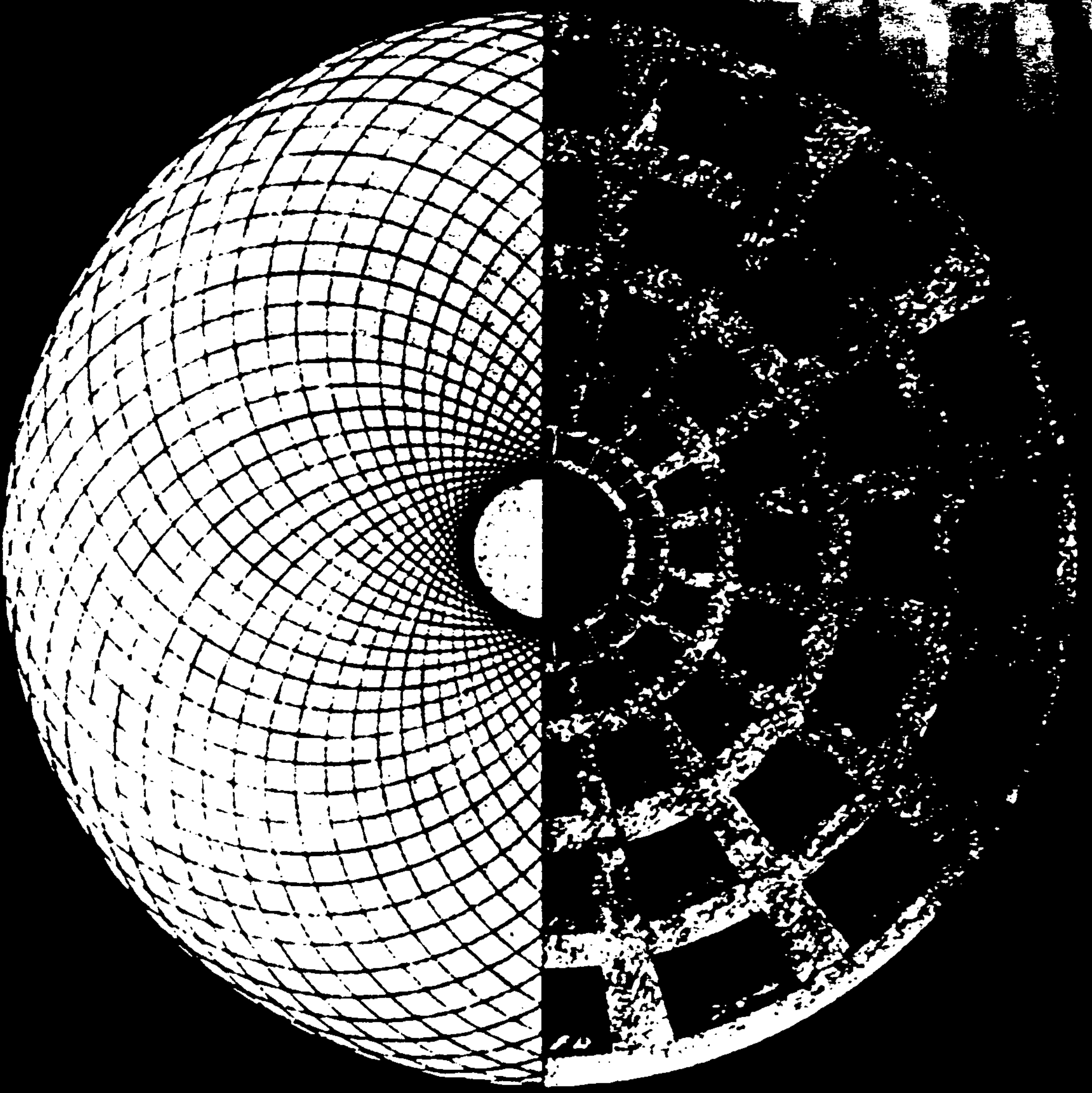
Geometry At Work

Papers in Applied Geometry



Catherine A. Gorini, Editor

MAA Notes Number 53

Geometry at Work

A Collection of Papers Showing Applications of Geometry

Catherine A. Gorini
Editor

Published and Distributed by The Mathematical Association of America

Geometry and Geographical Information Systems

George Nagy
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY
nagy@ecse.rpi.edu

Abstract

Most Geographic Information Systems (GIS) make use only of elementary geometric structures, but their intrinsic geometric simplicity is balanced by the need to manipulate large numbers of spatial entities efficiently and accurately. The emphasis in GIS is on robust geometric algorithms rather than on theorems. In GIS, little importance is attached to worst-case complexity analyses, randomized algorithms, and high-dimensional domains. The compilation of realistic geographic databases is beyond individual reach, therefore I provide pointers to several public-domain data sets. Then I single out three aspects of GIS for more extensive discussion: triangulations and proximity diagrams, geometric visibility, and map projections. I close with a list of additional GIS applications of computational geometry.

1 Introduction

Geometry and geography are daughters of Gaea who have grown apart but are now getting reacquainted. This paper presents a sample of geometric concepts that arise in the context of Geographic Information Systems. Because computerized information systems usually operate on large databases (typically millions of coordinates), the emphasis is on algorithms and procedures rather than on theorems and proofs. With few exceptions, only simple Euclidean constructs are invoked, but most of the tools required to manipulate them efficiently were recently developed in computational geometry.

Geographic Information Systems (GIS) are computerized means of storing, analyzing, and displaying data about the earth, its features, the distribution of life on it, and whatever affects human activity. The broad term GIS came into vogue in the mid-sixties, and may include the data as well as the methodology. GIS are designed to answer queries like: "How many towns with a population of over 10,000 are within 50 kilometers of a railroad line in the state of Nebraska?" or "What fraction of Illinois will be flooded if Lake Huron rises by 20 meters?" or "Which areas of Washington County are both in the floodplain and near railroad lines?"

The design of GIS draws on conventional database management systems (DBMS), operations research, numerical cartography, and computer graphics. Because towns may be represented as points, rivers as curves, and political boundaries as polygons, the essential relationships are geometric in nature. Only a small fraction of the computer code that constitutes a GIS may deal with geometric constructs (most of the code is probably buried in the graphical user interface or GUI), but the two- and three-dimensional geometric (and topological) relations are the conceptual core.

Two aspects differentiate a GIS from an ordinary DBMS such as might be used for payroll or class schedules. First, spatial coordinates do not support a global ordering: there is no unique sorting sequence. Second, spatial entities are not readily viewed as a discrete collection of objects: a continuous view of the world is essential.

Among popular applications of GIS are cadasters (property records); planning and management of transportation infrastructure and operations; mineral exploration; environmental monitoring; flood control and hydro power; retail marketing; the location of manufacturing, service and distribution facilities; and agriculture, pasture and timber management. The underlying spatial framework may be obtained from digitized maps, digital elevation models, and satellite images.

Many commercial GIS systems are available, ranging from simple digital atlases costing a few hundred dollars to elaborate multi-workstation networks for over $100,000. Among the best known ones are Arc/Info, Intergraph, TNT, ERDAS, and Smallworld. Systems designed for utility applications (internal and external plant, conduits, piping, power lines) increasingly resemble computer-aided design (CAD) systems, while those intended for making use of satellite data have complex-image processing components.

Advances in GIS research are presented at conferences.[1] and in three or four specialized journals.[2] Some useful introductory texts are [18, 39, 44, 53, 56] The Federal Geographic Data Committee (http://www.fgdc.gov/) promotes standards and publishes a newsletter. The National Center for Geographic Information and Analysis (NCGIA) at U.C. Santa Barbara distributes a curriculum and teaching materials. Many universities offer a course in computational geometry and several excellent texts are available [9, 29, 35, 40].

In the rest of this chapter, I will provide some sources of geographic data for study and experimentation, and discuss three components of GIS that have both intrinsic interest and solid geometrical foundations.

2 Geodata

In most countries, government organizations collect and disseminate large and medium scale maps, related geographic information, and topographic data in digital form [43]. In the United States, the principal players are the US Geological Survey (USGS), the National Imagery and Mapping Agency (NIMA), the National Oceanographic and Atmospheric Agency (NOAA), the Soil Conservation Service (SCS), and the National Aeronautics and Space Agency (NASA). Mapmakers, like Rand-McNally, Hammond, and Simon & Schuster, distribute highway and city maps in digital form. Specialized firms, like MapInfo and Intergraph, offer software, directory data, and detailed address-location information for computerized mapping and analysis of demographic data for marketing studies.

The USGS maintains an index of digital data on the World Wide Web at http://www.usgs.gov/. Among data sets they distribute are elevation values for the United States recorded at 3 arc-second intervals. Each file contains over one million points and corresponds to a 15-minute quadrangle in the 1:50,000 topographic map series. The NIMA ETOPOS5 dataset contains lower resolution (5 arc-minute) topography including bathymetry, for the entire world. A catalog of other elevation data may be obtained from http://www.geo.ed.ac.uk/home/ded.html.

The staggering amount of satellite imagery collected since the launch of the first Landsat (ERTS-A) in 1972 can be obtained (for a price) from the EOSAT Company in Lanham, MD. Accurate registration and rectification of the data, taking into account differences in surface elevation, requires complex geometric transformations and interpolation, and huge computing resources. Several commercial GIS allow the superposition of recent satellite images on digitized topographic maps. This technique is vital in monitoring changes such as coastal erosion, deforestation, desertification, overgrazing, floods, and forest fires. (Examples of beneficent changes are harder to come by.) Landsat images can be found at http://edcwww.cr.usgs.gov/Earthshots1.00/.

Most of these databases are built on a quasi-rectangular grid, such as latitudes and longitudes, or state-plane coordinates. Some of the older statewide GIS are based on 1 km × 1 km cells. Maps are also digitized into rectangular pixel arrays or bitmaps.

Techniques used for accelerating the processing of grid data include the Uniform Grid method for line intersections that takes advantage of the homogeneous distribution of the data [22], quad trees [46], and R-trees [24]. These techniques are all designed to speed up query processing by taking advantage of the locality of most entities of interest, i.e., their small size relative to the extent under consideration. Alternative methods, described in the next section, explicitly partition the entire extent during a preprocessing step, before any queries are processed. These non-grid-based tessellation methods are probably of greater interest to students of geometry.

3 Triangulated Networks and Proximity Diagrams

Aside from the rectangular grids discussed above, the most common data structures used in GIS are

[1] The International Symposium on Spatial Data Handling (IGU), the International Symposium on Large Spatial Databases (ACM Sigmod), the International Conference on GIS and Environmental Modeling (NCGIA), the National GeoData Forum, GIS/LIS, and the European Conference on GIS (EGIS Foundation) among others.

[2] *Int'l J. of GIS, Cartography and GIS, Cartographica, GIS World.*

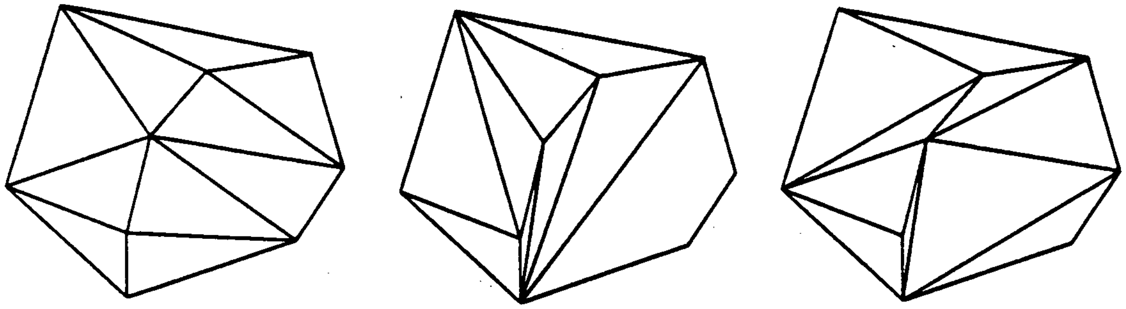


Figure 1: Several different triangulations of a set of nine points enclosed by their convex hull.

based on Delaunay triangles and Voronoi polygons. As we shall see, these two are intimately related.

3.1 Triangulated irregular networks

Instead of simply referring to locations of interest by their coordinates (e.g., latitude and longitude), it is convenient to partition the spatial domain of a geographic information system into mutually exclusive cells. An assembly of triangular cells, called a *Triangulated Irregular Network* (TIN), offers many advantages over more complex cell shapes [14, 19, 38]:

1. Triangles can compactly fill any area: they tessellate the plane.

2. Given a set of points in the plane, simple algorithms exist (try writing one!) for linking them into a TIN [45]:

3. Each cell can be represented by a simple data structure: given three vertices, there is no ambiguity about the edges. (An interesting problem is to devise a data structure where appending a triangle to an edge of the boundary requires adding only the information associated with the new node.)

4. The planar graph formed by the vertices and edges—the triangles are "faces"—has an almost homogeneous structure. On the average the nodes are of degree six, and n points give rise asymptotically to $3n$ edges and $2n$ triangles.

5. When an elevation value is associated with each vertex of a triangulation, the elevation value at any point of the resulting surface can be linearly interpolated from vertex elevations. This results in a piecewise-linear surface model. In contrast, elevations cannot be linearly interpolated in a rectangular grid model because four points overconstrain a plane.

6. TINs generalize readily to higher dimensions, where they are called *simplicial cell-complexes*. These find uses in solid models of mechanical parts, geological strata, and finite-element solutions of differential equations [4, 5, 49, 52].

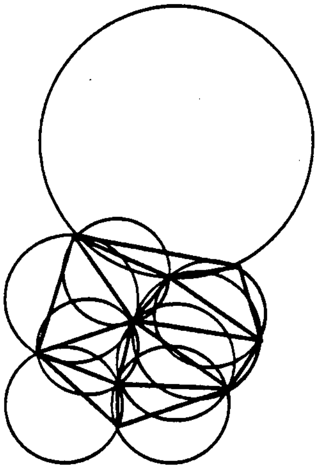


Figure 2: Delaunay triangulation of the nine points with their "empty" circumcircles.

Figure 1 shows several different triangulations of a set of nine points enclosed by their convex hull.[3] Among specific types of triangulation are the *minimum weight triangulation*, which minimizes a cost

[3] The convex hull is easily visualized as a rubber band around the points in 2-D, and as shrink-wrap in 3-D. Writing an efficient algorithm to find the convex hull of a set of points, especially in higher dimensions, is not easy. The computational geometry literature is littered with faulty attempts.

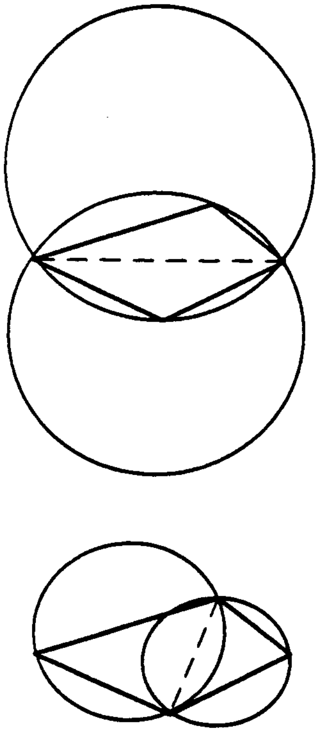


Figure 3: Swapping diagonals to obtain empty circumcircles increases the minimum interior angle.

function defined on the edges, *constrained triangulation*, which requires including some prespecified edges, and *Delaunay triangulation*, discussed below.

3.2 Delaunay Triangulation

In GIS, a particular type of triangulation, called Delaunay Triangulation, is standard. It is named after the author of a paper titled *On The Empty Sphere*. Presented in 1931 to the Soviet Academy of Science and dedicated to George Voronoi, it demonstrated the existence of a triangulation such that the circumcircle of any triangle contains no vertices other than those of its inscribed triangle[4] [7]. Figure 2 shows the Delaunay triangulation of the same nine points with their "empty" circumcircles.

From the point of view of interpolating terrain elevations, the chief merit of Delaunay triangulation is that among all possible triangulations, it maximizes the minimum interior angle of each triangle. This tends to equalize the lengths of the sides of the triangles, resulting in more robust interpolation. The minimax angle property also allows constructing a Delaunay triangulation from any arbitrary triangulation by simply swapping the diagonals of any quadrilateral (formed by two triangles that share a side) if the swap increases the smallest of the six interior angles (Figure 3).

Another unexpected property of Delaunay triangulation (that does not hold for arbitrary triangulations) is a partial ordering with respect to any "viewpoint" in the plane [10, 15]. The triangles can be assigned consecutive integer labels in such a way that an arbitrary point in any triangle can be joined to the viewpoint by a line segment that passes *only through triangles with a lower number.* In the Delaunay triangulation on the top of Figure 4 the viewpoint is one of the vertices (circled) of the triangulation. The numbering shown is consistent with the partial order. For instance, any point in triangle #8 can be joined to the viewpoint with a straight line segment either through triangles #5 and #2, or through triangles #4 and #1. In the bottom non-Delaunay triangulation of the same points, no such numbering exists. This property of Delaunay triangulation (called *acyclicity*) can be exploited for computing the region of the terrain (represented by a TIN) that is visible from a specified viewpoint; see Section 4.

The connectivity of the Delaunay graph has been extensively investigated. It can be shown, for instance, that it contains the minimum spanning tree, which is the straight-line connection among the

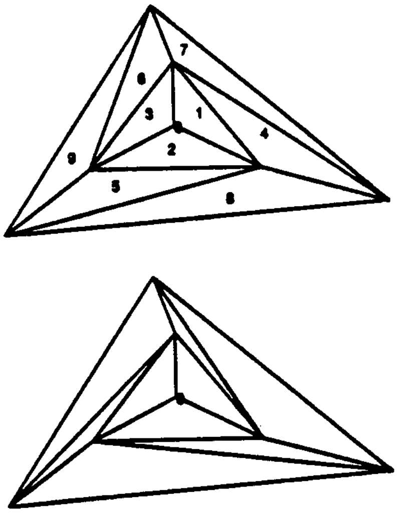


Figure 4: Above: a partial ordering of Delaunay triangles. Any ray from vertex 0 to a point within any triangle passes only through triangles with a lower number. Below: the triangulation is cyclic with respect to the vertex 0: no such numbering is possible.

[4] If the set of points is in general position, i.e., no three points are collinear.

points that has minimum length. It has also been shown that the Delaunay graph is not necessarily *Hamiltonian* (i.e., contains a cycle passing through all the points), but that it is so with high probability [8].

3.3 Voronoi Diagrams

The Voronoi diagram[5] defined by a set of points is a subdivision of the plane into polygons. Each polygon is associated with one of the given points called its *seed* or *site*, and encloses the convex region of the plane closer to its seed than to any other point. It is therefore convenient to think of the Voronoi diagram in terms of the intersections of the halfplane regions nearest to each of every pair of data points (Figure 5). For a scholarly survey of the properties of the Voronoi diagram, see [1].

Voronoi polygons are sometimes called *Thiessen polygons* in geography and *Dirichlet tessellation* in geometry. They can be used for nearest-point problems: if the seeds represent distributors, then to find out to which distributor a consumer is closest, it is sufficient to determine in which polygon the consumer is located (this is also called the *post-office problem*). Voronoi diagrams are also immediately useful for finding the closest pair among n sites, the *largest empty figure*, and *collision-free path-planning*.

The Voronoi diagram is the *straight-line dual* of the Delaunay triangulation. Each edge in the Voronoi diagram is the perpendicular bisector of an edge of a Delaunay triangle. In fact, some algorithms first construct the Voronoi diagram by a divide-and-conquer method [40], then convert it to the Delaunay triangulation by joining every pair of points that share a Voronoi boundary. "Dynamic algorithms" can modify an existing Voronoi diagram or Delaunay triangulation as points are added to the database.

The *kth-order Voronoi diagram* partitions the plane into regions where every point is nearer to a set of k points than to any other point. The number of Voronoi diagrams of all orders on n points is $O(n^3)$. The *Farthest Point Voronoi Diagram*, useful for locating facilities away from undesirable sites, is the Voronoi diagram of order k, with $k = n - 1$.

To determine the closest points to each road, we need a *Voronoi Line diagram*. The partitions here consist of a mixture of line segments and quadratic curves. Other useful extensions are the *Weighted-Distance Voronoi Diagram*, where the distances are weighted according to the site from which they are measured, and the *Voronoi Diagram with Barriers* [21], where a point cannot be considered near a site if there is an impenetrable barrier (e.g., a river) between them. Current research topics include efficient incremental modification of Voronoi diagrams (when points are added or deleted, or when they are in motion), and Voronoi interpolation.

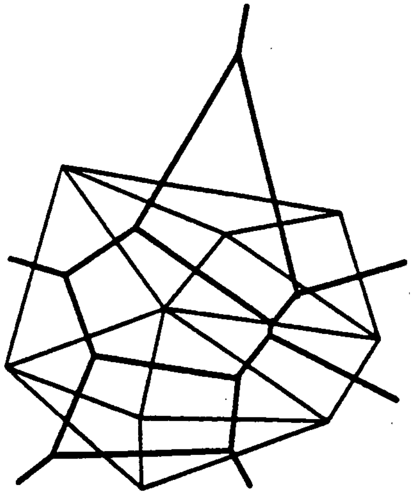


Figure 5: Voronoi diagram of the nine points.

Naturally, the Voronoi diagram also generalizes to higher dimensions. In 3-D, it is the dual of the Delaunay tetrahedralization. However, the computational resources required to compute and store it rise exponentially with the dimensionality. The maximum number of vertices of the Voronoi diagram for a set of N points in d-dimensional space is of the order of $(d/2)!N^{d/2}$. More efficient algorithms are now available to find the nearest neighbors of query points in the high-dimensional vector spaces found in pattern recognition applications.

4 Terrain Visibility

This section explores the application of geometric algorithms to problems involving lines of sight between points on the earth's surface. One might, for instance, compute from which of several alternative observation sites in San Francisco the largest portion of the Bay would be visible. The most important concept here is the *visible region of a viewpoint*, or that part of the surface that can be "painted" by a searchlight located at the viewpoint.

Geometric visibility is an abstraction based only on the intersection with the terrain of the lines of

[5] Named after the Russian turn-of-the-century mathematician.

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.

Digital elevation terrain models (DEMs) provide an abstract representation (*model*) of the surface of the earth by ignoring all aspects other than topography. 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.

Computer-aided visualization of geometric constructs [58] facilitates solving, by inspection, many problems of a geographic nature. Using geometric visibility, however, allows some of these problems to be solved by direct computation instead of inspection. Examples include locating fire towers and microwave transmitters and receivers; siting power 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. Visibility mapping plays a central role in scenic landscape assessment for establishing jurisdictions, quantifying impacted populations, exploiting sources of energy, and planning transportation corridors [13].

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.[6] In addition to computer graphics, spatial data processing and topographic analysis, geometric visibility bears on computational geometry, computer vision, and operations research. References to original research on the topics discussed below may be found in a recent survey by the author [31].

4.1 Basic visibility concepts

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 have the largest visibility indices [23].

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 (Figure 6). 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* (shaded maps) 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

[6] However, visibility in the plane, or polygon visibility, has been a popular topic in computational geometry. Most of the essential results can be found in [33, 34, 48].

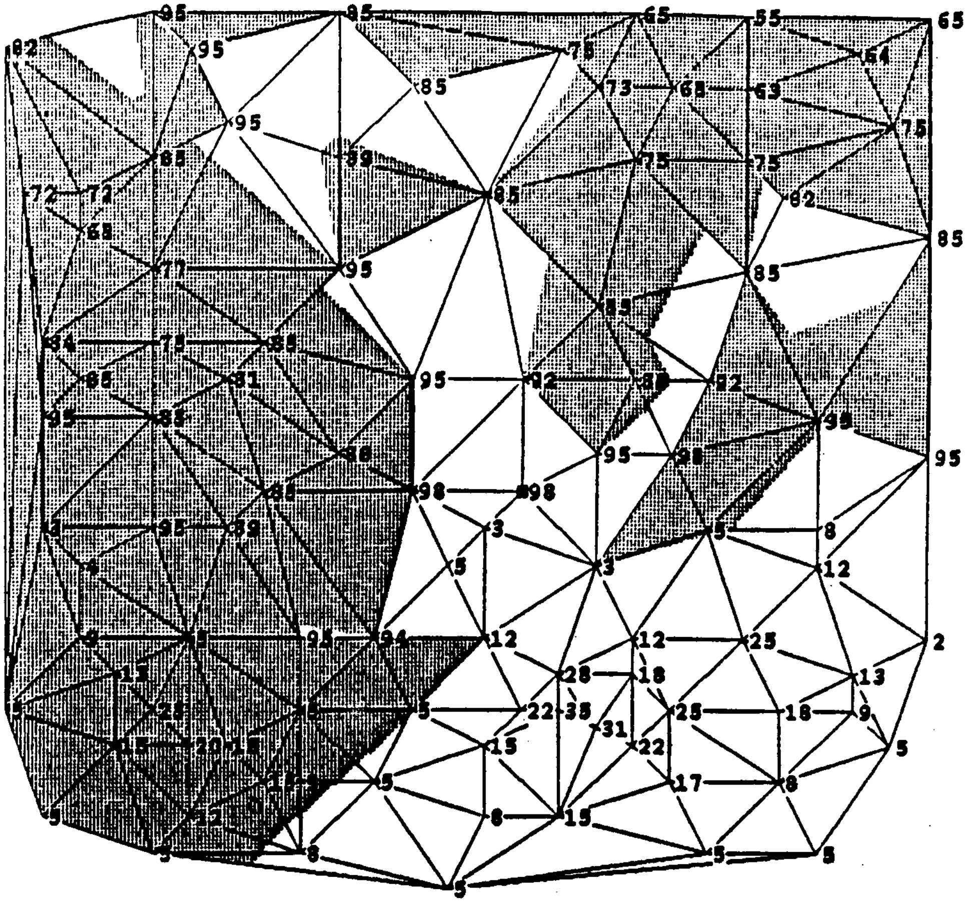


Figure 6: A visibility map. The figure shows the horizontal projection of a terrain on a Delaunay-triangulated irregular network. The dark areas are invisible from the viewpoint near the center, which is marked by a small square.

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.

4.2 Visibility Regions

As we have seen, a visibility map is a projection onto the horizontal plane of the 3-D curves (or line segments) that separate visible and invisible regions on a topographic surface. The boundaries of the regions visible from a given viewpoint, projected onto the x-y plane, may be divided into blocking segments and shadow segments. In a sectional view (a vertical section through the viewpoint) of the terrain, such as Figure 7, these segments are just points on the baseline. From the perspective of the viewpoint, a *blocking segment* represents the transition from a visible to an invisible region (again, projected onto the x-y plane). An example is the first ridge to the right of the viewpoint (projected onto the baseline). A *shadow segment* represents the transition from invisible to visible. In Figure 7, there are two shadow segments, each corresponding to the "shadow" of the ridge to its left.

Blocking segments typically correspond to ridges and shoulder lines that cross a *line of sight* (i.e., a ray through the viewpoint). Shadow segments correspond to a double projection: the orthogonal (vertical) projection on the horizontal datum of the central projection (from the viewpoint) of a ridge (or shoulder line) 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

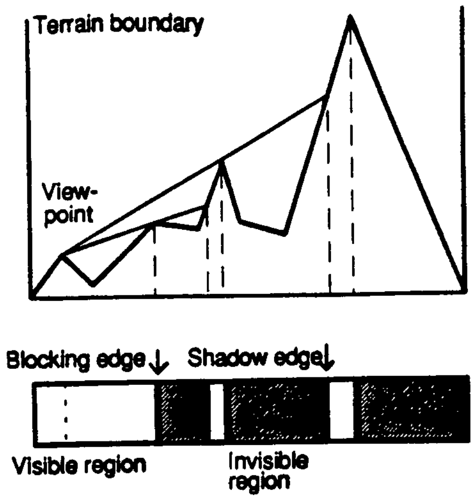


Figure 7: Sectional view: horizons. The viewpoint has three odd right horizons (blocking edges) and three even right horizons (shadow edges) including the terrain boundary. Its only left horizon is the terrain boundary. Terrain segments between odd and even horizons are invisible from the viewpoint, and segments between even and odd horizons are visible.

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 *horizon* is the set of ridges that corresponds to the blocking segments most distal from the viewpoint. It has been shown that the number of segments comprising the horizon is, in the worst case, only slightly supralinear in the number of terrain edges. The boundaries between visible and invisible regions are sometimes called odd and even order *horizons* with respect to the given observation point (Figure 7).

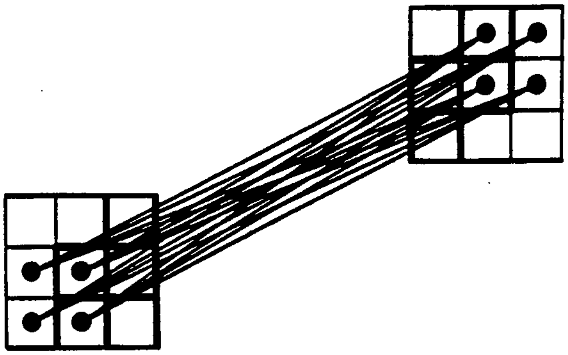


Figure 8: Visibility calculations on a grid. This figure shows the redundancy of computing visibility by following the line of sight from each observation point to each target point. Most grid-based visibility programs eliminate computing some of the redundant intersections of the lines of sight with the edges of the grid cells.

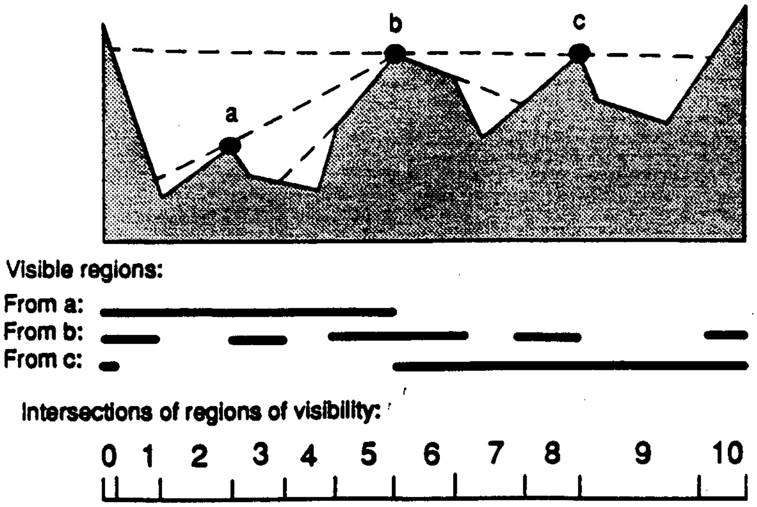
In order to program visibility computations, two questions must be laid at rest. The first question is: What happens beyond the boundary of the terrain, where we have no elevation information? We can assume, for instance, that the terrain is bounded by an infinitely high wall, or that it is surrounded 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.

4.3 Computing visibility

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 [16]. A simple algorithm for computing the complete visibility map on a TIN can be visualized as



	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

Figure 9: 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.

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. With a TIN, a triangle which may cast a shadow on another triangle must be processed first.[7] 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.

Grid-based algorithms all compute the intersection of radial lines of sight with the edges of the grid cells that they intersect [42]. The difference between algorithms lies mainly in the choice of rays (Figure 8). There is little loss of accuracy if the number of rays is reduced, provided that the points are weighted to account for the dispersion of the rays [23].

In the next several sections, we examine several specific applications based on visibility computations.

4.4 Observation points

A *shortest watchtower algorithm* determines the location of the point *with the lowest elevation above the surface* from which an entire polyhedral terrain is visible [47]. 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. It is also possible to determine efficiently whether any particular point is visible from a *single observation point* on or above the surface [6].

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. Topographic applications include the location of fire towers, artillery observers, and radar sites.

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. 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. An example showing a vertical section through the terrain is shown in Figure 9. Variations of the prob-

[7] The necessary ordering property was mentioned in Section 3.

lem include [27]:

a. Find the area visible from a fixed set of observation points.

b. Maximize the area visible from a fixed number 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. Depending on the application, a commanding vista may be called a *military crest* or a *panorama*.

4.5 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.

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. Finding the minimum number of relay towers necessary for line-of-sight transmission between two transceivers can be formulated as a shortest-path search 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.

Now consider the problem of locating relay towers to complete the line-of-sight network between several transceivers (Figure 10). This problem can be solved, under the restriction that the relay towers are located at vertices of the TIN. Here, instead of computing the shortest path, one must find the *Minimum Steiner Tree* (MStT) on the visibility graph. The MStT in this context is the acyclic subgraph of the visibility graph, with the minimum number of (unweighted) "intermediate" edges, that includes all the transceivers. 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 MStT 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 [17].

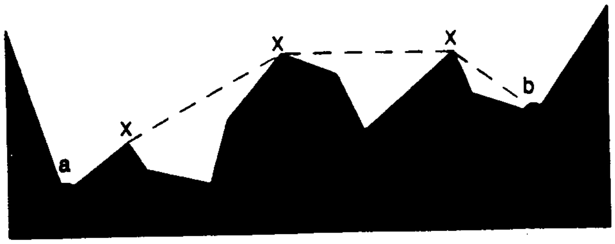


Figure 10: Relay towers for line-of-sight transmitters a and b. The x's indicate the minimum number of transmitters. Although a sectional view is shown here for ease of illustration, in actuality the transceivers and the relays would not be in the same vertical plane.

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).

4.6 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*. 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.

Iwamura and his colleagues demonstrate a GIS 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 [25].

4.7 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 sectional views of visibility-equivalent terrains shown in Figure 11. 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. As a corollary, the visibility functions on a $1\frac{1}{2}$-D terrain are invariant under any projective transformation, including translation and scaling of coordinates, rotations, and affine transformations.

Tools for computing the intervisibility of selected points have long been included in geographic information systems but a recent attempt 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.

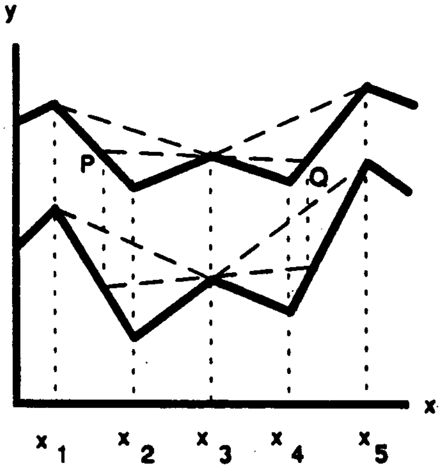


Figure 11: Two visibility-equivalent terrains in section. The locations of shadow edges are identical for every terrain whose elevations satisfy the projective relationship $(y_4 - y_1)/(y_4 - y_3) = (y_5 - y_2)/(y_3 - y_2)$ for each subset of five data points. For example, the shadow edge Q of the observation 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.

5 Map Projections (Geometry, Geography, and Geodesy)

When the extent of the database under consideration is larger than a city or county, we must allow for the sphericity of the planet. Although for internal computer processing a three-dimensional model for locating objects in terms of spherical coordinates is perfectly adequate, for display and mapping purposes there is no satisfactory alternative to projecting the earth's surface on a 2-D plane.

The ideal projection would preserve relative distances, angles, and areas, but no projection can preserve all of these. The distortions increase with the solid angle subtended at the center of the earth. No projection can show scale correctly throughout the map, but there are usually one or more lines on the map where scale remains true. *Equidistant projections* show true scale between one or two selected points and every other point. *Azimuthal projections,* which preserve the direction of all points with respect to the center, are usually projections on a tangent plane.

The most common mapping projections are the Lambert Conformal and the Transverse Mercator. Both of these are *conformal projections,* and therefore preserve local shape. The former projects the earth's surface on a cone centered about the polar axis with its surface tangent or secant to the extent under consideration. It is suited to regions elongated east to west, like Nepal. Distortions can be minimized by letting the cone cut the surface of the sphere at two parallel circles, and tilting its axis away from that of the earth. The Transverse Mercator is based on a cylinder with its axis perpendicular to the polar axis and its surface tangent to a meridian (longitude) near the center of the projected area. It is suitable for regions whose major dimension is north and south, like Norway. The cylinder of the Oblique Mercator touches the earth's surface along a great circle other than the Equator. While the transformation from Transverse to Oblique is easy for a sphere, it is more difficult for an ellipsoidal model of the Earth. As shown in Figure 12, both cone and cylinder are developed into a flat surface by cutting along a single line segment [51].

State Plane Coordinates (used, for example, by Roads departments) consist of a rectilinear grid defined on a map projection. The origin is placed outside the extent, hence locations can be identified

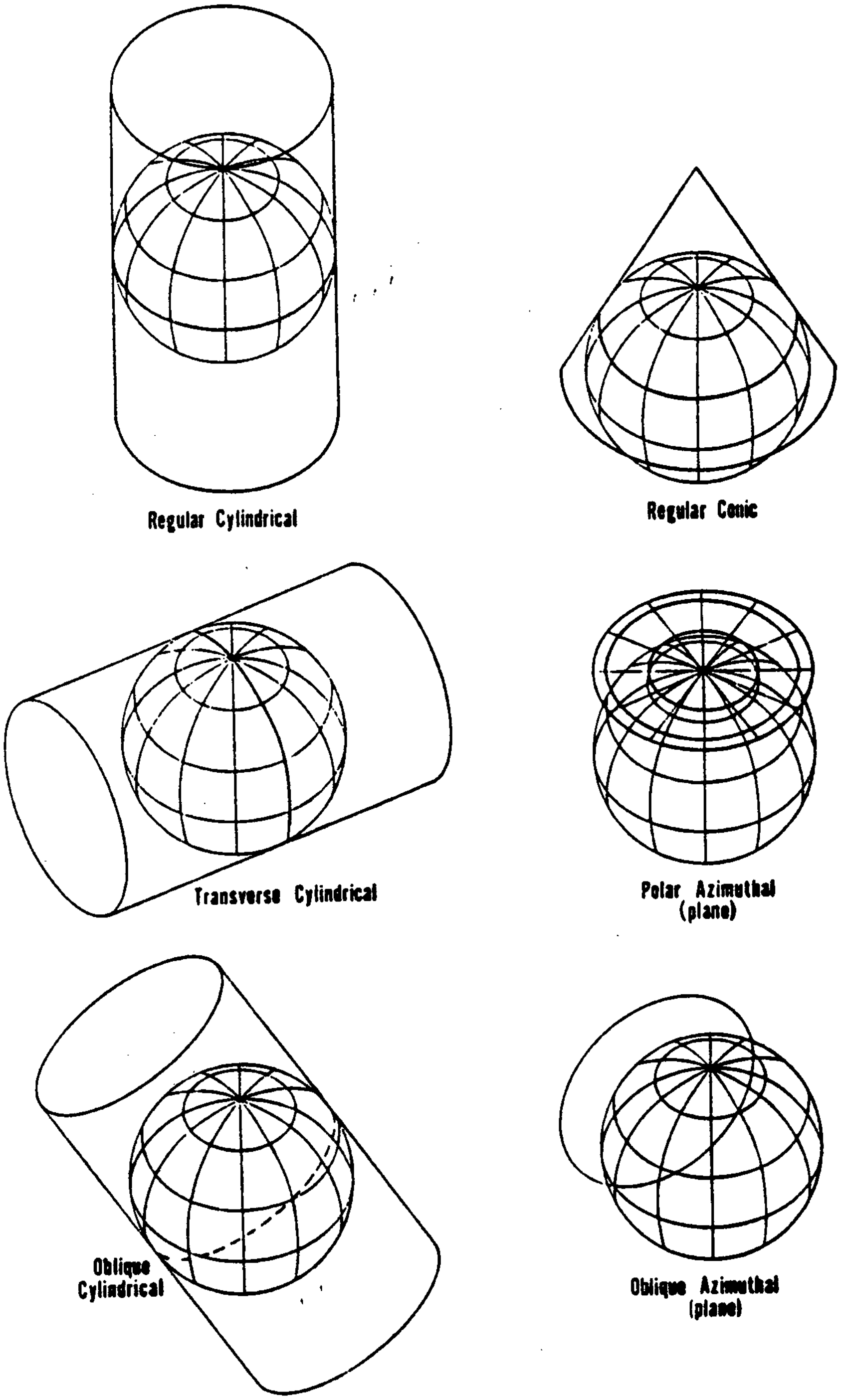


Figure 12: Projection of the earth on a cylinder, a cone, or a plane. For some maps, a central projection is used, but in most cases more complex mappings are necessary to obtain desirable properties.

entirely by pairs of positive integers. In large states, the state plane coordinates are based on more than one projection.

The newest projections are designed for satellite mapping. Since 1972 several mapping satellites have been launched into nearly circular sun-synchronous orbits inclined at 99 degrees to the Equator at about 919 km altitude. Because of the double motion of the satellite and the earth's rotation, these satellites track a continuous S-shaped swath (about 185 kms in width). In the case of Landsat, the course does not return to the same point for 251 orbits (18 days). For mapping, a "dynamic" map projection on which the groundtrack remains true-to-scale is required. The Space Oblique Mercator (SOM) projection was derived for this purpose from the Hotine Oblique Mercator projection (which Hotine called "rectified skew orthomorphic") [50].

This new projection can be adapted to orbits of any eccentricity and inclination. The central line of the SOM closely follows the groundtrack of the satellite, which is not a great circle but a path of constantly changing curvature. Figure 13 shows two orbits of the SOM projection. The forward and backward projection formulas, which require iteration and numerical integration, can be found in [50]. Prior to the introduction of these projections, the preparation of photo-mosaics for mapping purposes from satellite images was more an art than a science.

The shape of the earth is approximated by an oblate ellipsoid of revolution. The difference of 1/300th between the polar and equatorial radius must be taken into account in maps at a scale of 1:100,000 or larger. Furthermore, slightly different ellipsoidal surfaces are required to provide the best fit at different parts of the globe. Therefore the ellipsoid is used with an "initial point" reference location to provide the sea-level datum for mapping. The Clarke ellipsoid, used for the 1927 North American Datum, was supplanted in the 1983 North American Datum by the World Geodetic System ellipsoid. The new datum is based on satellite tracking data.

The distortions in distance, relative angle, and area can be readily computed and plotted for the simpler projections under the assumption of a spherical earth. They can also be studied by direct measurement on existing maps, making use of the known locations of the intersections of latitudes and longitude lines or ticks.

6 Ranging Further

Here we briefly list some operations that are commonly encountered in processing GIS queries and provide references for additional information.

6.1 Line intersections

Given a set of line segments in the plane, described by their endpoint coordinates, it is often of interest to find and report the location of every intersection among them. A number of fast techniques exist for accomplishing this, including line-sweep methods based on presorting the segments in one dimension [2, 3, 40], and uniform-grid techniques that presort the segments in both dimensions [22]. Polygon overlay is a generalization of line intersections [32]. Assume that one map shows diverse agricultural land use in a state, while another shows county boundaries. The task is to determine agricultural land use in each county. A special application of polygon intersection is polygon-to-grid transformation. This is useful, for instance, for transforming digitized soil maps into a cellular data structure for simpler query processing.

6.2 Proximity

Not all proximity problems are solved most efficiently by Voronoi diagrams. For instance, finding the closest, or farthest, pair of points among a set of points requires a specialized algorithm [40]. Shortest route problems, when the path is constrained to a surface, or to some graph imbedded in the surface, form a large class of their own [26].

6.3 Curves

Most GIS approximate curves by a sequence of straight-line segments because few geographic features can be described by simple mathematical curves. Irregular curves can, however, be approximated by *splines* of various families. Unfortunately, the economies of such representations are usually offset by the increased complexity of the processing algorithms. Currently, geometric curves are used only in applications where planar map projections are inadequate, as in long-distance navigation.

6.4 3-D GIS

While most GIS address essentially two-dimensional problems, in geology, mining, and some oceanographic problems the 3-D structure is essential. Rather than develop truly 3-D data

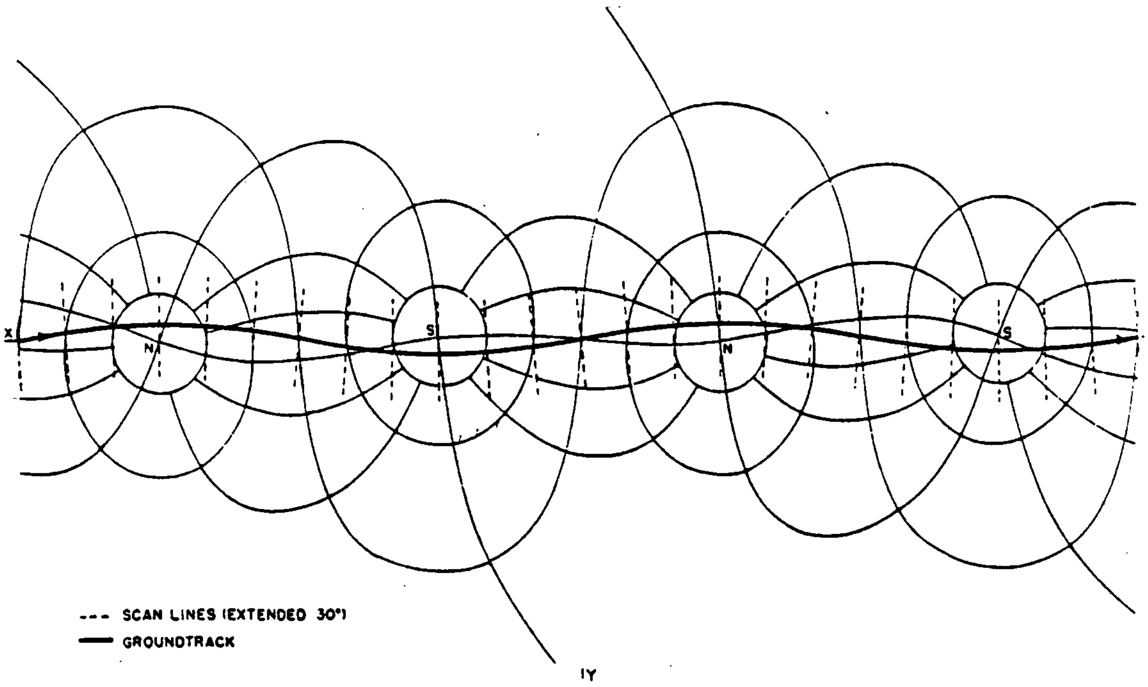


Figure 13: Two orbits of the Space Oblique Mercator projection, shown for Landsat. A narrow band along the groundtrack remains true to scale.

structures, a common approach is to consider parallel layers [41].

6.5 Interpolation

The simplest method to interpolate values defined on a grid is bilinear interpolation. Higher-order interpolation algorithms take into account more than the four immediate neighbors (eight in 3-D) of the query point. Triangulated surfaces allow linear interpolation. However, the resulting surfaces are not smooth, in the sense that they have discontinuities in their derivatives at the edges of the cells. When the data points are not uniformly distributed, Voronoi interpolation allows consideration of values at additional data points.

6.6 Physiographic features

The literature on landforms is remarkably short on algorithmic definitions. Most attempts to automate physiographic feature extraction have been based on discrete approximations to derivatives of the surface [12, 37]. 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. Metrics based on geometric visibility automatically take into account global relations.

Some examples of geometric visibility 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. Even if landforms cannot be determined entirely according to visibility criteria, these may generate useful measures for ranking them.

6.7 Watersheds

The study of watersheds, originated more than a century ago by James Clerk Maxwell, illustrates a mathematically advanced aspect of the study of physiography. Critical points (*peaks*, *pits*, and *passes*), *elevation contours* and *slope lines*, *ridge lines* and *course lines*, and *dales* (a dale is a district whose slope lines run to the same pit) all have precise mathematical definitions in terms of partial derivatives of the terrain function [30]. Elaborate analyses and simulations are routinely carried out in planning dams and canals to mitigate flooding.

6.8 Navigation

Automated path planning (collision-free paths, shortest paths, cheapest paths, safest paths) is already an important aspect of robotics. Preliminary *gross planning* typically takes into account fixed aspects of the robot's environment, while *fine planning* copes with unpredicted obstacles, such as other vehicles, that may be sensed by the robot in motion. As the range of autonomous vehicles expands, path planning may become a routine component of GIS.

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 [55]. The use of horizon lines for autonomous navigation by a Mars Rover has been considered.

6.9 Robust computation

There are two aspects to robust computation. One is the treatment of pathological cases. In geography, one cannot simply assume that points are in general positions and that special cases will not arise. The presence of collinear and co-circular points, exactly vertical or horizontal lines, duplicate entities, and so forth simply cannot be allowed to yield anomalous results or disrupt processing [20].

The second aspect is the accumulation of round-off error. Even if floating-point arithmetic is used, the finite word-length of computers allows only signed-integer computation, and overflow and underflow will result in arithmetic error. Consider, for instance, the intersection of two straight-line segments defined by endpoints placed on a grid. This grid corresponds to all the x and y coordinates that can be represented in a particular computer (if floating point arithmetic is used, then the size of the grid-cells changes in logarithmic increments). The intersection of the two line segments will not, in general, lie on the grid, and will thus have to be approximated by the coordinates of the nearest grid point. Therefore a straightforward arithmetic check for collinearity on the two endpoints of one of the segments and the computed point of intersection, will return an incorrect negative answer.

Arithmetic precision problems of this type can be circumvented by *rational arithmetic* [57]. However, on each successive intersection computation, the number of digits required in the numerator and the denominator tends to triple. An alternative, in the above problem, is to perturb the endpoints of the line segments in such a way that the intersection will fall on the grid. This can be accomplished with a method based on continued fractions, and results in a trade-off between positional accuracy and the preservation of collinearity relations [28]. A third method is demonstrated through the construction of a million-cell Voronoi diagram in [54].

Research on robust algorithms is booming because current methods often introduce slivered regions, region boundaries that double back on themselves or leak, and towns that are accidentally shifted from one bank of a river to the other.

References

[1] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric structure. *ACM Computing Surveys*, 23(2):354–406, September 1991.

[2] H. S. Baird. Fast algorithms for LSI artwork design. *Design Automation and Fault-tolerant Computing*, 2(2):165–178, May 1978.

[3] J. L. Bentley and T. A. Ottman. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Computers*, 28:643–647, 1979.

[4] J. C. Cavendish. Automatic triangulation of arbitrary planar domains for the finite element method. *Int. Jour. Numer. Meth. Engrg.*, 8:679–696, 1974.

[5] J. C. Cavendish, D. A. Field, and W. H. Frey. Automating three-dimensional finite element mesh generation. In K. Baldwin, editor, *Modern Methods for Automating Finite Element Mesh Generation*, pages 61–72. American Society of Civil Engineers, New York, 1986.

[6] R. Cole and M. Sharir. Visibility problems for polyhedral terrains. *J. Symbolic Computation*, 7:11–30, 1989.

[7] B. Delaunay. Sur la sphere vide. *Bull. Acad. Sci. USSR(VII), Classe Sci. Mat. Nat.*, pages 793–800, 1934.

[8] M. B. Dillencourt. A non-Hamiltonian, nondegenerate Delaunay triangulation. *Information Processing Letters*, 25:149–151, 1987.

[9] H. Edelsbrunner. *Algorithms in Computational Geometry*. Springer-Verlag, Heidelberg, 1987.

[10] H. Edelsbrunner. An acyclicity theorem for cell complexes in d-dimensions. *Combinatorica*, 10:251–260, 1990.

[11] H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Discrete and Computational Geometry*, 1:25–44, 1986.

[12] B. Falcidieno and M. Spagnuolo. Polyhedral surface decomposition based on curvature analysis. In T. L. Kunii and Y. Shinagawa, editors, *Modern Geometric Computing*, pages 57–72. Springer-Verlag, Tokyo, 1992.

[13] J. P. Felleman. Landscape visibility. In R. C. Smardon, J. F. Palmer, and J. P. Felleman, editors, *Foundations of Visual Project Analysis*, pages 47–62. Wiley and Sons, NY, 1986.

[14] L. De Floriani. Surface representations based on triangular grids. *The Visual Computer*, 3:27–50, 1987.

[15] L. De Floriani, B. Falcidieno, G. Nagy, and C. Pienovi. On sorting triangles on a Delaunay tesselation. *Algorithmica*, 6:522–532, June 1991.

[16] L. De Floriani and P. Magillo. Visibility algorithms on triangulated terrain models. *Int. J. Geographic Information Systems*, 8(1):13–41, 1994.

[17] L. De Floriani, G. Nagy, and E. Puppo. Computing a line-of-sight network on a terrain model. In *Proc. Fifth Int. Symp. Spatial Data Handling*, pages 672–681, Charleston, SC, August 1993.

[18] S. Fotheringham and P. Rogerson. *Spatial Analysis and GIS*. Francis & Taylor, London, 1994.

[19] R. J. Fowler and J. J. Little. Automatic extraction of irregular network digital terrain models. *Computer Graphics*, 13:199–207, 1979.

[20] W. R. Franklin. Cartographic errors symptomatic of underlying algebra problems. *Proc. Int. Symp. Spatial Data Handling, Zurich*, pages 190–208, 1984.

[21] W. R. Franklin, V. Akman, and C. Verrilli. Voronoi diagram with barriers and on polyhedra for minimal path planning. *Visual Computing*, 1(2):133–150, October 1985.

[22] W. R. Franklin, M. Kankahalli, and C. Narayanaswari. Geometric computing and the uniform grid data technique. *Computer Aided Design*, 21(4):410–420, 1989.

[23] W. R. Franklin and C. Ray. Higher isn't necessarily better: visibility algorithms and experiments. In T. C. Waugh and R. G. Healey, editors, *Advances in GIS Research, Sixth International Symposium on Spatial Data Handling*, pages 751–770. Taylor & Francis, September 1994.

[24] A. Guttman. *R*-trees: a dynamic index structure for spatial searching. In *Proc. SIGMOD Conference*, pages 47–57, Boston, June 1984.

[25] K. Iwamura, Y. Nomoto, S. Kakumoto, and M. Ejiri. Geographical feature analysis using integrated information processing system. In *IAPR Workshop on Machine Vision Applications*. Tokyo, November 1990.

[26] H. Booth J. O'Rourke, S. Suri. Shortest paths on polyhedral surfaces. In *Proc. Second Symp. Theoretical Aspects of Computer Science*, volume 182 of *Lecture Notes in Computer Science*, pages 243–254, New York, 1985. Springer-Verlag.

[27] J. Lee. Analysis of visibility sites on topographic surfaces. *J. Geographical Information Systems*, 5(4):413–429, 1991.

[28] M. Mukherjee and G. Nagy. Collinearity constraints on geometric figures. In B. Falcidieno and T. Kunii, editors, *Modeling in Computer Graphics*, pages 115–125. Springer-Verlag, NY, 1993.

[29] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1994.

[30] L. R. Nackman. Two-dimensional critical point configuration graphs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(4):442–450, 1984.

[31] G. Nagy. Terrain visibility. *Computers and Graphics*, 18(2):763–773, December 1994. Special issue on Modelling and Visualization of Spatial Data in GIS.

[32] J. Nievergelt and F. Preparata. Plane-sweep algorithms for intersecting geometric figures. *Comm. ACM*, 25(10):739–747, 1982.

[33] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, 1987.

[34] J. O'Rourke. Visibility graphs. *ACM SIGACT NEWS*, 24(1):20–25, 1993.

[35] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[36] M. H. Overmars and E. Welzl. A new method for computing visibility graphs. In *Proc. Fourth Annual ACM Symposium on Computational Geometry*, pages 164–171, Urbana-Champaign, IL, June 1988.

[37] 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.

[38] T. K. Peucker, R. J. Fowler, J. J. Little, and D. H. Mark. The triangulated irregular network. In *Proc. ASP-ACSM Symp. on DTMs*, St. Louis, 1978.

[39] D. J. Peuquet and D. F. Marble. *Introductory Readings in Geographic Information Systems.* Taylor & Francis, London, 1992.

[40] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[41] J. Raper. *Three Dimensional Applications in GIS.* Taylor & Francis, London, 1990.

[42] C. Ray. A new way to see terrain. *Military Review*, LXXIV(11), November 1994.

[43] D. Rhind. The information infrastructure of GIS. In *Proc. 5th Int. Symp. on Spatial Data Handling*, pages 1–20, Charleston, SC, 1992.

[44] D. Rhind, J. Raper, and H. Mounsey. *Understanding Geographical Information Systems.* Taylor & Francis, London, 1992.

[45] A. Saalfeld. Triangulated data structures for map merging and other applications in geographic information systems. *Proc. Int. Symp. on GIS*, 3:3–13, November 1987.

[46] H. Samet. *The Design and Analysis of Spatial Data Structures* and *Applications of Spatial Data Structures.* Addison Wesley, Reading, MA, 1989.

[47] M. Sharir. The shortest watchtower and related problems for polyhedral terrains. *Information Processing Letters*, 29:265–270, 1988.

[48] T. Shermer. Recent results in art galleries. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.

[49] P. P. Silvester and R. L. Ferrari. *Finite Elements for Electrical Engineers.* Cambridge University Press, second edition, 1990.

[50] J. B. Snyder. The Space Oblique Mercator Projection. *Photogrammetric Engineering and Remote Sensing*, 44(5):585–596, 1978.

[51] J .P. Snyder. *Map Projections Used by the U.S. Geological Survey.* Number 1532 in Geological Survey Bulletin. US Gov. Printing Office, Washington, 1982.

[52] V. Srinivasan, L. R. Nackman, J. M. Tang, and S. N. Meshkat. Automatic mesh generation using the symmetric axis transformation of polygonal domains. *Proceedings of the IEEE*, 80(9):1485–1501, September 1992. Special Issue on Computational Geometry.

[53] J. Star and J. Estes. *Geographic Information Systems: An Introduction.* Prentice Hall, Englewood Cliffs, NJ, 1990.

[54] K. Sugihara and M. Iri. Construction of the Voronoi diagram for "one million" generators in single precision arithmetic. *Proceedings of the IEEE*, 80(9):1471–1484, September 1992. Special Issue on Computational Geometry.

[55] Y. Ansel Teng, Daniel De Menthon, and L. S. Davis. Stealth terrain navigation. *IEEE Trans. Systems, Man, and Cybernetics*, 23(1):96–110, Jan/Feb 1993.

[56] D. Tomlin. *Geographic Information Systems and Cartographic Modeling.* Prentice Hall, Englewood Cliffs, NJ, 1990.

[57] P. Y. Wu and W. R. Franklin. A logic programming approach to cartographic map overlay. *Canadian Computational Intelligence Journal*, 6(2):61–70, 1990.

[58] W. Zimmerman and S. Cunningham. *Visualization in Teaching and Learning Mathematics.* Number 19 in MAA Notes. MAA, Washington, DC, 1991.