for specific viewpoints or the determination of a minimal set of viewpoints to see the entire surface, by combining knowledge in operations research, geographical information systems, and geographical research.

The aim of analyzing a surface by using visibility information is to produce a description that does not differ significantly from that given by a human observer viewing a surface. A visibility-based description of a surface consists of a collection of visible regions computed with respect to a set of points. The visible region of a viewpoint on a topographic surface, expressed as $z=f(x,y)$, is the set of projections on the x-y plane of the points on the surface that can be joined to the viewpoint by means of a straight-line segment that does not intersect the surface. By approximating surfaces with digital elevation models, visibility information may be computed for the following applications:

1) **Visibility coverage problems**. An example of these problems is the determination of the minimum number of observation points (e.g. watchtowers or firetowers) necessary to view an entire region. One might also be interested in scenic locations, or in paths with specified visibility characteristics.

2) **Line-of-sight communications**. One can determine locations for the minimum number of television transmitters for an area, or the optimal location of receivers. With portable transceivers, one might be interested in the locus of travel of a party, each of whose members must remain in uninterrupted communication with the others. The location of

radar, laser, and sonar surveillance systems also belong to this category.

3) **Orientation and navigation**. The profile of the horizon is a natural and simply extracted measurement, derivable directly from information of visibility regions that can be readily used by an observer to locate him/herself with respect to a topographic map. Other examples include weapons guidance systems, pilot-training simulations, and the like.

4) **Representation of physiography**. Visibility considerations may suggest which points can be eliminated from a digital elevation model while preserving visibility characteristics. Examples include landscape design and planning (landscape architecture), displaying thematic information (or combined with thematic data such as soils, land use or vegetation), and so on.

5) **Extraction of significant terrain features**. The location and relation of visible regions provide adequate information for determining important topographic terrain features, such as peaks, pits, ridges, and valleys. Some connections between visibility and topography are the following. "Points where the immediate neighbourhood is invisible are convexities: many adjacent convexities constitute

> a dome. Large, multiply-connected vistas are properties of dominant peaks and ridges. In pits and valleys, the prospect is singly-connected and tends to change gradually. Thus visibility considerations suggest where the valley ends and the mountain begins. If two points have the same visible region, then they are in the same valley; if they do not, then there is a ridge between them. Horizons that form the common boundary of the visible regions of many observation points are usually significant ridges." (de Floriani and Nagy 1989, pp. 7).

This thesis attempts to address visibility coverage problems on topographic surfaces. In this study, topographic surfaces are approximated by a set of vertices and facets constructed between these vertices. Various types of topographic surfaces are used to explore the relationship between the visibility information and geomorphological features of the surfaces. It is also the intention of this thesis to examine the association between terrain surfaces with various geomorphological features and different patterns of visibility information in the studied areas.

A visibility-based model on an approximated topographic surface, described in chapter III, was first suggested by de Floriani and others (1986), who also noted the possibility of

using a heuristic approach to solve the visibility coverage problems being discussed in this thesis. Since then several possible formulations and applications have been explored. Studies have been conducted in the area of extracting topographic features by using visibility information (Nair 1988) and in the area of visibility problems on polyhedral terrains (Cole and Sharir 1986, Sharir 1987). However, visibility coverage problems on rugged topographic surfaces are still unanswered as it would require more research work, not only in developing algorithms for solutions to the problems but also in the areas of better approximation of topographic surfaces. In addition, the potential of exploring the relationship between types of physiography and patterns of visibility information was never attempted until this study.

In the visibility model considered, the regions that are visible from each point can be computed from a faceted approximation of the surface such as one based on Delaunay triangulation of the projections of data points. Problems such as the best location for observation towers can be determined using the heuristic approach of set-covering algorithms.

This thesis first reviews the literature on related topics in chapter II, including discussion of the various forms of digital elevation models available to be adapted in this thesis; existing studies on the concepts and algorithms

of extracting visibility regions; and the origins of set-covering problems as well as usable heuristics for solutions of coverage problems. The chapter also contains a discussion of reasons for selecting a **Triangulated Irregular Network (TIN)** constructed by Delaunay triangulation as a better-suited approximation model for the topographic surfaces studied.

Chapter III starts with a definition of visibility and its functions. Visibility and related coverage problems are defined and classified, first based on a continuous visibility model and then revised for a discrete one. The definition and classification of the continuous model are basically a generalization of visibility coverage problems; while the revised discrete version of visibility coverage problems serves as a realization of the continuous model and reduces the problems to a manageable size.

Chapter IV discusses the sample problems, which include 30 simplified data sets and 4 U.S.G.S. 7.5 minute quadrangle DEMs of real topographic surfaces in the areas of Cedar Breaks, Utah, Krone Ranch, Montana, Herbert Domain, Tennessee, and Benndale SW, Mississippi. Data sets are used to verify the visibility concepts and to test the feasibility of algorithms proposed for the visibility coverage problems discussed in this thesis. The chapter also presents a new algorithm which is used to convert the U.S.G.S. grid DEMs to TINs. A comparison analysis between existing and new conversion algorithms is also included.

Chapter V presents algorithms for extracting the visibility regions of given viewpoints and algorithms for computing the minimal visible heights (defined in section 5.3) between viewpoints and regions on the surface. The following chapter describes how visibility information may be structured as visibility matrices. Two types of visibility matrices record either visibility information or the minimal visible heights between viewpoints and are in turn used to solve visibility coverage problems of fixed and variable heights. Solutions of these problems are presented in the same chapter.

Also in chapter VI, the solutions of visibility coverage problems indicate that viewpoints selected are geomorphologically significant points such as peaks or points along ridge lines. These findings strongly suggest the existence of a close relationship between geomorphological features on terrain surfaces and visibility information. In a broader sense, flat land surfaces undoubtedly require fewer viewpoints or watchtowers to monitor the surfaces; while land surfaces with more rugged reliefs will require more viewpoints or watchtowers for the same purpose. A similar contrast might also be found between fluvial and glacial land surfaces. Moreover, terrain surfaces with specific geomorphological features are assumed to be associated with different patterns of visibility information. For instance, to see the entire surface, a typical deep-sloped Karst landscape would require more viewpoints to be selected on peaks or pits, compared to

a typical drainage landscape where viewpoints would be found to be best located along either ridge lines or basin boundaries.

While the approach of using visibility information for terrain analyses is relatively new in the geographical research community, many potential applications are shown to be useful and feasible. The contributions of this thesis are both conceptual and empirical. The formulation of visibility coverage problems and verification of the feasibility of using heuristic approaches to solve them hint at a clear passage toward many practical applications, such as extracting significant terrain features, line-of-sight communications, orientation and navigation, or representation of physiography.

The development and the implementation of new algorithms for extracting visibility regions, of computing the minimal visible heights, and of converting gridded DEMs to TINs give good examples of how a visibility model may be used for terrain analyses.

Finally, and most importantly, the relationships between various geomorphological features and different patterns of visibility information hinted by the results of solving visibility coverage problems indicate a feasible research direction toward more exciting and meaningful geographical terrain analyses.

CHAPTER II LITERATURE REVIEWS

Although analyses of digital elevation models to obtain parameters such as slope, convexity, concavity, and aspect started in the early seventies, it is only recently that visibility coverage problems have been described. Using the visibility information to describe a land surface is useful for a variety of applications in geomorphology, navigation, or terrain exploration. Studies have used visibility information to detect nodal and linear features of a topographic surface (Nair 1988) and boundaries of visibility regions to assist navigation and terrain exploration (de Floriani and Nagy 1989).

Interest in visibility problems is not new; however, the approach of relating it to geographic analyses is more recent, as is the attempt to use heuristic approaches to solve particular problems. Therefore, the survey and review of literature draws on related work from each of the relevant research fields.

This chapter reviews existing literature on digital elevation models: origins, different forms of models, and suitability for this study. Studies relevant to visibility regions (visibility algorithms, problems, and applications) are also reviewed. Finally, the origin and development of coverage problems in operations research, from p-median

problems to set-covering problems and suggested heuristic solutions, are also reviewed.

2.1. Digital Elevation Models

Any digital representation of the continuous variation of relief of the earth's surface may be referred to as a **digital elevation model(DEM)**[1]. Digital elevation models can look back on a long history of growth and diversification. In the late 1950s, civil engineers used computers for the calculation of terrain profiles across highway trusses. Roberts and Suhrbier (1962) at MIT, were the first to incorporate these profiles to interpolate along and between profiles for the computation of cut-and-fill and other data useful for civil engineers. From then on, DEMs have received much attention in the research community, especially in geodesy and photogrammetry.

Digital elevation models have many uses in current geographical research and offer functions in storing, analyzing, and displaying information about a land surface. Among the most important uses are the following (from Burrough 1986, pp.39):

---

[1]The term, digital terrain model (DTM) is also commonly used but DTM implies attributes of a landscape other than the altitude of the land surface. The term DEM is preferred for models containing only elevation data (Burrough 1986).

1. Storage of elevation data for digital topographic maps in national databases.
2. Cut-and-fill problems in road design and other civil and military engineering projects.
3. Three-dimensional display of landforms for military purposes (weapon guidance systems, pilot training) and for landscape design and planning (landscape architecture).
4. For analysis of cross-country visibility (also for military and for landscape planning purposes).
5. For planning routes of roads, locations of dams, etc.
6. For statistical analysis and comparison of different kinds of terrain.
7. For computing slope maps, aspect maps, and slope profiles that can be used to prepare shaded relief maps, assist geomorphological studies, or estimate erosion and run-off.
8. As a background for displaying thematic information or for combining relief data with thematic data, such as soils, land-use or vegetation.
9. Provide data for image simulation models of landscapes and landscape processes.
10. By replacing altitude by any other continuously varying attribute, the DEM can represent surfaces of travel time, cost, population, indices of visual

beauty, levels of pollution, groundwater levels, and so on.

Good reviews and discussions of DEMs have been published by Mark (1978), Nagy and Wagle (1979), Peucker (1979), and Burrough (1986). Some of these have classified DEMs according to methods of spatial representation used. In general, DEMs can be presented either by mathematically defined surfaces or by point or line images, as in the classification shown in Table 2.1.

A. Mathematical Methods
  A1. Global method
    - Fourier Series
    - Multiquadratic polynomials
  A2. Local method
    - Regular patches
    - Irregular patches
B. Image Methods
  B1. Using point data
    Regular
    - Uniform density
    - Variable density (using regular or irregular grids)
    Irregular
    - Triangulation
    - Proximal networks (using triangles or hexagonal grids)
    Critical features
    - peaks, pits, passes, boundaries
  B2. Using line data
    Horizontal slices - (contours)
    Vertical slices - (profiles)
    Critical lines - ridges, stream courses, shorelines, breaks in slope.

Table 2.1. DEM Classification Framework
(Source: Mark 1978)

Given the problems specified in this study, it is necessary to review the suitability of each representation method available to us. Whether a representation method is suited for the type of problems in this study may be decided by several criteria:

The method to be chosen for this study must

1. store data to represent an arbitrary surface efficiently because study areas have a large numbers of data points,
2. preserve as much of the topographic features as needed to properly describe an arbitrarily rugged landscape,
3. be efficient in computing visible regions from any given observation point. Because of its expected high complexity of computation, redundancy of data storage and ambiguity of spatial representation must be avoided.

Most of the mathematical models tend to be more powerful in describing regular variation than in describing a land surface that is less regular. In addition, an equation that fits well to a real, non-regular land surface is often extremely complex and is difficult to obtain. For this reason the study excludes mathematical models for representing arbitrary land surfaces. Therefore, we are left with several

other commonly used approaches, such as contour, regular grid, and Triangulated Irregular Network (TIN).

The most common representation on a paper map is by contours, or lines connecting points of equal elevation. To construct a DEM from a contour map, one might simply digitize the contour lines as ordered sets of points, and assume adjacent pairs of points within each line to be connected by straight lines (e.g. Douglas and Peucker 1973, Davis 1975).

Although this model is the most widely used, users usually require some training to read contour maps. The major disadvantage of this approach as a digital representation is that it provides a very uneven density of information: uncertainty about a randomly chosen point's elevation is zero on each contour line, and rises directly with the point's distance from the nearest line. To obtain an accurate representation of an entire surface it is therefore necessary to use a large number of contours at very small intervals of elevation. However, this does not avoid at all the problem of carrying a huge amount of data.

The second alternative, the representation of the surface by a regular square grid of sample elevations (Collins 1975, Davis and McCullagh 1975, Sprunt 1975, and Mark 1983), gives a uniform intensity of sampling, and is therefore frequently used in practice. The regular grid model has historically been the most widely used primarily for two practical reasons:

1) it is compatible with the array data structure built into FORTRAN programming; and
2) it is compatible with a number of devices used for spatial data capture and display.

For this study, it is possible to calculate the area seen from each of the grid of sample points by interpolation. However, the ruggedness of any real topographic surface tends to vary from one part of the surface to another; some areas tend to be smooth, while in other areas elevation varies rapidly over short distances. It is for this reason that a uniform sampling density is inefficient compared to a design that responds to the variability in the surface by sampling more intensively in the more rugged areas. Moreover, it is not clear how the surface should be interpolated between grid points in a DEM of regular grid points. Every grid patch has four corner points and there is no obviously unique way of fitting a plane to this patch.

As to other representation methods belonging to this group, problems of inconsistent precision and accuracy during interpolation of elevations within each cell or tile of the surface approximation suggest that such methods are inappropriate for this study. These methods, including hierarchical grid-based representations, such as the quadtree or the k-d tree, are not adaptable directly to linear interpolation of elevation data.

The stated disadvantages of contour and grid models suggest a third alternative, which is known as the TIN or **triangulated irregular network** (see Figure 2.1). In this model the space is represented by a set of vertices and is divided into a set of triangles with shared edges, and the surface is modeled by the triangles as if they were mosaic tiles. Topology is relatively straightforward and easily verified. Each edge is defined by two vertices and is shared by exactly two triangles, with the exception of those edges which form the outer boundary of the network. Each vertex is shared by at least three triangles. In the simplest version, which is used in this study, the surface is assumed to be planar within each triangle.

The triangulated irregular network is a system designed by Peucker et al. (1978) for digital elevation modelling that avoids the redundancies of DEM data and is more efficient for many types of computation (such as slope or visibility) than systems that are based only on digitized contours.

To triangulate a set of irregularly spaced points, one may take either the approach of **Greedy Triangulation** or **Delaunay Triangulation**. Given a set of points, Greedy triangulation first computes all possible edges by connecting all pairs of points, and puts them into a stack after being sorted by distance. The algorithm begins with an empty set of candidate edges. At every iteration, an edge is selected and put into the list of candidate edges if it is the shortest

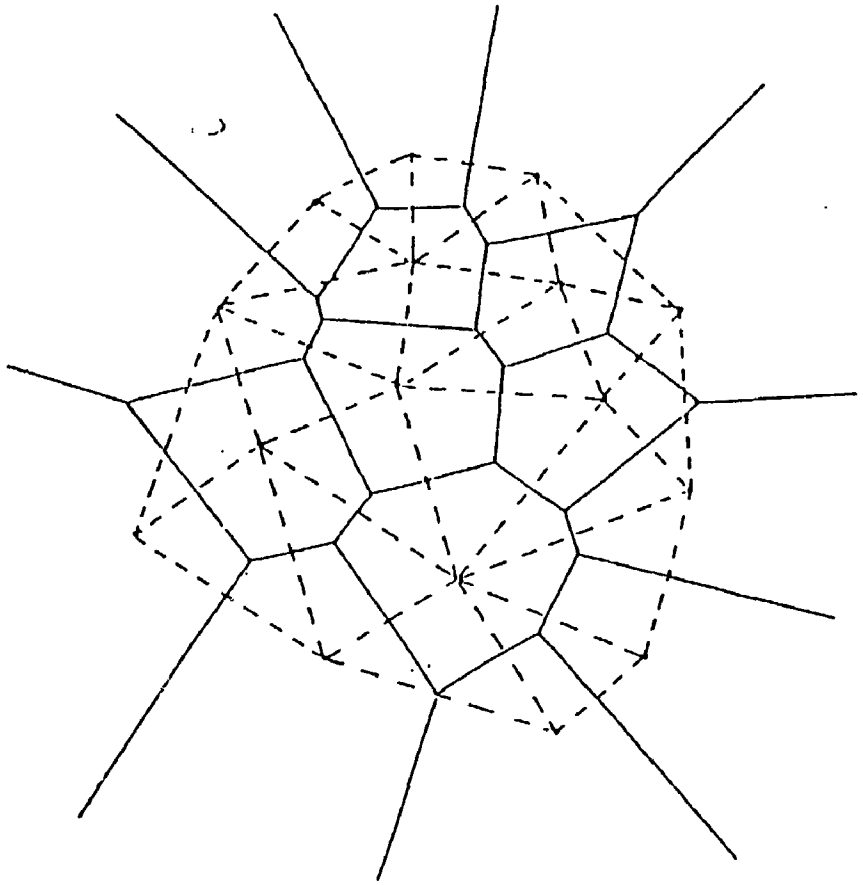


Figure 2.1. Dirichlet Regions and Delaunay Triangulation

The Dirichlet regions and Delaunay triangulation of a point set. The solid lines are the boundaries of the Dirichlet regions. The dash lines connect every two points of Dirichlet neighbours and form the Delaunay triangulation.

in the stack and does not intersect with any edge in the list of candidate edges. The algorithm terminates when all points are connected, and triangles are formed by connecting those edges. The triangulation created in this fashion would have shorter edges for triangles in the network than those created by other triangulation methods.

Delaunay triangulation is another procedure for defining the edges of the TIN. The triangulation is to connect all pairs of points that are Dirichlet neighbours. The Dirichlet region (also referred to as Voronoi region, Thiessen polygon) of each point is defined as that part of the plane which is closer to the point than to any other point. Two points are then Dirichlet neighbours if their respective Dirichlet regions share an edge. The result is the Delaunay triangulation of the sample points: each triangle has the property that its circumscribed circle contains no other point. The outer boundary of the Delaunay network is the convex hull of the point set (this property is also true for Greedy triangulation). Figure 2.1 shows the relationships between Delaunay triangles, the Dirichlet regions which are their topological duals, and the points which generate them. For algorithms of Delaunay triangulation, see Brassel and Reif (1979), Lee and Schachter (1980), Tarvydas (1983), de Floriani, Falcidieno, and Pienovi (1985a), and Preparata and Shamos(1985).

There is no unique method for triangulating the network. Any method that avoids long and thin triangles would be preferred. This ensures that the maximum distance of any interior point from a vertex of the triangle that encloses it is reduced, which in turn reduces the interpolation error when estimating an interpolation for an interior point from neighbouring points. Delaunay triangulation maximizes the minimum angle in each triangle, thus resulting in more equiangular triangles (de Floriani, Falcidieno, and Pienovi 1985a). Therefore, the Delaunay-triangulated network would be a suitable digital terrain model for the study.

The approach taken by this study to generate Delaunay triangles (adapted from Goodchild 1987) begins by finding the closest pair of points, i and j, which must be Dirichlet neighbours, and connecting them to give the first Delaunay edge. Two entries are then made into a stack, one for the ordered pair ij and the other for ji. In the main cycle of the algorithm an ordered pair ij is taken from the stack, and a search is made for the third vertex k, which will complete the triangle whose vertices are ijk when taken in clockwise order. The pair ij is then deleted from the stack, and jk and ki are added. By pre-sorting the points in x it is possible to process n points in the worst case in $O(n^2)$ time.

In summary, a TIN is a terrain model that uses a sheet of continuous, connected triangular facets based on a certain triangulation of irregularly spaced nodes. These points then

form the vertices of the TIN. The TIN model allows extra information to be gathered in areas of complex relief without the need for huge amounts of redundant data to be gathered from areas of simple relief. Consequently, the data capture process for a TIN can specifically follow ridges, stream lines, and other important topographical features that can be digitized to the accuracy required.

Moreover, several aspects of geomorphological features of landscapes have observed triangle facets. For instance, drainage basin, as well as the landscape of badlands, may be represented best by TINs as the erosion caused by run-off streams cuts land surface in acute angles rather than perpendicular or obtuse angles. In addition, the branches and main streams of rivers intersect also in acute angles. Therefore, the landscape will be eroded to surfaces whose facets are connected by acute angles.

The TIN model seems to be the best choice for the study. It is noted that the TIN model is capable of preserving topographic features and representing terrains without too much redundant data. Moreover, the data structure is straightforward: by keeping only triangles and nodes for each triangle, one can easily verify the topology because three nodes define a triangle and two nodes define an edge.

2.2. Visibility Regions

Visibility is defined as a function between two points in space where a straight line connecting these two points is not intersected or interrupted by any other object on the surface. The visibility region of an observation point on a terrain surface would then be defined as the union of the areas that are visible from the observation point.

Information on the visibility regions of viewpoints on terrain surfaces is useful for many applications, as previously stated in chapter I. However, it should be noted that research in this area is recent and limited.

The analyses of visibility regions are closely related to algorithms for detecting hidden lines and/or hidden surfaces. Literature in computer graphics has an enormous number of examples and studies on the subject (see Wright 1973, Weiler and Atherton 1977, El Gindy and Avis 1981, Anderson 1982, Burton and Smith 1982, McKenna 1987). Several rather complete reviews can be seen in Sutherland, Sproull, and Schumacker (1974) and in texts by Newman and Sproull (1979) and Foley and van Dam (1983). These hidden line/surface removal algorithms, however, are usually designed for geometric objects or regular grid meshes and thus are inappropriate for a study that uses the TIN model.

For studies of visibility regions, a visibility model was first introduced by de Floriani et al. (1986) to study the

visibility regions of given viewpoints on a land surface. An algorithm was proposed to compute the visibility regions for given viewpoints, using a triangulation network to partition the land surface.

De Floriani's visibility algorithm defines a set of functions between a vertex and the surface to be used to test visibility against all vertices and edges (details described in section 5.1.). The algorithm begins by classifying triangles into various types according to each triangle's inclination and orientation with respect to the viewpoint. The triangles are then 'depthsorted', which is a partial ordering of the edges according to blocking status.

A node or an edge is visible if there is no '**curtain**' in between to hide it from the viewpoint. The curtain is defined as an imaginary vertical trapezoid constructed downward from any edge whose height is high enough to intersect the line of sight connecting the viewpoint to the node or the edge being tested. As the testing moves outward in sorted sequence, every triangle has to be tested for visibility against all blocking edges between the viewpoint and itself. The algorithm takes $O(n^3)$ time for an point set with pre-processed triangulation and depthsorting.

Although the algorithm performs to some degree of satisfaction, the concept and functions and the required pre-processing needed are rather complicated and difficult to be adapted for computing very large data sets.

Cole and Sharir (1986) and Sharir (1987) contributed to the solution with algorithms on visibility problems for polyhedral terrains and related problems. The visibility problems discussed in their papers are similar to the visibility problems discussed in this study. However, both Cole and Sharir (1986) and Sharir (1987) used only hypothesized polyhedral surfaces, which consist of the upper facets of polyhedra produced by intersecting a small number of geometrically regular objects (such as pyramids or cubes). The surfaces studied in these two papers are all geometrically characterized so that they may be fitted by the mathematical equation of a polyhedron. Their studies also attempted to answer a subset of the visibility problems discussed in this thesis. However, their formulation of sample problems, which is limited to geometrically characterized polyhedral terrains, is not easily generalized to the case of real topographic surfaces of rugged reliefs. Finally, Nair's experiments (1988) concern the description of digital terrain models using visibility information. Nodal features, such as pits, peaks, etc., and linear features, such as ridge lines, or basin boundaries, etc., were extracted using visibility information.

These existing studies provide bases for further research in the area of visibility analyses on topographic surfaces. However, most of the visibility coverage problems are left unanswered. Moreover, the algorithm designed to compute visibility regions for specified viewpoints is not suitable

for large data sets and leaves room for further improvement. This thesis, therefore, intends to offer solutions to those unanswered visibility-coverage problems and to offer a better-fit algorithm for computing visibility regions.

2.3. Coverage Problems And Heuristics

Drawing from the literature of operational research, several topics in coverage problems that are closely related to visibility problems are reviewed in this section.

The development of the maximal-covering problem can be traced back to the classical **p-median** problem. The so-called p-median problem was first introduced by Hakimi (1964). This problem can be defined on a network of nodes and arcs in the following manner: minimize the total weighted distance by locating p facilities (or medians) such that each demand node is assigned to its closest facility. One of Hakimi's major theoretical results is that an optimal solution to this problem exists, which is comprised entirely of nodes of the network (Hakimi 1965). A set of p points that yield the smallest possible weighted distance is called an optimal solution set. The candidates for these p facilities are restricted to nodes of the network.

The p-median problem can be formulated as follows:

Minimize

$$\Sigma_i \Sigma_j A_i D_{ij} Z_{ij}$$

Subject to

$$\Sigma_j Z_{ij} = 1 \qquad \text{for all } i \in I$$

$$\Sigma_j Z_{jj} = p$$

$$Z_{jj} \geq Z_{ij} \qquad \text{for all } i \in I,\ j \in J$$

Z_{ij} = 1 if demand node i is assigned to facility j,
= 0 otherwise,

where
I = set of demand nodes,
J = set of facility nodes,
A_i = demand population in node i,
D_{ij} = distance between nodes i and j,
p = number of facility sites.

The objective is to minimize the total distance, weighted by the demand population. The first constraint ensures that each demand node is assigned to one and only one facility site. The second constraint states that the number of facility sites is restricted to p. Finally, the demand of a designated facility site should be serviced by itself.

Maranzana (1964) proposed an algorithm based on assigning demand nodes to their nearest facility sites. Teitz and Bart (1968) improved the algorithm with vertex substitution. ReVelle and Swain (1970), and Toregas et al. (1971) were the first to apply a simplex-type solution to the problem, and then Swain (1974) solved the p-median problem by

applying the decomposition principle and the generalized Lagrange multiplier technique. Many other heuristics have been developed for the p-median problem since then, including those developed by El-Shaieb (1973), Garfinkel et al. (1974), and Khumawala (1973).

Toregas and ReVelle (1972) introduced the **location set-covering** problem. It is to find the minimal number and the location of facilities such that no demand point will be farther than some pre-specified maximal service distance from a facility.

The problem can be mathematically formulated as follows:

Minimize

$$\Sigma_j X_j$$

Subject to

$$\Sigma_{j \in N_i} X_j \geq 1 \qquad \text{for all } i \in I$$

X_j = 1 if node j is a facility site, 0 otherwise,

where

$N_i = \{ j \mid D_{ij} \leq S \}$ and
S is the pre-specified maximal service distance.

The objective is to minimize the number of facilities necessary to cover the population of a set of n demand nodes. The set N_i for each i is the collection of nodes within the maximal service distance with respect to node i. The first constraint requires that each node i be covered by at least one facility site.

The combination of the p-median problem and the set-covering problem leads to the formulation of the maximal-covering location problem.

The **maximal-covering location problem (MCLP)** was first introduced by Church and ReVelle (1974). It seeks to maximize the population covered within a desired service distance S by optimally locating a fixed number of facilities. That is to:

Maximize

$$\Sigma_{i \in I} A_i Y_i$$

Subject to

$$\Sigma_{j \in N_i} X_j \geq Y_i \qquad \text{for all } i \in I$$

$$\Sigma_{j \in J} X_j = p$$

$X_j = 1$ if a facility is allocated to node j, 0 otherwise

$Y_i = 1$ if demand node i is covered, 0 otherwise,

where

N_i, D_{ij}, A_i, and p are as previously defined.

The first constraint allows a demand node to be covered by more than one facility within the maximal service distance while the second constraint limits the number of facilities to a pre-specified number.

Church and ReVelle (1976) discussed theoretical and computational links between the p-median, location set-covering, and maximal-covering location problems. The real

distances D_{ij} in the p-median problem can be transformed into a binary form via the following definition:

$$D_{ij} = \begin{cases} 0 \text{ if } D_{ij} \leq S, \\ 1 \text{ if } D_{ij} > S. \end{cases}$$

Thus, the objective of minimizing

$$\Sigma_i \Sigma_j A_i D_{ij} Z_{ij}$$

is equivalent to the minimization of the amount of population not covered within the maximal service distance S. They also reported a comparative study of the relative efficiency of several heuristic algorithms for solving the MCLP.

The p-median, set-covering and coverage problems all have similar form to the visibility problems defined in the previous chapter. Therefore, the heuristic used to solve these problems may be adapted to solve the visibility coverage problems. A chapter of discussion, detailed descriptions of heuristics and solution algorithms to our visibility coverage problems will follow the chapter on visibility regions and solutions.

2.4. Conclusions

The analysis of visibility regions on DEMs is a new approach in geographical research that is related to automated cartography, operations research, and computer graphics. It is the locational aspect of the research that links this study to the research community of geography. This review section

examined these fields individually and came to the following conclusions.

First, the suitable digital elevation model for the study is the TIN model, as discussed in the section on Digital Elevation Models (section 2.1.). It is the TIN's ability to represent an arbitrary terrain surface with less redundant data and a more efficient data structure along with its simpler geometric properties that make it more appropriate for this study than other DEMs.

Second, the review of the study of visibility regions calls for a new visibility algorithm because most of the existing hidden line/ surface detecting algorithms are designed for regular grid block diagrams or geometrically regular objects and thus are improper for our purposes. Furthermore, the only existing visibility algorithm used for TIN still leaves room for improvement because of its complexity and inefficiency in computing time.

Finally, the heuristic approach reviewed in this chapter demonstrated a feasible formulation of our visibility coverage problems and their solutions. Therefore, heuristics seem to offer a feasible way of approaching the related visibility coverage problems which are the focus of this thesis.

CHAPTER III VISIBILITY REGIONS AND COVERAGE PROBLEMS

A visibility model of a surface consists of a set of visible regions computed with respect to a set of points on the surface. The visible region of a viewpoint on a topographic surface is the set of the projections on the x-y plane of the points on the surface that can be joined to the viewpoint by a straight-line segment that intersects the surface only at its endpoints.

A topographic surface is usually continuous in space. However, there exist practical difficulties in storing, analyzing, or displaying topographic surfaces in continuous form. Therefore, some representation models, such as contouring, grid DEM, or TIN, etc., are used to approximate the continuous surface. For the purposes of this study, the Triangulated Irregular Network (TIN) has already been discussed and selected in the review chapter.

A framework suggested by de Floriani and Nagy (1989) is adapted and described to express and classify visibility problems. This follows a section that describes a discrete model to re-formulate visibility problems for topographic surfaces represented by a finite set of vertices. The discussion of visibility problems in continuous form is basically for the purpose of generality of the framework while the discussion of the discrete model is for practical reasons.

In both sections, the study assumes an arbitrarily rugged surface, and the ability to locate viewpoints on any vertex of the surface. The first assumption preserves the flexibility for analyzing visibility regions on any kind of land surfaces because, theoretically, a rugged surface is more complicated than a flat or regularly shaped one. The second assumption allows a free choice of observation points on the surface.

3.1. Continuous Visibility Model And Problems

As suggested by de Floriani and Nagy (1989), visibility problems are expressed by two subsets of the domain of the bivariate function that describes the surface and by using two visibility functions that account for the mutual visibility of points belonging to the two sets.

The following two sections are quoted directly from de Floriani and Nagy (1989). In them, the two visibility functions are defined and visibility problems are identified and classified.

3.1.1. Visibility functions

A **topographic surface** is a regular surface S described by a real-valued function of two variables $z=f(x,y)$ defined over a regular simply-connected domain R in E^2.

"The **visibility** is a binary relation I of R^2, which is reflexive and symmetric, defined as follows:

> (P,Q) in I iff $P=(x_P,y_P)$ in R and $Q=(x_Q,y_Q)$ in R and the straight-line segment P'Q', where $P'=(x_P,y_P,f(x_P,y_P))$ and $Q'=(x_Q,y_Q,f(x_Q,y_Q))$ lies above S.

If $P=(x_P,y_P)$ is in R, we call the **visible region** of P the subset VR(P) of R of the points Q in R such that (P,Q) or (Q,P) is in I.

We consider subsets R' of R which are either compact subsets of E^2 or finite sets of points of R (called **point sets**). In the first case, R' can be a regular domain (called a **region**, or a regular **curve**). We assume as **measure** of a subset R', denoted as m(R'), the cardinality of R', if R' is a (finite) point set; the length of R', if R' is a curve; or the size (area) of R', if R' is a region. In the following, we will use the term subset to denote a point set, or a region as defined above.

Let R_0 and R_1 be two subsets of R. We call R_0 the **reference set** and R_1 the **target set**. We assume that R_0 is a predefined subset of R, while R_1 can vary. We define the **reference visibility** and **target visibility** as follows:

> The **reference visibility** function, denoted $z=v[R_0](x,y)$ is a function $v[R_0]$: R_1 --> $[0,m(R_0)]$ such that for every $P=(x,y)$ in R_1 $v[R_0](x,y)=m(R')$, where $R' = VR(P) \cap R_0$ (i.e., R' is the subset of R_1 which is visible from P).

It is a measure of that portion of the reference set which is visible from a point of the target set.

> The **target visibility** function, denoted $z=v[R_1](x,y)$, for every subset R_1 of R, is a function $v[R_1]: R_0 \rightarrow [0,m(R_1)]$ such that for every $Q=(x,y)$ in R_0 $v[R_1](x,y)=m(R")$, where $R"=VR(Q) \cap R_1$.

It is a measure of that portion of the target set which is visible from a point in the reference set.

Visibility problems can be expressed in terms of visibility functions and criterion functions defined over them. Examples are the (reference) **size** functions defined below in terms of the reference visibility function.

> Let R_0 be a reference set and $0 <= p <= m(R_0)$. Let $R^* = \{ (x,y) \text{ in } R_1 \mid v[R_0](x,y) \ \alpha \ p\}$, where $\alpha = >, <, >=, <=, =$. A <u>size</u> criterion function, denoted $a[p,\alpha,R_0](R_1)$, is defined as
>
> $a[p,\alpha,R_0](R_1)=m(R^*)$" (de Floriani and Nagy, 1989, pp. 2-3).

3.1.2. Definition and classification of visibility problems

The two visibility functions quoted above may be used to formulate a wide variety of visibility problems. Usually constraints may be imposed on the reference set or the target set. It follows that a feasible solution and an optimal solution are both desired under imposed constraints, where the optimization may be achieved on the measure of the target set of visibility measures.

In most cases exact solutions may be found for both feasible visibility problems and optimal visibility problems. However, a thorough search for an exact solution for optimal

visibility problems tends to be extremely costly in many cases, especially when the surface S is represented by a large set of vertices. Therefore, solutions to optimal visibility problems are usually obtained through heuristic approaches which may not guarantee the optimality of the solution but may instead maximize the possibility of reaching an optimal solution.

The following definitions and discussion have been adapted and extended from de Floriani and Nagy (1989, pp. 4-6):

A class of **feasible visibility** problems can be defined in terms of the reference size function as follows:

> Given a reference set R_0, find a subset R_1 of R such that $a[P,\alpha,R_0](R_1)=m(R_1)$, where P and α are fixed.

Feasible visibility problems belonging to such classes are:

1. Find a region R_1 in R such that every point in R_1 is visible from each point in the reference set R_0.

   This problem can be expressed as:

   > Find R_1 in R such that $a[m(R_0),=,R_0](R_1) = m(R_1)$; or, equivalently, (in terms of the reference visibility function) $v[R_0](x,y)=m(R_0)$ for every (x,y) in R_1.

2. Find a region R_1 in R such that every point in R_1 is visible from at least one point in R_0.

   This problem can be expressed as:

   > Find R_1 in R such that $a[0,>,R_0](R_1) = m(R_1)$; or, equivalently, $v[R_0](x,y) > 0$ for every (x,y) in R_1.

3. Find a point P on S ($z=f(x,y)$) which can see the whole surface.

This problem may be formulated as follows:

> Find a point $P=(x^*,y^*)$ in R such that $v[R](x^*,y^*)=m(R)$;

Related **optimization** visibility problems can be defined as follows:

> Let S(R) be the collection of the subsets of R satisfying a given constraint. Find R_1 in S(R) such that
> $m(R_1)$= MIN (or MAX) ($m(R^*)$ | R^* in S(R) and $a[P,\alpha,R_0](R^*) = m(R^*)$).

Optimization visibility problems belonging to this class are the following:

1. The shortest invisible path problem, i.e., find an invisible path of minimum length between two points A and B on S with respect to a given reference set R_0.

   This problem may be formulated as follows:

   > Find a path Q between two points A and B on S such that $m(Q)$ = MIN ($m(Q^*)$ | Q^* path between A and B such that $a[0,=,R_0](Q^*) = m(Q^*)$).

   Related problems include finding a path such that the visible region from the path is maximized. Moreover, by combining more than one path, a minimum length patrol path can be defined such that the whole surface is under surveillance.

2. The largest invisible region problem, i.e., find the region R_1 of largest size containing a target point P which is invisible from a given reference set R_0.

   This problem may be formulated as follows:

   > Find a region (regular domain) R_1 in R such that $m(R_1)$ = MAX{$m(R^*)$ | R^* in R, P in R^* and $a[0,=,R_0](R^*)=m(R^*)$}

Other optimization visibility problems optimize on the measure of the visible portion of the target set. Such problems may be generally formulated as follows:

> Find a target set R_1 in S(R) such that $a[P,\alpha,R_0](R_1)$ = MIN (or MAX) $\{a[P,\alpha,R_0](R^*)$ for every R^* in S(R) $\}$

An example is the "hidden" path problem expressed as:

> Find a path Q on S between two points A and B such that as little as possible of Q is visible from a reference set R_0.

This problem may be formulated as follows:

> Find Q such that $a[0,>,R_0](Q)$ = MIN $\{a[0,>,R_0](Q^*)$ for every path Q^* on S between A and B$\}$.

A "dual" problem is to find a path Q between two points A and B on a surface S from which most of the surface defined on a reference set R_0 is visible.

This problem may be formulated as follows:

> Find a path Q on S between A and B such that $w[R_0](Q)$ = MAX $\{ w[R_0](Q^*)$ for every path Q^* on S between A and B $\}$, where $w[R_0](Q) = \int_Q v[R_0](x,y)dxdy$.

All the feasible and optimization problems considered so far are defined on the target set with respect to a (fixed) reference set. Such problems have the common characteristics that they can be solved by an a priori evaluation of $v[R_0](x,y)$ on the points of R. When, for instance, R_1 is a path between two points on the surface, the previous problems reduce to path finding, shortest path, or minimum visibility-cost path problems on a surface.

Classes of "dual" problems can be defined on the reference set with respect to the target set. An example of such problems is the selection of the smallest number of observation points needed to view the whole surface (or a portion of it defined by a reference set). This problem may be formulated as follows:

> Given a reference set R_0, find a point set R_1 in R such that
> $m(R_1) = MIN \{ m(R^*) \mid R^*$ point set and $a[0,>,R^*](R_0) = m(R_0) \}$.
>
> If R_1 must cover the whole surface, then R_0=R. The problem can be easily restated for discrete domains as the set covering problem.

A class of feasible visibility problems can be defined as follows in terms of the target size function:

> Given a reference set R_0, find a set R_1 in R such that
> $a[P,\alpha,R_1](R_1) = m(R_0)$, where P and α are fixed.

We can similarly define constrained visibility problems based on $v[R_1](x,y)$ in which we restrict the class of candidate target sets to be considered, and also optimization visibility problems in which the optimization is performed either on the measure of the target set or on the size function.

The framework described above defining a large class of visibility problems on a topographic surface is quite general, being independent of the discrete model used to approximate the surface and on the discretization of either the reference or the target sets. When we consider a finite point set as

the reference set (set of observation points) and a regular open curve on S (path) as the target set, the visibility problems defined in terms of the reference size function (both those searching for a feasible solution or those optimizing on the path length) reduce to path finding or shortest-path problems on a surface.

Ideally, these visibility problems should be solved in their most general form, on continuous topographic surfaces. However, this is impractical as the time and effort required would be too large. In addition, a discrete approximation to continuous topographic surfaces offers a high degree of validity if the discrete model preserves well the features of the landscape.

If the surface is in turn discretized, in the form of a triangular mesh, then classical graph-theoretic algorithms can be applied to produce "discrete" feasible or optimal solutions. For example, for a finite point set as the reference set (again, a set of observation points) and for a regular domain of surface on S (region, or triangles in this case) as the target set, the visibility problems defined by the target size function reduce to set-covering problems on a surface.

If we consider a discretized surface such as a topographic surface represented by a triangulated mesh, there may be some extensions to the optimization problems which add to the classes of visibility problems, such as allowing the

target set or the reference set to have variable heights (for example, building watchtowers at different heights). Furthermore, constraints may be imposed, in various forms, on allowable heights in the target or reference sets (for instance, imposing a budget on the construction cost of watchtowers).

The discussion and definitions, so far, are based on continuous topographic surfaces. In the next section, visibility problems are re-formulated for a discrete model of topographic surfaces, which is the TIN in this study.

3.2. Discrete Visibility Model And Problems

The discussion of visibility problems in this section will be based on an assumption that the topographic surface is approximated by a discrete model, which is the TIN model, selected as a result of the discussion in Chapter II. Assumptions for using rugged land surfaces and for maintaining the flexibility to locate viewpoints freely anywhere on the land surface remain in effect.

The visibility problems discussed in this section are intended to serve as prototype problems as it would be impossible to cover all potential variations. Each problem listed in this section is carefully selected to represent its class of visibility problems.

The formalization of discrete visibility problems follows the standard form of the location set-covering problems (Toregas and ReVelle 1972) and the maximum-covering location problem (Church and ReVelle 1974) respectively. Discrete visibility problems are discussed in two contexts:

1) viewpoints located on the topographic surface; and
2) viewpoints with variable heights.

3.2.1. Visibility problems of fixed heights

Given np points representing vertices of a landscape, $P_i=(x_i,y_i,z_i)$, where i=1,np, and let the presence of a facility (e.g. a firetower, a watchtower, or a radio transmission station) at vertex i be denoted by c_i, which is 1 if a facility is present and 0 otherwise. In addition, the whole landscape, S, is partitioned into T_j subregions (triangles in our case), where j=1,nt and nt denotes the number of subregions, and a visibility parameter V_{ij} = 1, denotes that all points of the subregion j are visible from vertex i (viewpoint), and 0 otherwise. Moreover, let A_j be the area of the subregion T_j (that is, $A_j=m(T_j)$ is the measure of T_j, following the notation of section 3.1).

The problems to be discussed in this study are:

Problem 1:

To find the area of visible regions for a given set of viewpoint(s), VP, that is to:

find

$$\Sigma_{j \in VR} A_j$$

where

$$VR = \{ j \mid \Sigma_i V_{ij} c_i \geq 1 \}$$

$c_i = 1$ if i belongs to VP; 0 otherwise.

This is the problem of extracting visibility regions for a given set of viewpoint(s). The objective is to find a union of visibility regions that are visible from those viewpoints specified. Since the visibility function is defined as a Boolean function, the subregion T_j, of area A_j, will be included in the solution set if both $V_{ij}=1$ (meaning that T_j is visible from viewpoint i) and $c_i=1$ (meaning that point i is included in the specified viewpoint set of VP). After this set has been found, the total area would be the union of the visible areas from the viewpoints in VP.

This problem has an exact solution, which does not involve any optimization, as the visible regions of a given set of viewpoint(s) may be computed precisely. The solution set may be a connected polygonal region or several disjoint polygonal regions; but there would always be a finite solution as those subregions immediately connected to the specified viewpoints are always visible. The algorithms for extracting visibility regions for a given set of viewpoints, as well as the solutions of Problem 1, are discussed in section 5.2.

Problem 2·

To find the area of invisible regions for a given set of viewpoint(s), VP, that is to:

find

$$\Sigma_{j \in VQ} A_j$$

where

$$VQ = \{ j \mid \Sigma_i V_{ij} c_i = 0 \}$$

Problem 2 is a parallel formulation to Problem 1. The purpose is to find the possible "hidden" regions against a given set of viewpoint(s). The solution is a set of subregions that are invisible from all viewpoints in a given set, VP. A subregion is included in the solution set if V_{ij} $c_i=0$ for all i in VP (meaning that subregion j is invisible from all viewpoints in VP). In other words, the solution set should include subregions, if any, which are in the intersection of invisible subregions from the given set of viewpoint(s).

The solution of Problem 2 may be an empty set, a connected polygonal region, or several disjoint polygonal regions. Algorithms for extracting visibility regions may also be used to find invisible regions for viewpoints. The solutions and algorithms are discussed in section 5.2.

Problem 3:

To minimize the number of facilities required to see the entire surface, that is to:

minimize

$$\Sigma_i c_i$$

subject to

$c_i = \{ 0,1 \}$ for all i

$\Sigma_i V_{ij} c_i \geq 1$ for all j

This is the classic watchtowers problem, i.e., to find a minimal set of locations from which the entire surface is visible. The first constraint requires an integer solution. The second constraint ensures that all subregions are covered. It is always possible to find a solution set of vertices as viewpoints to see the entire surface because the worst case is to include all vertices as viewpoints for the solution. However, the emphasis is on minimizing the number of viewpoints required to "see" the entire surface. The solution of this problem is discussed in section 5.2.1.

Problem 4:

To maximize the area covered by a given number of facilities F, that is to:

maximize

$\Sigma_j A_j \min(1, \Sigma_i V_{ij} c_i)$

subject to

$c_i = \{ 0,1 \}$ for all i

$\Sigma_i c_i \leq F$ for all i.

Problem 4 is a parallel formulation to the watchtower problem (Problem 3). The number of viewpoints are pre-defined to the limit of F. The entire surface is not necessarily covered by the pre-specified viewpoints. The

locations of the specified viewpoints are selected such that the union of visible subregions from all viewpoints is maximized. In other words, the solution may also be expressed as a set of selected vertices as viewpoints such that the total area of visible subregions from the selected viewpoints are maximized. A subregion is included in this case if and only if it is visible from at least one of the selected viewpoints. The solution of this problem is discussed in section 5.2.1.

So far the discussion has been limited to the case of viewpoints located on the topographic surface. Extensions of these problems are possible by allowing the height of a viewpoint to vary. This extension gives visibility coverage problems a new approach and therefore broadens their practical applications.

3.2.2. Visibility problems of variable heights

It is clear that visibility coverage may be increased by raising the height of the viewpoint; since the surface S is assumed to be single valued, there must exist a minimum height at each vertex from which a specified subregion is visible. In other words, one generalization of the visibility coverage problems is to allow the viewpoint to have variable heights. This allows us to define a whole set of issues related to Problem 1 to 5 described above.

Problems invoked by introducing the concept of viewpoints with variable heights include finding the minimal height for a viewpoint so that a fixed given set of vertices can be seen, or establishing a functional relationship between height and cost, and selecting locations and heights for the minimum-cost combination of watchtowers. The motivation will be to reduce the construction cost of watchtowers by minimizing required heights.

Let h_{ij} denote the minimum height above the vertex i from which an observer can see subregion j; the subregion will be visible from all elevations above h_{ij} and invisible at all lower elevations. Define h_{ij} as **the minimal visible height**. Assume that construction cost is a simple, monotonically increasing function of tower height, $C(H_i)$, where H_i denotes the height of the tower at vertex i. A vertex that is used as a viewpoint but has no tower will have $H_i = 0$. A vertex with a tower of height H_i should presumably incur a positive cost. A vertex that is not used as a viewpoint will be identified by a negative H_i. We can now introduce problems of visibility coverage in which the objective is to achieve maximum coverage at minimum cost.

Let $W_{ij} = 1$ if $H_i \geq h_{ij}$, 0 otherwise;

Problem 5:

To see the entire surface at minimum tower cost, that is to:

minimize

$$\Sigma_i \, C(H_i)$$

subject to

$$\Sigma_i \, w_{ij} \geq 1 \qquad \text{for all } j$$

where

$$C(H) > 0 \text{ for } H \geq 0, \; C(H) = 0 \text{ for } H < 0.$$

Problem 5 is to find an optimal solution that selects a minimum number of viewpoints and appropriate heights at the location of each selected viewpoint. The solution always exists when a watchtower can be built to any height. Theoretically, we can build our watchtowers to any desired height. However, physical constraints exist when the height of the tower is beyond the ability to build. In this case more than one tower or tower location will be needed. The solution of this problem is discussed in section 5.2.2.

Problem 6:

To maximize the area covered within a construction budget T, that is to:

maximize

$$\Sigma_j \, A_j \, \min(1, \Sigma_i \, w_{ij})$$

subject to

$$\Sigma_i \, C(H_i) \leq T.$$

Problem 6 is a parallel formulation to Problem 5 but differs because of a constraint on the construction budget to raise heights of viewpoint locations. By adding the budget constraint, the solution of Problem 6 might not be feasible at an acceptable cost. The solution is discussed in section 5.2.2.

The problems described so far are examples that serve as prototypes of the categories of problems discussed in section 3.1. In this study we do not plan to tackle all possible variations of the visibility problems. While the list might not be complete, most of the important characteristics of visibility problems have been already included.

3.3. Conclusions

With visibility problems defined in this chapter, it is appropriate to turn the focus on solutions and the implications of these solutions. Intuitively, the coverage problems identified in this study may be answered by using some searching techniques, such as exhaustive search, random search, dichotomous search, and the like (see Cooper and Steinberg 1970 for details). However, these methods are not really desirable as they tend to be too time-consuming when applied to large data sets such as those used in this study. Therefore, heuristic approaches are preferred as they usually find solutions in shorter time than any thorough search method.

The solutions, along with algorithms for computing solutions, are presented in the two chapters that follow the discussion of data and DEM conversion algorithms (Chapter IV). Chapter V focuses on visibility region problems

(Problem 1 and Problem 2) and Chapter VI focuses on coverage problems of both fixed and variable heights.

CHAPTER IV DATA AND CONVERSION ALGORITHMS

Before discussing solutions for visibility coverage problems and their relationships to various types of topographic surfaces, descriptions of the data sets used in this thesis are presented in this chapter. Both randomly generated data sets and real topographic DEM files are used to compare the feasibility of algorithms for finding solutions and testing the validity of visibility concepts. The random data sets are also used to examine patterns of visibility information over different topographic surfaces. Visibility information considered in this study includes visibility information between regions, the number of viewpoints required to see entire surfaces, their locations, and topographic features of the selected viewpoints.

The purpose of using randomly generated data sets is to ensure that the algorithms used are indeed capable of processing random surfaces while real topographic surface DEMs are used to verify the performance of the solution algorithms on real surfaces.

Topographic DEMs are available from U.S.G.S. digital elevation data files. Unfortunately the available DEM files were compiled by U.S.G.S. in grid format, so it is necessary to convert them into a format which is suitable for the data structure of a TIN model. A TIN converted from a dense grid

is actually a simplified version of the original surface depicted by the grid DEM. The success of any conversion method can be measured by the difference of elevations between the original grid surface and the surface approximated by the converted TIN. Therefore the objective is to design a satisfactory conversion method to minimize the difference between the original surface and the surface approximated by the converted TIN. A conversion algorithm was developed for this purpose and will be described in the later sections.

This chapter first describes the data sets used in this study: their structure, sizes, locations, and assignment to different physiographic regions in the United States. Advantages and disadvantages of the existing DEM conversion algorithms are reviewed. A new algorithm was developed to convert these DEM files to TIN files that aimed to improve the disadvantages of existing methods. The discussion of the conversion algorithms also includes tests and results. The analysis reveals that the algorithm proposed in this chapter performed better in general than existing algorithms for the specified objectives.

4.1. Data Sets

Data sets used for the study include thirty randomly generated sets and four U.S.G.S. 7.5 minute by 7.5 minute DEM

files. Each random data set contains 50 points. Random data sets were used to test the feasibility of algorithms for extracting visibility information and heuristic solutions. The topographic DEMs were used to explore the patterns of visibility information and their relationship to different types of topographic surfaces.

4.1.1. Surface description parameters

In this study, surfaces are represented by a set of vertices with x, y, and z coordinates, z=f(x,y). Statistics and other surface description parameters were calculated from the coordinates of vertices to describe a surface. This section discussed the following surface description parameters:

1) elevation statistics:
   a) mean of elevations of vertices (ZMEAN);
   b) standard deviation of elevations of vertices (ZSD);
   c) skewness of elevations of vertices (ZSKEW);
   d) kurtosis of elevations of vertices (ZKURT);
2) slope statistics:
   a) mean of slopes of triangle facets in TIN (TPSMEAN);
   b) standard deviation of slopes of triangle facets in TIN (TPSSD);
   c) skewness of slopes of triangle facets in TIN (TPSSKEW);
3) coefficient of dissection (ZCD);
4) spatial autocorrelation coefficient (SAC);
5) variogram parameter (VARI);
6) classifications of topographic features of vertices.

Elevation statistics

ZMEAN

Mean of elevations is defined as the arithmetic mean of all elevations of a set of vertices. More specifically, it may be computed by the following formula

$$ZMEAN = (\Sigma_i \ z_i \ / \ np)$$

This mean is a measure of 'average' which is the most commonly used measure of central tendency.

ZSD

Standard deviation of elevations in a set of elevations is defined as:

$$ZSD = [\ \Sigma_i \ (z_i - ZMEAN \ )^2 \ / \ np \ ]^{\frac{1}{2}}$$

It may be also referred to as the root mean square deviation. Standard deviation is the measure of dispersion most commonly applied to geographical data. It is used in this study to indicate the variation of reliefs on surfaces.

ZSKEW

Skewness measures the extent to which the majority of values in a distribution are concentrated to one side or the other of the mean. Used here to indicate the degree to which the distribution of elevations follows a normal distribution. If most of the elevations are less than the mean, the distribution is said to be

positively skewed. If there are more values greater than the mean, the distribution is negatively skewed. The skewness of a distribution is defined as:

$$ZSKEW = [\ \Sigma_i\ (z_i - ZMEAN)^3\ ]\ /\ (np\ ZSD^3)$$

ZKURT

Kurtosis measures the extent to which values are concentrated in one part of a distribution. If one group of adjacent values (or classes) in a distribution contains a large proportion of all the values in the distribution, then the distribution has a high degree of kurtosis. It is used in this study to measure the relative peakedness or flatness of the distribution of the elevations. A normal distribution is represented by a value of 3. Distributions with kurtosis values lower than 3 tend to be distributed more uniformly than those with kurtosis values higher than 3. The kurtosis of a distribution may be calculated by:

$$ZKURT = [\ \Sigma_i\ (z_i - ZMEAN)^4\ ]\ /\ (\ np\ ZSD^4\ )$$

Slope Statistics

As the surface is approximated by triangulated facets constructed from a set of vertices, the slopes of extended planes of these triangle facets would certainly affect

visibility information. That is, a surface with more steeper slopes tends to be associated with a wider relief variation.

The slope of a triangle plane is defined as the angle between the horizontal plane (z = 0) and the triangle plane. If the triangle plane is

$$ax + by + cz + d = 0;$$

the angle between z = 0 and the plane is:

$$\text{angle} = \cos^{-1} (c / (a^2 + b^2 + c^2)^{\frac{1}{2}})$$

TPSMEAN

The mean of slopes of triangle planes in a TIN may be calculated as the arithmetic mean of the slopes by using the formula for computing ZMEAN.

TPSSD

The standard deviation of slopes of triangle planes in a TIN may be calculated by using the formula for computing ZSD.

TPSSKEW

The skewness of slopes of triangle planes in a TIN may be calculated by using the formula for computing ZSKEW.

Coefficient of dissection

ZCD

Coefficient of dissection (ZCD) indicates the distribution of the landmass with elevation. The coefficient is defined as:

$$ZCD = (ZMEAN - ZMIN) / (ZMAX - ZMIN)$$

where ZMIN and ZMAX are the minimum and maximum elevation in a distribution. In a study of analysis of erosional topography Strahler (1952) found the following association between values of ZCD and stages of a landscape in its geomorphological cycles (also quoted by Klinkenberg 1988):

< .35	'old' stage, with monadnock masses present;
.35-.65	'mature' stage, or equilibrium stage;
> .65	'young' stage, or inequilibrium stage.

Spatial Autocorrelation Coefficient

SAC

Spatial autocorrelation (Cliff and Ord, 1973, 1981) is concerned with the degree to which objects or associated values at some places are similar to other objects or associated values located nearby. Both Geary's index (Geary 1954, Geary 1968) and Moran's index (Moran 1948) may be used as an index describing the distribution of attribute values of things in space. Of the two, Moran's index is used in this study for "it is arranged in a way such that its extremes match the notion associated with positive or

negative correlations" (Goodchild 1986, pp.16). Moran's index is positive when the associated attributes of nearby areas are similar, negative when they are more dissimilar, and close to zero when the associated attribute values are arranged randomly and independently in space. Two good reviews and discussions on the topic of spatial autocorrelation may be found in Goodchild (1986) and Griffith (1987).

This study adapted the extended spatial autocorrelation for point values suggested by Ebdon (1985), in which Moran's index was modified by using $1/D_{ij}$ as a weight between points P_i and P_j (where D_{ij} is the distance between P_i and P_j). Instead of considering the relationship between pairs of contiguous area values, we measure the relationship between all pairs of point values, taking into account the distance separating them. For np points, there are np(np-1)/2 possible pairs of points. A revised version of Moran's I was used:

$$I = \frac{np \sum_k w_{ij} (z_i - ZMEAN)(z_j - ZMEAN)}{(\sum_k w_{ij}) \sum_i (z_i - ZMEAN)^2}$$

where

i,j	$i,j=1,np$;
w_{ij}	is the reciprocal of the distance between the two points, P_i and P_j. If the distance between P_i and P_j is D_{ij}, then $w_{ij}=1/D_{ij}$;

Σ_k means that the value of numerator, $w_{ij}(z_i-ZMEAN)(z_j-ZMEAN)$, must be calculated for all pairs of values $np(np-1)/2$ and summed (Ebdon 1985, pp. 160).

Variograms

Variograms are defined by the variance or mean squared differences between attributes at given proximity to each other. Essentially this data analysis method assumes that the statistical variation in attributes between vertices is some function of the distance between them. The independent variable is the distance between pairs of vertices; the dependent variable is the variance of the difference in the attribute values for all vertices the given distance apart. Variograms are used as one component of the process of kriging (Krige 1966) for interpolation of elevation data as well as a tool to test fractal models (Mandelbrot 1975, 1977, Burrough 1983, Mark and Aronson 1984, Culling and Datko 1987, Goodchild and Mark 1987, and Klinkenberg 1988).

In this study, the attribute of concern is the elevation of vertices representing a surface. The variogram is defined as:

$$2\gamma(d) = E[z(x,y) - z(x+u,y+v)]^2,$$

where

$z(x,y)$ and $z(x+u,y+v)$ represent the elevations at point (x,y) and point $(x+u,y+v)$ respectively, and $d = (u^2+v^2)^{\frac{1}{2}}$.

This study calculated the variograms of the surfaces used by implementing the following steps:

1) the entire surface was included;
2) the maximum distance was determined for each data set;
3) this distance was divided into 25 distance classes using a geometric progression such that the difference between the natural logs of the widths of each distance class was the same;
4) for all possible pairs of vertices, P_i and P_j, where i<>j, i, j = 1, np, in the data set were computed for
   a) distance between P_i and P_j, D_{ij}; and
   b) elevation deviations, $Zdev=[z_i-z_j]^2$;
5) calculated distances, Zdev, and frequencies were kept for each distance class;
6) the averages of the deviations were calculated for each distance class;
7) a double logrithmic regression line was computed for the variances, i.e. the averages of the deviations in step 6, obtained for distance classes by using the least-square regression method (SPSS^x, SPSS Inc., regression procedures are used for this computation).

The slope of the regression line of log variances against log distance may be used to indicate how the elevations change

between vertices of different distances. A steeper slope of this regression line represents a larger variances change among various distance intervals.

Classification of topographic features of vertices

Each vertex in a data set may be classified as either a peak, a pit, a pass, a ridge point, a channel point, or other than these five types. The classification of a vertex into a specific feature type is based on the relationship between the elevation of the vertex and the elevations of its neighbour vertices. Since this study adopted Delaunay triangulation to create a network of vertices, the neighbour vertices for a vertex are defined as its Delaunay neighbours. Five types of topographic features may be categorized as follows (adapted from Peucker and Douglas 1975):

for vertex v:

n	number of neighbours of v;
Δ_i	difference in elevation between v and its ith neighbour, i=1,n - in clockwise sequence around v;
Δ_+	sum of all positive differences between v and all its neighbours;
Δ_-	sum of all negative differences between v and all its neighbours;
N_c	number of sign changes between 2 Δ_i's;
L_c	minimum number of points between two sign changes;

1)	peak	$\Delta_+ = 0$,	$\Delta_- > tp$,	$N_c = 0$;
2)	pit	$\Delta_+ > tp$,	$\Delta_- = 0$,	$N_c = 0$;
3)	pass	$\Delta_+ + \Delta_- > tps$,	$N_c = 4$;	
4)	ridge	$\Delta_- - \Delta_+ > tr$,	$N_c = 2$,	$L_c <> n/2$;
5)	channel	$\Delta_+ - \Delta_- > tr$,	$N_c = 2$,	$L_c <> n/2$;

(where tp, tps, and tr, are threshold values for peak/pit, pass, and ridge/channel respectively).

Similar definitions may be used for the vertices of gridded DEMs. However, the neighbours would be grid neighbours of a vertex. For example, the vertex at position (i,j) would have neighbours at positions (more discussion concerning the definitions of peaks, pits, ridge or channel points is in section 4.2.):

```
(i-1,j+1),     (i  ,j+1),     (i+1,j+1),
(i-1,j  ),     (i  ,j  ),     (i+1,j  ),
(i-1,j-1),     (i  ,j-1),     (i+1,j-1).
```

The percentages of each feature type of a data set represent another characteristic of the surface that the data set depicts and were used in this study as one of the parameters to describe surfaces.

In the following sections, the 30 random data sets and 4 U.S.G.S DEM data sets are described both by the parameters defined in this section and information from their corresponding 7.5 quadrangle topographic contour maps.

4.1.2. Random data sets

Random data sets were generated by an intrinsic routine (a CYBER FORTRAN subroutine) which generates pseudo random numbers with a uniform distribution in any prescribed interval, using variable seeds. For every point in the data sets, coordinates x, y, and z were generated independently so that the spatial autocorrelation among points is minimal. In

total, 30 data sets were generated and each contained 50 points.

These 30 data sets were generated by three different range of z values, set to be 10, 20, and 30 to allow for different degree of changes in relief. The labelling of these data sets was according to the ranges of z values as TIN101, TIN102, ..., to TIN110 for those with z values limited between 0 and 10; TIN201, TIN202, ..., to TIN210 for those with z values limited between 0 and 20; and TIN301, TIN302, ..., to TIN310 for those with z values limited between 0 and 30.

The data structure for the randomly generated data sets is relatively straightforward. Each point is represented by a triplet, (x,y,z), as its coordinate in 3D space. Each set contains 50 triplets of x, y, and z coordinates. Table 4.1. shows the surface description parameters for the 30 random data sets.

4.1.3. U.S.G.S. DEM data sets

The second data source used for testing the algorithms for extracting visibility information over various types of topographic surfaces consists of four digital elevation models (DEM). A DEM is composed of a regular array of elevations referenced to mean sea level with a horizontal

	MEAN	S.D.	C.D.	Skewness	Kurtosis	SAC
TIN101	5.40	3.02	0.489	-0.137	1.634	-0.050
TIN102	4.96	2.86	0.440	0.275	1.786	-0.021
TIN103	5.32	2.91	0.480	0.130	1.821	-0.081
TIN104	4.80	2.84	0.422	0.203	1.816	-0.008
TIN105	4.48	2.69	0.387	0.265	1.949	-0.060
TIN106	5.76	2.61	0.528	-0.275	2.017	0.002
TIN107	5.34	3.22	0.482	0.126	1.573	-0.016
TIN108	5.88	2.38	0.542	-0.187	2.180	0.016
TIN109	4.82	2.64	0.420	0.120	1.830	-0.055
TIN110	4.82	2.70	0.424	0.185	1.792	-0.119
TIN201	10.60	6.14	0.505	-0.212	1.703	-0.059
TIN202	9.86	5.84	0.466	0.187	1.820	-0.032
TIN203	10.46	5.74	0.498	0.006	1.827	-0.078
TIN204	9.52	5.77	0.448	0.125	1.769	-0.013
TIN205	8.62	5.43	0.401	0.245	2.111	-0.063
TIN206	11.50	5.14	0.553	-0.314	2.177	0.006
TIN207	10.64	6.45	0.507	0.081	1.593	-0.085
TIN208	11.58	4.76	0.564	-0.203	2.006	0.018
TIN209	9.58	5.39	0.452	0.022	1.861	-0.056
TIN210	9.62	5.67	0.454	0.199	1.803	-0.122
TIN301	16.04	9.13	0.519	-0.202	1.695	-0.062
TIN302	14.74	8.82	0.474	0.174	1.824	-0.030
TIN303	15.72	8.80	0.508	-0.010	1.853	-0.081
TIN304	14.22	8.62	0.472	0.077	1.788	-0.013
TIN305	12.90	8.16	0.425	0.165	2.020	-0.059
TIN306	17.14	7.79	0.576	-0.341	2.158	0.005
TIN307	15.94	9.63	0.515	0.068	1.608	-0.030
TIN308	17.38	7.18	0.553	-0.195	2.042	0.019
TIN309	14.32	8.12	0.476	0.018	1.848	-0.057
TIN310	14.38	8.52	0.461	0.162	1.821	-0.126

mean - mean of elevations of vertices
S.D. - standard deviation of elevations of vertices
C.D. - coefficient of dissection
Skewness - skewness of elevations of vertices
Kurtosis - kurtosis of elevations of vertices
SAC - spatial autocorrelation coefficient

Table 4.1. Surface Description Parameters, Random Data Sets

	VARI	TP MEAN	TP S.D.	TP SKEW	PK %	PT %	PS %	RG %	CH %	OT %
TIN101	0.108	47.550	26.730	-0.127	18	10	24	10	14	24
TIN102	0.524	44.562	26.320	-0.074	12	8	6	14	14	46
TIN103	-0.289	41.151	24.421	0.189	14	14	14	24	12	22
TIN104	0.244	41.929	24.263	0.336	14	12	14	12	8	40
TIN105	-0.109	42.796	24.551	0.153	16	6	12	8	12	46
TIN106	0.371	44.350	26.660	0.203	8	12	4	18	12	46
TIN107	0.017	48.354	26.319	-0.037	6	8	8	12	16	50
TIN108	0.340	45.090	23.075	-0.081	8	10	4	6	22	50
TIN109	0.142	50.785	23.309	-0.104	14	10	14	18	14	30
TIN110	0.277	44.587	24.768	0.257	14	12	16	16	16	26
TIN201	0.242	43.366	25.949	-0.051	22	14	28	8	20	8
TIN202	1.123	44.116	26.002	-0.148	14	14	14	12	16	30
TIN203	-0.500	42.438	27.137	0.214	14	18	22	20	16	10
TIN204	0.459	45.345	24.759	-0.008	16	14	20	12	18	20
TIN205	-0.240	47.288	26.717	-0.299	20	6	14	12	20	28
TIN206	0.791	46.315	26.934	0.008	12	14	8	26	14	26
TIN207	0.121	44.870	25.146	-0.085	8	12	20	18	18	24
TIN208	0.697	46.498	22.551	-0.076	6	14	12	16	26	26
TIN209	0.322	50.334	24.230	-0.164	16	14	16	20	22	12
TIN210	0.549	49.541	24.010	-0.372	14	14	18	18	18	18
TIN301	0.311	37.779	25.542	0.378	20	14	26	10	20	10
TIN302	1.774	42.724	28.502	0.080	14	16	16	12	20	22
TIN303	-0.864	38.855	27.054	0.181	16	18	22	22	16	6
TIN304	0.631	43.507	27.426	0.064	18	14	22	12	18	16
TIN305	-0.344	44.999	29.200	-0.024	20	8	16	12	20	24
TIN306	1.157	42.224	28.002	0.190	10	14	10	26	16	24
TIN307	0.197	39.413	25.096	0.316	10	14	20	20	22	14
TIN308	1.066	45.006	26.584	0.127	8	14	14	18	26	20
TIN309	0.459	45.163	25.852	0.139	14	16	18	18	24	10
TIN310	0.890	48.059	27.533	-0.213	14	16	20	22	16	12

VARI - slope of regression line of variogram
TPSMEAN - mean of slopes of triangle facets in TIN
TPSSD - standard deviation of slopes of triangle facets in TIN
TPSSKEW - skewness of slopes of triangle facets in TIN
PK - % of peak points
PT - % of pit points
PS - % of pass points
RG - % of ridge points
CH - % of channel points
OT - % of points of other types (unclassified).

Table 4.1. Surface Description Parameters for Random Data Sets (Cont'd)

spacing of 30 meters in both N-S and E-W directions, and a vertical resolution of 1 meter (see Elassal and Caruso, 1983, for a complete description of the format and content of a U.S.G.S. DEM).

The unit of coverage for a DEM is a standard U.S.G.S. 7.5 minute quadrangle, and elevations are referenced to the Universal Transverse Mercator (UTM) coordinate system. Thus, each DEM has a variable number of rows and columns, a result of the variable angle between true north (used as the reference for the quadrangle boundaries), and grid north (used as the reference for the UTM coordinate system). In this study, the central part of each DEM was taken to form a 350 by 350 grid DEM for further conversion to TIN models.

The U.S.G.S. is far from completing its task of producing a DEM for every 1:24,000 quad sheet that covers the United States, and the existing coverage varies widely -- from areas such as Florida which have very few models completed, to areas which have almost a complete coverage, such as the state of Wyoming. In order to explore fully the relationship between patterns of visibility information and different types of topographic surfaces, a random sampling stratified by physiographic regions is preferred. However, constraints on both data acquisition and computing resources precluded this. Therefore, data sets used in the following analyses were those which were freely available. Four DEMs were obtained from Dr. Brian Klinkenberg (University of

British Columbia) who obtained them originally from Dr. James Carter (University of Tennessee). The four DEMs are located in Krone Range, Montana (DEM1), Herbert Domain, Tennessee (DEM2), Benndale SW, Mississippi (DEM3), and Cedar Breaks SE, Utah (DEM4). Each of these DEMs has its distinct topographic characteristics. The complete detail of the four data sets is summarized in table 4.2. and table 4.3. More information about the four DEMs used, such as contour maps and 3D plots, will be presented when each DEM is discussed individually in the following sections.

Each of the four data sets represents a sample of different physiographic regions which reflects its overall physical characteristics. The division of the United States into natural regions has been intensely researched (e.g., Hunt 1974, Thornbury 1965, Atwood 1964). Hunt (1974) classified the United States into physiographic provinces so that each province has its own peculiar characteristics.

"The natural regions of the United States and Canada" (Hunt 1974) was used as the primary source for defining the physiographic divisions used in this thesis. The physiographic regions to which the four data sets belonged were generally easy to determine as the locations are found within the borders of their physiographic regions. The physiographic regions which are represented, and the DEMs which represent them, are presented in Table 4.2. In the

following discussion, the physiographic regions will be referred to by their names.

Physiographic Regions	DEM	DEM Locations
Great Plains	DEM1	Krone Ranch, MO
Appalachian Plateaus	DEM2	Herbert Domain, TN
Coastal Plain	DEM3	Benndale SW, MS
Colorado Plateaus	DEM4	Cedar Breaks SE, UT

DEM Name	QUAD Name, State	Longitude (W) ° ' "	Latitude (N) ° ' "
DEM1	Krone Ranch, MO	112 15 00	47 07 30
DEM2	Herbert Domain, TN	85 15 00	35 45 00
DEM3	Benndale SW, MS	88 52 30	30 37 30
DEM4	Cedar Breaks SE, UT	112 52 30	37 30 00

Corresponding topographic contour maps

DEM1	Comb Rock, MO	N4707.5-W11207.5/7.5
DEM2	Herbert Domain, TN	N3545 -W8507.5 /7.5
DEM3	Vestry, MS	N3037.5-W8852.5 /7.5
DEM4	Navajo Lake, UT	N3730 -W11252 /7.5

Table 4.2 DEM Locations and Their Physiographic Regions

DEM Name	Min	Max	Mean	S.D.	Skew-ness	Kurt-tosis
DEM1	1266.	1529.	1374.	48.60	.027	.409
DEM2	481.	646.	558.	39.43	.489	-.845
DEM3	9.	66.	34.	15.90	.412	-.531
DEM4	2446.	3163.	2889.	191.22	-.766	-.128

Table 4.3. Summary of DEM Elevation Data

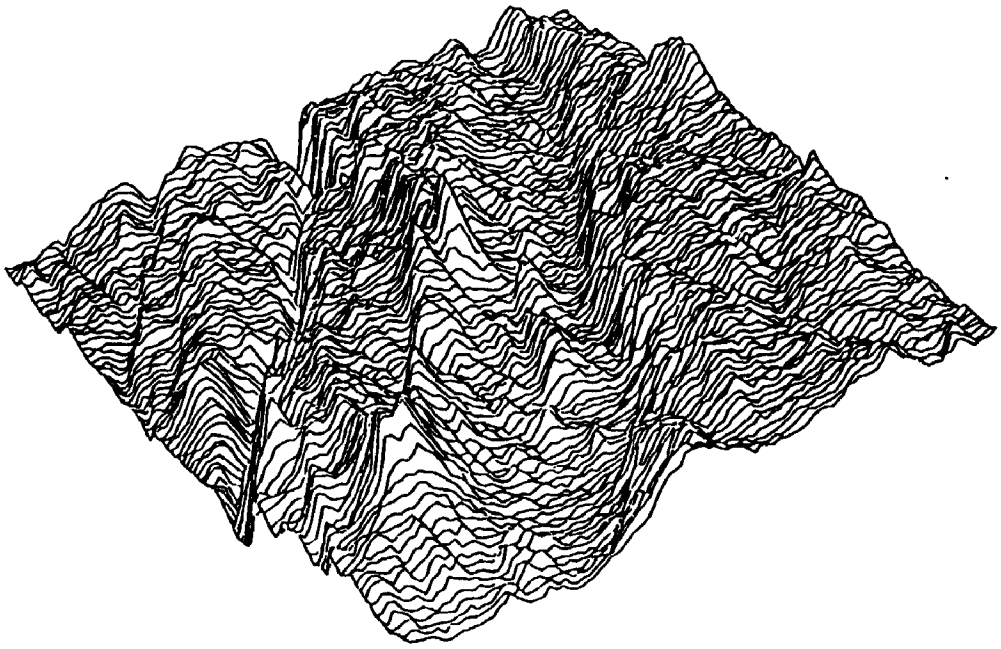


Figure 4.1. The Three Dimensional Plot of DEM1

Figure 4.2. The Three Dimensional Plot of DEM2

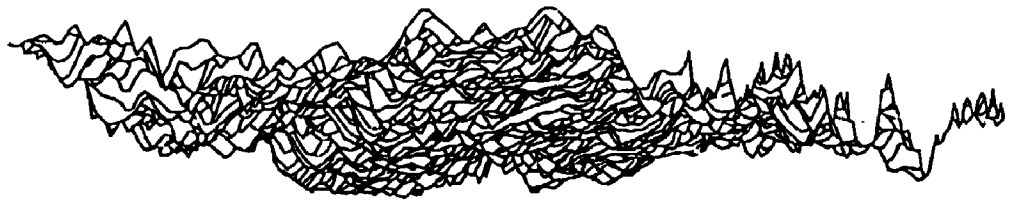


Figure 4.3. The Three Dimensional Plot of DEM3

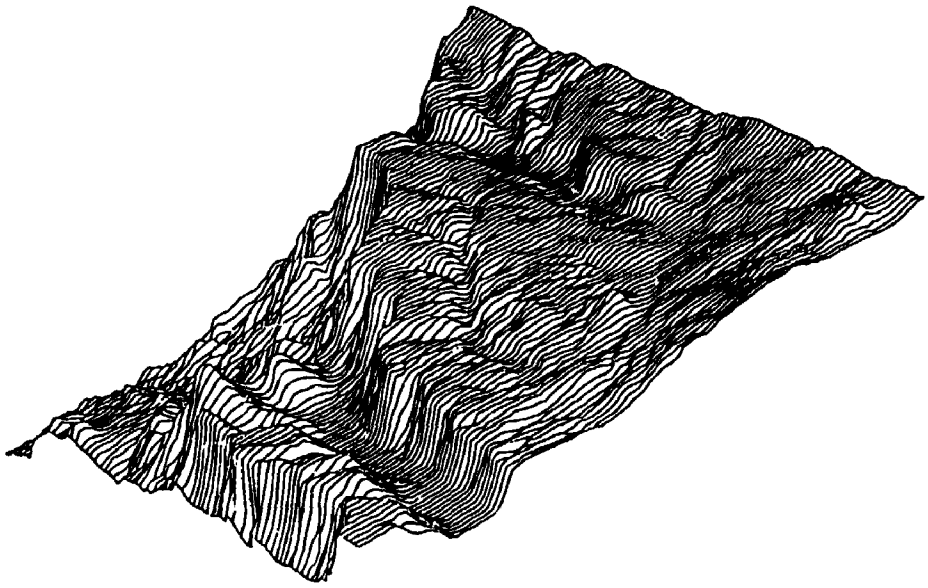


Figure 4.4. The Three Dimensional Plot of DEM4

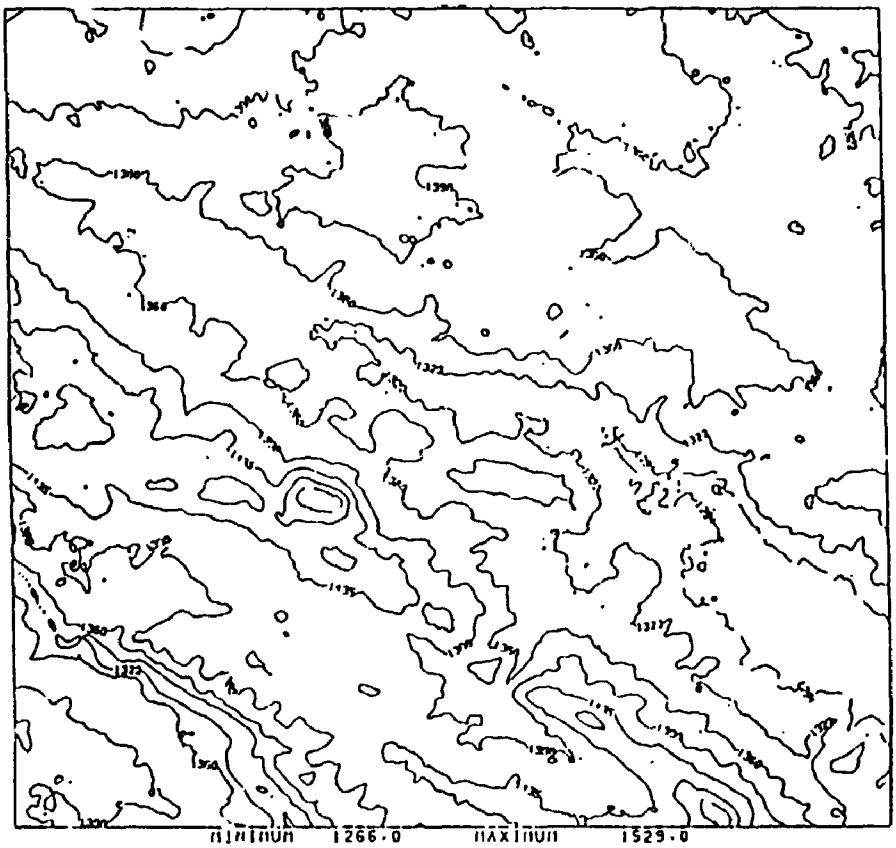


Figure 4.5. The contour Map of DEM1

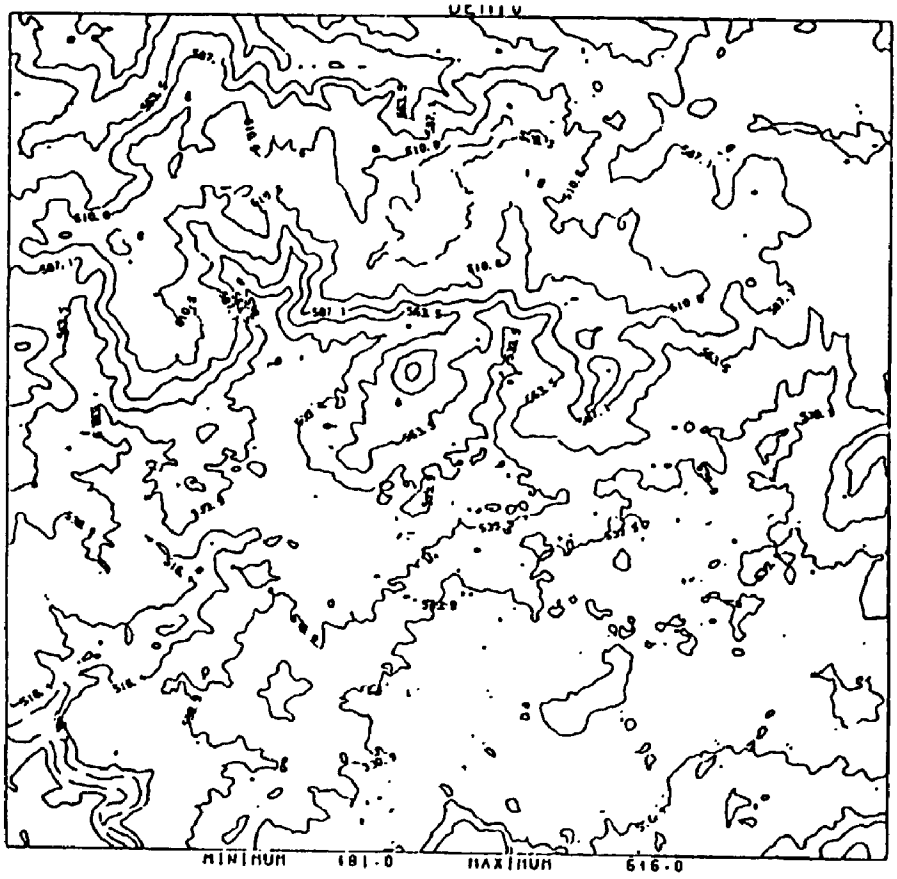


Figure 4.6. The Contour Map of DEM2

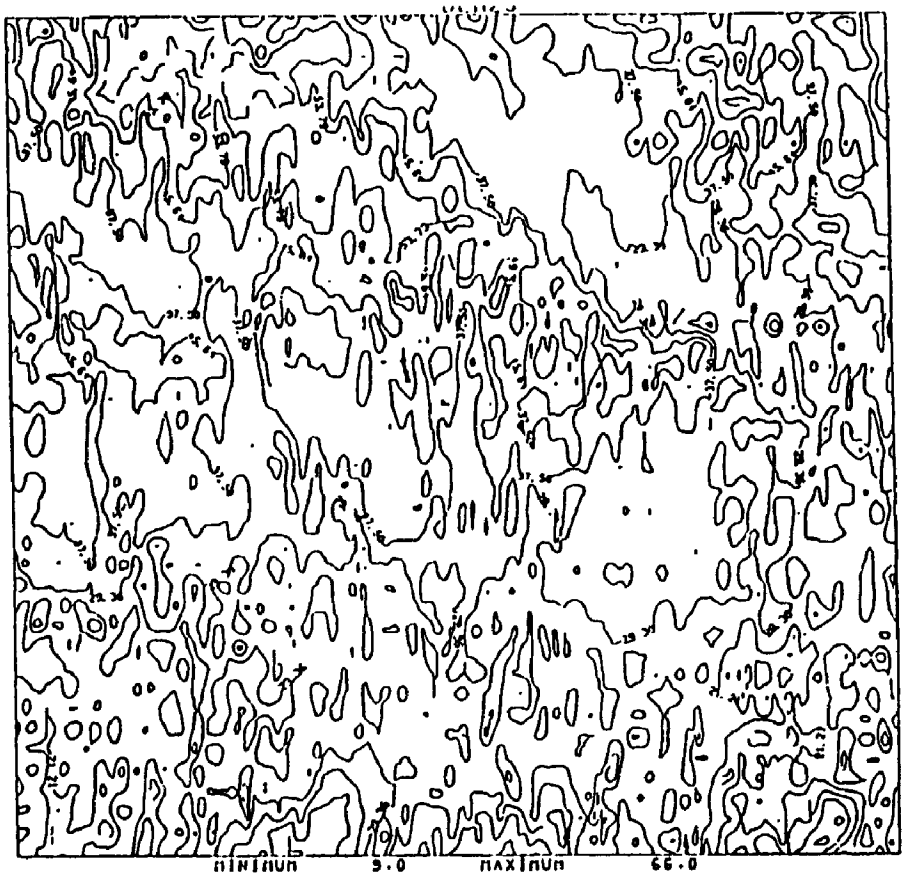


Figure 4.7. The Contour Map of DEM3

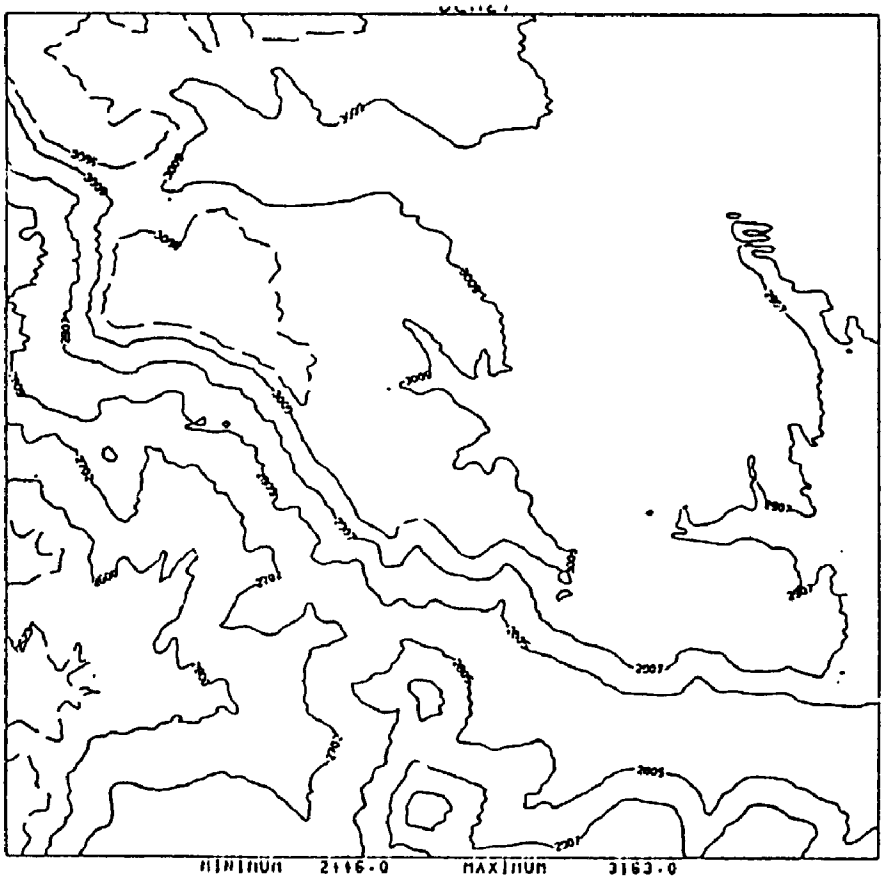


Figure 4.8. The Contour Map of DEM4

DEM1 is located near the northwestern edge of the Great Plains (Table 4.1., Table 4.2.). The Great Plains may be described as a broad plateau with slopes running eastward from the foot of the Rocky Mountains where elevations are still relatively high to the lower elevations of the central lowlands. In the northern part of this region the rivers can be entrenched a few hundreds of feet into the plateau, and dome mountains can rise to 1500 to 2000 feet above it. Broad anticlines and synclines may be found between the dome mountains, with cuestas clearly marking their limits (Hunt 1974, Thornbury 1965, Klinkenberg 1988). DEM1 from this region has low skewness and high kurtosis which reflects steep local reliefs spread widely over the DEM (Table 4.2., Figure 4.1., and Figure 4.5.). In fact, one can easily observe that DEM1 has ridges (The Reef, peaked at around 4600 to 4900 feet) and valleys running in a northwest-southeast direction, enclosing a basin of elevations around 4100 to 4250 feet (Nicholas Basin). The southwestern part of DEM1 has steeper slopes while the slopes in the northeastern part are gentler. In this sense, DEM1 has captured a representative portion of the range of relief present in this province.

DEM2 falls within the southern (Tennessee) section of the Appalachian plateau physiographic region (Table 4.1., Table 4.2.). The characteristics or this physiographic region are deeply incised valleys with steep hillsides. We

may also observe considerable local relief (Hunt 1974). The standard deviation of elevation lies between the standard deviations of the four DEMs, so in that sense DEM2 does not stand out (Table 4.2). However, considerable physiographic differences do exist within the region, as "some areas are deeply dissected with closely spaced valleys between narrow ridges ...; others may be equally deeply dissected by widely spaced valleys that are separated by broad, open uplands" (Hunt 1974, pp. 263, quoted by Klinkenberg, 1988). These characteristics may be represented by DEM2 (Table 4.3., Figure 4.2., and Figure 4.6.). The area covered by DEM2 is basically a portion of the Cumberland Plateau in Tennessee. Buck Ridge, with a direction closed to northwest-southeast, runs across the DEM2. It is about 200 feet higher than the adjacent lower land, where Herbert Domain is located.

DEM3 is centrally located within the East Gulf section of the Coastal Plains region (Table 4.1., Table 4.2.). This region has low topographic relief, a reflection of its origins as a sea bottom. The East Gulf section consists of a dissected, belted coastal plain with a series of cuestas and lowlands forming the dominant physiographic features (Hunt 1974, Klinkenberg 1988). DEM3 displays little topographic relief - note the very small range of elevations, and the very low standard deviation of elevations (Table 4.2, Figure 4.3., and Figure 4.7.). The area included in DEM3 is close to the southern border of Mississippi, covering places

such as the University of Mississippi Forestry and adjacent Desoto National Forest area. Gentle hill slopes are scattered all over the area. Red Creek runs across the northeastern part of the region. Although having smaller changes in local reliefs, the overall land surface depicted by DEM3 serves as a prototype of a flatter landscape to be tested for visibility problems.

DEM4 is located near the western edge of the Colorado plateau in the high plateau section (Hunt 1974, pp. 426, Klinkenberg 1988) (Table 4.2., Figure 4.4., and Figure 4.8.). This region contains the highest plateaus in the continent, with extensive igneous structures and steep-walled canyons. In the high plateau section lava capped plateaus are separated by wide, flat-bottomed north-south tending valleys. These fault-related features are found in this section of the province more than in any other part. DEM4 also displays considerable topographic relief, with both large standard deviations and a high range in its elevations. A clear boundary delimiting a plateau may be observed in DEM4. Several flat-bottom valleys are clearly shown in Figure 4.4. The boundary of the plateau (elevations around 9500 to 11000 feet) is running in northwest-southeast direction. On the plateau and also in a similar direction, a ridge line (Pink Cliffs) runs along Navajo Lake in the southeastern part of the region. Two other valleys (Midway Valley and Long Valley) may also be identified on the plateau. The

elevations of the very northwestern part of the area (Cedar Breaks) drop sharply to around 8800 feet. The relief displayed in DEM4, relatively flat plateau and fault-like elevation changes in the area, describe a rather different topography than those of the other three DEMs and serve as another type of surface to test the visibility concepts and algorithms.

In summary, random surfaces are used to test the feasibility and validity of visibility concepts and algorithms; while the four DEMs are used to represent a wide range of topographic features, such as ridges, valleys, plateaus, and various degrees of variations in reliefs. The intention is to examine the distribution and topographic features that may be associated with visibility information in this study.

In the following section, existing algorithms for converting gridded DEMs to TINs are reviewed. A comparison analysis of the advantages and disadvantages of the existing algorithms is conducted and presented with tests and results. A new algorithm that was developed to improve the shortcomings of the existing algorithms is also presented. The new algorithm was found to perform better for the objective of minimizing the difference of elevations between the original DEM surface and the surface approximated by the converted TIN.

4.2. DEM Conversion Algorithms

As described in the beginning of this chapter, available DEMs from U.S.G.S. were compiled in a grid format that must be converted to a data structure for TIN models. The objective of any algorithm used to convert grid DEMs to TINs is to minimize the differences of elevations between the original surface and the surface approximated by the converted TIN. The procedures involved during the conversion include a series of selection procedures by which each grid point is evaluated and then kept or discarded. The points that are kept by applying these criteria are then used to construct a TIN. Thus, the critical factors that affect the effectiveness and performance of a conversion algorithm are those involved with the selection procedures.

Different concepts lead to different selection processes and therefore produce different results. However, the basic philosophy is always based on the first law of geography, suggested by Tobler (1970): everything is related to everything else, but near things are more related than distant things.

The structure of a terrain surface can be characterized by a set of "surface-specific" points, such as the peaks, pits and passes, and a set of lines, the ridges and channels, that connect peaks, pits, and passes. Peaks, pits and passes form the major portion of the set of surface-specific points.

Peaks are points that are relative maxima, i.e. higher than all neighbours. Pits, likewise, are relative minima. Passes are saddle points on the surfaces. There are two alternate definitions of ridges and channels. Warntz (1966) defined a ridge as a line connecting adjacent peaks along a path of highest altitude. Similarly, he defined channels as lines connecting adjacent pits along paths of lowest altitude. Passes are the points at which ridge and channel lines intersect. These ridges and channels connect the peaks and pits into a network. While they connect into a network, they do not necessarily follow significant relief features in all areas. For example the border of a plateau will not likely be captured by this approach of connecting adjacent peaks or adjacent pits as points on the border are not necessarily peaks or pits.

Ridges and channels can also be defined as lines of divergent and convergent slope, respectively. These correspond well with intuitive notions of ridges and channels, including spur ridges and tributary channels, and should be represented with high accuracy for the research purposes of this study. Peaks, pits, passes, and ridge or channel lines are especially important to this study because of their significance as potential viewpoints (the operational definitions for various surface-specific points were included in section 4.1.1.).

In the following sections, three existing methods of selecting points to convert a grid DEM to a TIN are described and discussed. A comparison study is conducted to examine the performances of various methods. Finally an algorithm is developed to improve the shortcoming of the existing algorithms.

4.2.1. The Fowler/Little algorithm

The earliest attempt to extract a TIN model from dense raster data was suggested by Fowler and Little (1979). An initial approximation is constructed by automatically triangulating a set of feature points derived from the raster model. The method works by local incremental refinement of this model by the addition of new points until a uniform approximation of specified tolerance is obtained.

The first phase of generating a TIN from a very dense raster DEM is the extraction of the skeleton of surface-specific points and lines. Fowler and Little suggested a method to identify possible points on ridge or channel lines: for each point in the square grid, the heights of three cells adjacent to it are compared with the height of the central cell. If the cell is at grid location (i,j), the cells at (i,j+1), (i+1,j), and (i+1,j+1) are examined. The lowest and highest cells of the four are marked as not being ridge and channel candidates, respectively. The result is that cells

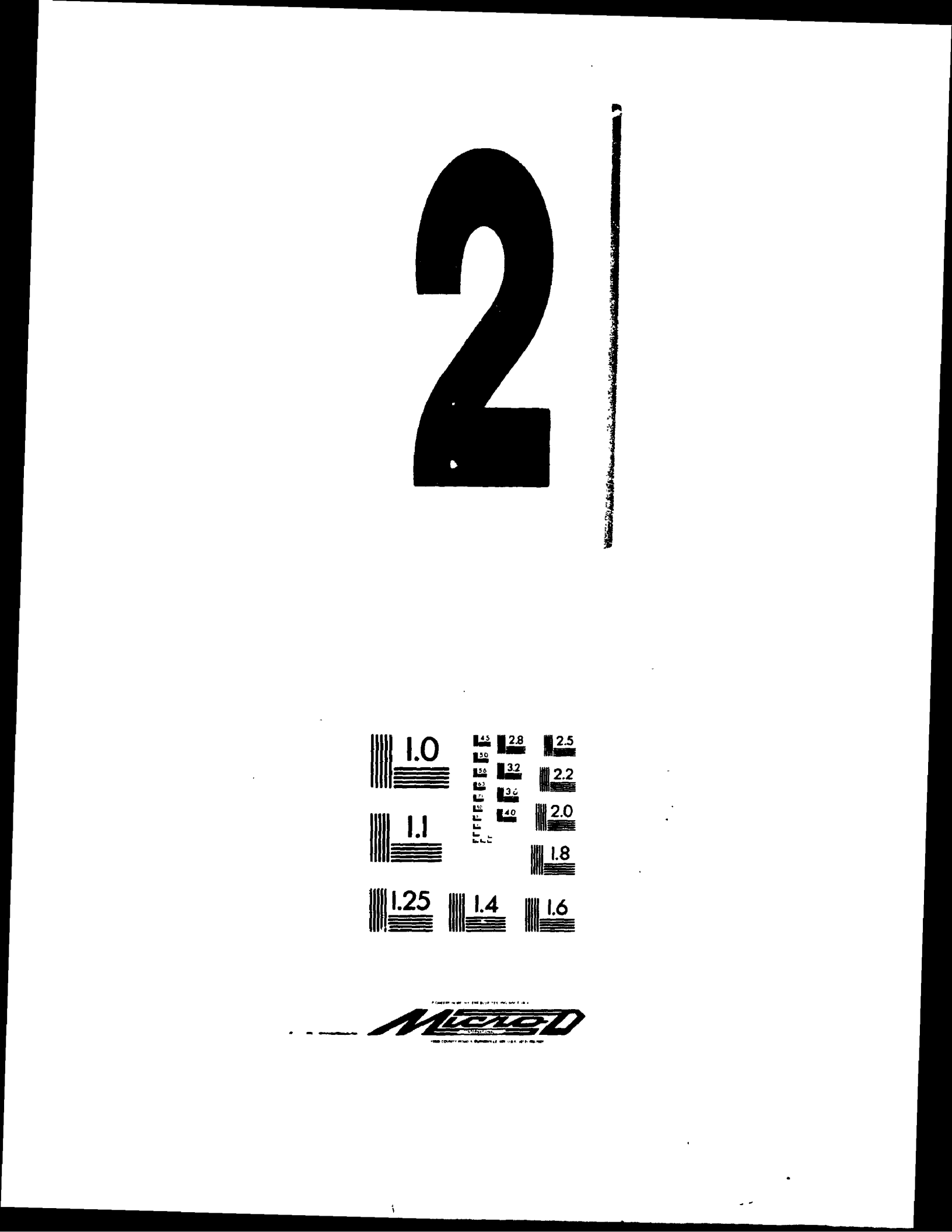
which are not marked are potential ridge (or channel) candidates.

These potential ridge and channel points lie on or around the ridges and channels; before they can be used, they must be connected into lines. The following is a description of the ridge finding process; a similar procedure with the operators inverted is used to construct channels. The connection process starts at 'pass' points and climbs to the neighbouring ridge candidates at the highest altitude. 'Pass' points are found by examining all ridge candidates and accepting all those which are lower than all neighbouring ridge candidates. From these starting points, the neighbouring candidate at the highest elevation is added to the growing ridge, continuing until the ridge terminates at a peak or joins an existing ridge.

The strings of cells extracted in this manner form the ridges, and a similar procedure constructs the channels. The denseness of the sampling of the source DEM results in many redundant points along these lines. To yield a more simple characterization of the surface graph, a line approximation or generalization procedure is used to drop those points within a small multiple of the model tolerance. The method suggested by Douglas and Peucker (1973) is recommended to reduce the number of points representing a ridge line or a channel line. Douglas and Peucker (1973) suggested an algorithm for generalizing cartographic lines. The set of

points which are left after simplification includes the peaks and pits, which are necessarily included as the endpoints of surface lines, and those points which are "significant" in the definition of the surface lines. These essentially lie at ridge or channel junctions and at significant bends in the surface lines. Together with the points selected on the border, these form the input to the triangulation step.

Once the border points and structural points have been processed by the Delaunay triangulation program, there exists a triangulated network which can be thought of as a first approximation to the DEM. The next stage is to add to this model a set of support points in order to ensure the statistical fidelity of the model as well as to force the structural points to be interconnected with structurally significant edges. The differences between the original surfaces and the surface approximated by the TIN generated by the first phase of the algorithm are computed for every location of grid points. If the difference is over the pre-specified tolerance at a location of grid point, the point is added to the output point set. After all points in the DEM set have been examined and no further points are found to be added to the TIN, the process is terminated


2
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0

4.2.2. The very important points algorithm (VIP)

Another algorithm for selecting points from a dense grid DEM relies on an estimate of the significance of each point in the grid (Chen and Guevara 1987). The significance is measured by calculating how well each point is approximated by its eight neighbours. If the point to be evaluated is located at (i,j), the opposite pairs include (i-1,j+1)-(i+1,j-1), (i-1,j-1)-(i+1,j+1), (i-1,j)-(i+1,j), (i,j+1)-(i,j-1). For each pair of neighbours, the distance from the centre point to the line connecting the neighbours is calculated; the shortest (perpendicular) distance is used instead of the simple vertical displacement, to compensate for the effect of different slopes. The (unweighted) average of the four distances is used as a measure of the significance of the central point. After calculating a significance for every point in the matrix, the least important points are discarded, based on either a predetermined significance level or a desired number of points. This routine, under the name of VIP (Very Important Points), was first presented in a conference, AUTO-CARTO 8, supported by American Congress of Surveying and Mapping and was then adapted in the TIN model of ARC/INFO by Environmental Systems Research Institute, Redlands, California.

4.2.3. The hierarchical triangulation algorithm

The third existing algorithm was introduced by de Floriani et al. (1985b), named hierarchical triangulation. This algorithm uses an additive procedure in which points are selected to "reduce the maximum error between a piecewise linear approximation o the surface using only the selected points and the elevations of the points not selected" (de Floriani et al. 1985, pp. 667).

The extraction of a TIN model from a dense DEM suggested by de Floriani et al. (1985b) consists of two steps:

1) Initial subdivision of the data points into mutually exclusive and completely exhaustive triangular subsets.
2) Subdivision of each subset into nested triangles by hierarchical triangulation.

A convex hull is first constructed around the data points. An internal point is then selected, and the triangles connecting the hull to the internal point are constructed. Ideally, this internal point is selected such that the maximum error within the resulting triangles is minimized. If, however, the total number of points is large, or if the region is not simply connected (for example, if the area of interest contains a lake) then we may have to accept a non-optimal subdivision, which can be obtained more rapidly.

The result of the initial triangulation of a simply connected region is a two-level tree of as many triangles as there are edges in the convex hull. In each of these triangles the surface, as represented by the elevation values of the data points, now will be approximated by a set of nested triangles. These triangles are obtained hierarchically; at each step of the algorithm one of the existing triangles is selected for subdivision. The triangle selected for subdivision is the one which has the maximum error where the vertical distance of the farthest point in each triangle is the error associated with that triangle.

A subdivision consists of joining the data point with the maximum error within the triangle to each of its vertices. The subdivision proceeds until one of the following three conditions is met:

1) All of the remaining data points are within a preset tolerance from the piecewise linear approximation represented by the existing triangulation.
2) All of the data points have been used. If this stopping rule is used, then the order in which the points are selected is preserved in the data structure, so it is possible to truncate the resulting tree to the equivalent of one generated with rule (1).
3) A preset number K of data points have been used. The number of points effectively used can be less

than K when this subset gives a better approximation.

4.2.4. Summary

In summary, the three algorithms described above each have their advantages and disadvantages. A common property of all of the algorithms is that the solutions are dependent on predetermined parameters: these are 2 tolerances in the case of Fowler/Little algorithm, and either a tolerance or a prescribed number of points in the case of VIP and hierarchical triangulation algorithms.

These three methods suggested that there may be two possible stopping rules when converting a dense grid to a TIN model:

1) a pre-set number of points to be selected and
2) a pre-set tolerance of difference of elevations between the original surface and the surface approximated by the converted TIN model.

While information loss must occur during the conversion process, it is not easy to choose one of the two possible stopping rules over the other in the cases of VIP and hierarchical triangulation. Either rule may be adapted for different purposes. Ideally, one might prefer to have as small a tolerance as possible while the size of the data sets is also as small as possible. Unfortunately there exists no

single solution to satisfy both requirements. Therefore, users will always have to find an appropriate middle ground between these two criteria for their own purposes.

The first algorithm suggested by Fowler and Little (1979), using surface-specific points, does not require a pre-specified number of points to be selected but does need two pre-set tolerance levels, a vertical tolerance to be used to decide where to stop adding support points and a horizontal tolerance to be used to reduce the number of points along ridge and channel lines. The recursive procedures of adding support points may fit the approximated surface very closely to the original surface under a small tolerance. However, the result will be a much larger number of points. In addition, there appear to be no unique, clear criteria to decide a proper tolerance level. The advantage of this algorithm and the hierarchical triangulation algorithm is that a user can control the difference of elevations between the original and the converted surfaces, but the disadvantage is the trade-off of obtaining a larger size of data sets.

The advantage of the second and the third algorithm, VIP and hierarchical triangulation algorithms, is the flexibility of controlling either the size of the output point set or a pre-set level of significance by which points with higher significance are kept and those with lower significance are discarded. The third algorithm, proposed by de Floriani et al. (1985b), also has the advantage of being able to control

the size of the output point set, but has the major disadvantage that very often it produces very long and thin triangles.

Among these three, the long thin triangles generated by the third algorithm are undesirable for computing visibility coverage. Therefore only the first two algorithms will be included for comparison with the new algorithm to be described in the next section.

4.3. The drop heuristic algorithm

The purpose of a conversion algorithm is to convert a dense grid DEM to a TIN model by properly selecting important points so that a surface approximated by the converted TIN model is as close to the surface of the original grid DEM as possible while the size of a converted data set is reduced to an acceptable level. An ideal conversion algorithm should have the ability to select only those important points and therefore to reduce the information loss. The success of conversion algorithms may be measured by the difference of elevations of the original and the converted surfaces, if the numbers of points selected are held constant.

Because of the problems associated with the three algorithms, the Fowler/Little algorithm, the VIP algorithm, and the hierarchical triangulation algorithm, a new conversion algorithm was developed for this study to avoid the tendency

of producing long and thin triangles, and to be able to control the size of the output point set while minimizing the information lost during the selection process. Unlike the VIP algorithm which establishes the importance of a point from the elevations of its grid neighbours, the new algorithm defines the importance of a point as the difference between its real elevation and an elevation interpolated from the approximated TIN surface.

This algorithm is based on the concept that the importance of one point in a set of points describing a topographic surface is closely related to the degree to which its associated elevation value may be interpolated or predicted from the elevations of its neighbours. This in fact is an extension of the first law of geography mentioned previously. Two critical factors here are 1) the definition of the neighbour relationship between points on the surface and 2) a proper interpolation method to be used to predict the elevation of a point from its neighbours.

The algorithm, **Drop Heuristic** is proposed here that attempts to improve the disadvantages associated with the three algorithms discussed in the previous section. The Drop Heuristic algorithm first includes all candidate points in the output point set and gradually drops those non-important points until a desired number of points is reached. The importance of a point is defined by the difference between its real elevation and the elevation interpolated after it has

been dropped and a new TIN using Delaunay triangles has been constructed. The larger the difference, the more important the point is.

The interpolation is described in Figure 4.9. First, the points are expressed in two dimensions (Figure 4.9 a), assuming that point O is found to have Delaunay neighbours A, B, C, D, and E. The dashed lines connect point O and its Delaunay neighbours. If point O is removed, Point A, B, C, D, and E will be connected by solid lines as the relationship of Delaunay neighbours would be re-defined by a new triangulation (solid lines). The next step is to locate point O with one of the newly constructed triangles. In the case of Figure 4.9b, it would be triangle (BCE). Construct a vertical line in space passing through point O. This vertical line must intersect the plane of the triangle (BCE). The absolute value of the difference between the elevations of point O and the intersection point O' is then defined as the difference between the real and interpolated elevations of point O.

The drop heuristic starts by adding a diagonal edge connecting the northeast point and the southwest point of each grid cell. Two triangles are created for each grid cell and the whole network serves as an initial output point set in which all points are initially selected and connected as Delaunay triangles. Each vertex in the output point set is then evaluated to determine how well it is approximated. If

the difference between the real and interpolated elevations for the point being evaluated is the smallest among all points being evaluated, the point is dropped from the output point set. The processes are repeated until a pre-determined number of points is reached or until no point is found to be associated with a difference of elevation which is smaller than a pre-determined threshold.

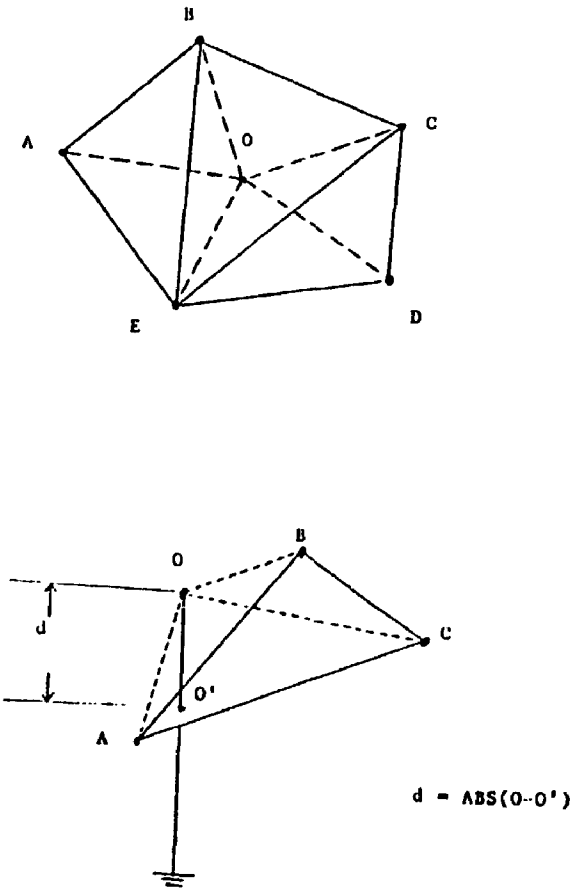


Figure 4.9. Interpolation Procedures of Drop Heuristic

The advantage of this algorithm is that it may be stopped either when a pre-set number of points is reached or when no point is found to have a difference of elevations smaller than a desired threshold. The threshold of this difference of elevations may be pre-set according to the accuracy of the original data sets. The number of points in the output point set may be controlled to fit the user's needs.

While this algorithm is able to avoid long and thin triangles (because the points are connected to Delaunay triangles) and to minimize the information loss at each step, it has the disadvantage of not re-computing at each step the differences in elevation for points which were dropped earlier, since these will change. Other procedures might improve the heuristic. One might pre-process the grid DEM by finding local maxima and local minima (peaks and pits) and force these to be kept for the output point set.

4.4. A Comparison Analysis of the DEM Conversion Algorithms

This section presents a comparison analysis of the performance of three algorithms, the Fowler/Little algorithm, the VIP algorithms, and the Drop heuristic, by applying all of them to DEMs used in this thesis.

The objective may be summarized as follows:

Given a grid DEM of n regularly spaced observation points and a pre-specified number of points of the output point set, or a difference threshold of elevation information loss, the objective is to select m points, where m << n, such that when the Delaunay triangulation of the m points is obtained, the interpolated elevations of the remaining n-m points are as close as possible to their observed values.

The algorithm which attempts to reach a solution close to optimum on this objective is as follows:

0. include all n points into the output point set initially;
1. to each grid cell, add a diagonal line segment connecting the upper left and lower right corners to construct the initial Delaunay triangulation for the grid DEM;
2. for every point, test the following steps:
   a. find its Delaunay neighbours;
   b. reconstruct a Delaunay triangulation among the Delaunay neighbours without the tested point;
   c. find out in which triangle the tested point is located;
   d. construct a vertical line passing through the tested point;
   e. compute the intersection of the vertical line and the plane of the triangle in which the tested point is located;
      (the elevation of the intersection found is the estimated interpolation of the tested point)
   f. compute the absolute difference between the real and estimated elevations for the tested point.
3. drop the point which has the least absolute difference between its real and estimated elevations from the output point set;
4. add edges of newly generated Delaunay triangles into the existing triangulation;
5. check if the desired number of points is reached, if so, stop, otherwise go to step 2.

Algorithm 4.1. Drop Heuristic Conversion Algorithm

Before any comparison may be conducted, however, there are a few parameters that needed to be held constant so that the comparison may be consistent. In the Fowler/Little algorithm, a vertical tolerance of elevation difference and a horizontal tolerance used in line simplification are needed in order to terminate the processes; while the VIP and Drop heuristic algorithms need a pre-defined number of points.

The selection of a proper vertical tolerance level for the Fowler/Little algorithm is never an easy task as there exists no directly related notion to support any pre-set tolerance level, except maybe the accuracy of the original data. Ideally, a suitable tolerance would at least be equal to or less than the data accuracy associated with the original DEM. However, this tends to produce a very large data set as the data accuracy is 1 meter for the 7.5 minute quadrangles DEMs (Elassal and Carusoit, 1983) and by using 1 meter as the vertical tolerance the Fowler/Little algorithm produces output data sets with large numbers of points, ranging from 4751 (DEM1), 3877 (DEM2), 4823 (DEM3), and 2633 (DEM4) points. In order to reduce the output data sets to a manageable size, the vertical tolerance was set to 10% of the elevation range (zmax-zmin). Horizontal tolerance was set to 30 meters - equal to the grid size of the original DEM.

Comparisons were conducted by first fixing the tolerance used by the Fowler/Little algorithm, then by using the number

of points produced as the pre-specified number of output points for the VIP and Drop heuristics. A vertical tolerance level of 10% of the elevation range (zmax-zmin) was used for each DEM, i.e. 26.486 meters (DEM1), 19.181 meters (DEM2), 12.00 meters (DEM3), and 166.40 meters (DEM4). These values served as the tolerance that was used to judge whether a support point from the original grid DEM will be added to the converted TIN model. The tolerance also represented the largest allowable error between the original and approximated surfaces. The numbers of points outputted from this algorithm, 1551 (DEM1), 1733 (DEM2), 1862 (DEM3), and 657 (DEM4), were then used as the pre-set number of points for the VIP and Drop heuristic algorithms. In order to investigate their performance at a different scale of problems, another set of TIN models was generated from the same four DEMs with the vertical and the horizontal tolerances doubled. The results are four smaller-sized data sets: 341 (DEM1), 425 (DEM2), 414 (DEM3), and 332 (DEM4) points.

After a TIN model was extracted from each DEM by the three conversion algorithms, differences of elevations between the original DEM surfaces and their corresponding surfaces approximated by converted TIN models were computed. These differences of elevations are defined as the error created when converting the grid DEMs to TIN models. Thus the errors may be obtained by computing the differences between the real

elevation and its interpolated elevation from the converted TIN model for every point on the original surface.

For any point of location (i,j) in the grid matrix, we first located the triangle on the approximated surface which contained the location (i,j) and computed the elevation difference as described in Figure 4.9b. The difference between the original elevation and the interpolated elevation on the surface approximated by the converted TIN was then defined as the error.

The performance of the three algorithms in converting four DEMs are summarized in Table 4.4, in which means and standard deviations are listed for all errors, positive errors (the elevation approximated by the triangulated surface is higher than the real elevation), and negative errors (the elevation approximated by the triangulated surface is lower than the real elevation).

In general, the Drop heuristic performed better than the other two algorithms by producing TIN models with lower mean errors and narrower standard deviations of errors for all four cases. The Fowler/Little algorithm performs better than the VIP algorithm for three cases (DEM2, DEM3, and DEM4) with the exception of DEM1. The topographic features in DEM2, DEM3, and DEM4 (Figure 4.6, Figure 4.7, and Figure 4.8) are clearly more suitable for the Fowler /Little algorithm which focuses more on linear topographic features than on local variations used by the VIP algorithm. Between the VIP and the

DEM1 Krone Ranch, Montana	Drop Heuristic	Fowler/ Little	VIP
Vertical Tolerance		26.48	
No. of Pts.	1551	1551	1551
All Error			
Mean	5.307	7.454	5.547
SD	10.979	22.264	14.269
Positive Error			
Mean	6.254	9.579	7.343
SD	13.122	25.327	17.951
Negative Error			
Mean	3.912	4.556	3.017
SD	6.430	16.800	5.047

DEM1 Krone Ranch, Montana	Drop Heuristic	Fowler/ Little	VIP
Vertical Tolerance		52.572	
No. of Pts.	341	341	341
All Error			
Mean	14.770	16.842	16.179
SD	14.922	19.180	20.887
Positive Error			
Mean	14.262	18.421	19.409
SD	14.556	22.047	24.273
Negative Error			
Mean	15.318	15.050	12.204
SD	15.287	15.090	14.811

Table 4.4. Results of Three Conversion Algorithms

DEM2 Herbert Domain, Tennessee	Drop Heuristic	Fowler/ Little	VIP
Vertical Tolerance		19.181	
No. of Pts.	1733	1733	1733
All Error			
Mean	3.229	3.561	3.737
SD	5.700	7.119	9.177
Positive Error			
Mean	3.210	3.070	3.525
SD	5.083	5.211	6.903
Negative Error			
Mean	3.250	4.010	3.951
SD	6.275	8.471	11.005

DEM2 Herbert Domain, Tennessee	Drop Heuristic	Fowler/ Little	VIP
Vertical Tolerance		38.362	
No. of Pts.	425	425	425
All Error			
Mean	11.679	12.853	13.147
SD	13.039	13.479	18.597
Positive Error			
Mean	10.736	13.505	15.021
SD	12.757	14.106	14.552
Negative Error			
Mean	12.370	12.198	12.687
SD	13.192	12.717	16.432

Table 4.4. Results of Three Conversion Algorithms (Cont'd)

DEM3 Benndale WE, Mississippi	Drop Heuristic	Fowler/ Little	VIP
Vertical Tolerance		12.000	
No. of Pts.	1862	1862	1862
All Error			
Mean	1.952	2.843	2.776
SD	3.513	4.968	5.706
Positive Error			
Mean	1.952	2.516	2.902
SD	3.972	5.388	6.402
Negative Error			
Mean	2.429	3.060	2.667
SD	3.589	4.657	5.029

DEM3 Benndale SW, Tennessee	Drop Heuristic	Fowler/ Little	VIP
Vertical Tolerance		24.000	
No. of Pts.	414	414	414
All Error			
Mean	5.489	6.379	6.383
SD	5.608	8.177	9.945
Positive Error			
Mean	4.913	4.692	6.229
SD	5.585	8.860	8.178
Negative Error			
Mean	5.877	7.030	6.515
SD	5.590	7.814	11.241

Table 4.4. Results of Three Conversion Algorithms (Cont'd)

DEM4 Cedar Breaks, Utah	Drop Heuristic	Fowler/ Little	VIP
Vertical Tolerance		166.40	
No. of Pts.	657	657	657
All Error			
Mean	19.022	26.158	26.551
SD	35.077	50.478	41.218
Positive Error			
Mean	22.494	26.541	28.389
SD	34.267	35.360	33.796
Negative Error			
Mean	14.245	25.660	18.509
SD	35.621	65.067	47.216

DEM4 Cedar Breaks, Utah	Drop Heuristic	Fowler/ Little	VIP
Vertical Tolerance		332.80	
No. of Pts.	332	332	332
All Error			
Mean	30.286	36.100	33.410
SD	42.050	54.666	48.791
Positive Error			
Mean	36.073	38.278	39.170
SD	41.043	43.094	47.710
Negative Error			
Mean	21.043	32.618	26.247
SD	40.534	69.120	49.172

Table 4.4. Results of Three Conversion Algorithms (Cont'd)

Fowler/Little algorithms, DEM1 (Figure 4.5), which has the highest kurtosis in elevation, is obviously more suitable to be converted by the VIP algorithm that primarily considers local variations rather than larger scale topographic features such as ridge and valley lines.

From the analysis in this section, it was decided that the four DEMs should be converted by using Drop Heuristic algorithm. Table 4.5. shows the surface description parameters for the four converted TIN. Figures 4.10. to 4.13. show the converted TIN models from the four DEMs networked by Delaunay triangulation.

	TIN1	TIN2	TIN3	TIN4
MEAN	1371.070	552.080	34.300	2779.100
S.D.	44.050	38.000	13.140	189.450
C.D.	0.362	0.539	0.496	0.469
SKEWNESS	0.616	-0.072	-0.053	-0.027
KURTOSIS	3.537	2.287	2.167	2.141
SAC	0.127	0.307	0.098	0.324
VARI	7.078	6.787	1.264	64.162
TPSMEAN	12.168	12.671	19.404	5.654
TPSSD	16.861	16.957	21.788	10.283
TPSSKEW	2.433	2.332	1.401	4.962
PEK %	7.500	12.000	14.000	6.000
PIT %	8.000	9.000	15.000	4.500
PAS %	16.000	23.500	20.000	12.000
RDG %	32.000	24.000	17.000	34.500
CHN %	30.000	28.500	21.000	37.000
OTH %	6.500	3.000	13.000	6.000

(Variables defined as in Table 4.1.)

Table 4.5. Surface Description Parameters for Four Converted TINs

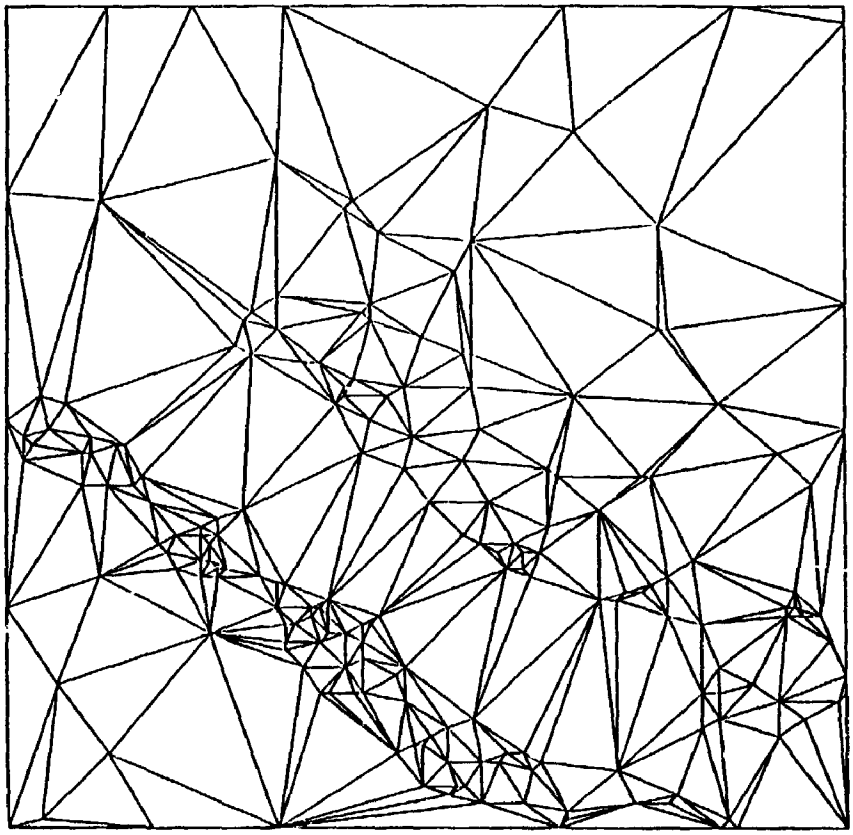


Figure 4.10. TIN Model Converted From DEM1.

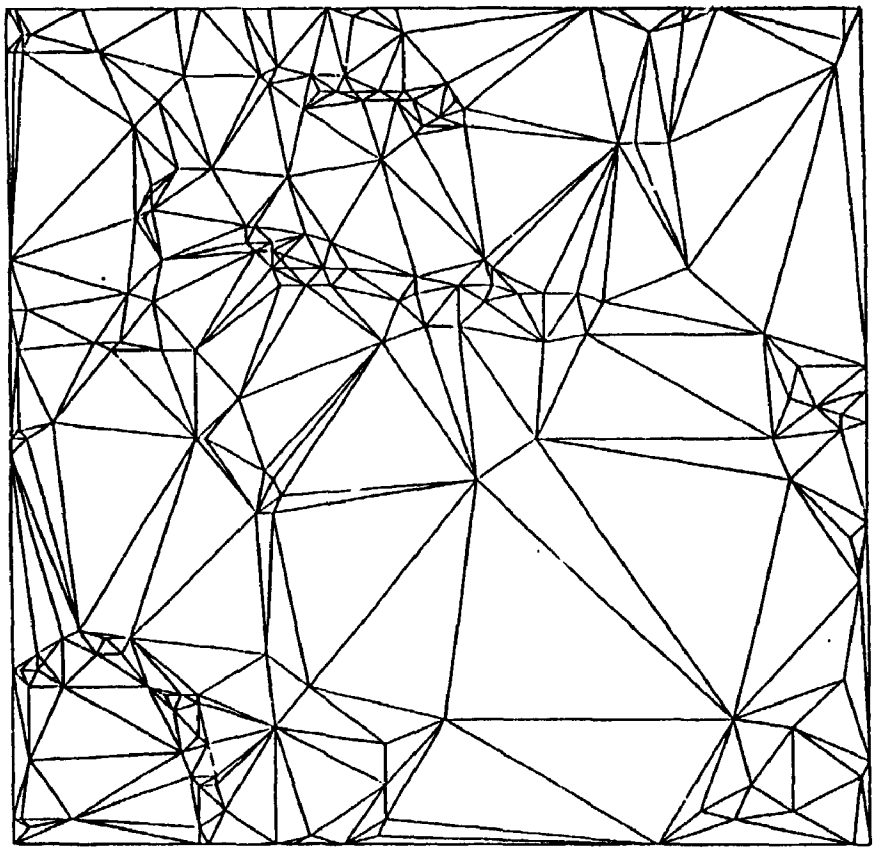


Figure 4.11. TIN Model Converted From DEM2.

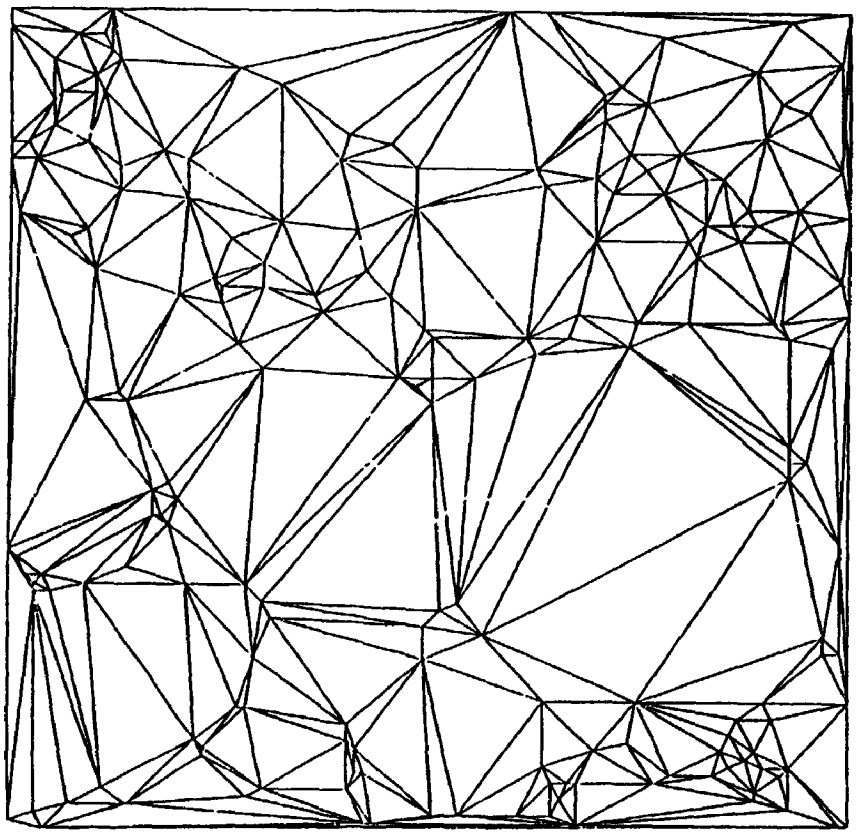


Figure 4.12. TIN Model Converted From DEM3.

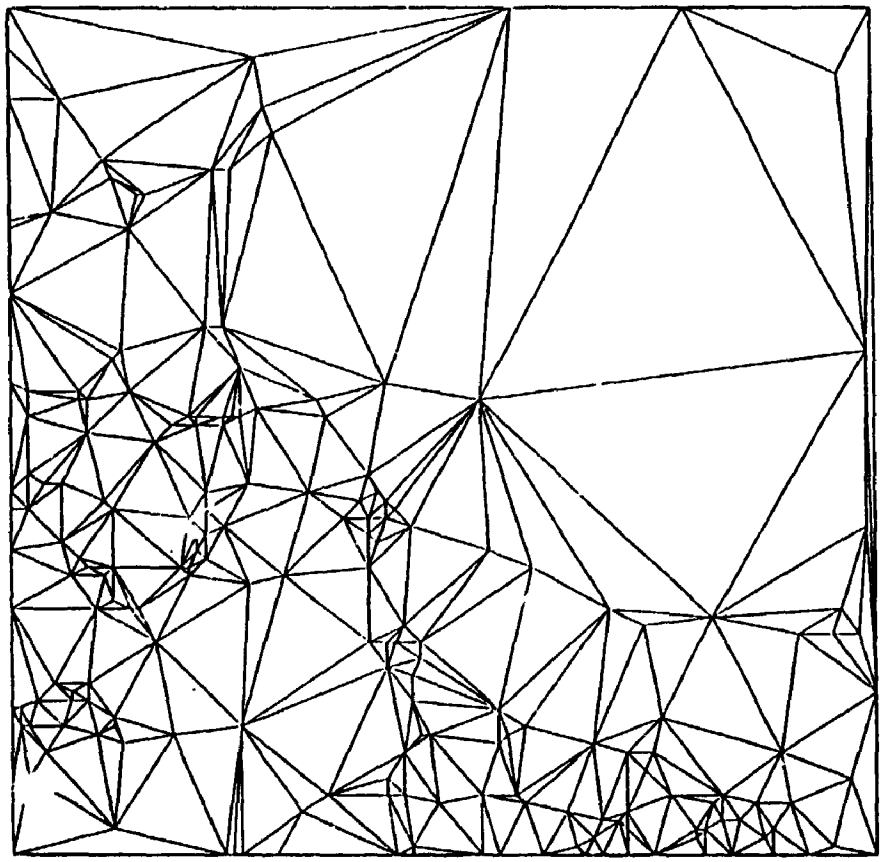


Figure 4.13. TIN Model Converted From DEM4.

Note that the basic statistics for the four TINs were different from those of the four DEMs because of the inevitable information loss during conversion processes which re-arranged the distribution of vertex elevations. For the converted TINs, the coefficients of dissection of four TINs are all within (0.35 , 0.65), indicating that none of the four TINs is either in their 'youthful' or 'old' stages. TIN1 has higher skewness and kurtosis of its elevations than the other three, indicating that TIN1 (see Figure 4.10.) has the highest peakedness and a higher proportion of its vertex elevations lower than the mean than those of the other three TINs. TIN2 (see Figure 4.11.) has a higher spatial autocorrelation coefficient but with a kurtosis that is just slightly higher than that of TIN3. It may also be observed that TIN2 has a steeper slope of the regression line on its variogram than TIN3, indicating that variance in elevation between far apart vertices are higher than those of nearby vertices in TIN2 than in TIN3. TIN3 (see Figure 4.12.), which has the lowest mean of elevations and the lowest standard deviation of elevations, was also found to have a low skewness and a low spatial autocorrelation coefficient in its elevations, indicating its relief variation is smaller than those of the other three TINs. It also has the lowest slope of the variogram regression line, indicating that the variances between distant vertices are not very much different from those between nearby vertices. With the highest slope mean, standard deviation,

and the lowest skewness in its angles of triangle planes, TIN3 would have more higher triangle slopes and more frequent but smaller relief changes. TIN4 (see Figure 4.13.) has the highest spatial autocorrelation coefficient and the steepest slope of its variogram regression line than those of the other three, indicating that the variances in elevation differences are much higher for distant vertices than for vertices that are nearby. In addition, the means and standard deviation of its triangle slopes are the lowest with most of its triangle slopes under the mean, compared to the other three, implying that more of the surface is gentle than the other three TINs. However, the high standard deviation of elevations of TIN4 implies that a dramatic elevation shift is also included.

4.5. Conclusions

From the above analysis, the drop heuristic is found to be the most satisfactory algorithm for converting a grid DEM to a TIN model for the objective addressed in this study. However, it should be noted that the conclusions here are based on the four U.S.G.S. DEMs. A more general conclusion can be made only by a larger number of examples, covering a wider range of topographies, geomorphological features, and terrain types at various scales. The primary advantage of the Drop heuristic is that it attempts to minimize the information loss during each step. Whether the point is actually a peak,

pit, pass, or a point on a linear topographic feature such as a ridge, valley, or channel, it can only be dropped if it can be closely interpolated by a surface modeled without it.

The other two conversion algorithms each have their strengths and weaknesses. The Fowler/Little algorithm has the advantage of capturing linear topographic features. The associated disadvantage of the algorithm is that there is not a clear and definite criterion to define a proper tolerance for the algorithm. Finally the VIP algorithm focuses more on local variations rather than topographic features of larger scale such as valleys, channels, or ridge lines. Its advantage is that the algorithm is fast and less complicated. But the disadvantage is the difficulty of defining a proper size for the output point set.

In conclusion, the four DEMs were converted by a new Drop heuristic to create four TIN models. These converted TIN models will now be used to compute visibility information for the analyses presented in the following two chapters. Chapter V reviews the existing algorithm for extracting visibility information and presents a new algorithm and an algorithm for computing the minimal visible heights for given viewpoints. Chapter VI describes solutions to the problems listed in section 3.2. and presents the relationship between extracted visibility information and topographic characteristics.

CHAPTER V VISIBILITY REGIONS AND MINIMAL VISIBLE HEIGHTS

How much of a surface one can see from any given point on that surface is important information. This visibility information may be used for a variety of applications, such as solving coverage problems and identifying significant topographic features. As discussed in the previous chapters, the surface, in this study, is approximated by a TIN model and the viewpoints are limited to the vertices of the TIN. The topographic surface is thus divided into subregions of triangles, using the Delaunay triangulation (the triangulation algorithm was described in section 2.1). The first step to compute visibility information is the extraction of visibility regions. With this, visibility information between triangles and given observation points may be further used to solve coverage problems (the solutions will be discussed in the next chapter) and for applications related to the identification of significant topographic features.

Largely due to interest in computer graphics, there is a great deal of literature on visibility and hidden line/surface removal. However, these algorithms were either

1) designed to determine only how an image of the surface model will appear from a given viewpoint;

they report the limits of visible areas as coordinates on an image and not on the model; or

2) designed for grid-based surface models.

Therefore, they are not appropriate for the extraction of visibility information in our case.

One specific algorithm, proposed by de Floriani et al. (1986) to be discussed in the next section, was specifically designed to extract visibility regions from a topographic surface approximated by a TIN model. There are, however, sufficient simplifications possible with less computational cost to warrant a new approach. The easiest, most beneficial simplifications are to note that there are no **bottom** surfaces; therefore, any triangle observed from the underside must be invisible, and the functions needed to determine the visibility between a triangle and a given viewpoint may be simplified to reduce the computational cost. In addition, the computation may be simplified by not computing visible or invisible parts of triangle with respect to a viewpoint but instead by only testing visibility of triangles as a binary attribute. This clearly involves an approximation, but for the purposes of this thesis the degree of approximation is assumed to be insignificant. If accuracy is important, it is always possible to have smaller triangles.

In the next section, the formalism and associated algorithm developed by de Floriani and others (1986) are

described as a basis for the discussion which leads to a new algorithm to be described in the following sections.

5.1. Existing Visibility Algorithms

Before we can describe the algorithm proposed by de Floriani (1986), a few definitions of required functions are necessary[2]. Note that in a topographic terrain model given by irregularly-spaced data points, each point consists of a coordinate triple (x,y,z). A triangulation of the data set divides the model into disjoint triangles by introducing edges between the vertices (data points). Each edge is adjacent to exactly two triangles unless the edge is on the convex hull of the data set, in which case it is adjacent to only one triangle. Two points on the surface are **mutually visible** if the line segment joining them does not pass through the surface.

De Floriani et at. (1986) suggested that a useful way of visualization would be to think of a **curtain** hung from each edge in the model. The curtain is defined as an imaginary vertical trapezoid constructed downward from any edge whose height is enough to intersect the line of sight connecting the viewpoint to the node or the edge being tested. Every point

---

2 This section is based on the paper by de Floriani et al., "A visibility-based model for terrain features". in Proceedings, the 2nd International Symposium on Spatial Data Handling, Seattle, Washington, 1986.

in the model which is invisible from a given viewpoint must be hidden by at least one of these curtains. More formally, the **curtain trapezoid** of an edge is the semi-infinite trapezoid formed by the edge and the two vertical rays dropped from its endpoints to z=-infinity.

The following functions are cited from de Floriani et al. (1986) so that the illustration of their algorithm may be consistent to their definitions:

> **ASPECT (P,T).** Every triangle is either edge-up or edge-down when viewed from P. T is EDGE-UP to P iff P is above the plane formed by extending T. T is EDGE-DOWN to P iff P is below the plane of T. (de Floriani et al, 1986, pp. 239). See Figure 5.1.
>
> **BLOCKS (E_1,E_2,P).** This Boolean function is true if there is some segment L from P to a point on E_2 which passes through the curtain trapezoid of E_1, and false otherwise. Intuitively, the curtain trapezoid of E_1 blocks some part of E_2 from view by an observer at P. The term "**2-blocking**" will also be used; this is the same definition applied to edges projected onto the horizontal plane (de Floriani et al. 1986, pp. 241). See Figure 5.2.
>
> **DEPTHSORTING.** This concept is entirely two dimensional, independent of elevation. A DEPTH SORTING of the edges E_0, E_1, E_2, ..., E_{n-1} with respect to P is a partial ordering of the edges such that for all i and j < n, if E_i 2-blocks E_j then i<j. Given this ordering, any edge which 2-blocks edge E has an index lower than the index of E. Also, any edge 2-blocked by E has an index higher than that of E. It must be observed that any edge which blocks another certainly 2-blocks it (de Floriani and Nagy, 1989, pp.241). See Figure 5.3.
>
> **SHADOW-SEGMENT (P,B,T).** Let PL be the plane formed by P and the two endpoints of B. Let R be the region of PL bounded by B and the rays drawn from P through the endpoints of B. If this semi-infinite trapezoid R intersects triangle T, then the line segment formed by the intersection is the SHADOW SEGMENT of B on T when viewed from P. If there is

no such intersection, the function returns NIL (de Floriani et al. 1986, pp. 243). See Figure 5.4.

SHADOW-SET (P,B). The SHADOW-SET of a blocking edge viewed from P is the set of all the shadow segments of P and B, that is, the union of SHADOW-SEGMENT (P,B,T) over all T (de Floriani et al. 1986, pp. 243).

IMMEDIACY (P,B,S) is a Boolean function where S is a shadow segment formed by P and B. S is an immediate shadow segment of B iff there is no segment from P to a point on S which intersects another element of SHADOW-SET (P,B) (de Floriani et al. 1986, pp. 243). See Figure 5.5.

SHADOW-CHAIN (P,B). A shadow chain is a maximal subset C of SHADOW-SET (P,B) such that every segment S in C can be reached from another segment in C by stepping between elements of C whose endpoints coincide (de Floriani et al. 1986, pp. 243). See Figure 5.6.

SHADOW RAY (V,B,Q). Q is any point on B or on a shadow segment. The point shadow of Q is a ray collinear with VQ, specifically the ray drawn from Q that does not pass through V. The ray is coplanar with all shadow segments of V and B, by definition. Note that the ray may pass freely above the model. If it intersects the model, then the point of intersection must belong to some shadow segment (de Floriani et al. 1986, pp. 243). See Figure 5.7 and 5.8.

Algorithm for exact visibility information

The following pseudocode demonstrates an algorithm proposed in de Floriani et al. (1986, pp. 245). Its input is a triangulated model consisting of np vertices P_i, i=1,np, and its output is a list of regions of invisibility R_i, one for each vertex P_i considered as a viewpoint.

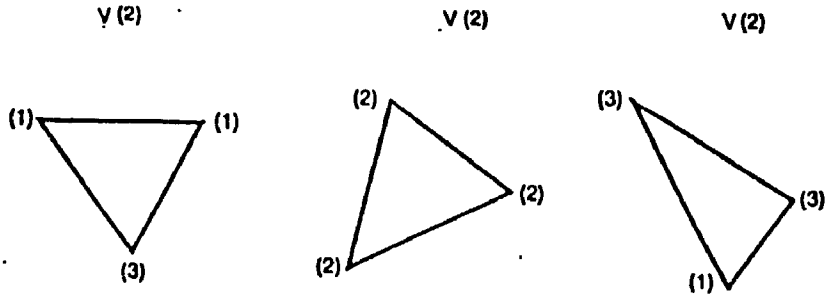


Figure 5.1. ASPECT of a triangle. The numbers in parentheses are elevations; V i the viewpoint. At the left is an EDGE-UP triangle; the centre triangle is EDGE-ON; the rightmost triangle is EDGE-DOWN.

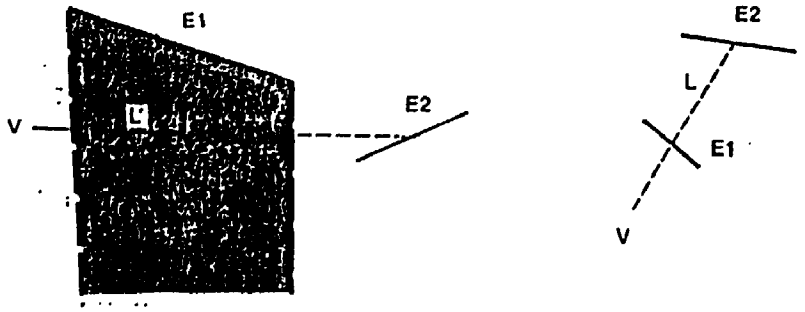


Figure 5.2. Edges BLOCKING one another. On the left, the curtain trapezoid of E1 obscures part of E2 from view by V. The segment L shows that at least one point on E2 is hidden. On the right, is a similar configuration in 2 dimension: E1 "2-blocks" E2.

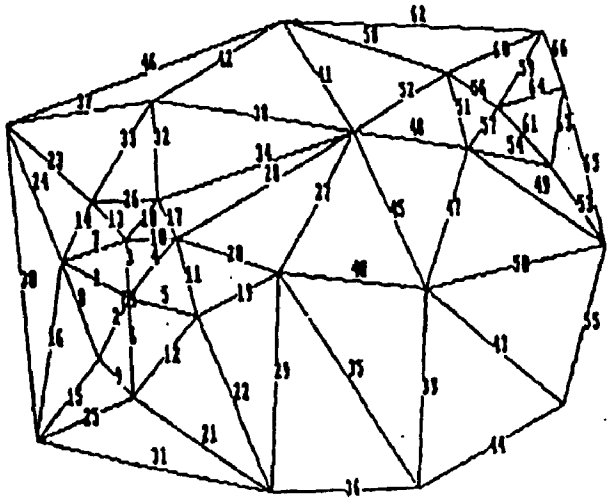


Figure 5.3. Output of algorithm DEPTHSORT. The viewpoint, left lower side, is marked by a square box. Each edge is labelled in a sequence which satisfies the depthsort criterion.

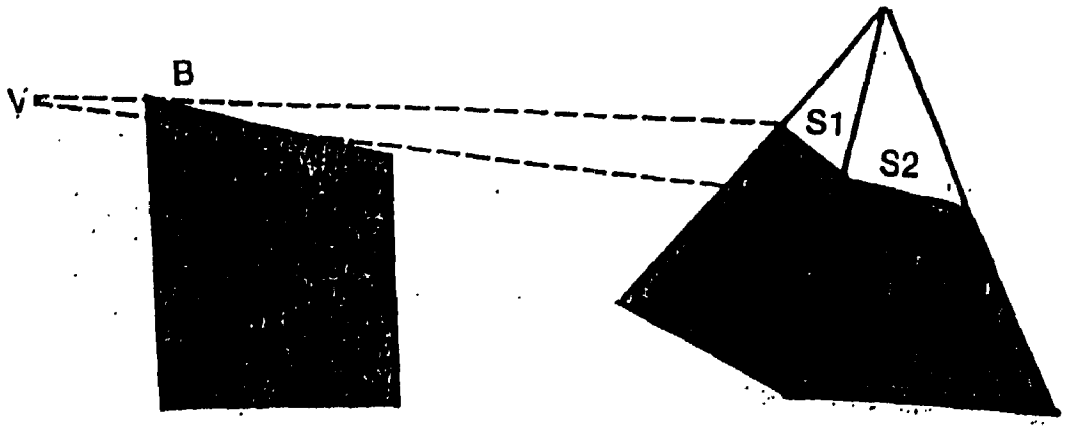


Figure 5.4. SHADOW SEGMENT. The two line segments shown are shadow segments on the triangles projected by blocking edge B when seen from viewpoint V. The shaded regions are invisible because of the curtain trapezoid of B.

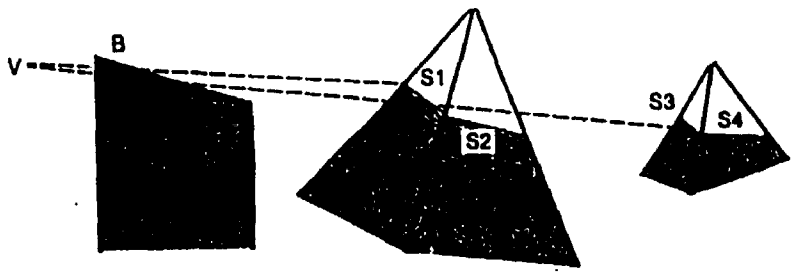


Figure 5.5. IMMEDIACY. There are four shadow segments. S1 and S1 and S2 are immediate but the S3 and S4 are not because segments which pass through S1 or S2 can be drawn from V to at least one point. on S3 and at least one on S4.

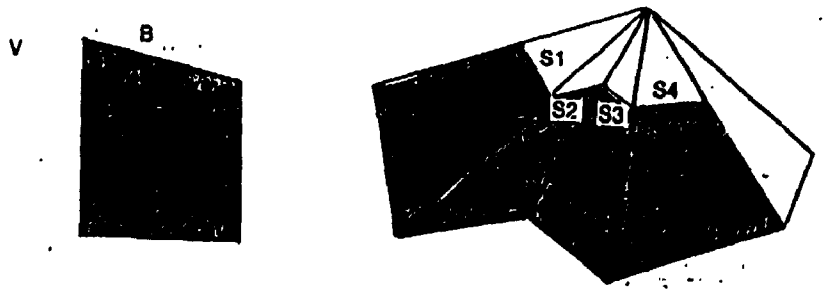


Figure 5.6. SHADOW CHAIN. The four shadow segments shown form a shadow chain because any segment can be reached from any other by moving along segments linked by coincident endpoints.

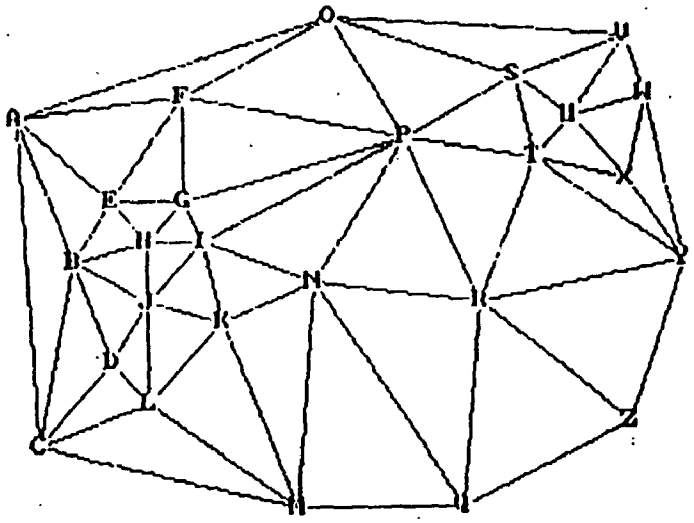


Figure 5.7. KEY FOR LABELS. All of the figures after this use the same model as an example (in Chapter iii). Each vertex is identified by a letter allowing standard geometrical notation for edge and triangle. In the remaining figures, the number of the vertices are elevations.

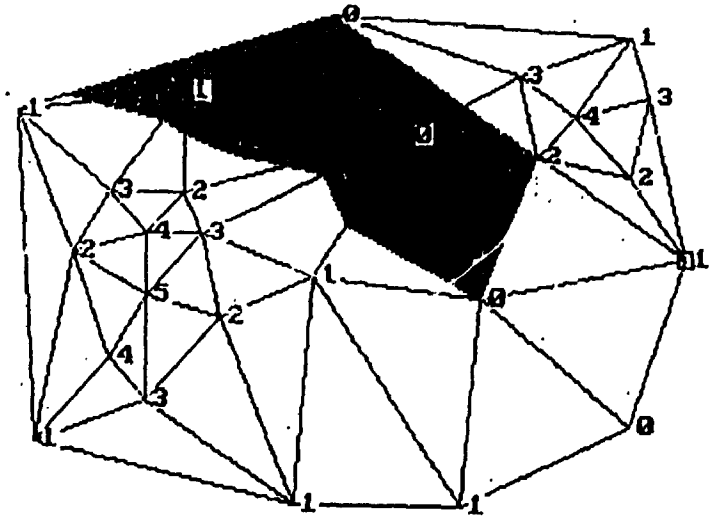


Figure 5.8. SHADOW RAY.
De Floriani et al. (1986) concluded that their implementation of the algorithm has a worst-case complexity of $O(np^3 M)$ for np points, where M is the expected number of blocking edges per viewpoint.

```
for each viewpoint P
     initialize R to empty
     depthsort edges
     for each edge E, considered in depth-sorted order
     examine ASPECT of both triangles adjacent to E
          if E is a blocking shadow set to empty
          for each triangle T
               add SHADOW SEGMENT (P,E,T) to shadow set
               form shadow chains from shadow set
               (* form partial region of invisibility *)
          start polygon edge list L at one endpoint of E
          initialize point p to the same endpoint of E
          while P is not the other endpoint of E do
               if P is endpoint of a shadow chain then
                    add shadow chain to L
               else
                    draw shadow ray from P.
                    add the segment between P and
                         intersection to L
                    add immediate portion of the shadow
                         chain to L
                    assign the final endpoint of the
                         chain to P
               endif
               (* the edge list L now forms invisible region
                    from P with respect to E *)
          merge polygon L with R
     endfor
endfor
```

Algorithm 5.1. de Floriani's Algorithm for Extracting Visibility Regions.

De Floriani's algorithm extracts visibility regions only in graphic form, that is, the visibility regions are only shown on the screen[3]. It lacks a proper data structure to

---

[3] The author wishes to express his appreciation to Dr. Georgy Nagy for kindly giving a complete set of his visibility programs at no cost. Without this generous offer, the comparison would not have been as straightforward.

store the visibility information. The algorithm is complicated; not only has the model to be pre-precessed for the ordering of edges in the network but also it has to be post-processed to determine shadow-segment, shadow-ray, shadow-chain, and shadow-set in order to form the invisibility regions and then visibility regions. Finally, it does not directly provide the visibility information in a data structure that is adaptable to this study. Therefore, new algorithms were developed to simplify and to speed up the computational procedures.

In the next section, a new algorithm is described. The concept required for this algorithm, are simply the status of 2-blocking between edges and the topology of the triangulation, which make this new algorithm easier to use and quicker in obtaining results.

5.2. New Algorithms For Extracting Visibility Regions

A revised algorithm, with simpler concepts, was developed to extract visibility information in a binary form by testing the visibility of all edges in the network against a given viewpoint. The visibility information of all edges in the network is then used to decide the visibility of a triangle according to whether three edges of that triangle are all fully visible with respect to a viewpoint. It first sorts all the edges in the triangulation into ascending order, according

to the edge's maximal distance to the given viewpoint. There are two endpoints for each edge and therefore two distances are possible when connecting the viewpoint to each of the endpoints. The maximal distance of an edge to the viewpoint refers to the longer one of the two possible distances between the two endpoints and the viewpoint and the minimal distance to the shorter. For any edge, E_i, to be tested for its visibility, the testing procedures will be applied only to those edges whose minimal distances to the viewpoint are shorter than the maximal distance of E_i. In other words, if a circle is drawn centred at the viewpoint with a radius of the maximal distance from the viewpoint to that edge, edges in the network will be used to test visibility only if they are at least partially located inside this circle.
visibility with retially located inside this circle.

The number of edges which are needed to test the visibility of E_i is further reduced in the algorithm when it proceeds to test the visibility of edge E_i against only edges which 2-block E_i. The function of 2-blocking status is as defined in section 5.1 and is described in Figure 5.9. Assuming that we have a viewpoint P, and two edges, E_i and E_j; we can connect each of the endpoints with point P to form four line segments (Figure 5.9). Thus, edge E_i is defined to be 2-blocked by edge E_j if one of the following conditions exists:

1) edge E_j intersects with either one of the two line segments, Pa, Pc, or both of them (Figure 5.9b);

2) edge E_i intersects with either one of the two lines, Pd, Pf, or both of them (Figure 5.9b); or

3) E_j lies totally inside the triangle formed by edge E_i and the viewpoint P (Figure 5.9a) (E_j totally 2-blocks E_i).

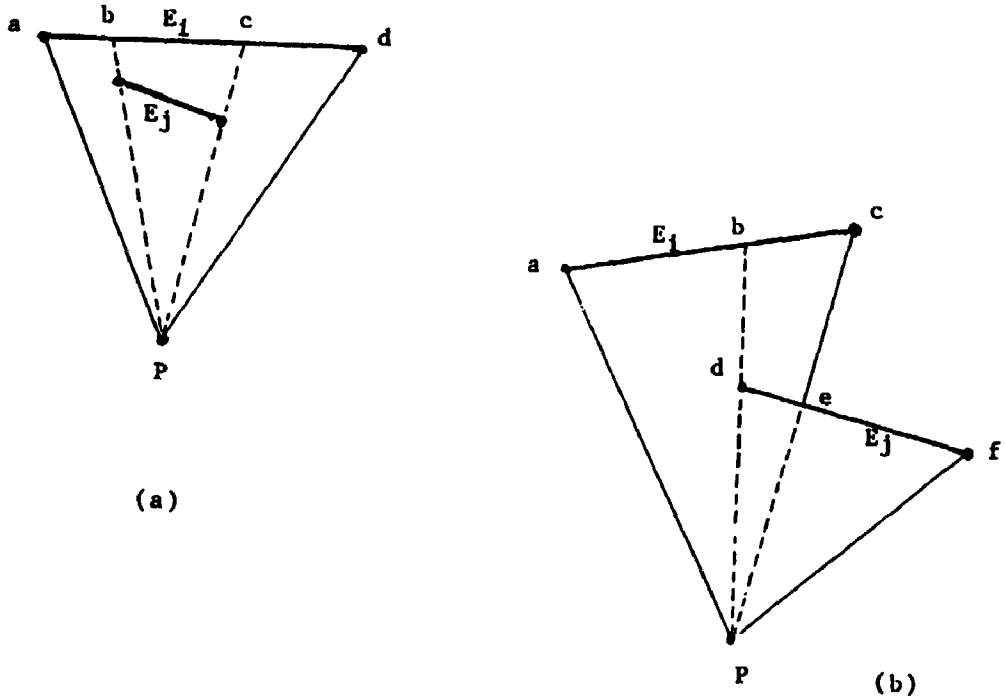


Figure 5.9. 2-BLOCKING Status Between Edges. In both cases, edge E_i is blocked by edge E_j. Only case (a) requires special treatment if checked for 2-blocking status by intersecting endpoints.

If E_j totally 2-blocks E_i with respect to P, E_i is visible if and only if both of the two endpoints of E_j are below the plane bounded by E_i and rays drawn from P through the endpoints of E_i. In the case of E_i partially 2-blocked by E_j (case 1 or 2 above), E_i will have to be decomposed into parts. The portion of E_i which is 2-blocked by E_j is then tested for visibility with respect to P against blocking edge E_j as if it was another edge which is totally 2-blocked by E_j (The portion not blocked by E_j will be tested for visibility against other 2-blocking edges, if any). The partition of E_i is decided by the number of intersections generated by line segments which are constructed by connecting the viewpoint and all four endpoints of E_i and E_j. In the case of Figure 5.9b, E_i will be partitioned into two parts only, namely ab and bc. The algorithm is expressed in pseudocode shown in Algorithm 5.2.

This algorithm determines visibility information of all edges in the network with respect to a viewpoint in $O(np^3)$ time since each edge must be compared to closer edges, and the number of edges is proportional to np. The visibility information of triangles in the network with respect to the viewpoint may be obtained from the visibility information of all edges. The procedures are then repeated for every candidate viewpoint. Figure 5.10 show an example of the result by using the revised algorithm to compute visibility regions on a sample data set.

```
for each viewpoint, P:
for all edges in the network, Ei, i=1,ne:
     compute the max-distance, DMi and DNi, from
     viewpoint P to Ei.
next Ei.
sort all edges in the network into ascending order according
to DNi;
initially mark all edges in the network fully visible;
for each edge, Ei, i=1,ne:
     for each edge, Ej, j=1,ne, i<>j;
          go to next Ej if DNj > DMi;
          test the 2-blocking status of Ej on Ei;
          case 1:
            Ej does not 2-block Ei;
               go to next Ej;
          case 2:
            Ej totally 2-blocks Ei;
               if two endpoints of Ei all
               visible?
               mark Ei invisible if at least
               one endpoint invisible.
          case 3:
            Ej partially 2-blocks Ei;
               if the portion 2-blocked by Ej
               all visible?
               mark Ei invisible if blocked
               portion is not visible.
     next Ej.
next Ei.
for every triangle, Ti, i=1,nt:
     if all three edges are still marked visible, the
     triangle is visible -  output the triangle as
     visible
next Ti.
record visibility information;
next P.
```

Algorithm 5.2. A New Algorithm For Extracting Visibility Regions

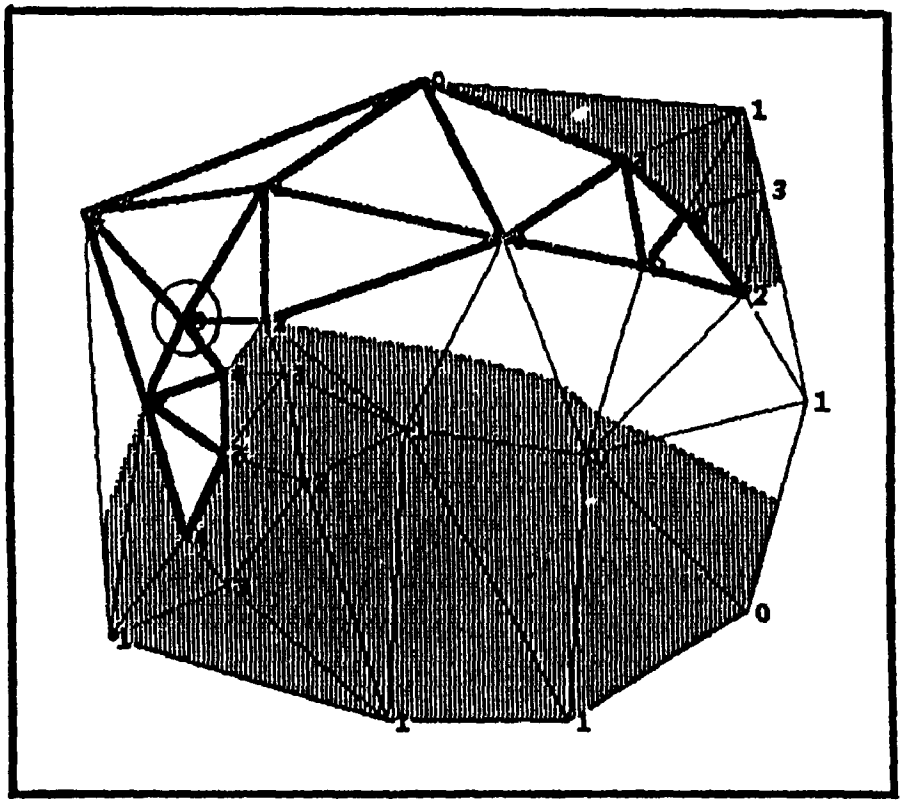


Figure 5.10. Visible Triangles of A Viewpoint. The visible region may be approximated either by the visible point set or by the set of triangle patches considered visible according to some predefined criteria.

Compared to the algorithm proposed by de Floriani, the revised algorithm requires has lower computational complexity, reducing from $O(np^3 M)$ to $O(np^3)$ (note that M will increase with np). The revised algorithm only needs one function, 2-blocking status, to perform visibility testing. This algorithm computes visibility regions by triangles: a triangle is visible iff the whole of a triangle is visible.

This algorithm was used to compute visibility information for all 30 random data sets and the 4 TIN data sets generated from U.S.G.S. grid DEM files. For each data set, the result of visibility information is kept in a visibility matrix, in which we have viewpoints as rows and triangles as columns and each element equals 1 if the corresponding triangle is visible from the corresponding viewpoint, 0 otherwise.

The viewshed of a point on an irregular topographic surface is defined as the area visible from the point. The area visible from a set of points is the union of their viewsheds. Thus our Problem 1, defined in section 3.2.1. is solved directly by applying the new algorithm for extracting visibility regions for a set of viewpoint(s). Similarly, the solution of finding invisible regions for a set of viewpoints, which is defined as Problem 2 in section 3.2.1., would be the intersection of the invisible regions from a set of points.

5.3. The Minimal Visible Height Algorithm

The visibility algorithms described so far are mainly designed for the extraction of visibility regions for a given set of viewpoints of ground heights. That is, the heights of all viewpoints considered so far are fixed to the ground elevations of the viewpoints. Visibility problems become more complicated but more realistic when the heights of viewpoints are allowed to be variable. For instance, a watchtower of some height may be built at a location to allow a viewer to see a larger part of a surface from the top of the tower. The visibility problems of variable heights are more complicated as it is not only necessary to find an optimal solution of selecting the locations of viewpoints but also their heights.

Recall that visibility was defined as $V_{ij} = 1$ when a straight line segment connecting vertex i and vertex j does not intersect the surface except at i and j, 0 otherwise. The minimal visible height, mvh(i,j) = h, is defined as the minimal height to which the viewpoint i needs to be raised such that $V_{ij} = 1$. The minimal visible height is set to 0 if j is visible from i from the ground height of i.

Consider the case between a viewpoint, P, and a triangle, T. In order for P to see T, the first criterion is that P has to be above the extended plane of T. If T has P as one of its vertices, T is immediately visible from P and thus mvh(P,T)=0. More steps are required to find the minimal visible height

when there are edges in the network which 2-block T with respect to P. First, consider edge E_{T1} as one of the three edges of T (E_{T1}, E_{T2}, and E_{T3}). For every other edge in the network that 2-blocks E_{T1}, the 2-blocking edge has to lie below the extended plane constructed by E_{T1} and P for E_{T1} to be visible from P. If, however, E_{T1} is not totally visible, P has to be raised to a height of the intersection between 1) a vertical line passing through P and 2) a plane constructed by the higher end of the 2-blocking edge and E_{T1}. This procedure has to be repeated for all three edges of T for every 2-blocking edge in the network. By testing one edge of T with one 2-blocking edge, an intersection along the vertical line passing through P is computed. After testing three edges of T for all 2-blocking edges, the highest among all intersections found along the vertical line passing through P is then defined as mvh(P,T).

After defining the minimal visible height between any pair of a viewpoint and a subregion, it is then possible to solve visibility problems of variable heights. First of all, the algorithm for extracting visibility regions for a given viewpoint needs to be modified to an algorithm for computing the minimal visible heights for all pairs of viewpoints and subregions on a topographic surface. Then the solutions may be found by searching combinations of viewpoints and heights. The modified algorithm is based on the revised visibility algorithm for extracting visibility regions. The basic

concepts and iterations are similar those used in the algorithm for extracting visibility regions. However, the output is a minimal visible height for each pair of viewpoints and subregions in the network.

The algorithm computes the minimal visible heights for all pairs of viewpoints and triangles. The minimal visible height is set to be -1.0 if it is infinity. The detailed procedures of the algorithm may be expressed as the pseudocode shown in Algorithm 5.3.

```
for every viewpoint P
     *set vheight=z(p)
     *construct a vertical line, L, passing P
     *for every triangle T
          *find all edges that 2-blocks at
               least one edge of T
          *for every edge of T, E_Ti, i=1,3
               *test the visibility of E_Ti
               against all 2-blocking edges
               *if E_Ti is totally visible ->
                    go to next edge if
                    visible
               *for every 2-blocking edge, E_j
                    *construct a plane,
                    PEE, passing the
                    higher endpoint of
                    E_j, and E_Ti.
                    *compute the
                    intersection, INT,
                    between L and PEE
                    *if (INT > vheight)
                         -
                         >vheight=I
                         NT
               *endfor
          *endfor
     *output mvh(P,T)=vheight
     *endfor
*endfor
```

Algorithm 5.3. Algorithm For Computing The Minimal Visible Heights

The output contains a minimal visible height for each pair of viewpoint and subregion in the network. With np vertices, the number of triangles is O(np). During every iteration, each edge of the tested triangle is tested against all 2-blocking edges to compute the minimal visible height, thus the computing time is $O(np^3)$ for all pairs of viewpoints and triangles.

5.4. Conclusions

In summary, this chapter presented two new algorithms for extracting visibility regions and computing the minimal visible heights between viewpoints and triangles. The algorithms were coded in FORTRAN and tested on a CYBER 835 for all 34 data sets.

In the next chapter, these algorithms are used to compute visibility information. The solutions to visibility coverage problems will be based on the visibility information obtained by the revised algorithms described in this chapter. The visibility information will be organized into visibility matrices as inputs to a variety of heuristics solving visibility coverage problems. Finally patterns of visibility information are analyzed with respect to different topographic surfaces.

CHAPTER VI VISIBILITY MATRICES AND HEURISTIC SOLUTIONS

Visibility algorithms described in Chapter V were tested on 30 random data sets and 4 TIN data sets converted from U.S.G.S. grid DEMs as described in Chapter IV to 1) verify the feasibility of the visibility algorithms, 2) solve the visibility coverage problems defined in Chapter III, and 3) examine the relationship between patterns of visibility information and types of topographic surfaces. The visibility algorithms, both for visibility regions and the minimal visible heights, were used to compute visibility information between viewpoints and subregions of the 34 sample data sets. Visibility information obtained from visibility algorithms may be expressed as rectangular matrices. The visibility matrices can then be used to solve visibility coverage problems.

This chapter first describes the structure and content of the visibility matrices. Secondly, three heuristics are used to solve visibility coverage problems: a Greedy Add, a Greedy Add with Swap, and a Stingy Drop. These heuristic methods are then compared for their performance on the tested data sets. In addition, solutions are presented according to the order of visibility coverage problems defined in sections 3.2.1. and 3.2.2. Finally, topographic characteristics of obtained visibility information are analyzed to explore the

relationship between visibility information and different types of landscape.

6.1. Visibility Matrices

Two types of visibility matrices, both of size np by nt, may be defined by information of visibility regions and the minimal visible heights between all viewpoints and triangles. These visibility matrices were used to solve visibility coverage problems of fixed and variable heights in later sections.

Visibility has been defined in Chapter IV as a binary function. That is, the visibility, V_{ij}, between a viewpoint, i, and a subregion, j, on a topographic surface is 1 if j is visible from i; 0 otherwise. The visibility matrix in this case will contain elements of 1's and 0's. For example, $V_{34}=1$ indicates that subregion 4 is visible from viewpoint 3.

In the case of a visibility matrix of minimal visible heights, the matrix contains elements which are the minimal visible heights required by the corresponding viewpoints (rows) in order to see the corresponding triangles (columns). The elements are heights and -1.0's. When the minimal visible heights are in fact infinity, i.e. the slope of the plane of a subregion is vertical and facing away from the viewpoint, an element of -1.0 would be entered to represent infinity.

Figure 6.1(a) and 6.1(b) show examples of visibility matrices described in this section.

	T1	T2	T3	T4	T5	T6	...	Tn
V1	1	1	1	0	0	1	...	0
V2	0	1	1	1	0	0	...	0
V3	1	0	1	0	0	0	...	0
V4	0	0	1	0	1	0	...	0
V5	0	0	0	1	1	1	...	0
...								
Vn	0	0	0	0	0	0	...	1

(a)

	T1	T2	T3	T4	T5	T6	...	Tn
V1	0.0	0.0	2.1	5.0	-1.0	11.0	...	41.0
V2	1.5	0.0	0.0	0.0	2.3	-1.0	...	-1.0
...								
Vn	-1.0	35.2	44.8	-1.0	-1.0	-1.0	...	0.0

(b)

Figure 6.1. Visibility Matrices. (a) An example of a visibility matrix recording visibility information between viewpoints and triangles. (b) An example of a visibility matrix recording the minimal visible heights.

6.2. Heuristic Solutions

Theoretically, it is always possible to obtain solutions to our visibility coverage problems by thoroughly searching all possible combinations (or subsets) of viewpoints on a landscape. However, a thorough search is unlikely preferred simply because the number of vertices representing a landscape often is enormous such that the effort becomes extremely costly. Therefore, a heuristic approach must be used to solve visibility coverage problems.

The desirable characteristics of a heuristic algorithm include:

a) execution in reasonable computational time,
b) solutions which are close to optimality,
c) only a small probability of any one solution being far below optimality, and
d) simplicity of both design and computational requirements.

The heuristic approach offers a fast, simple, and flexible way of handling complex and enormous data problems, but it is not without disadvantages. The main drawback is the possibility of reaching solutions below optimality by using the heuristic approach. However, a proper validation, such as testing with a small size data set of known solution, can increase, although not guarantee, the possibility of getting an optimal solution.

6.2.1. Solutions of visibility coverage problems of fixed heights

In total, three classes of heuristics, a greedy add, a greedy add with swap, and a stingy drop, were used to solve Problem 3 and Problem 4. Although Problem 3 and Problem 4 are different in their objectives, all can be solved by these three heuristics using different stopping rules. Both problems were defined in section 3.2.1. The objective of Problem 3 is to minimize the number of viewpoints required to view the entire surface while the objective of Problem 4 is to maximize the visibility coverage from a given set of viewpoints.

The nature of the visibility coverage problem raises the possibility of driving the heuristic with distinctive parameters, other than the objective function, as these may be easier to compute at each step. One might argue that vertex elevation plays a large part in intuitive solutions to these problems, and should be successful as a selection criterion. Considering a land surface where only one peak exists, the selection of the point with the highest elevation as a viewpoint might be the best choice to maximize the area of regions visible from one viewpoint. However, the selection process would not be as straightforward when the reliefs become more complicated on the topographic surfaces. Very often the selection of the highest n ($n > 1$) vertices does not

maximize the area of visible regions from these vertices simply because the area visible from these vertices is also related to topographic reliefs formed by other vertices on the surface. Since the association between elevations of vertices as viewpoints and the area of visible regions is not a simple relationship, vertex elevation will not be adapted to drive heuristics in this study.

The column total of the visibility matrix, or the number of vertices from which a given triangle can be seen, is an index of its general visibility, and might be a useful weight: it would be logical to select first those viewpoints which cover as many as possible of the less visible triangles. But for problems where the objective is to maximize the area of visible regions from each viewpoint, the number of vertices from which a given triangle can be seen is not simply related to the area of visible regions from these vertices, as the area of triangles is also important. That is, a viewpoint may see many triangles which are not visible from many other viewpoints but still have a small area coverage because those triangles visible from the viewpoint are all very small. Therefore it is also less preferred for use in driving the heuristics in this study.

The first heuristic considered is the **Greedy Add (GA)** algorithm. In order to optimize the objective function, the algorithm starts with an empty solution set and then adds to this set one at a time the best facility sites (location of

viewpoints). To solve Problem 3, the GA algorithm picks for the first viewpoint, the one that has the largest area of visible triangles. For the second viewpoint, GA picks the viewpoint that has the largest area of visible triangles not visible from the first viewpoint. Then, for the third viewpoint, GA picks the viewpoint that has the largest area of visible triangles not visible from the first and the second. This process is continued until the solution set just covers all triangles in the network. Details of the algorithm are given in Church (1974).

Problem 4 can be solved by similar procedures of selecting the best viewpoint locations which maximize the objective at each iteration. However, the process stops when the number of selected viewpoints, i.e. those which included in the solution set, reaches the pre-specified number. It should be noted that the one-viewpoint maximal coverage solution by the GA algorithm is optimal, for the one-viewpoint solution is by definition the viewpoint which covers the most triangles. However, for solutions where more than one viewpoint is derived, optimality is not guaranteed.

It is also important to note that the GA algorithm never removes viewpoints from the solution set. Therefore, it is possible that a viewpoint added to the solution set in the early iterations of the algorithm may not be justified later in the algorithm due to subsequent viewpoint selections. The presence of a "no longer justified" viewpoint in the solution

set would imply non-optimality. Theoretically the GA algorithm could be improved by including a technique that would reduce the probability of maintaining "no longer justified" viewpoints in the solution set.

The second heuristic which builds on the first is designated as the **Greedy Add with Swap (GAS)** algorithm. The GAS algorithm determines new viewpoints at each iteration just as the Greedy Add algorithm does, but in addition, seeks to improve the solution (according to the objective) at each iteration by trying to replace each viewpoint one at a time with a viewpoint outside the solution set. If improvement is possible, the viewpoint chosen to replace a particular viewpoint is the one which gives the greatest improvement in the objective. Like the GA algorithm, GAS does not guarantee any global optimality, either. The GAS algorithm is also outlined in Church (1974). Again, this algorithm can be used to solve both Problem 3 and 4 but stops when the entire surface is covered for Problem 3 and stops when the desired number of viewpoints is reached for Problem

The third heuristic used in the study is the **Stingy Drop (SD)** algorithm. In contrast to GA or GAS, the SD algorithm selects all viewpoints initially. At each step SD drops the vertex with the least deterioration on the objective function. The most suitable viewpoint to be dropped is the one which produces the least areas which are visible before dropping the point but are invisible after dropping the point. For Problem

3, the process is continued until the solution set just covers the whole network or the number of points remaining in the solution reaches the pre-specified number -for Problem 4. SD offers no better guarantee than GA or GAS for global optimality.

These three algorithms are expressed in pseudocodes in Algorithms 6.1., 6.2., and 6.3. These three heuristic, GA, GAS, SD, were coded in FORTRAN and tested on all 30 random data sets and 4 TINs converted from U.S.G.S. DEMs. The results are shown in Table 6.1.

As shown in Table 6.1., GAS performed the best as it reached the best solutions (comparing to those of GA and SD) the most often: 29 out of 34 cases (85.29 %). GA and SD reached the best solution an equal number of times: 22 out of 34 (64.71 %). Three methods tied in 9 out of 34 cases (26.47 %) and GAS tied with SD in 4 of 34 cases (11.76 %). Note that there are 4 cases in which SD alone reached the best solution among the three methods while in other 2 cases the best solutions were reached by GAS. Also note that for the 4 big TINs, SD only reached the best solution once, while the other three TINs were all tied between GA and GAS.

In general, GAS performed better than GA and SD as its chance of reaching a better solution is the highest among the three heuristics, while the performances of GA and SD tied in our 34 cases. However, the computing time required by the three heuristics to reach a solution set varies dramatically.

```
select first viewpoint if it has the largest area of visible
triangles;

check if the visible area of the first point is the entire
surface, output a single point solution if it can see the
entire surface;

while (the stopping criterion is not met) do
     select a viewpoint to be included into solution set
     if it has the largest area of visible triangles
     which are not visible from all of the viewpoints
     previously selected;

     compute the coverage of all viewpoints in the
     solution set;
endwhile
output solution set;

stopping criterion:
     for Problem 3 -
          the entire surface is totally covered by
          the viewpoints selected in solution set.
     for Problem 4 -
          the desired number of viewpoints is
          reached.
```

Algorithm 6.1. GA Algorithm.

```
select first viewpoint if it has the largest area of visible
triangles;

check if the visible area of the first point is the entire
surface, output a single point solution if it can see the
entire surface

while (the stopping criterion is not met) do
     select a viewpoint to be included into solution set
     if it has the largest area of visible triangles
     which are not visible from all of the viewpoints in
     solution set

     for every viewpoint included previously in the
     solution set, P
          for every viewpoint not included in the
          solution set, P'
               compute the coverage if
               swapping P and P'
                    swap P and P' if
                    coverage improved.
          endfor
     endfor

endwhile

output solution set

stopping criterion:
     for Problem 3 -
          the entire surface is totally covered by
          the viewpoints selected in solution set.
     for Problem 4 -
          the desired number of viewpoints is
          reached.
```

Algorithm 6.2. GAS Algorithm.

```
initially select all viewpoints into solution set

while (the stopping criterion is not met) do
     for every viewpoint, P, in the solution set
          compute visible area, VAP, from
          viewpoints in the solution set, including
          P;

          compute visible area, VA, from viewpoints
          in the solution set, without P;

          compute VAREA=VAP-VA
          mpt=P if P is associated the least VAREA
          among all viewpoints in the solution set;
     endfor

     drop mpt from solution set

endwhile

output the remaining solution set;

stopping criterion:
     for Problem 3 -
          the entire surface is totally covered by
          the viewpoints selected in solution set.
     for Problem 4 -
          the desired number of viewpoints is
          reached.
```

Algorithm 6.3. SD Algorithm.

	# OF VIEWPOINT			BEST	BEST-PERFORMED
	GA	GAS	SD	SOLUTION	METHOD
TIN101	10	9	8	8	SD
TIN102	8	8	8	8	GA/GAS/SD
TIN103	9	9	9	9	GA/GAS/SD
TIN104	7	7	8	7	GA/GAS
TIN105	8	8	8	8	GA/GAS/SD
TIN106	10	9	10	9	GAS
TIN107	9	9	9	9	GA/GAS/SD
TIN108	8	8	7	7	SD
TIN109	8	7	7	7	GAS/SD
TIN110	8	8	10	8	GA/GAS
TIN201	11	10	8	8	SD
TIN202	8	8	8	8	GA/GAS/SD
TIN203	10	9	9	9	GAS/SD
TIN204	7	7	7	7	GA/GAS/SD
TIN205	10	8	8	8	GAS/SD
TIN206	9	9	10	9	GA/GAS
TIN207	9	9	9	9	GA/GAS/SD
TIN208	9	9	9	9	GA/GAS/SD
TIN209	8	7	7	7	GAS/SD
TIN210	9	9	10	9	GA/GAS
TIN301	10	8	11	8	GAS
TIN302	8	8	9	8	GA/GAS
TIN303	10	11	9	9	SD
TIN304	7	7	7	7	GA/GAS/SD
TIN305	8	8	9	8	GA/GAS
TIN306	10	9	10	9	GAS
TIN307	9	9	9	9	GA/GAS/SD
TIN308	9	9	9	9	GA/GAS/SD
TIN309	8	8	8	8	GA/GAS/SD
TIN310	9	9	9	9	GA/GAS/SD
TIN1	25	25	27	25	GA/GAS
TIN2	32	33	31	31	SD
TIN3	33	33	34	33	GA/GAS
TIN4	25	25	27	25	GA/GAS

Table 6.1. Solutions to Problem 3

Among the three, GA is the quickest approach, followed by SD, while GAS took the longest time. The average computing time for reaching a solution for all 50-point data sets are:

GA	1.06 CPU seconds
GAS	2.85 CPU seconds
SD	1.64 CPU seconds.

For the four TINs converted from U.S.G.S. DEMs, the differences in computing time are enormous due to the larger numbers of vertices in the TIN models. For example, the computing time used to search for solutions for Problem 3 for TIN1 are:

GA	2.27 CPU seconds
GAS	9451.94 CPU seconds
SD	84.07 CPU seconds

The reason for these differences is that for a TIN with np vertices and a solution set of k viewpoints, GA has to search

$$np + (np-1) + (np-2) + \ldots + (np-(k-1))$$

times among all candidate vertices to reach a solution, while SD has to search

$$np + (np-1) + (np-2) + \ldots + (k+1)$$

times. As for GAS, the time is increased enormously because at each iteration GAS has to search among all candidate vertices trying to swap with viewpoints which were already selected; therefore it has to search

$$np + 2\ [\ (np-1) + \ldots + (np-(k-2))\ ] + (np-(k-1))$$

times among all candidate vertices to reach a solution.

Considering the purpose of using heuristics to solve our problems, a method which finds reasonable solutions more

quickly is certainly more preferred. In our case GA is preferred for this reason. However, GAS should also be used to look for any possible improvement if the size of data sets is small or there is no constraint on computing resources.

A further investigation on the viewpoints selected as solutions was also conducted. All viewpoints were classified according to the definitions described in section 4.1.1. to see whether they are peak, pit, pass, ridge, channel, or another type of points. The results are shown in Table 6.2. Numbers in parentheses are the percentages. Percentage averages were computed for each column. It may be observed that ridge points had higher percentages of being selected as solution viewpoints than peak points. The proportions between peak and ridge points of GA and GAS are more similar to each other than to those of SD. Among the three heuristics, GA selected more ridge points (67.06%) than GAS (66.5%) and SD (58.12%), while SD selected the highest percentage of peak points (41.88%). Although the results shown here are based solely on the 34 cases included in this study and may lack of generality, one comment which may be made from these results is that solution viewpoints are all either peak or ridge points.

GA, GAS, and SD may also be used to solve Problem 4, which has an objective of maximizing visibility coverage from a given set of viewpoints. In order to show the effect of dropping viewpoints even when the surfaces are not entirely

	GA		GAS		SD	
	PEAK	RIDGE	PEAK	RIDGE	PEAK	RIDGE
TIN101	3 (30)	7 (70)	2 (22)	7 (78)	3 (38)	5 (62)
TIN102	5 (62)	3 (38)	5 (62)	3 (38)	4 (50)	4 (50)
TIN103	3 (33)	6 (67)	2 (22)	7 (78)	2 (22)	7 (78)
TIN104	3 (43)	4 (57)	3 (43)	4 (57)	4 (50)	4 (50)
TIN105	3 (38)	5 (62)	2 (25)	6 (75)	5 (62)	3 (38)
TIN106	1 (10)	9 (90)	2 (22)	7 (78)	2 (20)	8 (80)
TIN107	1 (11)	8 (89)	1 (11)	8 (89)	3 (33)	6 (67)
TIN108	2 (25)	6 (75)	2 (25)	6 (75)	2 (29)	5 (71)
TIN109	2 (25)	6 (75)	2 (29)	5 (71)	3 (43)	4 (57)
TIN110	6 (75)	2 (25)	6 (75)	2 (25)	6 (60)	4 (40)
TIN201	4 (36)	7 (64)	3 (30)	7 (70)	3 (38)	5 (62)
TIN202	4 (50)	4 (50)	4 (50)	4 (50)	5 (62)	3 (38)
TIN203	3 (30)	7 (70)	2 (22)	7 (78)	3 (33)	6 (67)
TIN204	3 (43)	4 (57)	4 (67)	2 (33)	4 (57)	3 (43)
TIN205	5 (50)	5 (50)	3 (38)	5 (62)	6 (75)	2 (25)
TIN206	3 (33)	6 (67)	3 (43)	4 (57)	4 (40)	6 (60)
TIN207	1 (11)	8 (89)	1 (11)	8 (89)	2 (22)	7 (78)
TIN208	2 (22)	7 (78)	2 (22)	7 (78)	3 (33)	6 (67)
TIN209	2 (25)	6 (75)	2 (29)	5 (71)	3 (43)	4 (57)
TIN210	4 (44)	5 (56)	5 (56)	4 (44)	6 (60)	4 (40)
TIN301	4 (40)	6 (60)	3 (38)	5 (62)	4 (36)	7 (64)
TIN302	3 (38)	5 (62)	4 (50)	4 (50)	5 (56)	4 (44)
TIN303	4 (40)	6 (60)	4 (36)	7 (64)	3 (33)	6 (67)
TIN304	4 (57)	3 (43)	4 (57)	3 (43)	4 (57)	3 (43)
TIN305	3 (38)	5 (62)	3 (38)	5 (62)	7 (78)	2 (22)
TIN306	3 (30)	7 (70)	3 (33)	6 (67)	4 (40)	6 (60)
TIN307	1 (11)	8 (89)	2 (22)	7 (78)	3 (33)	6 (67)
TIN308	1 (11)	8 (89)	2 (22)	7 (78)	3 (33)	6 (67)
TIN309	1 (13)	7 (87)	1 (13)	7 (87)	1 (13)	7 (87)
TIN310	4 (44)	5 (56)	4 (44)	5 (56)	5 (56)	4 (44)
TIN1	7 (28)	18 (72)	6 (24)	19 (76)	7 (26)	20 (74)
TIN2	7 (21)	25 (79)	6 (22)	27 (78)	10 (32)	21 (68)
TIN3	14 (42)	19 (58)	11 (33)	22 (67)	12 (35)	22 (65)
TIN4	3 (12)	22 (88)	1 (4)	24 (96)	7 (26)	20 (74)
MEAN	(32.94)	(67.06)	(33.50)	(66.50)	(41.88)	(58.12)

Table 6.2. Topographic Features of Solution Viewpoints

covered, the number of viewpoints was set to be 6, which is smaller than the smallest size of solution sets shown in Table 6.1.

GA and GAS select the locations of viewpoint sites during each iteration which improve objective functions the most, but SD requires further criteria. Recall that SD initially included all candidate viewpoints into the solution set and dropped the viewpoint which produced the least reduction in the objective function. However, when more than one point may be dropped without reducing the objective function, another criterion is needed. It was decided that in each cases, we will drop whichever point has the least visibility coverage. The results of solutions by GA, GAS, and SD for Problem 4 are shown in Table 6.3. In general, GA and GAS outperformed SD in maximizing visibility coverage for 6 viewpoints in our 34 cases. GAS showed improvements but there are only three cases, TIN101, TIN102, and TIN205 where improvement occurred earlier than the 6th iteration. Note that these three improvements found by GAS are highlighted by bold and underline typesetting. Again, GA is the quickest approach but it is possible to improve the solution by using GAS. However, the computing time required to run GAS is very much more than GA, especially with a larger data set.

As discussed in the first chapter, the number of viewpoints required to see the entire surface is believed to be related to the topography of the surface. Flatter surfaces

	GA	GAS	SD
TIN101	35835.9	**35838.6**	31080.6
TIN102	44554.7	**44694.4**	35672.4
TIN103	43321.3	43321.3	31688.8
TIN104	49854.7	49854.7	45401.9
TIN105	54974.4	54974.4	53649.0
TIN106	46052.2	46052.2	45240.4
TIN107	46820.0	46820.0	43466.2
TIN108	51258.8	52158.8	50359.2
TIN109	42204.7	42204.7	40760.6
TIN110	45621.5	45621.5	44120.5
TIN201	35615.2	35615.2	34215.3
TIN202	44563.1	44563.1	41535.1
TIN203	42653.3	42653.3	38550.5
TIN204	49854.7	49854.7	47.53.2
TIN205	53054.5	**53582.6**	51919.5
TIN206	46286.5	46286.5	44649.5
TIN207	46238.0	46238.0	43466.2
TIN208	50980.1	50980.1	46411.2
TIN209	42472.4	42472.4	38827.9
TIN210	45045.2	45045.2	41457.9
TIN301	35780.1	35780.1	33744.7
TIN302	43840.3	43840.3	41115.3
TIN303	42510.3	42510.3	40673.4
TIN304	49854.7	49854.7	46788.9
TIN305	54010.9	54010.9	49422.2
TIN306	46061.0	46061.0	42759.1
TIN307	47087.7	47087.7	44015.1
TIN308	50980.1	50980.1	47077.2
TIN309	42194.9	42194.9	39947.8
TIN310	45459.7	45459.7	41542.9
TIN1	68740.0	68740.0	56761.5
TIN2	62795.0	62795.0	42125.0
TIN3	62156.5	62156.5	34477.5
TIN4	63586.0	63586.0	46593.0

Table 6.3. Solutions To Problem 4

with less relief would require a smaller number of viewpoints than surfaces of greater relief. An examination of the relationships between surface description parameters and the number of viewpoints of the best solutions produced by the three heuristics was conducted by using correlation analysis. The following ten surface description parameters and the distribution of point types in each TIN model were used in this analysis:

1) mean of elevations;
2) standard deviation of elevations;
3) skewness of elevations;
4) kurtosis of elevations;
5) mean of slopes of triangle planes;
6) standard deviation of slopes of triangle planes;
7) skewness of slopes of triangle planes;
8) coefficient of dissection of elevations;
9) spatial autocorrelation coefficient;
10) slope of the regression line of variograms;

11) peak % = (no. of peak points)/(total no. of pts.);
12) pit % = (no. of pit points)/(total no. of pts.);
13) pass % = (no. of pass points)/(total no. of pts.);
14) ridge % = (no. of ridge pts.)/(total no. of pts.);
15) channel % = (no. of channel pts.) / (total no. of pts.);
16) other % = (no. of other pts.)/(total no. of pts.).

Definitions of these ten parameters were discussed previously in section 4.1.1. All 34 data sets were used in this analysis. The correlation coefficient was used to measure linear relationships between each of the ten parameters and the number of solution viewpoints. Correlation coefficients were computed by using the following formula (between variables X and Y, each having N cases):

$$r = \frac{\Sigma_i \left(X_i - (\Sigma_i X/N)\right)\left(Y_i - (\Sigma_i Y/N)\right)}{(N-1)\, S_X\, S_Y} \qquad i=1,N$$

where S_X and S_Y are the standard deviations of the two variables.

Correlation coefficients between each of these parameters and the number of solution viewpoints are listed in Table 6.4. Only those correlation coefficients whose absolute values were higher than 0.45 were used to rank the associations between each parameter and the number of solution viewpoints. The final ranking is the average of ranking on three heuristics and is shown as follows:

RANKING OF VARIABLES

1. Mean of slopes of triangle planes in a TIN
2. Skewness of slopes of triangle planes in a TIN
3. Standard deviation of slopes of triangle planes in a TIN
4. Mean of elevations of vertices in a TIN
5. Percentage of channel points in a TIN
6. Standard deviation of elevations of vertices in a TIN
6. Kurtosis of elevations of vertices in a TIN
8. Percentage of ridge points in a TIN
9. Spatial Autocorrelation Coefficient
10. Slope of regression line of Variogram

The distribution of slopes of triangle planes in a TIN appears to be the most strongly associated with the number of solution viewpoints. Strong, positive correlations between means, and standard deviations of triangle slopes and the number of solution viewpoints imply that a surface with steeper slopes and with a wider range of triangle slopes represents a more complicated relief and would require more viewpoints to see the entire surface. Taking the four TINs converted from U.S.G.S. DEMs as examples, TIN4 which has a

	GA	GAS	SD	BEST	VH	VHS
Tri Pln Slp Mean (1)	.906 (1)	.910 (1)	.925 (1)	.911 (1)	.905 (1)	.890 (1)
Tri Pln Slp Skwns (2)	-.789 (2)	-.791 (2)	-.819 (2)	-.780 (2)	-.792 (2)	-.789 (2)
Tri Pln Slp SD (3)	.733 (3)	.734 (3)	.749 (3)	.740 (3)	.697 (3)	.690 (3)
Z Mean (4)	.591 (4)	.590 (4)	.627 (4)	.602 (4)	.593 (5)	.597 (5)
Chnl Pts% (5)	.582 (5)	.582 (5)	.586 (6)	.583 (5)	.637 (4)	.655 (4)
Z SD (6)	.548 (7)	.547 (7)	.580 (7)	.558 (7)	.581 (6)	.581 (6)
Z Kurt (7)	.571 (6)	.568 (6)	.596 (5)	.580 (6)	.490 (9)	.506 (8)
Rdge Pts% (8)	.486 (10)	.496 (8)	.525 (8)	.515 (8)	.498 (7)	.499 (10)
SAC (9)	.491 (9)	.470 (9)	.509 (9)	.504 (9)	.442 (10)	.544 (7)
Variogram Reg L Slp (10)	-.503 (8)	-.461 (10)	-.497 (10)	-.474 (10)	-.495 (8)	-.498 (9)
Peak Pts%	-.220	-.249	-.268	-.275	-.206	-.265
Pit Pts %	-.247	-.233	-.249	-.243	-.166	-.206
Pass Pts%	.189	.178	.141	.146	.240	.187
Z Skew	-.011	.011	.032	.037	.001	.043
Z CD	.033	.037	-.011	.021	.080	.064

Table 6.4. Correlation Analysis of Surface Description Parameters.

mean of triangle slopes of 5.65° and an S.D. of 10.28° was found to require 25 viewpoints to see its entire area, while on the other end, TIN3, whose mean of triangle slopes is 19.40° with an S.D. of 21.78° required 33 viewpoints. The strong, negative correlation between skewness of triangle slopes and the number of solution viewpoints also verified this relationship, indicating that a surface with more slopes below the mean of slopes required less viewpoints. There are positive correlations between the distribution of vertex elevations, the topographic feature types of vertices and the number of solution viewpoints for similar reasons. The positive correlation between standard deviation of elevations and the number of viewpoints indicates that surfaces with more dispersed elevations (higher S.D.) would require more viewpoints. Higher kurtosis and higher percentage of ridge and channel points were fouhd to be associated with higher numbers of viewpoints. Spatial autocorrelation coefficients also showed positive correlations with the number of solution viewpoints, with coefficients ranging from .44 to .54. Finally, the slopes of regression lines of variograms displayed a negative correlation.

Each of GA, GAS, and SD produced a set of viewpoints for each TIN model converted from DEMs. The solution sets and the heuristics which produced them were as follows:

TIN1	25 viewpoints	produced by GA/GAS
TIN2	31 viewpoints	produced by SD
TIN3	33 viewpoints	produced by GA/GAS
TIN4	25 viewpoints	produced by GA/GAS.

Solution viewpoints for the four TINs were plotted on their respective triangulated network by surrounding each viewpoint site with a square to indicate its location. Figure 6.2. to 6.5. show respectively locations of solution viewpoints for 4 TIN models.

Viewpoints selected for TIN1, Krone Ranch, Montana, are found mostly located along two main ridge lines (The Reef), running in a northwest-southeast direction, surrounding a basin (Nicholas Basin). Other viewpoints were located at peaks (including Comb Rock) in the northeastern part of the region where local reliefs are more gentle. The highest point, the peak of Limekiln Mountain, located in the southwest part of the region was not selected as one of the viewpoint sites as it is located right on the south border and its visibility regions were also covered by viewpoints located along nearby the ridge line of the Reef.

In the area of Herbert Domain, Tennessee, described by TIN2, viewpoints were found to be located mostly along Buck Ridge (running in a east-west direction), and along the border of Herbert Domain. Viewpoints located along the border of Cumberland Plateau and Buck Ridge provided visibility coverage for both sides of the ridge. The viewpoints located along the border of Herbert Domain

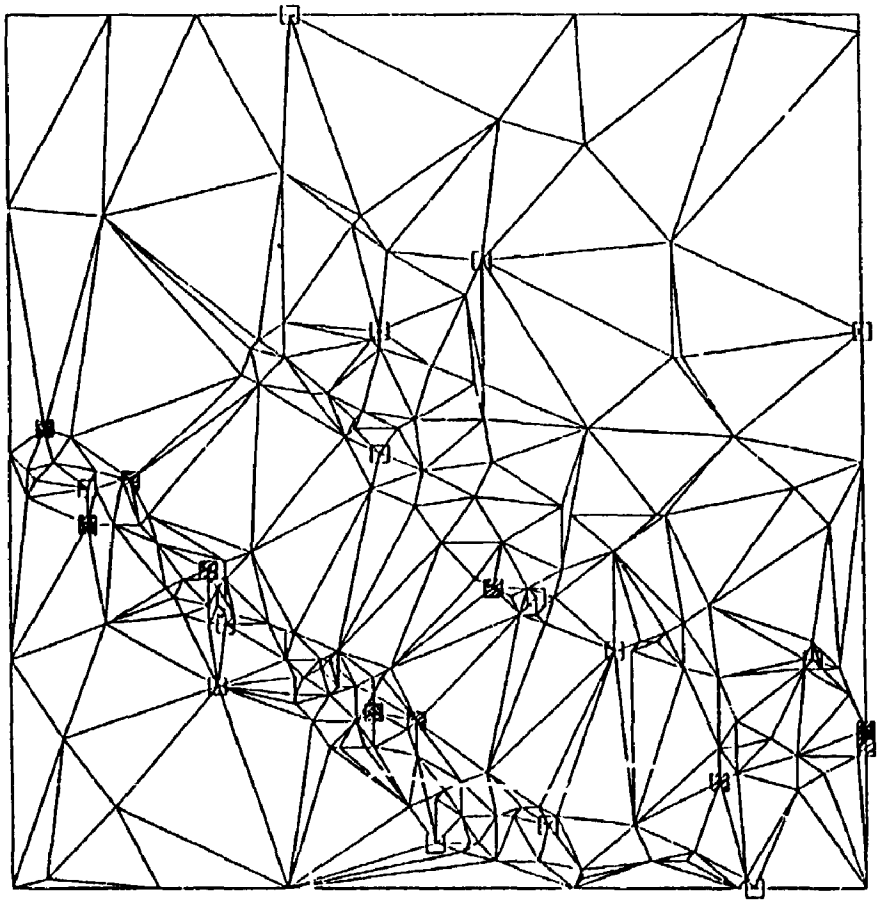


Figure 6.2. The Locations of Solution Viewpoints, TIN1.

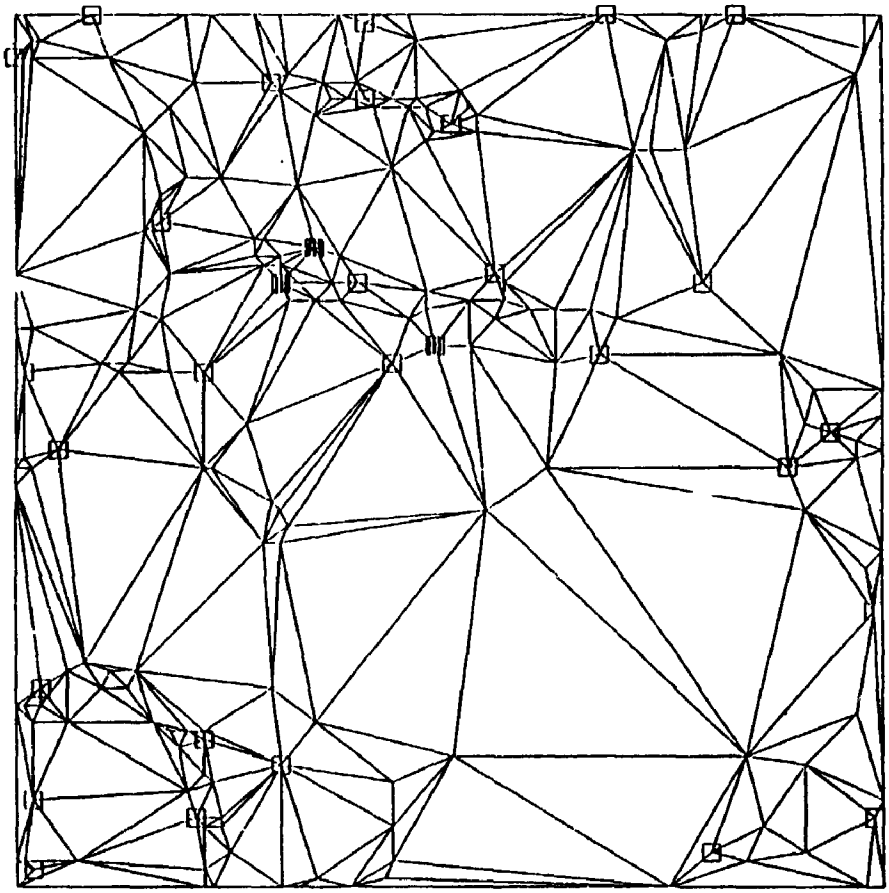


Figure 6.3. The Locations of Solution Viewpoints, TIN2.

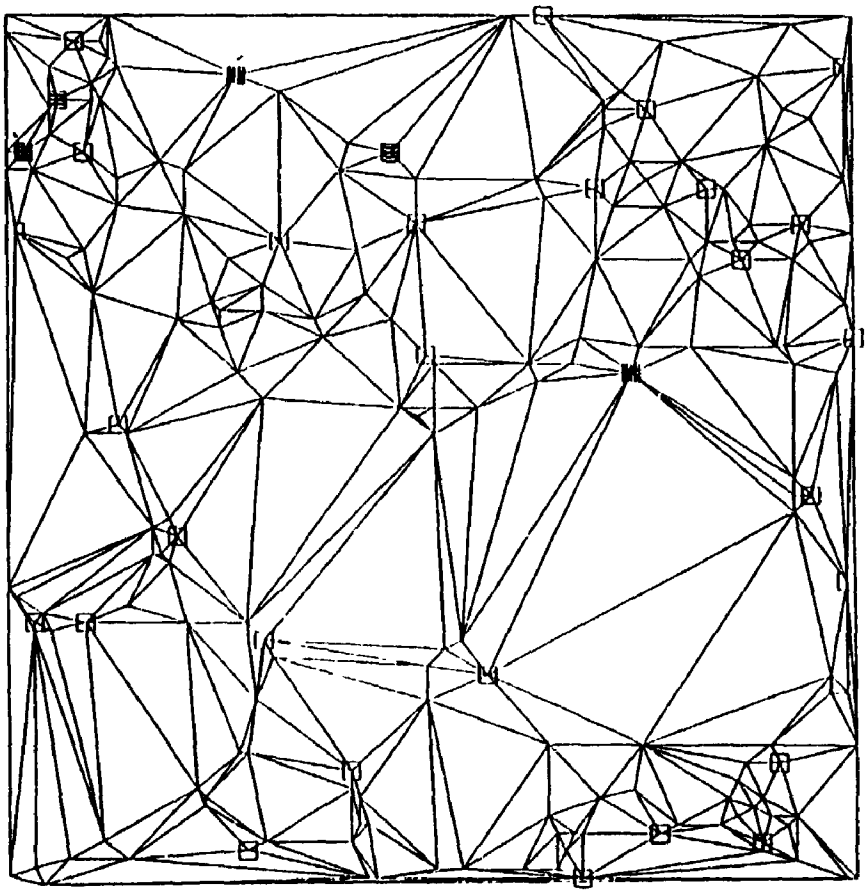


Figure 6.4. The Locations of Solution Viewpoints, TIN3.

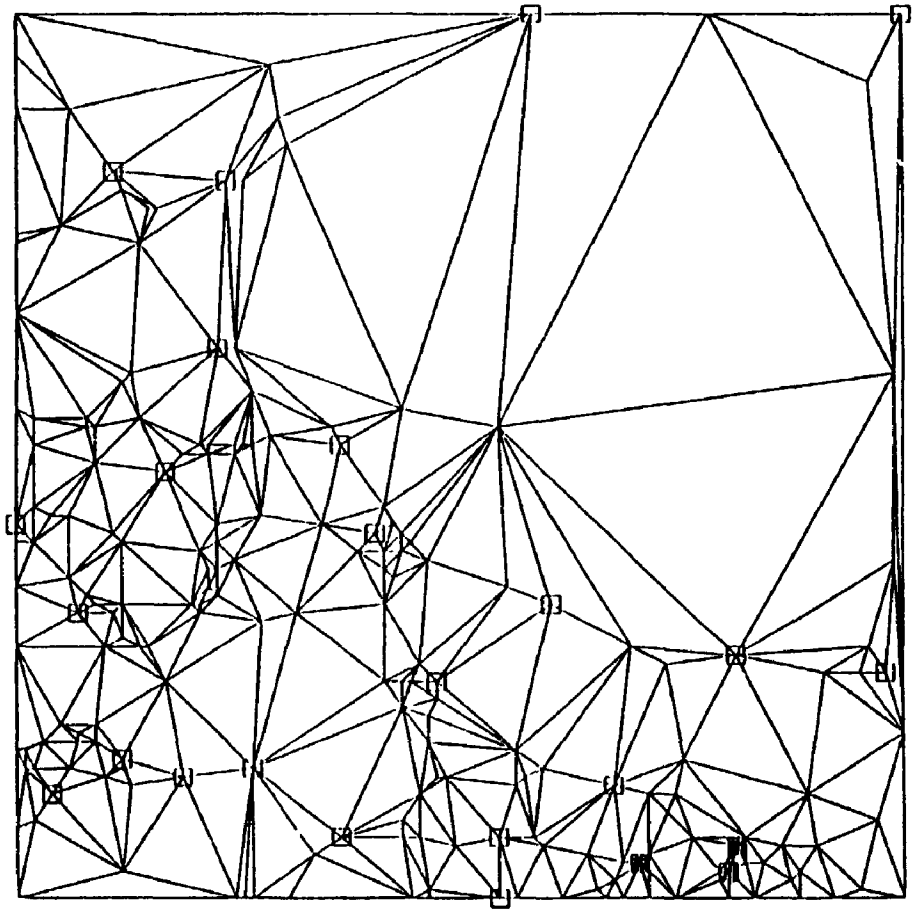


Figure 6.5. The Locations of Solution Viewpoints, TIN4.

overlooked the valley along Bee Creek in the southwest corner of the area covered by TIN2. The highest vertex is located in central northern part of the area and was selected as one of the viewpoints.

The area described by TIN3 is the southwestern part of Benndale, Mississippi. The topography in this area is mostly gentle hill slopes. Viewpoints selected were mostly local peaks (14 of a total of 33 solution viewpoints), the highest proportion among 4 TINs. Their locations were scattered all over the area, providing visibility coverage for the surrounding regions. The highest vertex, located in the northwestern area, was selected as one of the solution viewpoints.

TIN4 describes the Cedar Breaks area in Utah. Viewpoints were located along the border of a highland and nearby ridge lines. In the southeastern part of the region, viewpoints were located along Pink Cliffs overlooking Navajo Lake and adjacent regions. Other viewpoints were found along cliffs overlooking Cedar Breaks and adjacent areas in the northwestern corner of TIN4. Viewpoints located along the border of the highland provided visibility coverage over two valleys (Midway and Deer Hollow) on the highland. Again the highest vertex, located on the northern border of TIN4, was selected as one of the viewpoints.

In summary, the three heuristics used in this study are all capable of finding solutions for Problem 3 and Problem 4.

However, to compare the performances of these heuristics is not simple. It may be concluded that GA is the quickest approach to find a reasonable solution, while SD and GAS should also be used to see if there is any improvement possible when computing resources are not critical or the size of problems is smaller.

As to what types of vertices are most likely to be selected as solution viewpoints, peak and ridge points were found to be the only two types selected by these three heuristics. Finally, triangle slopes seem to be the most strongly associated with the number of viewpoints considering our 34 cases. Surfaces with more relief variation and steeper slopes were found to be associated with larger numbers of viewpoints. In addition, higher elevation S.D.'s, higher elevation kurtosis, higher percentages of ridge and channel points in a TIN, and higher spatial autocorrelation coefficients all showed positive correlations with the number of solution viewpoints required to see the entire surface.

6.2.2. Solutions of visibility coverage problems of variable heights

In section 6.2.1., visibility coverage problems of fixed heights (defined in 3.2.1.) were solved by using the heuristic methods of GA, GAS, and SD. The solutions of those problems are useful when there is no need to raise the

heights of the observation locations. However, very often we prefer to reduce the number of observation sites as both manpower and budget are usually the main sources of constraints. The number of viewpoints required to monitor a surface may be reduced if one could build watchtowers of specific heights on appropriated sites. This is because the visibility regions of a viewpoint may be increased by raising the viewpoint's height. Visibility coverage problems of variable heights will be of more interests as they offer a more realistic approach to visibility coverage problems.

The first visibility coverage problem of variable height, Problem 5 (defined in section 3.2.2.) is to minimize the total cost of a set of viewpoints with watchtowers of variable heights such that the entire surface is visible. The cost of constructing watchtowers, which is part of the formulation of Problem 5 and 6, is assumed to be a simple, increasing function of tower heights, expressed as:

$$C(H) = F + S\ H^{A}$$

where C(H) is the cost of a tower of height H; F is a fixed cost associated with each watchtower; S and A are constants. If there is no fixed cost F, the solution of Problem 5 will be to build towers of zero height as there always exists a solution of viewpoints of ground height.

The second problem of this category, defined in section 3.2.2. as Problem 6, is to maximize the area visible under a budget of construction cost of building towers. Being

slightly different from Problem 5 by adding a budget constraint, Problem 6 is even more realistic than Problem 5. The surface might not be entirely visible from the solution to Problem 6 if the imposed budget is too low such that no feasible solution to cover the entire surface may be found.

For $C(H)>0$ when $H=0$, we need to define F to be a positive value. For an increasing function of height, S should be a positive number and A should be limited within (0,1) to allow the function to increase when tower height increases but at a decreasing rate. Note that this function is only used for the purpose of illustration. Although it was assumed that the cost of building a tower increases at a decreasing rate when its height increases, it is expected that the cost will increase sharply when its height exceeds certain limit. For example, the cost of adding another foot to the height of CN tower in Toronto will be very much higher than the cost of building its first foot near the ground. Therefore, when calculating cost of tower construction, another cost function should be defined for towers whose heights exceed the specified upper limit. Accurate estimates of the values of F, S, A, and a proper height limit will be defined according to types of construction and materials used.

With cost function defined, it is then possible to solve Problem 5 and 6 by combinations of the three heuristics used in the previous section. The basic concept is to start with an initial solution set of viewpoints with zero heights from

which the entire surface is visible, then at each iteration drop the solution viewpoint which reduces the cost the most until no further cost reduction may be found. Each time a viewpoint is dropped, inevitably some triangles may become un-covered (not visible from any of the remaining solution viewpoints). These un-covered triangles may be re-covered by raising the heights of other viewpoints remaining in the solution set, the minimal visible heights and visibility matrix defined in section 6.1.

A viewpoint is considered to be droppable from the solution set only if the reduction in cost after dropping the point exceeds the cost of raising the heights of other solution viewpoints to re-cover those triangles which became un-covered when the point was dropped. At each iteration, cost is re-calculated for each droppable viewpoint to decide which point, if any, is to be dropped from the solution set. The heights to which other viewpoints have to be raised are also decided from the visibility matrix and are included when calculating the associated cost. These processes are repeated until no further cost reduction is possible. Similar to the concept of GAS, the solution may be improved by attempting to swap a solution viewpoint with another viewpoint candidate to see if the number of viewpoints or the total cost may be further reduced at each iteration. The algorithms, VH and VHS, are expressed by pseudocodes in Algorithms 6.4. and 6.5., which were coded in FORTRAN and tested on a CYBER 835.

1. input the solution found by GA for Problem 3 as the initial solution set, S

   compute the objective function, $OB=\Sigma_i C(H_i)$, $i \in S$.

2. drop a point, P_d, from S, which has the least areas that are visible before dropping P_d and are invisible after dropping P_d.

3. find all triangles that became invisible because of dropping P_d.

   for each such triangle, along the column of the triangle in the minimal visible height matrix, find a viewpoint in the solution set whose minimal visible height to the triangle is the least within that column and raise the height of the viewpoint by the minimal visible height recorded in the matrix.

   repeat the process until all uncovered triangle is covered again.

   compute the objective function, $OB1=\Sigma_i C(H_i)$, $i \in S$, $i <> d$.

   if OB1 < OB, set OB=OB1.

4. for problem 5 -
   if no P_d found to have OB1 < OB, stop and output solution set.

   for problem 6 -
   if OB1 <= T, stop and output solution set.

Algorithm 6.4. VH Algorithm

1. input the solution found by GA for Problem 3 as the initial solution set, S

   compute the objective function, $OB=\Sigma_i C(H_i)$, $i \in S$.

2. drop a point, P_d, from S, which has the least areas that are visible before dropping P_d and are invisible after dropping P_d.

3. find all triangles that became invisible because of dropping P_d.

   for each such triangle, along the column of the triangle in the minimal visible height matrix, find a viewpoint in the solution set whose minimal visible height to the triangle is the least within that column and raise the height of the viewpoint by the minimal visible height recorded in the matrix.

   repeat the process until all uncovered triangle is covered again.

   compute the objective function, $OB1=\Sigma_i C(H_i)$, $i \in S$, $i <> d$.

   if OB1 < OB, set OB=OB1.

4. for every point P_k, k=1, ..., k<>S, k<>d, attempt to swap P_k with P_m, $m \in S$.

   for every swap attempt, compute $OB2=\Sigma_i C(H_i)$, $i \in S$, $i <> k$, $i = m$. If OB2 < OB, set OB=OB2 and swap P_k with P_m.

   for problem 5 -
   if no P_m found to have OB2 < OB, stop and output solution set.

   for problem 6 -
   if OB2 <= T, stop and output solution set.

Algorithm 6.5. VHS Algorithm

Theoretically, one can always find a single point solution for each surface, if F is large enough and S and A are small enough. However, these solutions very often are found to be associated with extremely high values of height, implying such values are inappropriate. Therefore, suited values for an applicable cost function will need to be based on practical experiences and prices of construction materials used to build watchtowers. For the purpose of illustration, F, S, and A were chosen as 1000., 100., and .50 respectively. The following discussion are based on these values.

The number of solution viewpoints selected by VH and VHS are shown with the highest tower height in the selected solution set and the associated cost. For the 30 random data sets, VHS improved solutions by swapping solution viewpoints with those not included except for 11 cases. For the 4 TINs VHS improved TIN1 but not TIN2, TIN3, or TIN4.

Six viewpoints were dropped from the initial solution set selected by GA in TIN1, Krone Ranch, Montana (See Figure 6.5.[4]). These dropped points were found to be located along the ridge lines of the Reef, surrounding Nicholas Basin. A total of 5 viewpoints were raised, to heights from 3.5 meters to 45.21 meters. These tower locations, selected by VH, were also located on ridge lines of the Reef and are close to the

---

[4]The locations of dropped viewpoints are shown in the respective figures by a shaded square. The locations of tower sites are shown by cross hatched squares.

F=1000 S=100 A=0.5	VH		VHS	
	NOS OF PTS. (HIGHEST MVH IN SOLUTION)	OBJ. FUNC. COST	NO. OF PTS. (HIGHEST MVH IN SOLUTION)	OBJ. FUNC. COST
TIN101	4 (53)	4405.29	1 (355)	2884.55
TIN102	8 (0)	8000.00	8 (0)	8000.00
TIN103	6 (48)	6699.76	2 (168)	4116.32
TIN104	1 (574)	3397.11	1 (574)	3397.11
TIN105	3 (202)	5736.58	2 (82)	3342.02
TIN106	3 (6?)	5018.80	1 (355)	2884.55
TIN107	7 (10)	7325.55	6 (10)	6327.00
TIN108	5 (9)	5756.17	4 (11)	4789.76
TIN109	3 (18)	4092.92	1 (185)	2360.61
TIN110	2 (208)	4260.20	1 (104)	2356.97
TIN201	8 (19)	9032.75	1 (808)	3843.18
TIN202	5 (34)	6001.94	3 (45)	4211.55
TIN203	9 (8)	9286.14	9 (8)	9286.14
TIN204	3 (71)	4409.90	1 (623)	3496.08
TIN205	3 (262)	5516.68	3 (193)	5007.98
TIN206	5 (41)	6531.19	4 (35)	5851.51
TIN207	8 (13)	8594.12	8 (7)	8503.18
TIN208	6 (27)	7062.67	6 (21)	6855.91
TIN209	7 (0)	7000.00	7 (0)	7000.00
TIN210	7 (33)	7477.51	5 (9)	6068.91
TIN301	8 (31)	8801.33	8 (31)	8607.62
TIN302	7 (11)	7338.53	7 (11)	7338.53
TIN303	10(10)	10318.09	2 (166)	4334.59
TIN304	7 (0)	7000.00	7 (0)	7000.00
TIN305	8 (0)	8000.00	8 (0)	8000.00
TIN306	6 (62)	7375.19	5 (48)	6880.14
TIN307	8 (15)	8698.30	8 (11)	8647.30
TIN308	7 (13)	7363.83	7 (13)	7363.83
TIN309	8 (13)	8369.73	4 (139)	5755.92
TIN310	7 (36)	7976.70	7 (36)	7976.70
TIN1	19(45)	20653.03	19(22)	20110.97
TIN2	30(5)	30244.74	30(5)	30244.74
TIN3	30(21)	31055.95	30(21)	31055.95
TIN4	24(25)	24854.97	24(25)	24854.97

Table 6.5. Solutions To Problem 5

dropped viewpoints. VHS swapped another viewpoint located on the Reef to reduce the largest tower height to 22.48 meters.

In the case of TIN2, Herbert Domain, Tenn., two viewpoints on Buck Ridge were dropped to reduce cost. One tower of height 5.99 meters was raised by VH on Buck Ridge located close to the dropped points to re-cover the visibility regions previously covered by the two dropped points. VHS did not improve the solution found by VH in this case.

For TIN3, SouthWest Benndale, Miss., 3 viewpoints were dropped by VH (and VHS); these were found to be classified as ridge points according to the definitions in section 4.1.1. Three tower sites of heights from 6.5 to 21.12 were selected by VH (and VHS). Again the tower sites selected were found to be close to those dropped points. VHS did not improve the solution found by VH in this case.

For, TIN4, Cedar Breaks, Utah, VH dropped one viewpoint from the initial solution set. The dropped point is on the ridge line of Pink Cliffs located in the southeastern corner of the area. Two tower sites of heights 25 and 12.6 meters are located nearby, also on Pink Cliffs. Again, VHS did not improve the solution in this case.

In summary, viewpoints which were dropped from the initial solution set were found to be mostly ridge points. The sites selected to build towers were found mostly close to the dropped points. The swapping process was found to be very effective in improving the solutions in the 30 random cases.

However, the solutions found by VH for TIN2, TIN3, and TIN4 were not improved by VHS.

Finally, correlation coefficients were calculated between cost and the surface description parameters to search for possible relationships. The results are also listed in Table 6.4. Relationships between surface parameters and cost were found to be similar to those found previously for the number of solution viewpoints, indicating that higher relief variations, steeper slopes, higher spatial autocorrelation, and more rapid variation on variograms are associated with higher cost.

6.3. Conclusions

This chapter offered solutions to all visibility problems defined in chapter III as algorithms. It was shown that the three heuristics, Greedy Add, Greedy Add with Swap, and Stingy Drop, are capable of solving visibility coverage problems defined by this study. Although it was not possible to find optimal solutions to each problem, the heuristics for set-covering problems appear to work well and offer lower-cost, and quicker solutions than any exhaustive searching method on both randomly generated surfaces and real topographic surfaces.

The geographical patterns of selected viewpoints for visibility coverage problems were shown to be closely related

to the type of topographic surfaces and their topographic features. Parameters describing surfaces such as the distribution of triangle slopes, the distribution of elevations and their degree of dispersion, the spatial autocorrelation coefficient and the slope of regression lines of variograms were found to have strong association with the number of viewpoints selected in solution sets and watchtower cost in our 34 cases. The viewpoints selected are the topographic feature points of peaks or points along ridge lines. Plots of locations of selected viewpoint reveal that the geomorphological structure of the surface affects the distribution of viewpoints. Locations such as the borders of plateaus, ridge cliffs, or fault-like elevation shifts shown in our four TIN models were found to be the best places to put viewpoints to see the surfaces.

CHAPTER VII SUMMARY AND CONCLUDING REMARKS

This thesis has presented a systematic study of visibility coverage problems on topographic surfaces. Various representations of topographic surfaces, including contouring, grid DEMs, Triangulated Irregular Networks (TIN), were reviewed and it was concluded that the TIN model, which partitions the surfaces into triangular faces, is best suited for the study because of its efficiency in storing data and in accommodating irregularly spaced data points.

The sample data sets used in this study consisted of 30 randomly generated data sets and 4 U.S.G.S. grid DEMs. The U.S.G.S. grid DEMs were later converted into TIN models by using a drop heuristic conversion algorithm as a result of a comparison with existing conversion algorithms. Many types of research activities have become more feasible than before as a result of increasing coverage and accessibility of U.S.G.S. DEMs. Therefore, an effective conversion algorithm for extracting TIN models from grid DEMs to represent topographic surfaces has become more and more important. The Drop heuristic algorithm proposed in this thesis was found to be satisfactory in attempting to minimize the differences between the surfaces described by the DEM and the surface described by the extracted TIN.

Although it is possible to find precise definitions of visibility in the context of a specified model of a surface such as the TIN used in this thesis, no such precise definition exists for real terrain. We have already seen that the digital elevation model can only approximate the real surface: although elevations at vertices may be exact, the surface between them is assumed to vary linearly. To obtain an accurate representation it is necessary to choose a large number of appropriately located sample points. However, an accurate DEM is no guarantee that the set of triangles visible from a point will be an accurate representation of the seen area. Apart from artifacts such as trees which may inhibit visibility independently of the topography itself, a small error in elevation is assumed to produce very large errors in visibility. For example, a difference of a few centimetres in a horizon close to the viewpoint can produce a difference of many square kilometres in the visible area.

The relationship between a viewpoint and its associated visible regions is expected to be affected by not only the accuracy of elevation data but also by the accuracy lost during conversion processes. This calls for an investigation of the relationship between heights of viewpoints and associated areas of visible regions over various degrees of accuracy in elevation data.

A set of visibility coverage problems of fixed and variable heights, the main concern of the current study, was

defined both on a continuous model of a surface and on a discrete model which approximates a continuous surface through a set of vertices and facets of triangles. The algorithms used to solve visibility coverage problems were based on visibility information and the minimal visible heights between viewpoints and subregions (triangles) on the surfaces. Two visibility algorithms were developed to extract visibility information in this study. Visibility information was then recorded in matrices to be used to solve coverage problems.

The visibility coverage problems discussed included two fixed height problems: the determination of a minimum number of viewpoints to see the entire surface and the determination of viewpoint sites to maximize the visibility coverage for a given number of viewpoints; and two variable height problems: the determination of locations and heights of a set of towers to minimize the cost and the determination of locations and heights of a set of towers to maximize coverage under a cost budget.

These visibility coverage problems were found to have the same formulation as that of set-covering problems. Therefore, heuristic algorithms which have been suggested to solve set-covering problems were adapted to solve visibility coverage problems. Among the three used, Greedy Add, Greedy Add with Swap, and Stingy Drop, Greedy Add offered the quickest approach to a solution over 34 sample data sets. The next quickest heuristic was found to be SD. GAS was found to have

the highest chance of reaching the best solution. However, the computing time required by GAS is very much more than that required by GA or SD.

It was concluded that GA is preferred but SD and GAS should be also used to see if there is any improvement possible when computing resource is not a constraint or when processing smaller size data sets. Problems of variable heights may be solved by some combination of the three heuristics. It was concluded that the extent to which cost was reduced by dropping viewpoint sites and raising heights at other sites depends on the cost function used. Generally greater cost reduction is associated with greater fixed cost and lower effect of height on cost.

Although the procedures for extracting visibility information from a surface might be computationally intensive especially when a large data set is used, knowledge of visibility information allows us to pursue a wide variety of applications. In this thesis, solutions were presented to some problems in facilities location.

Other examples for future research on visibility path problems can also be formulated in route finding, in autonomous vehicle navigation, and in laying out unsightly utilities, such as pipe lines and transmission lines so as to minimize their scenic impact. Furthermore, visibility information may also be used for 3D display of triangulated elevation networks.

In this thesis, patterns of visibility information have been shown to be closely tied to different types of topographic surfaces. It was found that peaks and points on ridge lines were most likely to be good candidates as solution viewpoints. The number of viewpoints required to see a surface entirely were found to be strongly associated with types of topography. Parameters describing a surface with less relief variation were found to be strongly correlated with smaller numbers of viewpoints. The surface description parameters which demonstrated strong positive correlations with the number of solution viewpoints included mean, standard deviation, and kurtosis of elevations, and mean and standard deviation of triangle slopes, the spatial autocorrelation coefficient, and the slope of regression lines of variograms for each surface. Similar relationships were found between these parameters and the cost associated with locating watchtowers on surfaces.

In addition to random surfaces, a variety of topographic surfaces were used in this study, including plateaus, plains with small local variation of elevations, and areas with significant valleys and ridges. It was concluded that best locations to be selected as viewpoints to monitor a surface are usually the locations along sharp changes of elevations, along major topographic features (such as ridges), and on nodal topographic features (such as peaks).

Although the findings are useful in exploring the relationship between surface topography and solutions of visibility coverage problems, they were made from only 30 random data sets and 4 real surfaces. For future research, a systematic sampling scheme over various types of topography might be included to support more generalized conclusions.

REFERENCES

Anderson, D. P., 1982, "Hidden line elimination in a projected grid surface". ACM Trans Graphics, Vol. 1, No. 4, pp. 274-291.

Atwood, W. W., 1964, The Physiographic Provinces of North America. London: Blaisdell Pub. Co.

Brassel, K. and D. Reif, 1979, "A procedure to generate Thiessen polygons". Geographical Analysis, Vol. 11, pp. 289-303.

Burrough, P. A., 1983, "Multiscale sources of spatial variation in soil". Journal of Soil Science, Vol. 34, I:pp. 577-597, II:pp. 599-620.

Burrough, P. A., 1986, Principles of Geographic Information Systems for Land Resource Assessment. Monographs on Soil and Resource Survey, No. 12, London: Oxford University Press.

Burton, R. P. and D. R. Smith, 1982, "Hidden-line algorithm for hyperspace". SIAM. J. Comput., Vol. 11, pp. 71-80.

Chen, Z-T. and J. A. Guevara, 1987, "Systematic selection of very important points (VIP) from a digital terrain model for constructing triangular irregular networks". AUTO-CARTO 8, ACSM, Baltimore, MD, pp. 50-56.

Church, R., 1974, "Synthesis of a class of public facility location models". Ph.D. thesis, The John Hopkins University, Baltimore, MD., 1974.

Church, R. and C. ReVelle, 1974, "The maximal covering location problems". Papers, Regional Science Association, Vol. 32, pp. 101-118.

Church, R. and C. ReVelle, 1976, "Theoretical and computational links between the p-median, location set-covering and the maximal covering location problem". Geographical Analysis, Vol. 8, pp. 406-415.

Cliff, A. D. and J. K. Ord, 1973, Spatial Autocorrelation. London: Pion.

Cliff, A. D. and J. K. Ord, 1981, Spatial Processes: Models and Applications. London: Pion.

Cole, R. and M. Sharir, 1986, "Visibility problems for polyhedral terrains". Tech. Rep. No. 32, Courant Institute, New York University, New York.

Collins, S., 1975, "Terrain parameters directly from a digital terrain model". Canadian Surveyor, Vol. 29, No. 5, pp. 507-518.

Cooper, L. and D. Steinberg, 1970, Introduction to Methods of Optimization. London: W. B. Saunders Co.

Culling, W. E. H. and M. Datko, 1987, "The fractal geometry of the soil-covered landscape". Earth Surface Processes and Landforms, Vol. 12, pp. 369-385.

Davis, J. C., 1975, "Contouring algorithms". Proceedings, AUTO-CARTO II, ACSM, pp. 352-59.

Davis, J. C. and M. J. McCullagh, eds., 1975, Display and analysis of spatial data. New York: John Wiley & Sons.

De Floriani, L., B. Falcidieno, C. Pienovi, 1985a, "Delaunay-based representation of surface defined over arbitrarily shaped domain". Computer Graphics, Vision and Image Processing, 32, pp. 127-140.

De Floriani, L., B. Falcidieno, C. Pienovi, and G. Nagy, 1985b, "Efficient selection, storage, and retrieval of irregularly distributed elevation data". Computers and Geosciences, Vol. 11, No. 6, pp. 667-673.

De Floriani, L., B. Falcidieno, C. Pienovi, D. Allen, and G. Nagy, 1986, "A visibility-based model for terrain features". Proceedings, Second International Symposium on Spatial Data Handling, Seattle, Washington, pp. 235-250.

De Floriani, L. and G. Nagy, 1989, "Visibility problems on a topographic surface". Unpublished manuscript.

Douglas, D. and T. K. Peucker, 1973, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature". Canadian Cartographer, Vol. 10, No. 2, pp. 112-122.

Douglas, D., 1986, "Experiments to locate ridges and channels to create a new type of digital elevation model". Cartographica, Vol. 23, No. 4, pp. 29-61.

Ebdon, D., 1985, Statistics in Geography. 2nd ed. Blackwell, Oxford.

Elassal, A. A. and V. M. Caruso, 1983, "Digital Elevation Models". in R. B. McEwen, R. E. Witmer, and B. S. Ramey, editors, U.S.G.S. Digital Cartographic Data Standards. Geological Survey Circular 895-B, U.S.G.S.

El Gindy, H. and D. Avis, 1981, "A linear algorithm for computing the visibility region of a polygon from a point:. J. of Algorithm, Vol. 2, pp. 186-197.

El-Shaieb, A. M., 1973, "A new algorithm for locating sources among destinations". Management Science, Vol. 20, pp. 221-31.

Fowler, R. J. and J. J. Little, 1979, "Automated extraction of irregular network digital terrain models". Computer Graphics, Vol. 13, No. 2, pp. 199-207.

Foley, J. D. and A. van Dam, 1983, Fundamentals of interactive computer graphics. Addison Wesley, Reading, Massachusetts.

Garfinkel, R., A. Neebe, and M. Rao, 1974, "An algorithm for the median plant location problem". Transportation Science, Vol. 9, pp. 217-236.

Geary, R. C., 1954, "The contiguity ratio and statistical mapping". Incorporated Statistician, Vol. 5, pp. 115-41.

Geary, R. C., 1968, "The contiguity ratio and statistical mapping". in B. J. L. Berry and D. F. Marble, eds., Spatial Analysis: A Reader in Statistical Geography, Prentice-Hall, Englewood Cliffs, N. J., pp. 461-478.

Goodchild, M. F., 1986, Spatial Autocorrelation. Concepts and techniques in modern geography (CATMOG), No. 47., Norwich: Geo Books.

Goodchild, M. F., 1987, Final report on Thiessen polygon project. Department of Geography, The University of Western Ontario.

Goodchild, M. F. and J. Lee, 1987, Network algorithms for the Electronic Atlas. Final report of contract 2ER.23242-6-2385 with Geographical Service Division, Surveys and Mapping Branch, Energy, Mines and Resources Canada.

Goodchild, M. F. and D. M. Mark, 1987, "The fractal nature of geographic phenomena". Annals of the Association of American Geographers, Vol. 77, pp. 265-278.

Gold, C. M., 1987, "Spatial ordering of Voronoi networks and their use in terrain data base management". Proceedings, AUTO-CARTO VII, ACSM, Baltimore, MD, March 1987, pp. 185-194.

Griffith, D. A., 1987, Spatial Autocorrelation: A Primer. Resource Publications in Geography, AAG, Washington, D. C.

Hakimi, S. L., 1964, "Optimal location of switching centres and the absolute centres and medians of a graph". Operations Research, Vol. 12, pp. 450-59.

Hakimi, S. L., 1965, "Optimum distribution of switching centres in a communication network and some related graph theoretic problems". Operations Research, Vol. 13, pp. 462-75.

Hunt, C. B., 1974, Physiography of the United States. London: Freeman & Co.

Johnston, E. G. and A. Rosefeld, 1975, "Digital detection of pits, peaks, ridges, and ravines". IEEE Trans. System, Man and Cybernetics, Vol. 5, pp. 472-480.

Khumawala, B. M. 1973, "An efficient algorithm for the p-median problem with maximum distance constraints". Geographical Analysis, Vol. 5, pp. 309-321.

Klinkenberg, B., 1988, Tests of Fractal Model of Topography. Ph.D. Thesis, Department of Geography, The University of Western Ontario, London, Ontario, Canada.

Krige, D. G., 1966, "Two-dimensional weighted moving average trend surfaces for ore valuation". Journal of the S. African Inst. of Mining & Metallurgy: Symposium - Mathematical Statistics & Computer Applications in Ore Evaluation.

Lee, D. T. and B. J. Schachter, 1980, "Two algorithms for constructing a Delaunay triangulation". International Journal of Computer and Information Science, Vol. 9, pp. 219-242.

Mandelbrot, B. B., 1975, "On the geometry homogeneous turbulence, with stress on the fractal dimension of the isosurface of scalars". Journal of Fluid Mechanics, Vol. 72, pp. 401-416.

Mandelbrot, B. B., 1977, Fractals: Form, Chance and Dimension. San Francisco: Freeman.

Marazana, F., 1964, "On the location of supply points to minimize transport cost". Operational Research Quarterly, Vol. 15, pp. 261-270.

Mark, D. M., 1975, "Computer analysis of topography: a comparison of terrain storage methods". Geografiska Annaler, Vol. 57A, No. 3-4, pp. 179-188.

Mark, D. M., 1978, "Concepts of Data Structure for DTM". Proceedings of the DTM Symposium, St. Louis, MO, ASP-ACSM, pp. 24-31.

Mark, D. M., 1983, "Automated detection of drainage networks from digital elevation models". Proceedings, AUTO-CARTO VI, Vol. 2, pp. 288-289.

Mark, D. M. and P. B. Aronson, 1984, "Scale-dependent fractal dimensions of topographic surfaces: an empirical investigation, with applications in geomorphology". Math. Geology.

McKenna, M., 1987, "Worst-case optimal hidden-surface removal". ACM Trans. on Graphics.

Moore, E. F., 1957, "The shortest path through a maze". Proceedings, International Symposium on the Theory of Switching. Harvard University Press, Cambridge, MA., pp. 285-292.

Moran, P. A. P., 1948, "The interpretation of statistical maps". Journal of the Royal Statistical Society, Vol. 43, pp. 321-328.

Nagy, G. and Wagle, 1979, "Geographic data processing". ACM Computing Surveys, Vol. 11, No. 2, pp. 139-81.

Nair, H., 1988, A High-level Description of Digital Terrain Models Using Visibility Information. Master's Thesis, Department of Electrical, Computer and System Engineering, Rensselaer Polytechnic Institute, Troy, New York.

4.2.3. The hierarchical triangulation algorithm

The third existing algorithm was introduced by de Floriani et al. (1985b), named hierarchical triangulation. This algorithm uses an additive procedure in which points are selected to "reduce the maximum error between a piecewise linear approximation o the surface using only the selected points and the elevations of the points not selected" (de Floriani et al. 1985, pp. 667).

The extraction of a TIN model from a dense DEM suggested by de Floriani et al. (1985b) consists of two steps:

1) Initial subdivision of the data points into mutually exclusive and completely exhaustive triangular subsets.
2) Subdivision of each subset into nested triangles by hierarchical triangulation.

A convex hull is first constructed around the data points. An internal point is then selected, and the triangles connecting the hull to the internal point are constructed. Ideally, this internal point is selected such that the maximum error within the resulting triangles is minimized. If, however, the total number of points is large, or if the region is not simply connected (for example, if the area of interest contains a lake) then we may have to accept a non-optimal subdivision, which can be obtained more rapidly.

Newman, W. M. and R. F. Sproull, 1979, Principles of Interactive Computer Graphics. McGraw-Hill, New York.

Peucker, T. K. and D. Douglas, 1975, "Detection of surface-specific points by local parallel processing of discrete terrain elevation data". Comput. Graph. Image Processing, Vol. 4, pp. 375-87.

Peucker, T., K. Fowler, J. Little, and D. Mark, 1978, "The triangulated irregular network". In Proceedings of the DTM Symposium, ASP-ACSM, St. Louis, Missouri, pp. 24-31.

Peucker, T. K., 1979, "Digital terrain model: an overview". Proceedings, AUTO-CARTO IV, ACSM, pp. 97-107.

Preparata, F. P. and M. I. Shamos, 1985, Computational Geometry: an introduction. Springer Verlag, New York.

ReVelle, C. S. and R. W. Swain, 1970, " Central facilities location". Geographical Analysis, January, 1970. pp 30-42.

Roberts, O. and S. Suhrbier, 1962, DTM location system, 20K program manual, Research Report R 62-8, MIT. Department of Civil Engineering, M.I.T., Boston, Mass, April, 1962.

Sharir, M., 1987, "The shortest watchtower and related problems for polyhedral terrains". Tech. Rep. NO. 334, Courant Institute, New York University.

Sprunt, B., 1975, "Relief representation in automated cartography: an algorithm approach". in J. C. Davis and M. McCullagh, eds., Display and Analysis of Spatial Data. London: John Wiley and Sons.

Strahler, A. N., 1952, Hypsometric (Area-altitude) analysis of erosional topography. Bull. of the Geological Soci. Am. Vol. 63, pp. 1117-1142.

Sutherland, I. E., R. F. Sproull, and R. A. Schumacker, 1974, "A characterisation of ten hidden-surface algorithms". Comput. Surveys, Vol. 6, No. 1, pp. 1-55.

Swain, R., 1974, "A parametric decomposition approach for the solution of uncapacitated location problems". Management Science, Vol. 21, pp. 189-98.

Tarvydas, A., 1983, "Terrain approximation by triangular facets". Proceedings, AUTO-CARTO VI, Ottawa.

Teitz, M. and P. Bart, 1968, "Heuristic methods for estimating generalized vertex median of a weighted graph". Operations Research, Vol. 16, pp. 801-1092.

Thornbury, W. D., 1965, Regional Geomorphology of the United States. New York: Wiley.

Tobler, W. R., 1970, "A computer movie simulating urban growth in the Detroit region". Economic Geography (Supplement), Vol. 46, pp. 234-240.

Toregas, C., R. Swain, C. ReVelle, and L. Bergman, 1971, "The location of emergency service facilities". Operations Research, Vol. 20, pp. 1363-1374.

Toregas, C. and C. ReVelle, 1972, "Optimal location under time or distance constraints". Papers of Regional Science Association, Vol. 28, pp. 133-143.

Weiler, S. and D. Artherton, 1977, "Hidden surface elimination using polygon area sorting". Proc., SIGGraph, pp. 214-222.

Wright, T. J. 1973, "A two-space solution to the hidden-line problem from functions of two variables". IEEE Trans. Comp., Vol. 22., pp. 28-33.

Warntz, W., 1966, "The topology of a socio-economic terrain and spatial flows". Papers of Regional Science Association, Vol. 17, pp. 47-61.