Annals of Operations Research, 18 (1989) 175–186

COVERAGE PROBLEMS AND VISIBILITY REGIONS ON TOPOGRAPHIC SURFACES

Michael F. GOODCHILD and Jay LEE

Department of Geography, University of Western Ontario, London, Ontario, Canada N6A 5C2

Abstract

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. We consider the problems of locating the minimum number of viewpoints to see the entire surface, and of locating a fixed number of viewpoints to maximize the area visible, and possible extensions. We discuss alternative methods of representing the surface in digital form, and adopt a TIN or triangulated irregular network as the most suitable data structure. The space is tesselated into a network of irregular triangles whose vertices have known elevations and whose edges join vertices which are Thiessen neighbours, and the surface is represented in each one by a plane. Visibility is approximated as a property of each triangle: a triangle is defined as visible from a point if all of its edges are fully visible. We present algorithms for determination of visibility, and thus reduce the problems to variants of the location set covering and maximal set covering problems. We examine the performance of a variety of heuristics.

1. Introduction

The problems considered in this paper have two simple motivations, and we suspect that others could be found without much difficulty. Consider first the problem of locating fire towers in a rugged landscape so that it is possible to see every part of the landscape from at least one tower. Alternatively, given a fixed number of fire towers to be located, one might wish to maximize the area seen on the assumption that no locations could be found which would provide complete surveillance. Similar problems might arise in the context of security surveillance: for example, it might be necessary to locate a minimum number of observation posts from which watch could be kept over a sensitive military area. In another version, not considered in this paper, one might wish to locate the minimum number of points along a given linear feature, for example a fence or wall, such that the entire linear feature could be seen from at least one, or perhaps at least two points. In all of these examples we assume an arbitrarily rugged surface, and the ability to locate anywhere on the surface. We also assume initially that the viewpoints are located on the surface itself; towers are either of height zero, or not sufficiently high to affect the area seen within the spatial resolution of the analysis. Later in the paper we discuss problems in which tower height affects area seen, and in which the objective is to achieve maximum visibility coverage with minimum expenditure on tower construction.

The first section of the paper considers three alternative approaches to the representation or modelling of the given topography. In reducing an arbitrary surface to digital form it is necessary to consider both the accuracy of the representation, since any model must in principle approximate the real surface, and also the model's suitability for use in solving the particular problem of visibility coverage. The second section discusses the algorithm used to generate each point's viewshed or visible area. It is necessary that the algorithm be as efficient as possible in view of the potential volumes of data to be processed in problems of realistic size. Given the ability to compute each point's visible area, the third section discusses the structure of the optimization problem, and compares it to two well-known set covering problems. The fourth section of the paper reviews possible heuristics and examines their performance on sample problems, and the fifth section discusses possible extensions.

1. Representations of surfaces

There are three commonly used approaches to the digital representation of an arbitrary surface (see for example [2]). We refer to this process of representation as the construction of a digital elevation model or DEM. Since the commonest representation of topography on a paper map is by contours, or lines connecting points of equal elevation, we 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. 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.

The second alternative, the representation of the surface by a regular square grid of sample elevations, gives a uniform intensity of sampling, and is therefore frequently used in practice. For example, elevations to the nearest metre are available for the entire U.S. at a sampling interval of 30 m. For our purposes it would be 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. For this reason a uniform sampling density is inefficient compared to a design which 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: for example there is no uniquely appropriate way of fitting a plane in each grid cell.

This leads logically to the third alternative, known as the TIN or triangulated irregular network [9]. In this model the space is divided into a set of irregular triangles with shared edges, and the surface is modelled by the triangles as if they were mosaic tiles. Each edge is shared by exactly two triangles, with the exception of those whose edges form the outer boundary of the network. Each vertex is shared by at least three triangles. In the simplest version, which we have adopted here, the surface is assumed to be planar within each triangle: this would not be possible with tiles of more than three vertices. In more elaborate versions of the TIN model the surface within each triangular tile is defined by a polynomial function of the x and y coordinates, with the constraint that the surface be zero-, first- and second-order continuous across each edge.

The construction of a TIN model begins with the selection of an irregularly located sample of point elevations. For maximum economy the points will be more densely sampled in areas of rugged terrain. By using pits, peaks and other critical surface points on ridges and in valleys it is possible to achieve an adequate representation of a surface with far fewer sample points than with either of the previously discussed alternatives. These points then form the vertices of the TIN. One unambiguous and frequently used procedure for defining the edges of the TIN is to connect all pairs of points which are Thiessen neighbours. The Thiessen region (also known as the Voronoi or Dirichlet region) 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 Thiessen neighbours if their respective Thiessen regions share an edge. The result is the Delaunay triangulation of the sample points: each triangle has the property that the circle defined by its vertices contains no other point. The outer boundary of the Delaunay network is the convex hull of the point set. The relationships between Delaunay triangles, the Thiessen regions which are their topological duals, and the points which generate them are illustrated in fig. 1.

A number of algorithms for Delaunay triangulation have been published [1,4,6,8,11], any of which would be appropriate in this context. Our approach is not the most efficient, but is adequate for our purposes and relatively easy to program. It begins by searching the set for the closest pair of points, i and j, which must be Thiessen 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 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 presorting the points in x it is possible to process n points in the worst case in $O(n^2)$ time.

In summary, then, the TIN representation used in this paper is constructed by first determining elevations at an irregular sample of points, including peaks, pits

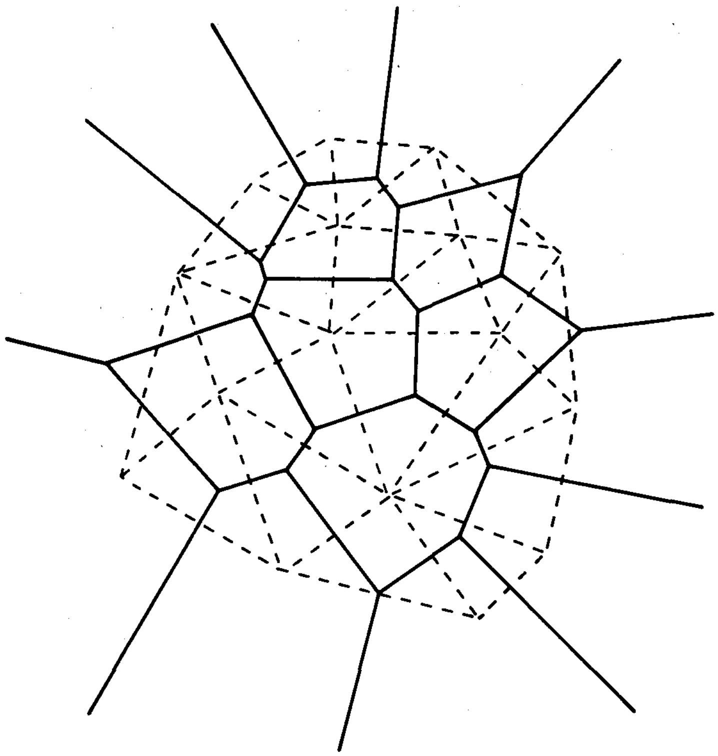

Fig. 1. The Thiessen regions (solid lines) and Delaunay triangles (dashed) of a point set.

and with heavy sampling on ridges and in valleys; second, by connecting these points as vertices into a network of triangles, using the Delaunay triangulation; and third, by assuming the surface within each triangle to be modelled by the plane defined by its vertices.

The TIN model is an efficient digital representation of an arbitrary surface, but it is also particularly suited to the problem of visibility coverage. To make the coverage problem tractable, it will be necessary to limit the search to a finite set of locations, which should include all of the peaks of the surface. Thus the process of selection of TIN vertices is efficient both as a means of representing the surface and as a method of selecting a discrete set of locations to be searched for visibility coverage.

Two vertices of the TIN are intervisible if a straight line between them lies entirely on or above the surface. We define the visible area of a vertex as a subset of the Delaunay triangles rather than as a subset of the vertices. There is no requirement that the subset be connected. However it will necessarily contain the viewpoint since we define the triangles adjacent to the viewpoint as visible.

For a triangle to be fully visible, it must be true that any line drawn from a point within the triangle to the viewpoint vertex must lie entirely on or above the

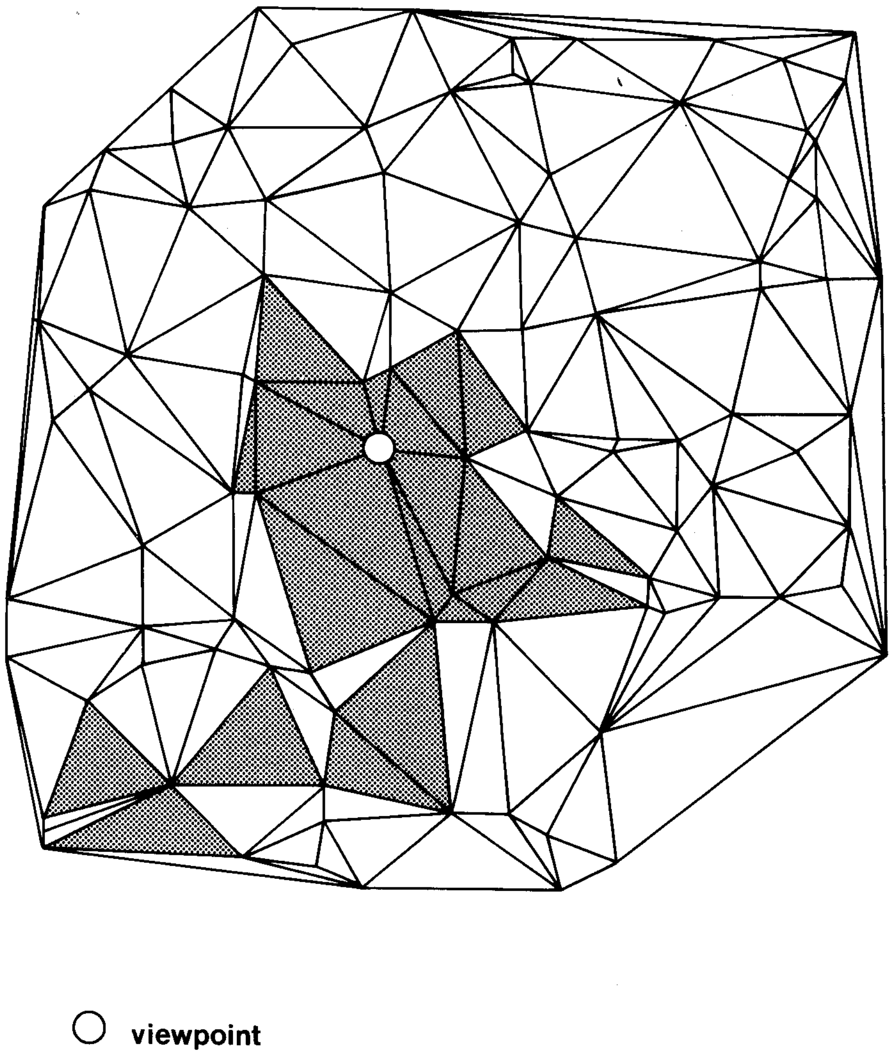

Fig. 2. Visible set of triangles from a single vertex on a 100-vertex simulated TIN.

surface. If we regard the surface as a function $z(x, y)$, then because of the obvious requirement that the function be everywhere continuous and single-valued, it follows that a triangle is fully visible if and only if its three edges are fully visible. An example of the set of triangles fully visible from a single vertex on a sample TIN of 100 vertices is shown in fig. 2.

2. The visibility algorithm

The visibility problem is clearly closely related to the detection of hidden lines, which is a well-known problem in computer graphics. The specific problem of determining the set of triangles of a TIN which are visible from a TIN vertex has been discussed by de Floriani et al. [5], who also noted the possibility of using their approach to solve the visibility coverage problems being discussed in this paper.

The algorithm begins with the selection of a TIN vertex as viewpoint. We assume for the present that the elevation of the viewpoint is the same as the elevation of the vertex. Consider the set of vertices each connected by a single edge to the viewpoint. These vertices form a polygon, all of whose edges are visible from the viewpoint, and all of the triangles contained within the polygon and sharing the viewpoint vertex are also visible. We define the *curtain* as an imaginary vertical fence constructed around the boundary of this polygon, with its *foot* following the horizontal projection of the polygon. The top or *horizon* of the curtain passes through each of the vertices of the polygon, and follows the connecting edges between those vertices. It follows that a point on the far side of the curtain from the viewpoint is invisible if a line drawn from it to the viewpoint passes through the curtain below the horizon.

The algorithm proceeds by keeping the curtain vertical, but moving its foot outwards from the viewpoint, one triangle at a time, with the constraint that a line drawn from any point on the curtain to the viewpoint must not intersect another point on the curtain. Consider the horizontal projection of an edge ij forming part of the foot of the current curtain. It will be shared by two triangles, one inside the curtain and one outside. Denote the latter by ijk. Then the foot of the curtain may be moved from the horizontal projection of edge ij to the projections of ik and kj if and only if the horizontal projection of a line drawn from k to the viewpoint intersects the horizontal projection of the edge ij.

As the curtain is moved outwards, its upper edge is adjusted to form the new horizon visible from the viewpoint. This is computed first by reprojecting the current horizon to allow for the new position of the foot, and second by a union operation with the horizon formed by the new edges ik and kj. The horizon is maintained as an ordered set of (x, y, z) triples which are assumed to be connected by straight lines, and in general the length of the list grows as the curtain is moved outward. The algorithm terminates when the curtain reaches the outer boundary or convex hull of the TIN.

A triangle can be identified as visible or invisible immediately the foot of the curtain has been moved over it. It is visible if and only if vertices i and j occured in sequence in the old horizon with no intervening triples, and if vertices i, k and j similarly occur in sequence as triples in the new horizon, with no intervening triples. This ensures that all three edges are fully visible from the viewpoint.

The algorithm is executed fully for every vertex, as there appear to be no simple theorems which would allow information about one vertex's visible area to be derived from another. With n vertices, the number of triangles is $O(n)$, and this determines the number of steps in the algorithm. However the number of triples on the horizon is less predictable. Let us suppose, as a provisional guess, that in the worst case the number of points which must be reprojected in calculating the new horizon in each step is $O(n)$, so the computational complexity of the identification of one visible area is $O(n^2)$. To find visible areas for all vertices thus requires $O(n^3)$ time in the worst case.

3. The coverage problem

The results of calculating the set of triangles visible from each of the TIN vertices can be expressed as a rectangular matrix with each row representing a viewpoint vertex and each column a triangle. Each element x_{ij} of the matrix is set to 1 if the triangle i is visible from the vertex j and 0 otherwise. Each triangle can be weighted by its area if the objective is concerned with the area visible, rather than with the number of triangles visible.

It is possible that rows can be eliminated as potential viewpoints because they are dominated by other vertices. The necessary condition for the dominance of vertex i by vertex k is simply:

$$x_{kj} \geqslant x_{ij} \quad \text{for all } j \tag{1}$$

Unfortunately it appears that dominance is unlikely in practice. It requires situations in which an observer is able to move on a landscape without bringing new areas into view: in practice moves almost always result in a changing field of view, with the addition of some areas and the deletion of others. The TIN representation ensures that the field of view of each vertex is almost certainly unique, and that dominance almost never exists. The search for coverage must therefore consider all vertices.

The formalization of the visibility coverage problems follows the standard form of the location set-covering [12] and maximum covering location [3] problems respectively. Let the presence of a facility at vertex i be denoted by y_i, which is 1 if a facility is present and 0 otherwise. To minimize the number of facilities required to see the entire surface:

$$\text{Minimize} \quad \sum_i y_i \tag{2}$$

$$\text{subject to} \quad y_i = \{0, 1\} \quad \text{for all } i \tag{3}$$

$$\sum_i x_{ij} y_i \geqslant 1 \quad \text{for all } j. \tag{4}$$

To maximize the area covered by a given number of facilities p, let the area of triangle j be denoted by A_j.

$$\text{Maximize} \quad \sum_j A_j \min\left(1, \sum_i x_{ij} y_i\right) \tag{5}$$

$$\text{subject to} \quad y_i = \{0, 1\} \quad \text{for all } i \tag{6}$$

$$\sum_i y_i = p. \tag{7}$$

4. Heuristics

A number of data sets of different sizes were simulated by generating (x, y, z) point elevation triples, all three coordinates being independent and uniformly distributed in the range (0, 1). The visibility algorithm was used to calculate coverage matrices as input to a variety of heuristics.

The dominance condition was found to be very rare, which is understandable given the independent elevations generated for neighbouring TIN vertices. A simulation of 30 points produced one case of dominance, and a 100 point surface was found to have none. The dominance test was subsequently deleted from the analysis.

Three broad classes of heuristics were tested: a greedy add (GA), in which viewpoints were added one at a time, on each step selecting that vertex which maximized some conveniently computed parameter; a stingy drop (SD), with all vertices selected initially and then dropped one at a time based on the minimum value of a parameter; and a greedy add with swaps (GAS), where an attempt was made to improve the objective function by exchanging each vertex in the solution with one not in the solution after the addition of each new vertex.

To solve the location set covering problem each heuristic was continued until the solution just covered the set. The maximal covering problem was approached by running the heuristic until the prescribed number of facilities had been located. Additions and deletions were driven both by change in total area visible, and by change in total number of triangles visible.

The combinations of three basic heuristics and two driving parameters generates a total of 6 methods. These were tested on 10 sample problems of 30 vertices, and the results are shown in table 1. The most successful heuristic was found to be GA; swapping never produced improvement in performance. Of the driving parameters the total number of triangles visible was never outperformed, whereas total area visible was outperformed in one case. We should emphasize, however, that the use of simulated TINs with independently generated elevations may be unrepresentative of real conditions, where substantial amounts of spatial autocorrelation are to be expected between vertex elevations.

Although these conclusions appear to be robust for the random TINs used in this analysis, it would be difficult to obtain general results for the performance of

Table 1
Number of times each heuristic was outperformed by another in solving 10 sample problems of 30 vertices

Heuristic	*Driven by area visible*	*Driven by triangles visible*
Greedy add	1	0
Greedy add with swaps	1	0
Stingy drop	9	8

heuristics. This would require first that analyses be made over a full range of types of topography. Although suitable stochastic models of spatially autocorrelated terrain exist [7], they are clearly appropriate only for a limited range of terrain types. But a more serious problem is the difficulty of removing the effects of the selection of TIN vertices on the results. A heuristic which gives good performance when TIN vertices are restricted to pits and peaks of the surface may be poor when vertices are allocated according to a random Poisson process, for example.

5. Extensions

The nature of the visibility coverage problem raises the possibility of driving heuristics with distinctive parameters. One might argue that vertex elevation plays a large part in intuitive solutions to these problems, and should therefore be successful as a selection criterion. The column total of the x 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. More specifically, let W_j be the visibility of a triangle, defined as follows:

$$W_j = \sum_i x_{ij}. \tag{8}$$

Then the heuristic-driving parameter V_i is obtained as:

$$V_i = \sum_j (B - W_j) x_{ij} \tag{9}$$

where B is a number at least as large as the largest W, $B \geqslant W_j$ for all j.

Thus far the discussion has been limited to the case of viewpoints located on the topographic surface. Visibility coverage can clearly be increased by raising the viewpoint; since the surface $z(x, y)$ is assumed to be single valued, there must exist some minimum height at each vertex from which it is possible to see the entire surface.

Let w_{ij} denote the minimum height above the vertex i from which an observer can see triangle j; the triangle will be visible from all elevations above w_{ij} and invisible at all lower elevations. 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 which is used as a viewpoint but has no tower will have $H_i = 0$, and will presumably incur positive cost. A vertex which 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

$$x_{ij} = 1 \quad \text{if } H_i \geqslant w_{ij}, \text{ 0 otherwise} \tag{10}$$

$$y_i = 1 \quad \text{if } H_i \geqslant 0, \text{ 0 otherwise} \tag{11}$$

To see the entire surface at minimum tower cost:

$$\text{Minimize} \quad \sum_i C(H_i), \quad C(H) > 0 \quad \text{for } H \geqslant 0, \quad C(H) = 0 \quad \text{for } H < 0 \tag{12}$$

$$\text{subject to} \quad \sum_i x_{ij} y_i \geqslant 1 \quad \text{for all } j. \tag{13}$$

To maximize the area covered within a construction budget T:

$$\text{Maximize} \quad \sum_j A_j \min\left(1, \sum_i x_{ij} y_i\right) \tag{14}$$

$$\text{subject to} \quad \sum_i C(H_i) \leqslant T. \tag{15}$$

Revelle [10] has suggested a discrete formulation of these problems which has certain advantages. Let $s_{ik} = 1$ denote the presence of a tower of height class k at vertex i, otherwise $s_{ik} = 0$, and q_{ik} the cost of the tower. Let U_j denote the set of ordered pairs $\{i, k\}$ which can see triangle j, and let $r_j = 1$ if triangle j can be seen by one or more towers, otherwise $r_j = 0$. Then we have two sets of constraints:

$$r_j \leqslant \sum_{(i,k) \in U_j} s_{ik} \quad \text{for all } j \tag{16}$$

$$\sum_k s_{ik} \leqslant 1 \quad \text{for all } i. \tag{17}$$

The objectives can now be expressed as:

$$\text{Maximize area seen:} \quad Z_1 = \sum_j A_j r_j \tag{18}$$

$$\text{Minimize cost:} \quad Z_2 = \sum_i \sum_k q_{ik} s_{ik}. \tag{19}$$

If the problem is solved as an LP with weighted objectives:

$$\text{Maximize } Z = \alpha Z_1 - (1 - \alpha) Z_2 \tag{20}$$

then solutions are likely to be integer [10].

The determination of the set U_j, or w_{ij} in the earlier continuous formulation, requires a modification of the visibility algorithm, as it is no longer appropriate to move a curtain outwards from each vertex. A similar curtain is constructed around each triangle and moved outwards. Whenever a vertex is encountered it is then possible to calculate the increase in elevation necessary to make the vertex visible over the top of the curtain, in other words the tower height.

Many other problems of spatial search and optimization can be formulated within the general framework of visibility on topographic surfaces. One might wish to find a minimum set of viewpoints to cover some prescribed subset of the surface, or to find the location from which a prescribed area can be seen using a tower of minimum height. Other objectives derive from a desire for concealment, for example to find the p most concealed vertices, by maximizing the area from which none can be seen. Given a suitable operationalization of the visibility of a path, one might search for the most concealed path between two specified vertices, or for vertices which best cover a given path.

6. Concluding remarks

Although it is possible to find precise definitions of visibility in the context of a specific model of a surface such as the TIN used in this study, 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 can 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 use of the TIN digital elevation model, the restriction of the search space to the vertices of the TIN, and the definition of each vertex's visible area as a set of fully visible triangles produces a tractable version of the visibility problem which is suitable for application to site selection. It can readily be generalized to the case where viewpoints are raised above the surface, and standard heuristics for set covering appear to work well. However the solution is clearly sensitive to the accuracy of the underlying TIN in representing the true topography.

This paper has introduced an extension of the concept of set covering to cases where coverage is determined by field of view on a topographic surface. As with all location problems, the search for optimality must be conducted on a model of reality rather than on reality itself; in this case the TIN serves as a model of the real topographic surface. However the effects of modelling are unusually explicit in this case, and the ability to investigate these effects is one of the more interesting aspects of this class of problems.

Acknowledgements

This research was supported by the Natural Science and Engineering Research Council of Canada through grant A7978. We also wish to thank Charles Revelle for extensive and helpful comments.

References

[1] K.E. Brassel and D. Reif, A procedure to generate Thiessen polygons, Geographical Analysis 11 (1979) 289.

[2] P.A. Burrough, *Principles of Geographic Information Systems for Land Resources Assessment* (Oxford University Press, 1986).

[3] R.L. Church and C.S. Revelle, The maximal covering location problem, Papers, Regional Science Association 32 (1974) 101.

[4] L. de Floriani, B. Falcidieno and C. Pienovi, Delaunay-based representation of surfaces defined over arbitrarily shaped domains, Computer Graphics, Vision and Image Processing 32 (1985) 127.

[5] L. de Floriani, B. Falcidieno, C. Pienovi, D. Allen and G. Nagy, A visibility-based model for terrain features, *Proceedings*, Second International Symposium on Spatial Data Handling, Seattle (1986) 235.

[6] D.T. Lee and B.J. Schachter, Two algorithms for constructing a Delaunay triangulation, International Journal of Computer and Information Sciences 9 (1980) 219.

[7] B.B. Mandelbrot, *The Fractal Geometry of Nature* (Freeman, San Francisco, 1982).

[8] D.M. Mark, Recursive algorithm for determination of proximal (Thiessen) polygons in any metric space, Geographical Analysis 19 (1987) 264.

[9] T.K. Peucker, R.J. Fowler, J.J. Little and D.M. Mark, The triangulated irregular network, Proceedings, Digital Terrain Models Symposium, ASP/ACSM (1978) 516.

[10] C. ReVelle, Personal communication, 1987

[11] A. Tarvydas, Terrain approximation by triangular facets, Paper prepared for AutoCarto 6, Ottawa, 1983.

[12] C. Toregas and C. ReVelle, Optimum location under time and distance constraints, Papers and Proceedings, Regional Science Association 28 (1972) 133.