Pergamon

Comput. & Graphics, Vol. 18, No. 6, pp. 763–773, 1994
Copyright © 1994 Elsevier Science Ltd
Printed in Great Britain
0097-8493/94 $6.00 + .00

0097-8493(94)00115-4

Modelling and Visualization of Spatial Data in GIS

TERRAIN VISIBILITY

GEORGE NAGY
Rensselaer Polytechnic Institute, Troy, NY 12180, USA, e-mail: nagy@ecse.rpi.edu

Abstract—Visibility indices, matrices, graphs, and maps are defined for digital elevation terrain models and their properties are established. Several potential applications of geometric terrain visibility in geographic information systems are presented. Direct applications include the location of observation points and hiding places, the determination of line-of-sight relay networks for microwave communications, scenic and hidden surface paths, and aids to navigation. Indirect applications allow inferences about physiographic terrain features: the mutual visibility among surface points yields considerable information about the landscape. The preservation of the visibility model also provides a check on data reduction (compression) in digital elevation models. Algorithms for computing terrain visibility on both triangulated irregular networks and rectangular grids are reviewed and experimental results from several sources are reported. It is shown that terrain visibility is invariant under certain projective transformations, and therefore relative elevations can be reconstructed from visibility maps.

1. INTRODUCTION

Computational tools for visualizing terrain from different perspectives facilitate solving, *by inspection*, many problems of a geographic nature. Using geometric visibility, some of these problems can be solved *by direct computation* instead of inspection. Examples include locating observation towers and line-of-sight transmitters and receivers; siting transmission lines, pipelines, roads, and rest-stops; navigation and orientation by reference to the horizon; the identification of certain topographic features; and, of course, a host of military emplacement problems. According to Felleman and Griffin, who also reviewed early visibility studies, "visibility mapping plays a central role in scenic landscape assessment including the delineation of jurisdictional and management zones, quantification of impacted viewer publics, and selection of visual control and simulation positions"[39].

Digital elevation terrain models (DEMs) provide an abstract representation (*model*) of the surface of the earth by ignoring all aspects other than topography. For instance, the elevation may be specified on a set of grid points (with a stipulated method of interpolation). Visualization tools, on the other hand, generate a display under some simplified assumptions (*models*) of surface reflectance, illumination, light transmission, and viewing mechanism. For instance, a surface may be visualized using a finite number of colors (that indicate land cover), lambertian reflectance, point-source illumination, and stereographic observation.

Because the visualization of terrain properties plays such an important role in geographic information systems, we are compelled to study the relationship between terrain elevation and visualization. The resulting abstraction, called *geometric visibility*, is based only on the intersection with the terrain of the lines of sight emanating from each viewpoint. Surface attributes, vegetation, atmospheric diffraction, and light intensity are neglected. While visualization shows the *appearance* of the terrain to an observer, geometric visibility is concerned only with the *extent* visible from given observation points. The output of a visualization program is intended for display for human assimilation, but the output of a visibility program can be channeled to another program for further calculation of visibility-based attributes.

The computation of terrain visibility is affected by the choice of the underlying computer representation of the terrain. Only rectangular grids and triangulated irregular networks (TINs) have been used so far. The preferred Delaunay triangulation is the dual of the Voronoi (or Thiessen) tessellation of the projections of the data points on the horizontal datum[14, 16, 45, 74].

Most elevation data, including that provided by the United States Geological Survey, is distributed in a uniform rectangular or quasi-rectangular (geodesic) grid format. Since four points cannot generally be fitted by a plane, this leads to a more complex, nonlinear approximation of the terrain. The complexity of the approximation depends on the degree of surface continuity desired. Tools are, however, available for reducing grid data to a triangulated irregular network which preserves certain terrain properties[55]. (We became interested in geometric visibility precisely because the selection of surface-specific points for a TIN, under a least-mean-square error criterion, often distorts the appearance of the terrain[13, 15]. Recent algorithms that take changes in the gradient or the gaussian curvature into account fare much better[32–37].)

Although the display algorithms that form the core of computer graphics are based on geometric visibility, the application of geometric visibility to terrain models is relatively new[1, 11]. In addition to computer graphics, spatial data processing and topographic analysis, it bears on computational geometry, computer vision, and operations research. The purpose of this paper is to summarize the results to date, show several useful applications, and present some algorithms for computing visibility.

2. BASIC VISIBILITY CONCEPTS

The visibility of the edges and vertices of planar polygons has been extensively studied in computational geometry both because of its intrinsic geometric interest and because of its applicability to art gallery, prison guard, robot motion, and tool-path planning problems[3, 4, 9, 48, 54, 61, 62, 67, 70, 80]. Most of the two-dimensional results have so far resisted generalization to three dimensions: O'Rourke's excellent 1987 monograph devotes only a few pages to 3D problems[68]. His recent survey on visibility graphs is also restricted to 2D[69], as is Shermer's update on art gallery problems[84].

Terrain visibility is usually called a *two-and-a-half dimensional* problem. For our purposes, a *terrain* is a topographic surface whose elevation above a horizontal datum is a single-valued function of x and y (no overhangs). Two points on such a surface are said to be *mutually visible* if the line segment that joins them does not pass below the surface. The intervisibility of a pair of points is a Boolean function of *four* scalar variables, or a mapping from $[R^2 \times R^2]$ to $\{0, 1\}$.

Given a terrain model on which surface-points, lines, and regions can be specified, the intervisibility of the various types of entities is represented by the corresponding Boolean *visibility function* defined on a product space of the entities. Among the nine visibility functions that can be defined among point, line, and region entities, the most useful are the *point-point* and *point-region* visibility functions.

Any visibility function can be represented by a *visibility graph* with arcs that link the nodes corresponding to intervisible entities. The visibility graph for point-point visibility is straightforward because any given point is either visible or invisible from any other point. However, edges and regions may be *partially* visible.

The *point-point visibility* among every pair of data points can be represented by a Boolean array of size N^2, called the *visibility matrix*, or by the corresponding *visibility graph* with N nodes and up to N^2 arcs, where N is the *total* number of data points. The visibility matrix is symmetric (under the assumption of zero observation height). The row and column sums (projections) of the visibility matrix correspond to the number of data points visible from each data point of the terrain. These *visibility indices* provide useful and relatively compact information about the terrain. In a bowl-shaped terrain, all points are intervisible; on a dome, none are. The highest points don't necessarily

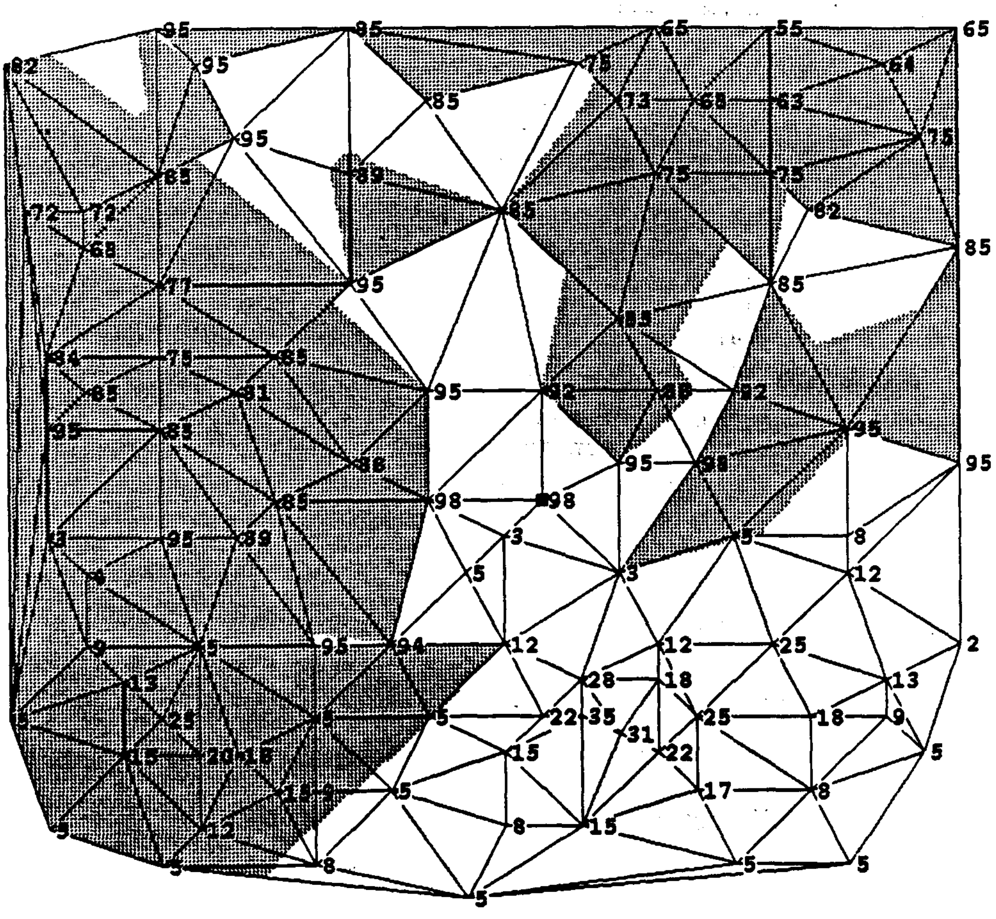


Fig. 1. A visibility map. The figure shows the horizontal projection of a terrain on a Delaunay-triangulated irregular network[52]. The dark areas are invisible from the viewpoint near the center, which is marked by a small square.

have the largest visibility indices[76]. The intervisibility of *two* sets of points is represented by a Boolean matrix with rows corresponding to one set, and columns to the other, or by the equivalent graph.

Point-region visibility can be represented by a set of two-dimensional *visibility maps* showing the vertical projection on the horizontal datum of the visible and invisible parts of the terrain from a specified viewpoint (Fig. 1). A visibility map is required for each observation point. In cartographic terms, most viewshed maps, from turn-of-century military conventions to the present, are binary choropleths of visible and invisible zones. The earliest visibility maps were generated by the military using a defilade approach consisting of radial samples of vertical cross-sections derived from topographic contours. The intersection points of the lines of sight were projected back to the original map, and interpolated. Regions of visibility and invisibility may be nested, as in the case of a mountain peak—that itself contains an invisible crater—which is visible beyond a ridge.

The projected boundaries of the regions may be divided into blocking segments and shadow segments. From the perspective of the viewpoint, a *blocking segment* represents the transition from a visible to an invisible region. A *shadow segment* represents the transition from invisible to visible. Blocking segments typically correspond to ridges and shoulder lines that cross a *line of sight* (*i.e.*, a ray contained in a radial vertical plane through the viewpoint). Shadow segments correspond to a double projection: the orthogonal projection on the horizontal datum of the central projection (from the viewpoint) of a ridge onto the terrain on the far side of the ridge.

The boundary of a connected region of visibility or invisibility that does not contain the viewpoint must consist of alternating chains of blocking segments and shadow segments. Any single chain consisting only of blocking segments or only of shadow segments must be a single-valued radial function of the azimuth, and may therefore form a closed curve only if it encloses the viewpoint. Furthermore, along any ray from the viewpoint on the visibility map, blocking and shadow segments must strictly alternate. (But vertical edges and surfaces tangent to a line of sight can give rise to anomalous radial boundaries between visible and invisible regions.)

If the terrain model consists of planar approximations, such as a TIN, then the projections on the horizontal datum of both the visible and invisible regions of an observation point consist of polygonal areas, and each blocking or shadow chain is a piecewise linear curve. Each blocking segment consists of edges of the triangulation. An edge of the triangulation may be part of a shadow segment only if the plane that contains the corresponding terrain edge and the viewpoint also contains a more proximal terrain edge. The maximum number of regions visible and invisible from a single viewpoint is quadratic in the number of faces[11].

The *horizon* is the set of ridges that corresponds to the blocking segments most distal from the viewpoint.

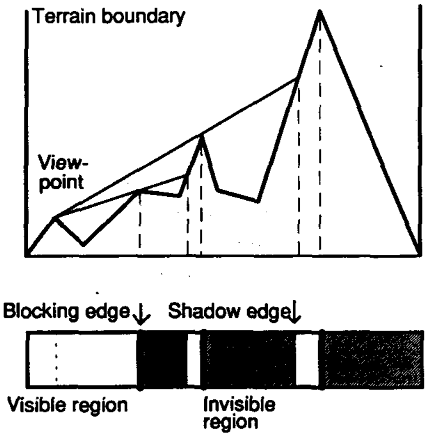


Fig. 2. Horizons in $1^{1/2}$-D. The viewpoint has three odd right horizons (*blocking edges*) and three even right horizons (*shadow edges*) including the terrain boundary. Its only left horizon is the terrain boundary. Terrain segments between odd and even horizons are invisible from the viewpoint, and segments between even and odd horizons are visible.

It has been shown that the number of segments comprising the horizon is, in the worst case, only slightly superlinear in the number of terrain edges[83, 11, 28, 29]. The boundaries between visible and invisible regions are sometimes called odd and even order *horizons* with respect to the given observation point (Fig. 2). A simple data structure for efficient manipulation of these boundaries, called the *influence tree*, is presented in [22].

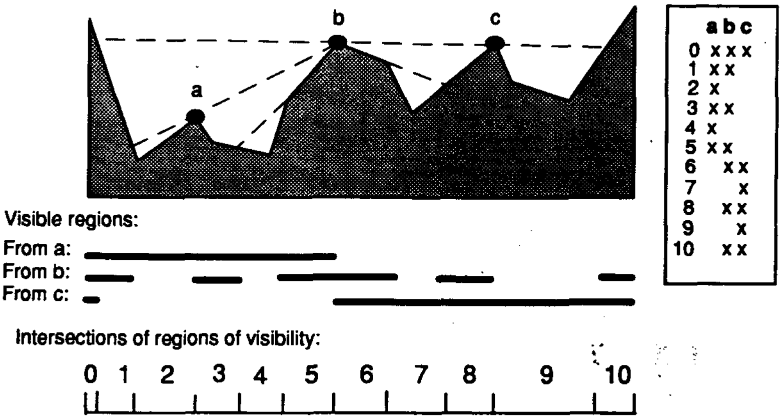
In order to program visibility computations, two questions must be laid to rest. The first question is: What happens beyond the boundary of the terrain, where we have no elevation information? We can assume, for instance, that the terrain is bounded by an infinitely high wall, or that it is surrounded by a flat ocean. Alternatively, we can model the curvature of the earth, which will ensure that visibility from every point is limited, or else simply set an arbitrary limit on the maximum distance from which a point may be visible.

The second question concerns collinear points. Are surfaces tangent to a line of sight visible? How we settle these questions won't have any significant impact on the methods or conclusions that we present, but computer implementation requires unambiguous specifications.

A further assumption may be made with regard to the height of the observer above ground. In most instances, assuming ground-level observation is not realistic. Assuming some *observation height* is essential for some problems, but optional for others. We defer discussion of algorithms for computing visibility matrices and visibility maps in order to consider some useful applications.

3. OBSERVATION POINTS

Sharir's *shortest watchtower algorithm* determines the location of the point *with the lowest elevation above the surface* from which an entire polyhedral terrain is



	a	b	c
0	x	x	x
1	x	x	
2	x		
3	x	x	
4	x		
5	x	x	
6		x	x
7			x
8		x	x
9			x
10		x	x

Fig. 3. Location of fire towers. Points *a*, *b*, and *c* are candidate locations. The regions visible from each point are shown below the terrain, followed by the ten elementary regions (generated by pairwise intersections) that are each completely visible or completely invisible from all candidate points. Visible elementary regions are marked with an *x* in the table on the right. In this simple example, it is clear that points *a* and *c* are sufficient to see the entire terrain.

visible[83]. Such a point must exist because the terrain elevation is a single-valued function, and therefore entirely visible from any point sufficiently far above it. The computational complexity is $O(n \log^2 n)$, where n is the number of polyhedral faces, so the algorithm is computationally practicable. It is also possible to preprocess the terrain to determine efficiently whether any particular point is visible from a *single observation point* on or above the surface[11].

Finding the location of the minimum set of observation points on the surface from which the entire surface is visible (*guard allocation*) is much more time consuming. Even for 2D polygons, the minimum vertex guard problem is of exponential complexity[68]. Topographic applications include the location of fire towers, artillery observers, and radar sites.

For finding the minimum set of observation points on a triangulated terrain, it is customary to restrict consideration to viewpoints located at vertices of the triangulation. First, the area of interest must be partitioned so that each partition is either completely visible or completely invisible from each viewpoint (Fig. 3). The required partitions are obtained by successive intersections of the visibility maps. Now finding the smallest number of observation towers can be stated as a *set-covering* (or *facilities-location*) problem of operations research, which is of exponential complexity[11, 49].

Tserkezou solves a small fire-tower problem, using matrix-reduction heuristics similar to those of the Quine-McCluskey algorithm for Boolean minimization[91]. The two steps of the iterative reduction process are:

1. Eliminate any viewpoint (column) that sees only a subset of the partitions seen by some other viewpoint.
2. Eliminate any partition (row) that is seen by a superset of the viewpoints that also see some other partition.

Once no further rows or columns can be eliminated from the partition-viewpoint matrix, all remaining combinations of viewpoints must be tested to determine the minimal set. If the residual set after reduction is too large for the available computing resources, then a sub-optimal solution with some redundant towers must be accepted.

Instead of using the matrix-reduction algorithm, Lee compares three different heuristic methods for finding the minimum number of observation towers on a 200-point TIN. The algorithms find almost the same number of towers (25, 25, and 27), but the computing times between the best and worst differ by a factor of 5000 to 1[56]. Variations of the problem considered in the same study include:

a. Find the area visible from a fixed set of observation points.
b. Maximize the area visible from a fixed set of observation points.
c. Given some cost function related to tower height, locate the towers so as to see the entire area at minimum cost.
d. Given some cost function related to tower height, locate the towers that maximize the area visible at a fixed cost.

Landscape analysis is less easy to formalize, but modern scenery analysis distinguishes between superior, normal, and inferior positions relative to local relief[12, 38, 58, 92]. Depending on the application, a commanding vista may be called a *military crest*, or a *panorama*.

4. LINE-OF-SIGHT COMMUNICATION

An obvious application of geometric visibility is the location of microwave transceivers for telephone, FM radio, television, and digital data networks. Of course, a realistic solution must take into account the height of the towers, the diffraction from intermediate ridges, and the distance limit imposed by the inverse-square law of electromagnetic propagation. So far, only the tower-height has been considered.

If the towers are restricted to the vertices of a polygonal terrain, then the only information that is required for line-of-sight computations is the visibility matrix or graph. The location of the viewpoints is immaterial, since a line-of-sight link may well start in the direction opposite to that of the terminus. (This happens when, for instance, a viewpoint located on the side of the transmitter away from the receiver can see both the starting point and the terminus.)

Finding the minimum number of relay towers necessary for line-of-sight transmission between two transceivers can be formulated as a shortest-path search[25] on the visibility graph. The overall computation can be accelerated by computing dynamically only the portions of the visibility graph that are required at any stage of the shortest-path search[8].

Now consider the problem of locating relay towers to complete the line-of-sight network between *several* transceivers. This problem is solved, under the restriction that the relay towers are located at vertices of the TIN, in [21]. Here, instead of computing the shortest path, one must find the *Minimum Steiner Tree* on the visibility graph. The Steiner problem is of exponential complexity in terms of its input, but efficient heuristics have been developed[53]. Because the overall computational complexity is the product of the cost of computing the visibility graph on the TIN and of the cost of computing the Minimum Steiner Tree on the visibility graph, the size of the underlying triangulation must be reduced as much as possible. Then, a conservative solution is obtained by adding the known bound on the resulting elevation error to the heights of the relay towers.

Finally, suppose that identical transmitters are to be located so as to broadcast to a fixed set of receivers. Specifically, it is required to locate the minimum number of transmitters so that each receiver can "see" at least one transmitter. This problem is similar to the fire-tower problem, and can be reduced to set covering on the visibility matrix itself (without intersecting any visibility maps).

5. SURFACE PATHS

The shortest path from one viewpoint to another along the edges of a triangulated terrain, such that none of the viewpoints traversed is visible from a given observation point, is called a *smuggler's path,* while a path on which every vertex is visible is a *scenic path.* We can find such a path (if one exists) by determining either the viewpoints that are visible from the observation point, or those that are not, and applying a standard shortest-path algorithm to the edges that connect them[81]. (For planar graphs, Dijkstra's algorithm is worst-case optimal[25], but since the location of the viewpoints provides a lower-bound on the length of the path to the terminus, the A* algorithm may sometimes be faster.)

We may restrict the location of the path to a *target region* and, instead of a single observation point, specify a *set of observation points.* Now, instead of computing the visibility from all the observation points ahead of time, we determine dynamically the visibility of candidate points along the path as they are expanded. The problem is completely specified by two $N \times N$ matrices: the adjacency matrix of the terrain vertices, and their visibility matrix. We may also seek a *scenic* path such that each vertex on it is *visible* from some (or all) of the specified observation points, or a path that has the maximum *cumulative* visibility[24].

Iwamura and his colleagues demonstrate a geographic database system for interactive planning of scenic paths. Constraints on the path include length, slope, and cost of construction. For any observation point along a candidate path, both a "visual range map" (the representation of a viewshed using radial lines from the viewpoint) and a bird's eye view of the terrain can be displayed[50].

6. PHYSIOGRAPHIC FEATURES

The literature on landforms is remarkably short on algorithmic definitions[10, 72, 75, 93]. Most attempts to automate physiographic feature extraction have been based on discrete approximations to derivatives of the surface[32–37, 46, 51, 63, 71, 73]. These features are generally quite local and highly scale-sensitive, whereas the significance of terrain features depends on their size and location relative to similar features in the entire area[58]. Metrics based on geometric visibility automatically take into account global relations. Human observers find it easy to classify significant features on the basis of visibility, though it is surprisingly difficult to trace the *significant* ridges on a contour map.

Some examples of visibility concepts applied to topographic features are the following. The visibility region of significant peaks tends to be large, and includes most of the visibility regions of lesser peaks. Significant peaks also have many blocking segments and multiply-connected visible regions, which distinguishes them from points in broad valleys that also have high visibility. Ridges block the horizons of many observation points. Points that are intervisible are in the same valley; otherwise, they are separated by ridges. In pits and valleys, the prospect is singly connected, and tends to change gradually. Lee presents a statistical analysis of elevation, visibility dominance, and landform category on a raster DEM that shows significant visibility differences among peaks, pits, ridges, and channels[57]. Even if landforms cannot be determined entirely according to visibility criteria, these may generate useful measures for ranking them.

Small-scale experiments on finding ridges, peaks, and pits using visibility criteria were first described in [65, 81]. Similar visibility-based methods have also been applied to the analysis of gray-scale images, where image intensity is considered the equivalent of elevation[19, 20]. The correlation between visibility and topography is explored from a military perspective by Ray[77].

7. NAVIGATION

Horizons do provide an important clue for navigation in mountainous terrain. Discontinuities in visibility can be readily determined under poor conditions by a variety of sensors, and matched to stored or computed horizons to determine the location of the observer. Discernible terrain features guide airborne military vehicles[89]. The use of horizon lines for autonomous navigation by a Mars Rover has been considered[64], but much further work remains to be done.

8. ALGORITHMS FOR VISIBILITY DETERMINATION

In principle, one could use any object-space hidden-surface algorithm from computer graphics[2, 59, 85, 94] to determine terrain visibility. (*Object-space algorithms*, in contradistinction to *image-space* algorithms like the popular Z-buffer method, determine the visible portions of the object rather than its projection on a display screen. *Ray tracing* exploits the finite resolution of the display. For visibility maps, we need the projection of the visible areas on the horizontal datum rather than on a view plane.) Most general-purpose hidden surface algorithms are inefficient for terrain visibility because they do not take advantage of the fact that there are no *bottom* surfaces. An efficient output-sensitive hidden-surface algorithm and its parallelization are reported by Reif and Sen[79], but implementation remains problematical. Recently, algorithms intended specifically for geographical terrains have been reported for computing visibility maps and visibility matrices (or graphs) on TINs, and for computing visibility matrices on grids.

The computation of the visibility matrix on a triangulated irregular network is conceptually straightforward, unless one attempts to exploit the obvious *coherence* in the visibility of neighboring viewpoints. Lee's algorithm is designed for "binary visibility" determination: a triangular facet is considered "visible" from a viewpoint if all three of its edges are completely visible[56]. Otherwise, the facet is considered "invisible." The edges are first sorted according to their minimum distance from the viewpoint, then compared pairwise to determine whether the proximal edge blocks the distal edge. The azimuths of the end-points of the edges are used to screen out pairs that cannot block one another. Lee also computes how much the viewpoint could be raised or lowered with the entire triangle still visible, and makes use of this information in solving the variable-height tower problems described earlier.

Our earlier algorithm for computing the complete visibility map on a TIN is conceptually quite similar[18, 52]. It can be visualized as a searchlight, located at the viewpoint, which illuminates the terrain in a progressive outward spiral. As the beam is raised, it encounters ridges that cast shadows on the terrain farther from the light. The endpoints of the ridge and shadow segments, which form the boundary between the visible and invisible regions, are recorded. Adjacent viewpoints (vertices of the triangulation) are considered intervisible.

In terms of triangulated terrain representation, the triangles must be examined in a sequence such that a triangle which may cast a shadow on another triangle is processed first. The necessary ordering property, called *acyclicity*, has been shown to hold for Delaunay triangulation[17, 30]. The visible portion of each triangle is determined by projecting on it the dominant blocking edges between the triangle and the viewpoint. New blocking edges are introduced whenever a partially or fully visible triangle is followed by an invisible triangle. The current horizon of dominant blocking edges is maintained in a data structure to avoid checking for each triangle all of the previous blocking edges.

Empirical observations show that the average performance of the program, for a single viewpoint, is linear in the number of vertices, because the visibility of each triangle is computed in constant time regardless of the size of the input. The computation of the N visibility maps for a terrain with N vertices is thus of $O(N^2)$[52].

More complex algorithms for triangulated terrain models are surveyed by De Floriani and Magillo from a computational-geometry perspective[23]. Many of these algorithms are based on computing the upper envelope of the visible ridges, which can be computed efficiently using divide-and-conquer and sweep-line techniques. Data structures for building the envelope include the *horizon tree* and the *conflict graph*. A randomized algorithm is also presented.

Grid-based algorithms all compute the intersection of radial lines of sight with the edges of the grid cells that they intersect. The elevation of each edge is assumed to vary linearly between that of its endpoints. This may, however, lead to inconsistencies because the elevation *within* the grid cell is not modeled. The difference between algorithms lies mainly in the choice of rays (Fig. 4).

Shapira computes a distinct ray to every grid point and abandons the ray as soon as a blocking ridge is found[81]. His algorithm, which is a modification of Anderson's hidden-surface method for viewpoint-viewpoint and directional visibility, was coded with care using only integer arithmetic[82]. It has been run on terrains with up to 14,000 vertices (but the output, a 14,800 × 14,000 visibility matrix, could not be saved). The average complexity of the algorithm is $O(N^{2.25})$

Rather than compute rays to every grid point, it is more economical to compute rays only to vertices of the grid that lie on the boundary of the region under

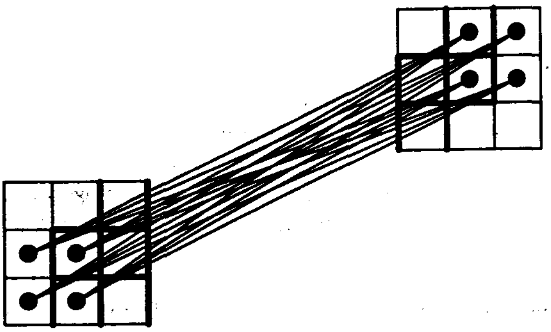


Fig. 4. Visibility calculations on a grid. This figure shows the redundancy of computing visibility by following the line of sight from each observation point to each target point. Most grid-based visibility programs eliminate computing some of the redundant intersections of the lines of sight with the edges of the grid cells.

consideration[7, 78]. The elevation angle of each ray is progressively raised to clear intersecting blocking edges, and edges below the ray are marked "invisible." The visibility of each grid point inside the region is computed from that of its incident edges. Ray compares the accuracy of using rays to *every grid point* to that obtained by using only rays to *selected boundary points*[78]. He shows that the differences are minimal. Furthermore, there is little additional loss of accuracy from reducing the number of rays even more provided that the points are weighted to account for the dispersion of the rays. Using only 32 rays from each observation point on a 100-m resolution grid is almost as good as using 128 rays when visibility is limited to 40 km (with observer and target heights of 5 m and 25 m, respectively). However, Ray is interested primarily in selecting high-visibility sites, and therefore computes only the visibility index for selected observation points rather than the complete visibility matrix.

9. PARALLEL COMPUTATION

Because visibility computations are so computer-intensive, they form a natural target for parallelization. Elegant algorithms are being developed for 2D, $2^{1/2}$-D, and 3D visibility[6, 7, 60, 87]. Whether these will prove to be of practical significance remains to be seen, since visibility computations can also be readily and efficiently parallelized by either terrain segment or by viewpoint. Furthermore, for a single viewpoint the average complexity tends to be linear in both input and output, so little gain can be expected from clever algorithms. The most massive computations to date, coarsely parallelized on networked Sparcstations, were reported by Ray on a 28-million point data set on the Korean peninsula[78].

Nevertheless, in computing the visibility matrix, it is possible to take advantage of the coherence of the visibility from adjacent view points, because rays from adjacent observation points to a given target point tend to traverse the same set of edges. This property has been carefully defined and exploited for both visibility index and visibility matrix computations[86]. The algorithms are parallelized for a CM-2 hypercube computer by assigning several processors to each ray. The visible segments of each ray from a given viewpoint to a target point are derived from the properties of rays from adjacent viewpoints, using a sweep algorithm. For 64 × 64 point source and target regions (which may overlap), a 16K-processor hypercube computes the visibility maps in about 30 seconds. Almost half of the total time is input/output.

The above algorithm requires some global communication among processors, which may be unacceptable for mesh-connected computers. An algorithm specifically designed for mesh-connected computers is based on the propagation of shadows in a manner similar to that of the TIN-oriented methods above, but it assumes a constant elevation within each cell[88]. On a grid, however, the direction of propagation is restricted by the cell connectivity and cannot follow exactly the line of sight. This introduces errors unless one keeps track of the partial visibility of each grid cell. The ray-sweep method and the propagation method were compared on three real scenes and two artificial scenes with respect to both accuracy and timing, but both methods were implemented only on the hypercube[88].

10. ACCURACY

All visibility calculations are very sensitive to changes in elevation near the viewpoint because the line-of-sight to distant targets magnifies these changes in proportion to the distance. Because of the dominant effect of errors near the observation point, Felleman and Griffin advocate a two-tiered model where the topography near the observation point is known five or ten times more accurately than in the rest of the viewshed[39].

The influence of possible elevation errors and of numerical errors generated in the computations is examined in[39, 40–44, 60, 88]. Mills, Fox, and Heinsbach compute in 30 minutes the complete intervisibility of the 86 × 76 data-point DEM (Howe Hill in Massachusetts) on a Connection Machine-2. Randomly distributed errors are then added to the elevation values, and the calculations are repeated. The results indicate that "intervisibility may be overestimated when the effect of DEM errors is not considered"[60]. Similar conclusions are stated in[76].

Fisher simulates *correlated* DEM errors in viewshed determination. He shows that the spatial correlation *reduces* uncertainty in comparison to independent errors, and conjectures that the areas visible from higher elevations are less susceptible to such errors than those visible from a depression. His work suggests that errors induce an *underestimate* of the visible area. (Consider a flat, completely visible area. Any variation in elevation due to error will *decrease* the computed visibility.) Because of the disproportionate effect of small errors, Fisher advocates using a probabilistic formulation to take into account uncertainties in elevation. He attempts to formulate a quantitative notion of *fuzziness* as the lack of clarity due to observation conditions[40–44].

A conservative strategy is to compute *ternary* visibility maps which show the area that is certainly visible, the area that is certainly invisible, and the region of uncertainty. Quoting Felleman and Griffin[39] again:

> The Monte Carlo analyses showed that expected variations in the DEM have potentially major effects on the resultant viewshed. Rather than the conventional paradigm of a single crisp black and white (visible or not visible) deterministic pattern, viewshed maps intrinsically contain an extensive, mottled, rich 'gray' set of 'potentially visible or hidden' zones. GIS analyses which ignore the magnitude and character of this information are necessarily suspect.

11. VISIBILITY INVARIANTS

Visibility functions do not define a terrain uniquely: several different terrains may have the same visibility map. We call these terrains visibility-equivalent. To gain some insight into what characterizes this equivalence relation, consider the two $1^{1/2}$-D visibility-equivalent terrains shown in Fig. 5. For simplicity, the x-coordinates of the points are uniformly spaced. Under these conditions, the y-coordinates are subject to the following equation:

$$(y_4-y_1)/(y_4-y_3) = (y_5-y_2)/(y_3-y_2).$$

This equation defines the cross-ratio of four segments, which is invariant under a projective transformation. Once we fix one of the elevations arbitrarily, any one of the others can be calculated from the three remaining elevations by solving a linear equation.

As a corollary, the visibility functions on a $1^{1/2}$-D terrain are invariant under any projective transformation, including translation and scaling of coordinates, rotations, and affine transformations. We therefore believe that the visibility map preserves an important property of the terrain that is related to *shape*, and may therefore be used as a criterion function for lossy compression algorithms. (Note, however, that the converse is not true: not all visibility-equivalent terrains can be derived by a projective transformation.)

The projective relationship can be extended to $2^{1/2}$-D triangulated terrains. It yields a set of linear equalities and inequalities that have been used to compute the visibility-equivalent terrains of a set of visibility maps using linear programming software [66]. The inequalities arise because the elevation of any point which is invisible except to its neighbors is only bounded, rather than determined exactly, by the elevations of other points (Fig. 6). For instance, every convex (concave) bowl has the same visibility map. The terrain reconstruction is computationally intensive: a terrain derived from a 9 × 9 rectangular grid converted into 128 triangular facets required the simultaneous solution of over 9000 equalities and inequalities.

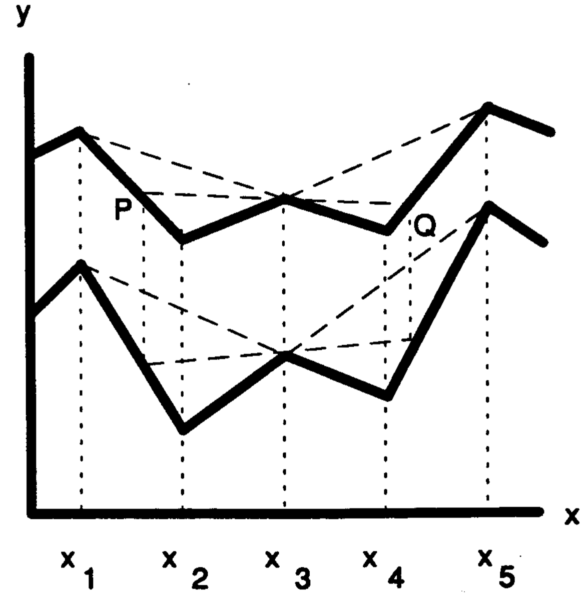


Fig. 5. Two visibility-equivalent terrains in $1^{1/2}$-D. The locations of shadow edges are identical for every terrain whose elevations satisfy the projective relationship $(y_1 - y_4)/(y_4 - y_3) = (y_5 - y_2)/(y_3 - y_2)$ for each subset of five data points. For example, the shadow edge Q of the observation point P has the same x-coordinate in both terrains. Consequently, the intervisibility of any pair of points can be readily determined from the locations of the horizons.

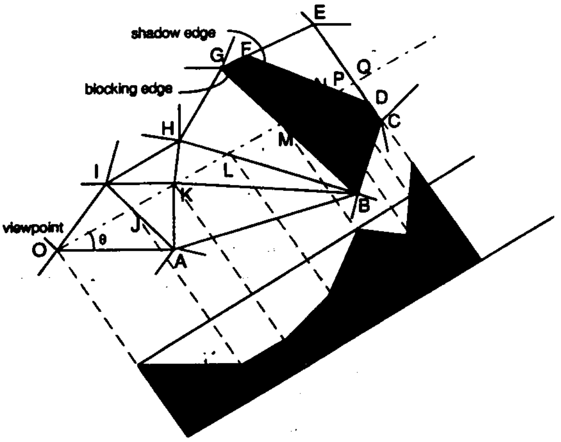


Fig. 6. Constraints for $2^{1/2}$-D terrain reconstruction. The viewpoint on this horizontal projection is vertex O. The ray OQ defines a line of sight with azimuth θ through the horizontal projection of point K, which is a vertex of the triangulation. The elevation of this line of sight is defined by its intersection with point M on ridge BG. The horizontal projection of BG is the first horizon of O, and DF, its induced shadow edge, is the second horizon (in direction θ). The visibility constraints associated with this ray are: (1) the elevations of points J, L, Q, and N (on edge CG) are *below* that of the line of sight; (2) points M and P are *on* the line of sight, and (3) point Q lies *above* the line of sight. Such constrains define a redundant set of inequalities and equalities that yield feasible values for the elevations of all the vertices. A vertical radial cross-section through OQ that satisfies the constraints is shown below the plan view.

To avoid the solution corresponding to a completely flat terrain that trivially satisfies all equalities and inequalities, a small constant threshold must be introduced in each inequality. The threshold specifies how much (at least) an invisible datapoint must be below the line of sight, or a visible datapoint above it. (This threshold plays a role similar to that of *slack variables*.) As a check on the reconstruction, the visibility matrices of the terrains were computed and compared to those of the terrain from which the visibility map was originally generated. Numerical inaccuracies sometimes cause minor differences between the matrices of the original and reconstructed terrains. The discrepancies decrease and eventually disappear as the threshold is increased, but if the threshold is too large, then the linear system does not have any solution.

The visibility matrix gives even less information than the visibility map about topography. We do not yet understand, even in $1^{1/2}$-D, under what conditions does a given visibility matrix correspond to a realizable terrain.

12. CONCLUSION

Efficient algorithms have been developed for computing geometric terrain visibility on both triangulated networks and regular grids. The visibility map of several hundred potential observation sites can be computed for a terrain defined by millions of elevation data points in a few hours on any contemporary workstation. Although parallel algorithms have been applied to advantage, viewpoint-to-viewpoint *coherence* has been exploited only at the grid-level rather than the terrain level. The visibility indices, matrices, graphs, and maps that such computations yield are useful for visualizing visibility-related terrain properties and also form the basis for several applications where direct computation is substituted for inspection. The major driving force has been geographic information systems for military applications.

Efficient solutions to the *single observation tower* problem have been published. The problem of locating the *minimum number of viewpoints* from which the entire terrain is visible can be solved by a *set-covering* algorithm. Some *shortest surface-path* problems constrained by visibility criteria and *line-of-sight path* problems for microwave communications are computationally tractable, and programs have been tested on small examples. Visibility methods are beginning to be applied to the identification and ranking of *physiographic features* such as ridges, but the experiments reported are difficult to evaluate in the absence of concrete criteria for descriptive topography. The application of visibility methods for navigation for *civilian* purposes has barely been initiated.

On polyhedral terrains, the visibility map is sufficient to define the terrain to within a projective invariant. However, the only method demonstrated so far for the reconstruction of visibility-equivalent terrains is far too costly for large-scale application. If a computationally less intensive method is found, it may have applications to terrain data compression by retaining only terrain features that have a significant impact on visibility.

Tools for computing the intervisibility of selected points have long been included in geographic information systems[5, 26, 27, 31, 47, 90], but a recent attempt by Felleman and Griffin[39] to compare eight software packages for viewshed determination led to inconsistent results. We expect, however, that the next generation of GIS will offer a number of robust visibility-related application programs.

Acknowledgements—The author's research on geometric visibility has been conducted from the outset in collaboration with L. De Floriani of the Department of Computer and Information Science of the University of Genoa, and B. Falcidieno, and C. Pienovi of the Genova Institute of Applied Mathematics of the Italian National Research Council. The work was supported by the National Science Foundation under US-Italy Collaborative Research Program grant INT-8714578 and Information, Robotics and Intelligent Systems grant IRI-8704718, and by the Italian National Research Council. Also gratefully acknowledged are an interesting and productive afternoon with Micha Sharir, discussions with W. Randolph Franklin, Jay Lee, Shashank Mehta, M. Fisher, Enrico Puppo, Clark Ray, and Y. Ansel Teng, and the essential contributions of former Rensselaer students D. Allen, P. Jeanne, D. Jung, H. Nair, N. Narendra, A. Shapira, and P. Tserkezou. An anonymous referee gave useful criticisms on an early draft.

REFERENCES

1. D. Allen, L. De Floriani, B. Falcidieno, G. Nagy, and C. Pienovi, A visibility-based model for terrain features. *Proc. International Symposium on Spatial Data Handling,* Seattle, 235–250 (1986).
2. D. P. Anderson, Hidden line elimination in projected grid surfaces. *ACM Trans. on Graphics* **1**(4), 274–291 (1982).
3. D. Avis and G. T. Toussaint, An optimal algorithm for determining the visibility of a polygon from an edge. *IEEE Trans. Comput.* **C-30,** pp. 910–914 (1981).
4. D. Avis, T. Gum, and G. T. Toussaint, Visibility between two edges of a simple polygon. *The Visual Computer* **2,** 342–357 (1986).
5. J. Berry, PMAP, The professional map analysis package. *Papers in Spatial Information Systems,* Yale University School of Forestry and Environmental Studies, New Haven, CT (1986).
6. P. Bertolazzi, S. Salza, and C. Guerra, A parallel algorithm for the visibility problem from a point. *J. Parallel and Distributed Computing* **9,** 11–14 (1990).
7. G. E. Blelloch, *Vector Models for Data-Parallel Computing.* The MIT Press, Cambridge, MA (1990).
8. M. Cazzanti, L. De Floriani, G. Nagy, and E. Puppo, Visibility computation on a triangulated terrain. *Proc. Sixth Int. Conf. Image Processing and Analysis,* 721–728, Como, Italy, September 1991.
9. W. Chin and S. Ntafos, Optimum watchman routes. *Information Processing Letters* **28,** 39–44 (1988).
10. R. J. Chorley, *Spatial Analysis in Geomorphology,* Methuen and Co. Ltd (1972).
11. R. Cole and M. Sharir, Visibility problems for polyhedral terrains. *J. Symbolic Computation* **7,** 11–30 (1989).
12. W. Creamer, The Upper Klethia Valley: Computer generated maps of site location. *Proc. SAA Meeting,* Denver (1985).
13. L. De Floriani, B. Falcidieno, G. Nagy, and C. Pienovi, A hierarchical structure for surface approximation. *Computers and Graphics* **8**(2), 183–193 (1984).
14. L. De Floriani, B. Falcidieno, and C. Pienovi, Delaunay-based representation of surfaces defined over arbitrarily shaped domains. *Computer Vision, Graphics, and Image Processing* **32,** 127–140 (1985).
15. L. De Floriani, B. Falcidieno, G. Nagy, and C. Pienovi, Efficient selection, storage, and retrieval of irregularly distributed elevation data. *Computers and Geosciences* **11**(6), 667–673 (November 1985).
16. L. De Floriani, Surface representations based on triangular grids. *The Visual Computer* **3,** 27–50 (1987).
17. L. De Floriani, B. Falcidieno, G. Nagy, and C. Pienovi, On sorting triangles on a Delaunay tesselation (CNR-IMA Technical Report #9), Genova, Italy, January 1988; *Algorithmica* **6,** 522–532 (June 1991).
18. L. De Floriani, B. Falcidieno, G. Nagy, and C. Pienovi, Polyhedral terrain description using visibility criteria (Technical Report #17/89), Istituto per la Matematica Applicata, CNR, Genova (October 1989).
19. L. De Floriani, P. Jeanne, and G. Nagy, Visibility characteristics of grey-scale images. In *Progress in Image Analysis and Processing,* V. Cantoni, L. P. Cordella, S. Levialdi, and G. Sanniti di Baja (Eds.), World Scientific, 435–442 (1990).
20. L. De Floriani, P. Jeanne, and G. Nagy, Visibility related image features. *Pattern Recognition Letters* **40,** 137–139 (November 6, 1992).
21. L. De Floriani, G. Nagy, and E. Puppo, Computing a line-of-sight network on a terrain model. *Proc. Fifth Int. Symp. Spatial Data Handling,* 672–681, Charleston, SC, (August 1993).
22. L. De Floriani and P. Magillo, Computing point visibility on a terrain based on a nested horizon structure. *ACM Symp. on Applied Computing,* Phoenix, AZ (March 1994).
23. L. De Floriani and P. Magillo, Visibility algorithms on triangulated terrain models. *Int. J. Geographic Information Systems* **8**(1), 13–41 (1994).
24. P. Dietrich, S. Hoffman, R. Moen, T. Prescott, J. Schimtff, D. Schwartz, L. Soon-Wha, and R. Trowbridge, *Managing*

scenic beauty along the lower Wisconsin River, U. Wisconsin Dept. Landscape Architecture, Madison, WI, (October 1988).

25. E. W. Dijkstra, A note on two problems in connection with graphs. *Numerische Mathematik* **1,** 269–271 (1959).
26. Douglas, Viewblock: A computer program for constructing perspective view block diagrams, *Revue de Geographie de Montreal XXVI,* 102–104 (January 1972).
27. R. Eastman, *IDRISI—A grid-based geographic analysis system,* Clark University, Worcester MA (1989).
28. H. Edelsbrunner, L. J. Guibas, and M. Sharir, The upper envelope of piecewise linear functions: Algorithms and applications. *Discrete and Computational Geometry* **4,** 311–336 (1989).
29. H. Edelsbrunner, The upper envelope of piecewise linear functions: Tight bounds on the number of faces. *Discrete and Computational Geometry* **4,** 337–343 (1989).
30. H. Edelsbrunner, An acyclicity theorem for cell complexes in d-dimensions. *Combinatorica* **10,** 251–260 (1990).
31. ESRI, TIN—Users Guide, Environmental Systems Research Institute, Redlands, CA (1987).
32. B. Falcidieno and C. Pienovi, Natural surface approximation by constrained stochastic interpolation. *Computer-Aided Design* **22**(3), 167–253 (1990).
33. B. Falcidieno and C. Pienovi, A feature-based approach to terrain surface approximation. *Proc. Fourth Int. Symp. Spatial Data Handling,* 190–199, Zurich (1990).
34. B. Falcidieno and M. Spagnuolo, A new method for the characterization of topographic surfaces. *Int. J. Geographical Information Systems* **5**(4), 397–412 (1991).
35. B. Falcidieno, C. Pienovi, and M. Spagnuolo, Discrete surface models: Constraint-based generation and understanding. In *Computer Graphics and Mathematics,* B. Falcidieno, I. Herman, and C. Pienovi (Eds.), Springer-Verlag, 245–261 (1992).
36. B. Falcidieno and M. Spagnuolo, Polyhedral surface decomposition based on curvature analysis. In *Modern Geometric Computing,* T. L. Kunii and Y. Shinagawa (Eds.), Springer-Verlag, Tokyo, 57–72 (1992).
37. B. Falcidieno and M. Spagnuolo, Geometric reasoning for the extraction of surface shapes. In *Communicating with Virtual Worlds,* N. Magnenat Thalmann and D. Thalmann (Eds.), Springer-Verlag, Tokyo, 166–178 (1992).
38. J. P. Felleman, Landscape visibility. In *Foundations of Visual Project Analysis,* R. C. Smardon, J. F. Palmer, and J. P. Felleman (Eds.), Wiley and Sons, New York, 47–62 (1986).
39. J. P. Felleman and C. Griffin, *The role of error in GIS-based viewshed determination—a problem analysis* (TR EIPP-90-2), Institute for Environmental Policy and Planning, State University of New York (1990).
40. P. F. Fisher, Simulation of the uncertainty of a viewshed. *Autocarto* **10,** 205–218 (1991).
41. P. F. Fisher, First experiments in viewshed uncertainty: The accuracy of the viewshed area. *Photogrammetric Engineering and Remote Sensing* **57**(10), 1321–1327 (1991).
42. P. F. Fisher, First experiments in viewshed uncertainty: Simulating fuzzy viewsheds. *Photogrammetric Engineering and Remote Sensing* **58**(3), 345–352 (1992).
43. P. F. Fisher, Probable and fuzzy models of viewshed operation. In *Innovation in GIS,* M. F. Worboys (Ed.), Taylor and Francis, 161–175 (1994).
44. P. F. Fisher, Algorithm and implementation uncertainty in viewshed analysis. *Int. J. Geographic Information Systems* 7(4), 331–347 (1993).
45. R. J. Fowler and J. J. Little, Automatic extraction of irregular network digital terrain models. *Computer Graphics* **13,** 199–207 (1979).
46. A. Frank, B. Palmer, V. Robinson, Formal methods for the accurate definition of some fundamental terms in physical geography. *Proc. Second Int. Symp. on Spatial Data Handling,* Seattle, 583–599 (1986).
47. W. R. Franklin, PRISM—A prism plotting program. *Mapping Software and Cartographic Databases, Harvard Library of Computer Graphics Mapping Collection Vol.* 2, 75–80 (1979).
48. L. P. Gewali and S. Ntafos, Covering grids and orthogonal polygons with periscope guards, *Computational Geometry: Theory and Applications* **2,** 309–334, Elsevier (1993).
49. G. Handler, P. Mirchandani, *Location on networks: Theory and algorithms,* MIT Press, Cambridge, MA (1979).
50. K. Iwamura, Y. Nomoto, S. Kakumoto, and M. Ejiri, Geographical feature analysis using integrated information processing system. *IAPR Workshop on Machine Vision Applications,* Tokyo (November 28–30, 1990).
51. E. Johnston and A. Rosenfeld, Digital detection of pits, peaks, ridges and ravines. *IEEE Trans. on Systems, Man and Cybernetics* **5,** 672–680 (1975).
52. D-M. Jung, *Comparisons between algorithms for terrain visibility,* MS Thesis, Rensselaer Polytechnic Institute (August 1989).
53. L. Kou, G. Markowsky, and L. Bermann, A fast algorithm for Steiner Tree. *Acta Informatica* **15,** 141–149 (1981).
54. A. Laurentini, The visual hull and its computation in 2-D. In *Visual Form: Analysis and Recognition,* C. Arcelli, L. P. Cordella, and G. Sanniti di Baja (Eds.), Plenum Press, 355–362 (1992).
55. J. Lee, Comparison of existing methods for building triangular irregular network models of terrain from grid digital elevation models. *Int. J. Geographical Information Systems* **5**(3), 267–285 (1991).
56. J. Lee, Analysis of visibility sites on topographic surfaces. *Int. J. Geographical Information Systems* **5**(4), 413–429 (1991).
57. J. Lee, Visibility dominance and topographic features on digital elevation models. *Proc. Fifth Int. Symp. Spatial Data Handling,* 622–631, Charleston, SC, August 1993, *Photogrammetric Engineering and Remote Sensing* **60**(4), 451–456 (1994).
58. A. MacEachren and J. Davidson, Sampling and isometric mapping of continuous cartographic surfaces. *The American Cartographer* **14**(4), 299–320 (1987).
59. M. McKenna, Worst-case optimal hidden-surface removal. *ACM Transactions on Graphics* **6,** 19–28 (1987).
60. K. Mills, G. Fox, and R. Heimbach, Implementing an intervisibility model on a parallel computing system, (TR 150b), Syracuse Center for Computational Science, 1992. *Computers and Geosciences* **18**(8), (1992).
61. J. S. B. Mitchell and E. L. Wynters, Optimal motion of covisible points among obstacles in the plane. *Proc. Second Canadian Conference on Computational Geometry,* 116–119 (1990).
62. J. S. B. Mitchell and E. L. Wynters, Watchman routes for multiple guards. *Proc. Third Canadian Conference on Computational Geometry,* 126–129 (1991).
63. L. Nackman, Two-dimensional critical point configuration graphs. *IEEE Trans. Pattern Analysis and Mach. Int.* **6**(4), 442–450 (1984).
64. G. Nagy and C. N. Shen, Autonomous navigation to provide long-distance surface traverses for Mars Rover sample return mission. *Fourth IEEE Int'l. Symp. on Intelligent Control,* Albany, 362–367 (September 1989).
65. H. Nair, *A high-level description of digital terrain models using visibility information,* MS Thesis, Rensselaer Polytechnic Institute (May 1988).
66. N. C. Narendra, *Reconstruction of terrain features from horizons,* PhD dissertation, Rensselaer Polytechnic Institute, Troy, NY (1992).
67. S. Ntafos, The robber route problem. *Information Processing Letters* **34,** 59–63 (1990).
68. J. O'Rourke, *Art gallery theorems and algorithms,* Oxford University Press (1987).
69. J. O'Rourke, Visibility graphs. *SIGACT News* **24**(1), 20–25 (1993).
70. M. H. Overmars and E. Welzl, A new method for computing visibility graphs, *Proc. Fourth Annual ACM Sym-*

posium on Computational Geometry, Urbana-Champaign, IL, 164–171 (June 6–8, 1988).

71. B. Palmer, Symbolic feature analysis and expert systems. *Proc. First Int. Symp. on Spatial Data Handling*, Zurich (1984).
72. Pearson, R. N., *Physical Geography*, Barnes and Noble (1968).
73. T. K. Peucker and D. H. Douglas, Detection of surface specific points by local parallel processing of discrete terrain elevation data. *Computer Graphics and Image Processing* **4**, 375–387 (1975).
74. T. K. Peucker, R. J. Fowler, J. J. Little, and D. H. Mark, The triangulated irregular network. *Proc. ASP-ACSM Symp. on DTMs*, St. Louis (1978).
75. R. Pike and W. Rozema, Spectral analysis of landforms. *Annals of the Association of American Geographers*, **65**(4), 499–516 (1975).
76. C. Ray, W. R. Franklin, and S. Mehta, *Geometric algorithms for siting of air defense missile batteries* (TR 2756), Rensselaer Polytechnic Institute (1992).
77. C. Ray, A new way to see terrain. *Military Review* **LXXIV**(11), (November 1994).
78. C. Ray, *Representing Visibility for Siting Problems*, PhD Dissertation, Rensselaer Polytechnic Institute, Troy, NY (1994).
79. J. H. Reif and S. Sen, An efficient output-sensitive hidden-surface removal algorithm and its parallelization, *Proc. 4th Annual ACM Symp. Computational Geometry*, Urbana-Champaign, IL, 193–200 (June 1988).
80. J. Rudiger Sack and S. Suri, An optimal algorithm for detecting weak visibility of a polygon, *IEEE Trans. Computers* **39**(10), 1213–1219 (October 1990).
81. A. Shapira, *Visibility and terrain labeling*, MS Thesis, Rensselaer Polytechnic Institute (1990).
82. A. Shapira, Fast line-edge intersections on a uniform grid, *Graphics Gems*, Andrew Glassner (Ed.), Academic Press, 29–36 (1990).
83. M. Sharir, The shortest watchtower and related problems for polyhedral terrains, *Information Processing Letters* **29**, 265–270 (1988).
84. T. Shermer, Recent results in art galleries, *Proceedings of the IEEE* **80**(9), 1384–1399 (1992).
85. I. Sutherland, R. Sproull, and R. Schumacker, A characterization of ten hidden-surface algorithms, *Computing Surveys* **6**(1), 1–55 (1974).
86. Y. Ansel Teng, D. De Menthon, and L. S. Davis, Region-to-region visibility analysis using massively parallel hypercube machines (University of Maryland Technical Report CAT-TR-578 [CS-R-2754]) (September 1991).
87. Y. Ansel Teng and L. S. Davis, Parallel visibility analysis on digital terrain models. *Proc. International Conference on Parallel and Distributed Systems*, 606–613 (1992).
88. Y. Ansel Teng and L. S. Davis, Visibility analysis on digital terrain models and its parallel implementation (University of Maryland Technical Report CAT-TR-625 [CS-TR-2900]) (May 1992).
89. Y. Ansel Teng, Daniel De Menthon, and L. S. Davis, Stealth terrain navigation. *IEEE Trans. Systems, Man, and Cybernetics* **23**(1), 96–110 (January/February 1993).
90. M. R. Travis, G. H. Elsner, W. D. Iverson, and C. D. Johnson, *VIEWIT* (USDA Forest Service General Technical Report PSW-11/75). Pacific Southwest Forest and Range Experiment Station, Berkeley, CA (1975).
91. P. Tserkezou, *Characteristics of visibility models*, MS thesis, Rensselaer Polytechnic Institute, Troy, NY (May 1988).
92. W. M. Turnbull, T. W. Maver, and I. Gourley, Visual impact analysis—A case study of a computer-based system. *Proc. Auto Carto*, London, 197–206 (1986).
93. S. D. Tuttle, *Landforms and Landscapes*, Wm. C. Brown Co. (1970).
94. K. Weiler and P. Atherton, Hidden surface elimination using polygon area sorting. *Proc. SIGGRAPH* 1977, 214–222 (1977).