THE EXTRACTION OF TOPOGRAPHIC FEATURES IN SUPPORT OF AUTONOMOUS UNDERWATER VEHICLE NAVIGATION

Don J. Orser and Martin Roche

Robot Systems Division
National Bureau of Standards
Gaithersburg, Md 20899

ABSTRACT

We describe in an expository manner ongoing research concerned with the identification and extraction of topographic features relevant to automated navigation algorithms for an autonomous underwater vehicle. These features are presented within the framework of the *extremal point topography network* (EPTN), an idea going back to Arthur Cayley and J. Clerk Maxwell. The computational problems addressed here are the reconstruction of the surface terrain from irregular spaced bathymetric data and the subsequent extraction of the EPTN. While clearly no single best method exists for this latter step, we present here a description of several methods we have tried with some success. The data used for this research is that for a selected area of Lake Winnipesaukee, New Hampshire.

Table of Contents by Section

"Hence the whole earth may be naturally divided into Basins or Dales, and also, by an independent division, into hills, each point of the surface belonging to a certain dale and also to a certain hill." JAMES CLERK MAXWELL, 1870.

1. INTRODUCTION

An autonomous underwater vehicle (AUV) has two types of information available to it: (1) *A priori* information incorporated into its design representing an initial state for both it and its envionment, and (2), *sensory space* information acquired by sensors as it moves through space-time. From the designer's point of view, desirable features of a priori information are (1), its static persistence over time, and (2), its representation of environmental features outside any temporally fixed sensory space frame, i. e., "global" information. These two characteristics of a priori information are of fundamental importance to the success of global (spatial) long range (time) navi-

This work was supported by the Defense Advanced Projects Agency (DARPA), Naval Technology Office, Arlington, Va. 22209-2308.

gation for an AUV.

For AUV *route planning* and *self localization*, an obvious instance of such desirable a priori information is the topographic features of the bottom surface terrain: They do not change within the timeframe of task execution, and in general, relevant features extend well beyond the range (from any given location) of sensors designed for their detection. If such features are not available a priori, then intelligent behavior most surely will be to acquire such features as an adjunct to whatever other tasks are assigned it [18,24]. In this paper we are assuming that a priori data is available.

Topographic features enter into AUV route planning (with respect to carrying out a given task), by giving *utility* to some features and *avoidance* to others. For example, within a search task, the "ridges" provide utility by indicating a route which maximizes the area searched by "line of sight" techniques. Conversely, ridge avoidance may be indicated if the AUV is the search target, in which case "passes" through the ridges connecting "ravines" may then have utility. Such considerations lead naturally to the idea of representing, not only the elevation terrain function, but also the utility/avoidance terrain functions, which may be thought of as global nonlinear transformations of the elevation function [19,28].

Secondly, topographic features enter into global navigation by potentially providing for matches between observed features and a priori stored ones. If the sensors used in acquiring these landmarks are passive, this provides a passive solution to the self localization problem.

Analogous global route planning/localization by humans is performed from pictorial navigation charts. Presented with such a chart, a trained human must visually interpret [17] what he sees in order to make the information mentally available for use in navigation. An autonomous vehicle algorithmic navigator has no comparable visual interpretive system, and hence merely storing "digitized" charts is inadequate. A much more explicit representation of chart data for AUV navigation is needed.

Navigation chart data is of two types: *Primary* data conveying the location of physical features which are detectable from physical measurements made on the actual topography, and (2), *secondary* data, corresponding to geographical, political, military etc. boundaries, not inherently detectable by physical measurement. Here we will be concerned exclusively with the algorithmic extraction of primary information from data obtained by physical measurement of terrain.

Specifically, the process by which this topographic a priori information is to be made available to the AUV is dealt with here. This process may be divided up into the following steps:

(1) The *reconstruction* of the surface terrain from bathymetric data.

(2) The *identification* of the topographic features relevant to the class of tasks to be undertaken by the AUV.

(3) The *extraction* of these features in which their implicit iconic representation is made into an explicit symbolic one.

(4) And finally, their various potential *representations* as computer data structures in which the requirements of the specific algorithms involved are taken into account.

In this paper we will be primarily concerned with (1) and (2).

2. THE EXTREMAL POINT TOPOGRAPHY NETWORK

More specifically, the problem addressed in this research is the following: Given a collection of irregularly spaced depth soundings, algorithmically generate a topographic feature network map consisting of the following:

(1) Rank ordered elevation minima, where rank is determined by "watershed" collecting at that minimum.

(2) Ravines connecting adjacent elevation minima, together with "blind" ravines.

(3) Rank ordered elevation maxima, ranked analogously to elevation minima.

(4) Ridges connecting adjacent elevation maxima, together with ridge "bifurcations".

(1)-(2) and (3)-(4) each define the nodes and directed edges of disjoint networks except at points of edge intersections which are a fifth type:

(5) Passes, or saddle points, which are ridge minima through which ravine maxima pass.

Imagine a vertical plane through an elevation minimum or maximum: It intersects the terrain surface in a curve which must have at least one inflection point to either side of the minimum or maximum. As the plane is imagined to rotate about the minimum/maximum, it generates a closed curve made up of these points. (In order to maintain closure, portions of these curves may be forced to coincide with ridges and ravines respectively.) These points form a sixth class:

(6) The closed curve surrounding each elevation minimum/maximum made up of the inflection points or "parabolic" points surrounding it.

These curves provide a consistent definition for the area surrounding each minimum and maximum, which in turn gives a specific interpretation for the terms "valley" and "hill". For example, the ratio of the average difference in height between this closed curve and the minimum or maximum, and the area inside the curve provides a consistent method of distinguishing between "pits" and "shallow valleys", and "peaks" and "hills", respectively.

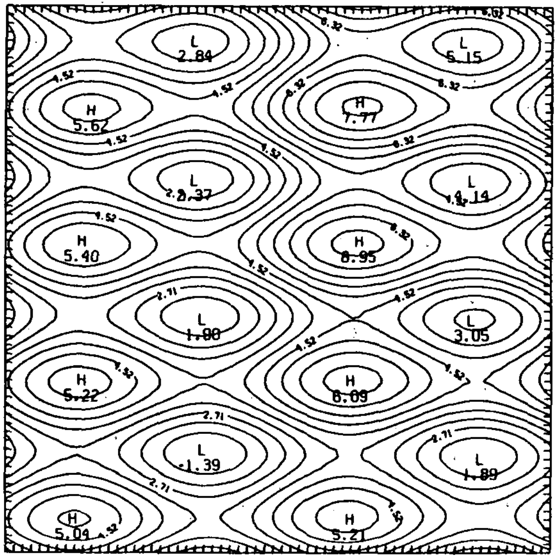

Figure 1 - Contour plot of synthetic surface consisting of "rolling hills" terrain. A perspective view is given in figures 7, 8 and 9.

In figure 1 we depict a perspective view of a 51 by 51 grid of sampled values taken from the surface:

$$z = \sin(x)+\sin(y)+\sin(.01xy)+.025xy. \qquad 0\leq x,y\leq 4\pi$$

In figures 2 and 3 the ravines and ridges, respectively, of this surface are shown as sequences of vectors terminating at minima and maxima. These ravines and ridges were extracted from the 51 by 51 grid using the descending/ascending count method described in Section 6. In figure 4, contour lines and undifferentiated ravines and ridges are superimposed using data to be discussed in a later section.

The characterization of topography by (1)-(5) has been known since the time of Cayley [6] and Maxwell [20]. More recently, those doing research in computer vision have applied these labelings to imagery[14,15]. Since terrain is continuous, concepts from elementary differential geometry seem readily applicable [16,22], and lend support to the ideas of (1)-(6). A grammar [27] would even make possible the "parsing" of terrain. This network we call the *(elevation) extremal point topography network* (EPTN).

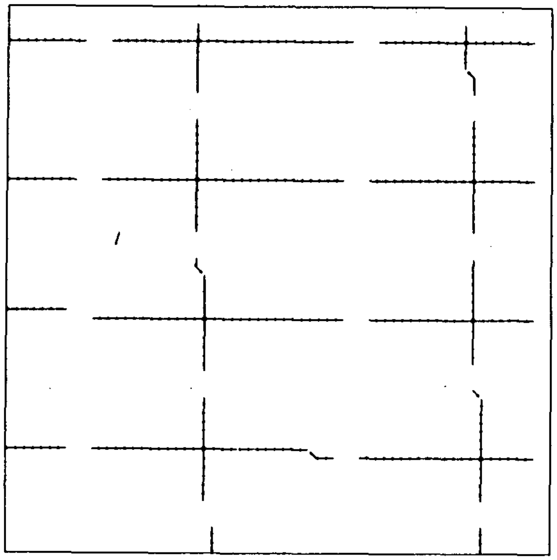

Figure 2 - The ravines of the synthetic surface shown in figure 1 are shown as arrows in the direction of their respective minima.

There are two methods for giving a formal definition to these extremal point classes: (1) The differential geometry definition is based on *intrinsic* properties (principle curvatures) independent of any coordinate system [8], while (2), for purposes of terrain classification, these extremal points may be defined in terms of terrain height and gradient (z coordinate), and hence are *extrinsic* in their definition. We will not attempt a formal definition here, but rather rely on the reader's intuition.

One useful analog in this respect is to think of the terrain as being rained on and to imagine the resulting water flow. In this hydrophysical model [25], ridges separate watershed regions, while ravines join them. Note that the ridge and ravine networks are duals of one another as may be seen by inverting the terrain so that the hills become the valleys and vice versa.

One point of clarification should be made however: A plane surface, horizontal or not, has neither ridges or ravines, nor does a hemispherical surface (either upper or lower portion). These surfaces are unique with respect to this propoerty. In differential geometry terms, this is due to the two principal curvatures being equal at all points (called umbilic points) of such surfaces [16]. For terrain, spherical

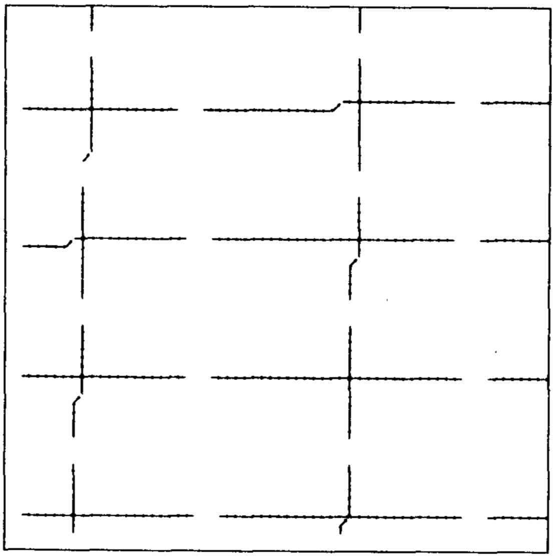

Figure 3 - Same as figure 2, but showing the ridges. Note that the ravines and ridges are disjoint.

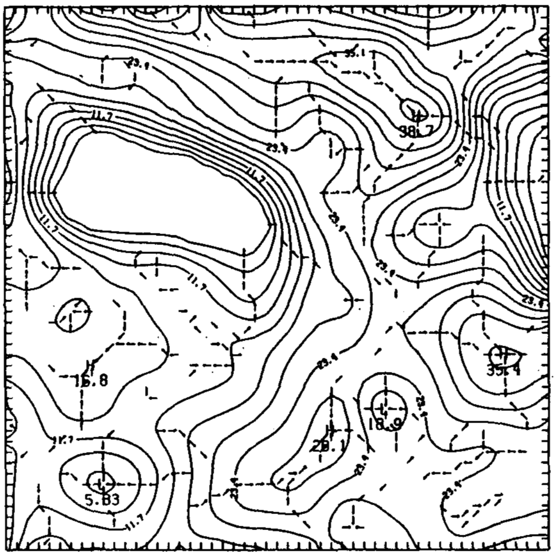

Figure 4 - Contour lines and undifferentiated ravine and ridge lines are shown superimposed using data from Lake Winnipesaukee, New Hampshire.

and flat topographic features may be of interest, but will not show up as features in the EPTN.

The actual number of minima and maxima on any portion of terrain is unbounded in number as one increases the resolution of measurement. As a result, the EPTN becomes more and more dense, and hence some method of truncating it to the desired resolution is necessary. This is accomplished in the following way.

Two minima (maxima) are *adjacent* if there is a ravine (ridge) connecting them through a single pass. A minima (maxima) adjacency matrix is formed by listing the minima (maxima) across the top and down the side in order of decreasing rank. An entry of one or zero indicates that the respective minima (maxima) are adjacent or not. Truncation to some desired resolution is then accomplished by truncating the matrix to some upper left hand portion.

Associated with the original terrain is the two dimensional gradient surface, which when treated in an analogous way through the use of (1)-(6) results in the *gradient extremal point topography network.* In the gradient network, minima and maxima correspond to level and steepest regions respectively.

Even more generally, the EPTN may be calculated for any two dimensional scalar field. For example, the EPTN may be calculated from a mapping of the magnitude of the deviations in the earth's magnetic field, and then subsequently used as navigation landmarks [32].

The point of view taken here is that these extremal point networks provide most of the information, to the extent that topographic features are to be taken into account, needed for navigation. The practical computational problem to be solved is that of automatically generating this network from a finite discrete set of irregularly spaced depth/elevation measurements.

There are at least two distinct approaches to the problem of identifying and extracting the extremal point network: (1) Methods which classify each "pixel" based on taking into account only a limited number of adjacent neighboring values, and (2), methods which are global in the sense that the labeling at a point may be affected by values arbitraily distant.

In this paper we describe a combination of local and global strategies for extracting the extremal point network and the result of their application to actual bathymetric data. This data is that used in the Multiple Autonomous Underwater Vehicle (MAUV) [1] Project and consists of a selected area of Lake Winnipesaukee, New Hampshire.

3. RECONSTRUCTION OF THE ELEVATION TOPOGRAPHY FROM BATHYMETRY

By bathymetric data is meant a finite sample of three dimensional points (z_i, x_i, y_i), taken from an unknown surface terrain function $z = f(x, y)$. For purposes here, we will also assume that f is single valued, e. g., there are no vertical "cliffs" or "caves". The value z is the depth and x and y may be thought of as latitude and longitude. Hence

$z = f(x, y)$ may be thought of as the *terrain elevation function*. We are usually also given x and y range limits, $x_{\min} \le x \le x_{\max}$, and $y_{\min} \le y \le y_{\max}$.

Given such a set of data, the task is to find a function $\tilde{f}$ which "best" fits the data, subject to $z_i = \tilde{f}(x_i, y_i)$ and hence is an interpolating approximation of f. We may also want an approximation of the gradient ∇f as well as the first several derivatives of f. For some purposes it may be sufficient to approximate $\tilde{f}$ and $\nabla\tilde{f}$ by a regular m by n grid of values $\tilde{z}_{ij}$ and $\nabla\tilde{z}_{ij}$, $1 \le i \le m$, $1 \le j \le n$.

This is the *bivariate interpolation problem* [3,4,12,13], and has a large literature associated with various methods for its solution. Note that this is a surface representation problem, and is not to be confused with the surface modeling problem arising in, for example, computer aided design.

For bathymetric data, the sample points, obtained from sonar ranging for example, may or may not satisfy the following conditions:

(1) Regular Spacing: The sampled points fall on the vertices of a straight line grid with variable (or constant) spacings between the lines.

An additional consideration for (1) is whether data is available for all points or just some sample.

(2) Uniformly Distributed: The sampled points are uniformaly distributed over the region, or are distributed according to some other a priori known probability distribution.

Note that if (1) is true, then (2) will also be true, though the resulting distribution may be complicated if the grid spacings are not constant.

In addition, it is often the case that bathymetric data is available in the form of contours drawn as a 2-dimensional map overlay, and the original data points are unavailable. Contour data, in addition to explicitly providing z_j samples, also provide information about the direction of the gradient, i. e., the gradient at a point on the contour is perpendicular to the contour at that point.

For elevation data obtained by stereometry applied to aerial photographic pairs, conditions (1) and (2) may be satisfied. However, for bathymetric data this is typically not the case and hence the case of generating regularly spaced interpolating values from irregularly spaced sample points is important. The strategy used here is described in [11,29] and is as follows.

Since three points determine a plane, a *bivariate linear interpolation* is obtained by finding three data points in the vicinity of x, y, passing a plane through them and evaluating it at x, y to obtain the interpolated value $\tilde{z}$.

More specifically, given the data depths z_0, z_1 and z_2 at three non-collinear points (x_0, y_0), (x_1, y_1) and (x_2, y_2), respectively, the interpolated depth $\tilde{z}$ and gradient $\nabla\tilde{f}$ at an (interior) point (x, y) of the triangle formed by these three points may be calculated as follows: First, let

$$A = [(y_1-y_0)(z_2-z_0)-(z_1-z_0)(y_2-y_0)]$$

$$B = [(z_1-z_0)(x_2-x_0)-(x_1-x_0)(z_2-z_0)]$$

$$C = [(x_1-x_0)(y_2-y_0)-(y_1-y_0)(x_2-x_0)]$$

$$D = -[Ax_1+By_1+Cz_1].$$

The equation of the plane is then

$$Ax+By+Cz+D = 0,$$

and we have the following interpolated values at position x, y:

$$\tilde{z} = \frac{Ax+By+D}{-C}$$

$$\nabla\tilde{f} = -\frac{A}{C}\vec{I}_x - \frac{B}{C}\vec{I}_y$$

$$|\nabla\tilde{f}| = \left[\left[\frac{A}{C}\right]^2 + \left[\frac{B}{C}\right]^2\right]^{1/2}$$

$$\text{Gradient direction } \theta = \tan^{-1}\left(\frac{B}{A}\right)$$

Note that adjacent triangles meet along a common edge so that the resulting interpolated values are continuous over the entire collection of triangles. However, the gradient across triangle boundaries is discontinuous. If continuity is desired here, one must go to higher order methods [2,31].

Since for a given x, y position, there is in general more than one triple of data points forming a triangle "containing" x, y, the problem then becomes one of determining, in general, which triple of data points gives the "best" fit.

Rather than find this triangle each time a location x, y is given, we desire a "triangulation" of the region to be interpolated. That is, we desire a collection of triangles of data points, together with a data structure, which will for location x, y, point to a triangle containing x, y. One way which gives good results is the *Delaunay triangulation* [7], also called the *Thiessen triangulation* [9]. Intuitively, the Delaunay triangulation is one in which the triangles are as equiangular as possible so as to avoid long thin triangles. It may be generated by taking the graph dual of the *Voronoi diagram* [23] of the data points. See figure 5.

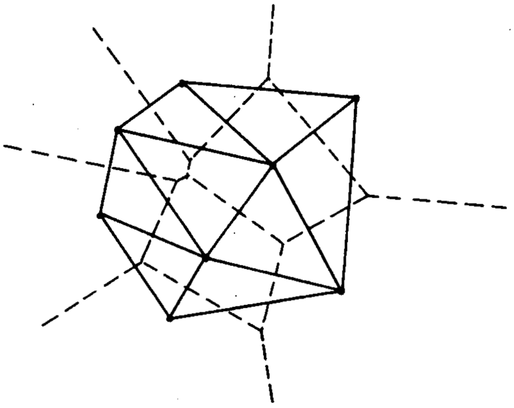

Figure 5 - A Delaunay triangulation (solid lines) and its graph dual, the Voronoi diagram (dashed lines) of the given data point soundings.

The Voronoi diagram is a graph in which each region contains one data point along with all those points of the plane which are closer to that data point than any other data point. The desired dual graph of the Voronoi diagram is then formed by connecting up those data points whose corresponding regions are adjacent in the Voronoi diagram. This divides that portion of the plane "interior" to the data points into disjoint triangles so that any given x, y location falls into a unique triangle formed by three data points[9].

Once such a triangulation is available, it then becomes easy to generate a regularly spaced grid of values $\tilde{z}_{ij}$ for any grid increment. Higher order methods, e. g., biquadratic and bicubic methods often start with a grid obtained in such a way.

There are a number of implementations of these ideas which meet various additional requirements, such as incremental addition of new data points. In [9,10,29,30] both theory and the description of a practical implementation is given. For our purpose we have chosen to use an implementation authored by Javier Bernal of the Center for Applied Mathematics of the National Bureau of Standards [5]. This program was designed with relatively large data sets in mind, and utilizes efficient methods for computing the Voronoi diagram.

For AUV navigation we contend that there is little or no need to store the entire surface explicitly, but rather what is needed is the ability to evaluate the functions $\tilde{f}$ and $\nabla\tilde{f}$ at arbitrary points. This is accomplished by storing just the coefficients of the splines (as is done in [10]) and not their values at some fixed grid resolution. Initially, a preprocess phase would do the fitting from whatever data is available, and the result incorporated in the run time system of an AUV. The ability to incrementally add data points without a complete refitting is also possible.

This method of representing the terrain elevation function, combined with the explicit storage of the extremal point topography network, also obtained offline in a preprocessing step, not only conserves storage, but potentially simplifies and speeds up AUV global route planning/self localization algorithms by already having extracted out the most pertinent information.

4. IDENTIFICATION OF THE EPTN: NEIGHBOR DIFFERENCING METHOD

In this method our objective is to extract global information through local operations. Every (grid) point will be classified on the basis of an analysis of its neighboring points (Figure 6). The difference in height of a point with each of its eight neighboring points is determined. These differences then, in turn, determine the features assigned to the point. The method used to analyze a point's neighbors is an extension of that followed in [26].

p_6	p_7	p_8
p_5	p_0	p_1
p_4	p_3	p_2

Figure 6 - Location of eight neighbors for pixel p_0.

For a given point P_0 and its neighbors P_1, P_2,P_3,P_4,P_5, P_6,P_7, and P_8 we set:

D_i = the difference in height between p_0 and its i-th neighbor (i = 1,...,8) ;

D^+ = the negative of the sum of all negative differences;

D^- = the sum of all positive differences;

NC = number of sign changes of D_i (with NC initialized at 1);

LC = number of points between two sign changes.

Using these we give classification criteria for elevation maxima, elevation minima, and passes (saddle points).

Letting tp, tf, tps, and ts be thresholds (or error tolerance values), the following point classifications are made:

(1) if $D^+ <$ tf and $D^- >$ tp, the point is an elevation maxima;

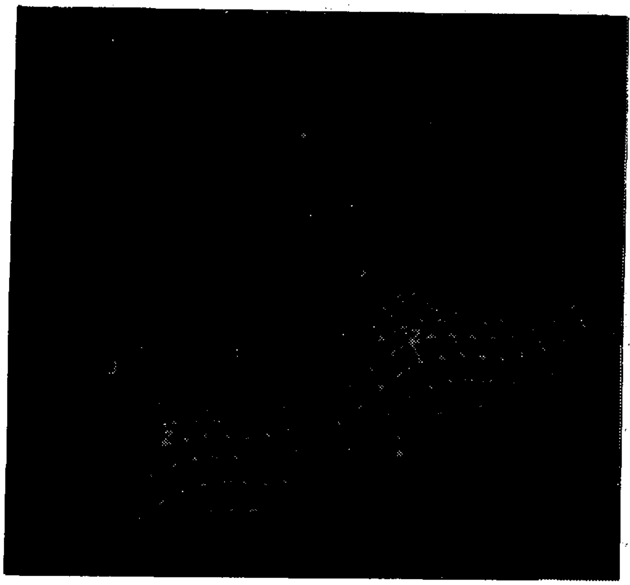

Figure 7 - This is a screen photo of a surface with elevation minima, elevation maxima, and the passes being determined by the neighbor differencing method and denoted by the numbers 1, 2, and 3 respectively.

(2) if $D^+ > tp$ and $D^- < tf$, the point is an elevation minima;

(3) if $D^+ + D^- < tf$, the point is a flat;

(4) if $NC \geq 4$ and (3) does not hold, the point is a pass;

(5) if $| D^+ - D^- | < ts$, $D^+ + D^- > tss$, $NC = 2$, and $LC = 4$, the point is a plane (with a non zero slope).

In our implementation we have assigned the following values or ranges to the threshold (or error tolerance) variables; $0 \leq tf \leq 1.$, $tp = 1.0$, and $ts = tss = 10.0$.

To determine if a point should be labeled a ravine point, we have broken with the Peucker [26] procedure. Using the notation of Figure 6, and defining points $P_9 = P_1, P_{10} = P_2$, $P_{11} = P_3$, and $P_{12} = P_4$ for notational convenience, we say that a point P_0 will be labled a ravine point if the following conditions hold:

(a) There is a point P_i, $1 \leq i \leq 4$, such that $h_i > h_0$ and $h_{i+4} > h_0$, where h_k is the height (or depth) component for the point P_k;

(b) For the i of part (a) , there is a j, $i< j <i+4$, such that $h_{j-1} > h_j$ and $h_{j+1} > h_j$;

(c) There is a number k, $i+4 < k < i+8$, such that $h_{k-1} > h_k$ and $h_{k+1} > h_k$

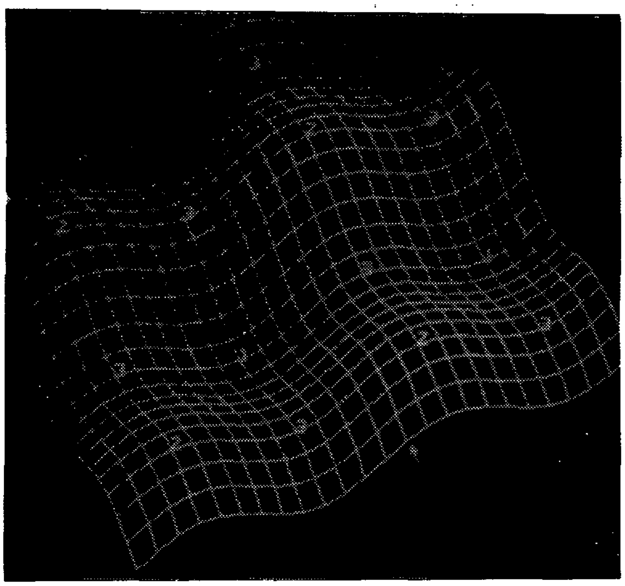

Figure 8 - This is a screen photo of the same surface as figure 7 with the extremal points being determined by the local surface fitting method.

A point is labeled a ridge point if the inequalities of (a), (b), and (c) can be reversed. Using the above notation, we consider the point P_0 and its neighbors. Then we say P_0 is a ridge point if:

(d) There is a point P_i, $1 \leq i \leq 4$, such that $h_i < h_0$ and $h_{i+4} < h_0$;

(e) For the i of part (d) , there is a j, $i < j <i+4$, such that $h_{j-1} < h_j$ and $h_{j+1} < h_j$;

(f) There is a number k, $i+4 < k < i+8$, such that $h_{k-1} < h_k$ and $h_{k+1} < h_k$.

5. IDENTIFICATION OF THE EPTN: LOCAL SURFACE FITTING METHOD

This approach has been somewhat motivated by the work of Haralick [14,15]. His procedure to label a pixel (a grid location) was based on an analysis of a cubic patch defined with the pixel as its center. We opted for a quadratic patch and performed an analysis to determine the elevation maxima points, elevation minima points, and passes (saddle points). This approach is based on a distance weighted, least-squares approximation technique [21]. This procedure seems suitable since the data is of non-mathematical origin for which numerical approximations are not usually appropriate. We consider a point P_i and its eight neighboring points or twenty-four neighboring points. With the total of nine (or twenty-five) points we fit a least-square quadratic surface to these points. That is, our aim is to find a polynomial P(x,y), which is of degree two:

$$P(x,y) = C_{00} + C_{10}x + C_{01}y + C_{20}x^2 + C_{11}xy + C_{02}y^2 \qquad (5.1)$$

which will be as accurate a fit as possible, in the usual least-squares sense, to the grid data points (x_j,y_j) with associated z_j value.

The steps to be taken are:

(1) At a given point of the mesh (not a mesh boundary point) determine a quadratic least-squares fit for the point and its neighbors for the polynomial equation (5.1). If (x_i,y_i) is the grid point being considered, then the quadratic patch to be considered in the next step is the quadratic defined by the least-square procedure with domain

$$x_i - h/2 \le x \le x_i + h/2 \qquad (5.2a)$$

$$y_i - h/2 \le y \le y_i + h/2 \qquad (5.2b)$$

where h is the distance between adjacent grid mesh points.

(2) Determine the critical points for the quadratic of (5.1). Note there is but one critical point for this equation and it is determined by solving two linear equations for the zero. The two equations are:

$$dP(x,y)/dx = C_{10}+2C_{20}x+C_{11}y \ ;$$

$$dP(x,y)/dy = C_{01}+C_{11}x+2C_{02}y \ .$$

The coordinates of their common zero are:

$$x_0 = (C_{01}C_{11}-2C_{02}C_{10})/(4C_{20}C_{02}-C_{11}^2)$$

$$y_0 = (C_{10}C_{11}-2C_{20}C_{01})/(4C_{20}C_{02}-C_{11}^2).$$

(3) Determine if the critical point lies in the domain given in (5.1). That is, determine if x_0 and y_0 satisfy equations (5.2a) and (5.2b).

(4) If the critical point is in the domain, then the determination of whether it is a local maxima, minima, or saddle point is made. If $(4C_{20}C_{02}-C_{11}^2) < 0.0$, then we have a saddle point. If $(4C_{20}C_{02}-C_{11}^2) > 0.0$ and $C_{20} > 0.0$, then we have a local minimum. If $(4C_{20}C_{02}-C_{11}^2) > 0.0$ and $C_{20} < 0.0$, then we have a local maximum.

The approaches of Sections 4 and 5 have been compared on test data. Either of the procedures can determine the elevation minima and elevation maxima with ease (figures 7, 8 and 9). The method of Section 5 has been found to capture the pass (saddle point) locations much better than the above local difference method of Section 4 (figures 7 and 8).

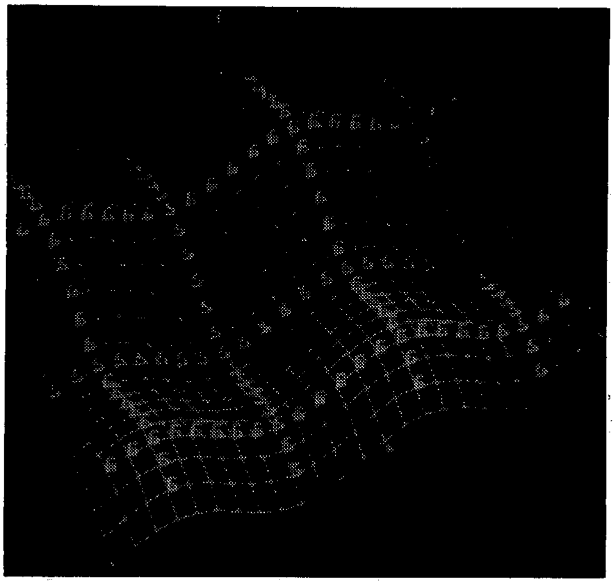

Figure 9 - This is a screen photo of the same surface as figure 7 with the ravine and ridge points being determined by the neighbor differencing method and denoted by the numbers 5 and 6 respectively.

Additional related work yet to be completed: If we wish to determine the curve (that passes through a pass) for the ridge connecting adjacent elevation maxima or determine the curve (that passes through a pass) for the ravine connecting adjacent elevation minima, it would seem reasonable to attempt an optimization approach. Since we are seeking to define a continuous curve that passes through a particular point one could:

(1) determine the passes (saddle points) by the above least square procedure of Section 5 (because of its reliability);

(2) use each pass (saddle) point as the start point for an optimization procedure where the optimization surface is locally defined by the above least square patch fit.

This approach has been very successful with the simplicial search optimization method being applied on a single patch. We have not yet implemented this procedure using a multi least square patch fit surface. Lastly we point out that this method, or any of a number of optimization methods could be applied easily (with good results) if the surface was approximated by a piecewise bicubic Hermite interpolant.

Another problem to be explored is the minimum path length between two points on the bottom surface while surface contact is maintained. That is two distinct points on a curved surface can be connected by many different paths on the surface; each path has, in general, a different length and the minimum one is the one of interest. Optimizing motion planning across a curved surface for a semi-autonomous or autonomous underwater vehicle (that is the vehicle travels close to the bottom surface) is an example of a practical application. In terms of this problem we can state our special interest is in the pass locations. Given the location of an underwater vehicle that is hugging the ocean floor, and below the pass points, what is the best path to take that minimizes the distance traveled from the present position to the nearest pass.

6. IDENTIFICATION OF THE EPTN: DESCENDING/ASCENDING COUNT METHOD

The method described next is based on a literal application of the "rainfall" model: For any pixel, water will flow from it in the (negative) direction of the gradient. For a discretized grid, in which each neighbor is treated as having eight neighbors, this will be in the direction of the neighbor whose elevation value is least among the eight.

From the viewpoint of this least or minimal valued neighboring pixel, it may be the minimal neighbor for others as well. We call the number of times this occurs the *descending minimal neighbor count* for this pixel. It is the number of times this pixel serves as the minimal pixel for its eight neighbors, and will vary between zero and eight: Zero if it does not serve as a minimum for any neighbor, (the most frequent case) and a maximum of eight at an elevation minimum. Intermediate values (three through seven) indicate ravine like pixels in which water flows into them from three or more adjacent pixels.

If this count is made on an entire grid of elevation pixels and the resulting grid of values thresholded to include only those values in excess of two, the resulting grid should indicate those pixels forming ravines. Figure 2, previously noted in Section 2, depicts the result of performing this operation for the synthetic image of figure 1. The direction of the arrows is determined by the direction of the minimal neighbor.

The analogous count obtained by finding the maximal valued neighbor we call the *ascending maximal neighbor count.* Values greater than three correspond to ridges. Figure 3, also previously noted in Section 2, depicts this for figure 1.

Note that this method does not provide a means for labeling passes, but of course may be combined with a method for doing this.

In order to turn the above idea into an efficient algorithm, the following should be noted. In passing through the raster of pixels, rather than finding the number of pixels for which the current one is the minimum, the following is more efficient:

(1) Attach to each pixel an initial count of zero.

(2) In raster order, for each pixel p_0 (skipping border pixels), determine among its eight neighbors, the neighbor p_m with the smallest value.

(3) If p_m is higher than p_0, then p_0 is a local minimum, and there is nothing to do.

(4) Otherwise, p_m is lower than p_0 and hence add one to p_m's count.

7. IDENTIFICATION OF THE EPTN: ACCUMULATIVE GRADIENT METHODS

The methods described here result from pushing the "rainfall' model even further: For each pixel we count the number of pixels "above" it whose "runoff", determined by always following the minimal neighbor, will pass through it. This value is a vector whose direction at this pixel is in the direction of its minimal neighbor. Its magnitude is proportional to the accumulative descending minimal neighbor count. When adding to the count of the minimal neighbor, we add in the accumulative descending minimal neighbor count (which must have already been calculated), of the pixel for which this pixel is the minimal neighbor, plus one. We call the value calculated in this way the *descending accumulative gradient.*

Hence, at a local elevation minimum, the accumulative gradient will be the number of pixels in the "watershed" area for which it is the collecting minimum, and serves to rank order the collection of minima.

More precisely, let g_0 be the accumulative gradient for p_0 and g_i, i=1,8, the yet to be calculated accumulative gradient for its eight neighbors. Then if p_i is the minimal neighbor of p_0, denoted by $p_i \leftarrow p_0$, p_0's contribution to g_i is g_0+1. The total accumulative gradient is then

$$g_i = \sum_{p_i \leftarrow p_0} (g_0+1)$$

where the summation is performed by letting each of the neighbors of p_i take the role of p_0.

While the above serves to define the accumulative gradient, it does not provide a method of calculating it: In order to calculate the accumulative gradient at a point p_0 it potentially must have already been calculated at each of its neighbors, so that when the minimal one is found, its accumulative gradient can be added in to p_0's. While it is probably possible to use an iterative "relaxation" method to achieve the same result, we have resorted to sorting (which requires $n\log n$ steps), into descending order the entire grid of elevation values. This insures that when a point is processed, the accumulative gradient of all points "above" it in the same watershed have made their contribution, and is thus available to "flow" to its minimal neighbor.

Similiar remarks as those given for the accumulative count algorithm apply here: Rather than find all the pixels that flow to the pixel under scrutiny, it is more efficient to just find the minimal neighbor and add into its accumulative gradient the contribution made by this pixel.

Clearly, the inverse operation of accumulating the gradient in the reverse direction is possible by moving in the direction of the maximal neighbor rather than the minimal neighbor. We call the resulting vector field the *ascending accumulative gradient.*

While these accumulative gradients serve to rank order the minima and maxima, they potentially have other uses as well. One such application is motivated by the need to represent the ability of the terrain to provide protection from sonar detecting devices. This is accomplished by accumulating the gradient in a non-linear way.

For example, if it is desired to distinguish between two comparable ravines on the basis of their respective width to height ratios, the accumulative gradient of the steeper sloped ravine can be "enhanced" by adding in a squared term representing this steepness. If p_i is the minimal neighbor for p_0, let $h = p_0 - p_i$. Then the expression for g_i is

$$g_i = \sum_{p_i \leftarrow p_0} [g_0 + (1+\frac{h}{d})^2]$$

where d is the distance between grid points. Note that $\frac{h}{d}$ is the tangent of the angle of "grade".

8. APPLICATION TO LAKE WINNIPESAUKEE BATHYMETRY

To meet the needs of the MAUV project [1], approximatly 13,000 irregularly spaced soundings, spread over a 1,700 meter by 1,700 meter region of Lake Winnipesaukee, New Hampshire and containing Diamond Island and the tip of Rattlesnake Island, were acquired by Alan Bieber & Associates of Lyme, Connecticut. Approximatly 11,000 of these soundings, together with 140 "soundings" of depth zero outlining the island boundaries, were used in the triangulation procedure described in Section 3 to reconstruct a 1,632 meter by 1632 meter region of the lake bottom. This reconstruction is in the form of a 816 by 816 grid of interpolated points with a two meter intergrid distance. Figure 10 depicts a perspective view of the resulting surface.

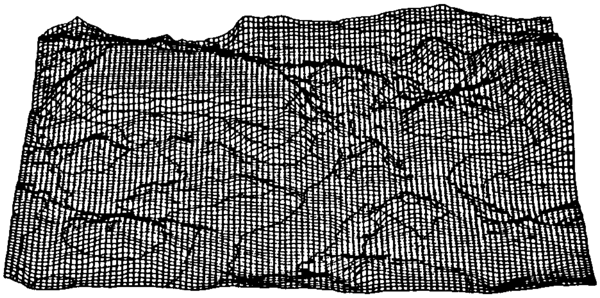

Figure 10 - The reconstruction of a 1632 meter by 1632 meter region of Lake Winnipesaukee based on approximately 11,000 irregular spaced "soundings". The height is scaled up by a factor of 5.

In order to overcome the problem of "noisy" local surface structure, the grid was "smoothed" by iteratively passing a 3 by 3 Gaussian convolution over it whose "kernel" was

$$\begin{bmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{bmatrix}$$

The result after four iterations is depicted in figure 11 as a perspective view and in figure 12 as depth contours.

The desending/ascending count method for extracting the ridges and ravines was applied to the unsmoothed as well as smoothed reconstructed surface. By observing the number of isolated fragments and the lengths of the resulting ridges and ravines, an optimal number of iterations appears to be five. The resulting ridges and ravines, extracted from the smoothed reconstruction, are shown in figures 13 and 14 respectively.

The methods of Sections 4 and 5 have been applied to unsmoothed as well as smoothed reconstructed surfaces.

As can be seen (figure 15), the ridge and ravine points can be easily determined when the lakebed points originate from a smooth reconstructed surface. For the reconstruction that was not

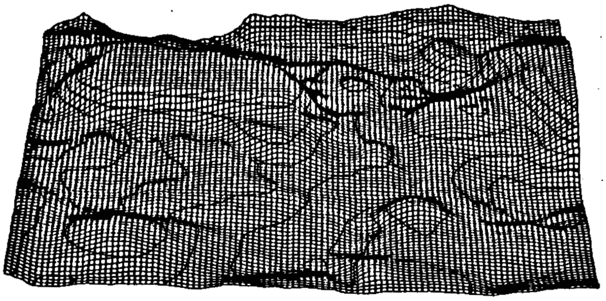

Figure 11 - The results of "smoothing" the reconstruction depicted in figure 10. The smoothed version is used only for the purpose of extracting topographic features.

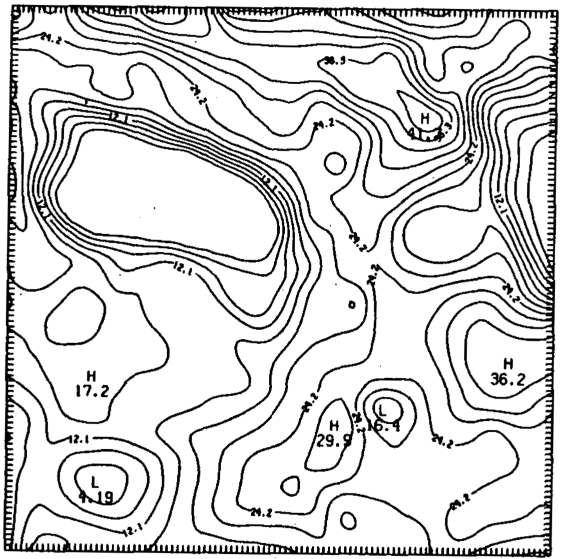

Figure 12 - A contour plot of the "smoothed" reconstruction depicted in figure 10. Depths are in meters. Note that "L" refers to a numerical minimum (elevation maximum), while "H" refers to an elevation minimum.

smoothed, the results are less favorable (figure 16). We prefer the approach of Section 5 over that of Section 4 for computing saddle points if computer time is not a consideration. For either of these procedures, it is preferable to have the data processed so as to have it appear to be from a smooth surface.

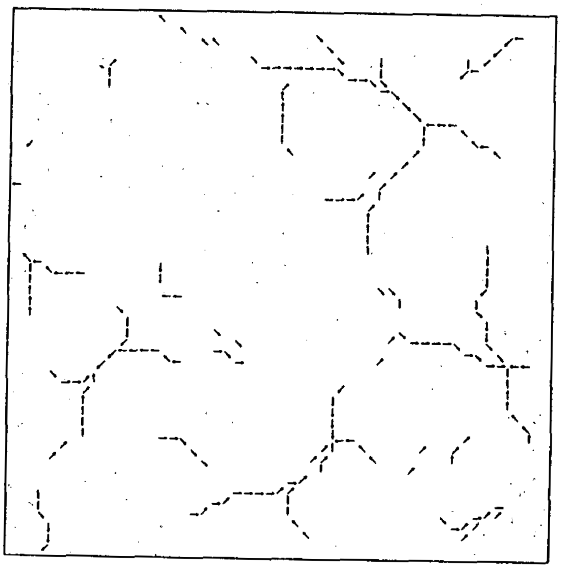

Figure 13 - The ravines of the recostructed area of Lake Winnipesaukee as identified by the descending count method of Section 6.

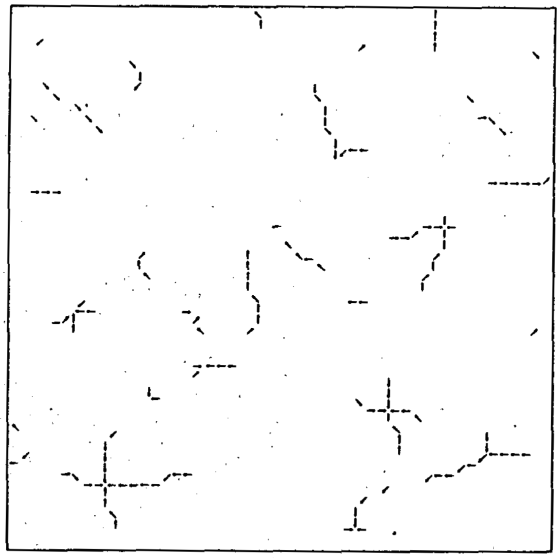

Figure 14 - The ridges of the reconstructed area of Lake Winnipesaukee as identified by the ascending count method of Section 6.

9. SUMMARY

We have given an informal description of ongoing work concerned with the use of bathymetric data for storage aboard an AUV for the purpose of supplying information relevant to algorithmic global navigation. The feasibility of identifying the EPTN has been demonstrated on actual data. The extraction of the EPTN from a grid and its conversion into an encoding of the resultant graph structures, as well as their compilation into data structures appropriate for their use by a planning algorithm has yet to be accomplished, but is relatively straight forward.

The usefulness of the EPTN for solving the route planning and self localization problems for an AUV has yet to be demonstrated, but we believe that

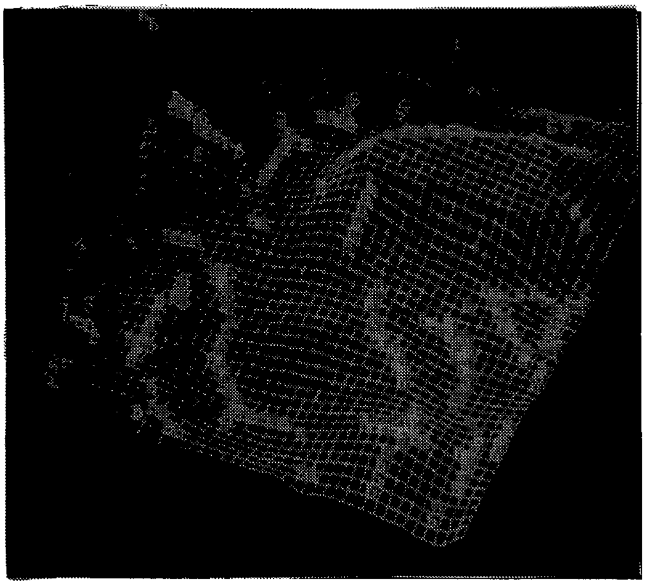

Figure 15 - The smooth reconstruction of figure 11, with the ridge and ravine points labeled by the neighbor differencing method.

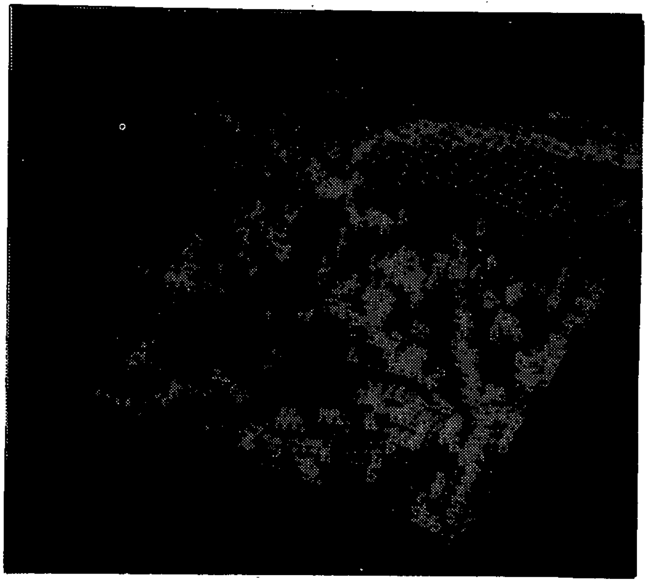

Figure 16 - The non-smooth reconstruction of figure 10, with the ridge and ravine points labeled by the neighbor differencing method.

as AUV control algorithms become more sophisticated, these methods will become more important.

ACKNOWLEDGMENTS

The authors would like to thank Javier Bernal for his help and the use of his program for doing the triangulation and interpolation of the Lake Winnipesaukee data. Javier, as well as Hoosh Abrishamian of the Robot Systems Division, also suggested a number of corrections for an earlier draft. Christoph Witzgall, also of the National Bureau of Standards, is responsible for suggesting the idea for the accumulative gradient. We also wish to thank members of the Robot Systems Division for many stimulating discussions concerning the needs of the MAUV project.

REFERENCES

[1] Albus, James and Blidberg, D. Richard, "A Control System Architecture for Multiple Autonomous Undersea Vehicles," *Proceedings of the Fifth International Symposium on Unmanned Untethered Submersible Technology,* Merrimack, New Hampshire, June 22-24, 1987.

[2] Akima, Hiroshi, "On Estimating Partial Derivatives for Bivariate Interpolation of Scattered Data," *Rocky Mountain Journal of Mathematics,* Vol. 14, No. 1, Winter 1984, pp. 41-52.

[3] Barnhill, R. E., "Representation and Approximation of Surfaces," *Mathematical Software III,* J. R. Rice (ed.), Academic Press, NY, 1977, pp. 69-120.

[4] ————, and Little, F. F., "Three- and Four-Dimensional Surfaces," *Rocky Mountain Journal of Mathematics,* Vol. 14, No. 1, Winter 1984, pp. 77-102.

[5] Bernal, J. and Howe, S. E., "Expected $O(N)$ and $O(N^{\frac{4}{3}})$ Algorithms for Constructing Voronoi Diagrams in Two and Three Dimensions,". Submitted to *SIAM Journal on Computing.*

[6] Cayley, Arthur, "On Contour and Slope Lines," *London, Edinburgh, and Dublin Philosophical Mag. and J. Sci.,* Series 4, Vol. 18, Dec. 1859, p. 264.

[7] De Floriani, L., Falcidieno, B. and Pienovi, C., "Delaunay-based Representation of Surfaces Defined over Arbitrarily Shaped Domains," *Computer Vision, Graphics, and Image Processing* 32, 1985, pp. 127-140.

[8] do Carmo, Manfredo P., *Differential Geometry of Curves and Surfaces,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976, pp. 134-216.

[9] Cline, A. K. and Renka, R. L., "A Storage-Efficient Method for Construction of a Thiessen Triangulation," *Rocky Mountain Journal of Mathematics,* Vol. 14, No. 1, Winter 1984, pp. 119-139.

[10] ————, *FITPACK: A Software Package for Curve and Surface Fitting Employing Splines Under Tension*, Department of Computer Sciences, The University of Texas at Austin, to be published.

[11] Foley, Thomas A., "Three Stage Interpolation to Scattered Data," *Rocky Mountain Journal of Mathematics*, Vol. 14, No. 1, Winter 1984, pp. 141-149.

[12] Franke, R., "A Critical Comparison of some Methods for interpolation of Scattered Data," Naval Postgraduate School Technical Report NPS-53-79-003, 1979.

[13] ————, "Scattered Data Interpolation: Tests of Some Methods," *Mathematics of Computation*, Vol. 38, No. 157, January 1982, pp. 181-200.

[14] Haralick, R.M., "Ridge and Valleys On Digital Images," *Computer Vision, Graphics, and Image Processing* 22, 1983, pp. 28-38.

[15] Haralick, R. M., Watson, L. T. and Laffey, T. J., "The Topographic Primal Sketch," *The International Journal of Robotics Research*, Vol. 2, No. 1, spring 1983, pp. 50-72.

[16] Hilbert, D. and Cohn-Vossen, S., *Geometry and the Imagination*, Chelsea Publishing Co., New York, 1952, pp. 171-235.

[17] Ichikawa, Tadao and Chang, Shi-Kuo, eds., *Proceedings of the 1984 IEEE computer Society Workshop on Visual Languages*, Hiroshima, Japan, December 6-8, 1984.

[18] Iyengar, S. S., Jorgensen, C. C., Rao, S. V. N. and Weisbin, C. R., "Robot Navigation algorithms Using Learned Spatial Graphs," *Robotica*, Vol. 4, 1986, pp. 93-100

[19] Krogh, B. H., "A Generalized Potential Field Approach to Obstacle Avoidance Control," *Robotics International Robotics Research Conference*, Bethlehem, Pa., August 1984.

[20] Maxwell, James Clerk, "On Hills and Dales," *London, Edinburgh, and Dublin Philosophical Mag. and J. Sci.*, Series 4, Vol. 40, Dec. 1870, pp. 421-425.

[21] McLain, D.H., "Drawing Contours from Arbitrary Data Points," *The Computer Journal*, Vol. 17, No. 4, November 1974.

[22] Nackman, Lee R., "Two Dimensional Critical Point Configuration Graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No 4, July 1984, pp. 442-449.

[23] Preparata, Franco P. and Shamos, Michael Ian, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985, p. 198.

[24] Rao, S. V. Nageswara, Iyengar, S. S., Jorgensen, C. C. and Weisbin, C. R., "Concurrent Algorithms for Autonomous Robot Navigation in an Unexplored Terrain," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, Calif., April 7-10, 1986, pp. 1137-1144.

[25] O'Callaghan, John F., "The Extraction of Drainage Networks from Digital Elevation Data," *Computer Vision, Graphics and Image Processing*, **28**, 1984, pp. 323-344.

[26] Peucker, T.K. and Douglas, D. H., " Detection of Surface-Specific Points by Local Parrallel Processing of Discrete Terrain Elevation Data," *Computer Graphics and Image Processing* (1974), 4, pp. 375-387.

[27] Pfaltz, John L., "A Graph Grammar that Describes the Set of Two-Dimensional Surface Networks," in *Graph-Grammars and Their Application to Computer Science and Biology*, Proceedings of a Workshop Held in Bad Honnef, West Germany, October, 1978, edited by G. Goos and J. Hartmanis, pp. 379-388.

[28] Pugh, George and Krupp, Joseph, "The Control of Autonomous Underwater Vehicles Through a Hierarchical Structure of Value Priorities," *Proceedings of the Fifth International Symposium on Unmanned Untethered Submersible Technology*, Merrimack, New Hampshire, June 22-24, 1987.

[29] Renka, R. J., and Cline, A. K., "A Triangle-Based C^1 Interpolation Method," *Rocky Mountain Journal of Mathematics*, Vol. 14, No. 1, Winter 1984, pp. 223-237.

[30] ————, "Algorithm 624: Triangulation and Interpolation at Arbitrarily Distributed Points in the Plane," *ACM Transactions on Mathematical software*, Vol. 10, No. 4, December 1984, pp. 440-442.

[31] Stead, Sarah E., "Estimation of Gradients from Scattered Data," *Rocky Mountain Journal of Mathematics*, Vol. 14, No. 1, Winter 1984, pp. 265-279.

[32] Tyrén, Carl, "Magnetic Terrain Navigation for Underwater Vehicles," *Proceedings of the Fifth International Symposium on Unmanned Untethered Submersible Technology*, Merrimack, New Hampshire, June 22-24, 1987.