Chapter 2.
3D Surface Representations

The computing community recognizes the tremendous impact that data structures have on the ability to meet the demands of a particular application. Data structures affect retrieval and processing times, degree of accuracy, scalability, and ease with which data may be manipulated, modified, and otherwise enhanced for greater applicability. Therein lies the goal of this work: to define a three-dimensional surface representation — and techniques for developing and manipulating this structure — that can support real-time applications.

Descriptions of spatial data representations abound in the computing literature. Yet the focus of my work has been on polygonal surface representations. My decision to focus on these structures was driven by the requirements of the applications I was working on: primarily cartographic analysis and simulation. Because these applications are only concerned with the surface of objects, I did not consider volumetric representations. Performance requirements of these real-time applications further constrained the set of possible solutions to representations that could be analyzed and rendered quickly. This eliminated most curved surface representations.

This chapter discusses several aspects of three-dimensional surface representations. First is an overview of potential sources for 3D surface data, followed by a discussion of surface characteristics that are critical to the modeling process. Next I present the criteria by which polygonal surface representations may be judged. A review of polygonal surface models concludes this chapter.

2.1. Sources of surface data

Digital surface data may be obtained from a variety of sources. The most common sources are digital terrain models, imagery, and range data.

2.1.1. Digital sources

The most common source of digital terrain elevation data is the gridded digital elevation model (DEM). The U.S. Geological Survey (USGS) produces and distributes DEMs covering the entire United States at a variety of scales. These DEMs are derived from contour maps, stereomodels of high-altitude photographs, or the digital line graph (DLG) hypsography data, another digital source which includes contours [USGS90]. The Defense Mapping Agency (DMA) produces DEMs with world-wide coverage at a variety of scales as well, although distribution is limited. Some of the digital DMA products in this category are Digital Terrain Elevation Data (DTED), World Mean Elevation Data (WMED), and Digital Bathymetric Data Base (DBDB) [DMA90b]. DTED is the most commonly used of these. It forms the basis of many other DMA digital products, and is included in others still. It is also an important source for the weapon system simulator database (Project 2851)[1].

Digitized contours are another source of digital height data, although their use is less common. In addition to the USGS's DLG hypsography data, the DMA includes height data in several of its vector data sources. Digital Feature Analysis Data (DFAD) and Digital Chart of the World (DCW) are two examples. These digital products are generally hand-traced off of charts or images; accuracy of the product depends on the scale of the map or resolution of the photograph.

2.1.2. Models derived from imagery

Surface data may also be derived from stereo imagery. Given precise camera parameters such as location, look direction, attitude, and timing, height at a point on the surface may be measured by finding the disparity between the correlated point locations in the two images [EW85, Coc87]. If the stereo images were captured at different times, other factors such as season, time of day, and cloud cover must be considered.

The Earth Observation Satellite Company (EOSAT) distributes imagery from the Landsat satellites [EOSAT85, EOSAT92]. The Thematic Mapper scanning optical sensors of Landsat 4, 5, and 6 detect six spectral bands at 30 meter resolution, one spectral band at 120 meter resolution, and (on Landsat 6) one panchromatic band at 13-15 meter resolution [EOSAT85, EOSAT92]. Although the sensors only scan the ground directly beneath in a fixed-width swathing pat-

1. Project 2851 is a joint service program to develop a standard simulator digital data base and common transformation program for all future training systems [ISWG88].

tern, the swaths overlap. This provides the different viewing angles required for measuring disparity in the stereo images.

SPOT Image Corporation (Satellite Pour l'Observation de la Terre) satellites carry two High Resolution Visible (HRV) instruments for capturing and recording radiometric values. These instruments employ a pushbroom-type scanner with adjustable viewing angles, which facilitates the capture of stereo imagery [SPOT88]. The detector array records panchromatic radiometric values at 10-13 meter resolution, and multi-spectral values at 20-27 meter resolution. SPOT Image Corporation also recently began distributing 5 meter resolution photographs from the Russian Soujuzkarta program and 15-30 meter synthetic aperture radar (SAR) imagery from the Russian Almaz satellite [SPOT92]. Presumably these may also be used to derive height information.

2.1.3. Range data

Unlike photographic images that measure radiometric reflections from a surface, range images measure the distance from the camera to an object. This true three-dimensional product is used to capture objects in several applications. For example, laser radar is used in conjunction with other sensors for target detection. CAD/CAM systems also make use of this data, using technologies such as the triangulation based White Scanner deployed at Michigan State University's Pattern Recognition and Image Processing Laboratory [LS91], or Cyberware Laboratory, Inc.'s rotating sensor [HDDMS91].

Other truly volumetric digitization devices are also commercially available from companies such as Science Accessories Corporation and Pixsys. These typically measure the three-dimensional position of a locator within a calibrated space.

2.2. Critical surface features

The key to an accurate surface model is having all critical features of the surface represented by the model. Yet this raises the questions: what are the critical features of a surface? and how are they detected?

The literature includes several treatises [PD75, Nac84, Dou86, FB89] which define six surface-specific features as critical: peak, pit, and saddle points; ridge and channel lines; and break points and lines. Peaks are defined as local maxima, pits are local minima, and saddle points or passes are maxima in one direction and minima in the other.

Table 2.1 illustrates the correspondence of these points to two measures of curvature — Gaussian (K) and mean (H) — that are invariant under rotation and translation and therefore useful for feature detection [Bes86]. Note that both curvatures are required to accurately classify the points.

	K > 0	**K = 0**	**K < 0**
H > 0	pit	valley	saddle valley
H = 0	—	flat	saddle
H < 0	peak	ridge	saddle ridge

Table 2.1. Indications of Gaussian (K) and mean (H) curvature

Ridge and channel lines are the slope line paths that connect critical points. A slope lines is the line on a surface of steepest gradient through a point, crossing contour lines at right angles [Nac84, Dou86]. A ridge line starts at a peak, descends to a pass, then ascends to another peak; likewise, a channel or course line starts at a pit, climbs to a pass, and then descends to another pit. Ridge lines also form the boundary between basins or dales, which are regions defined by all points whose slope lines descend to the same pit. Channel lines form the boundary between hills, which are regions defined by all points whose slope lines ascend to the same peak.

Break points are points of high curvature that do not fall into the categories of peak, pit, or pass [PD75, CG87]. Break lines are continuous patterns of break points that define edges on the surface separating regions with very different slope gradients. Examples of break lines and points are the edge of a cliff, the corner of a box, and the base of a building.

Any good surface model must contain these surface-specific points and lines in order to retain accuracy in the representation. Therefore triangle vertices must include pits, peaks and saddle points, and triangle edges must approximate ridge, channel, and break lines. Yet few have actually done this. Instead, most papers in this category focus on the task of classifying individual points with respect to their local neighborhoods, and then identifying the most critical of those for inclusion in a surface model. At ESRI, for example, TINs were generated from a set of “very important points” distinguished by values from a spatial differential high-pass filter [CG87]. Goldgof et al. use Gaussian and mean curvature measures to classify and select critical points for their surface model [GHL89]. Fowler and Little defined an elegant method for tracing ridge and channel lines [FL79], but then discard the linear connections. Peucker and Douglas defined eight classifications of points, based on local neighborhoods, which were then used in a triangulation [PD75]. All of these algorithms then use a nearest-neighbor triangulation scheme to connect the points, literally ignoring the critical lines on the surface.

2.3. Criteria for judging surface models

Before deciding on a “best” surface model, one must first define what “best” means. In the remaining chapters of this dissertation, the quality of a surface model is judged based on several factors:

- Numerical accuracy of the model measured as the difference between truth and the model;
- Visual accuracy of the model verified by inspection and by measuring sliveriness of the polygons, a common cause of visual artifacts;
- Size of the model; and
- Other factors affecting processing speed.

Each of these factors is described in more detail below.

2.3.1. Numerical accuracy

All applications relying on surface data expect some level of integrity in the model. Although each application has different precision requirements — plotting a flight path requires less accurate terrain models than plotting the path of a tank, and scientific visualization and finite element analysis require even more accurate data — they all have accuracy requirements that must be strictly adhered to.

Accuracy is generally measured by finding the distance between some point on the true surface and its projection onto the model. Having found a uniformly distributed set of the differences, these measurements may then be compiled in a variety of ways to formulate a single accuracy value. Some measures which have been used in the past include maximum error found, total, mean, and standard deviation of error [Lee91]. These quick calculations are capable of providing a good indication of how well a surface model fits the truth.

2.3.2. Visual accuracy

Visual accuracy measures how closely the model resembles reality. We perceive this accuracy in terms of generalized shape, receiving clues from the shading and outlines of objects. One might assume that if the surface is mathematically accurate, the shapes will be correct. Yet it is entirely possible for a model to be mathematically accurate, but still not “look right”.

The most obvious way to judge visual accuracy is by inspection. Although such judgements are subjective by definition, they are appropriate for applications designed to provide visual information to a human.

Certain aspects of visual accuracy may also be measured numerically. One factor that affects visual quality is the shape of the polygons. Long and thin polygons — also known as slivers — can produce serious visual artifacts due to aliasing. The way a surface mesh looks may affect more than the visual quality. For some finite element methods, small angles in the surface patches (elements) lead to ill-conditioned linear systems that are not easily solved with accuracy [Fri72]. Still others benefit from or even require a model with no obtuse angles in the triangulation. Yet although slivery triangles — and sometimes even obtuse angles — are generally undesirable, they are sometimes unavoidable given a particular set of data. The only way to eliminate these slivers without impacting accuracy is to split up the undesirable triangles into a series of patches [BGR88, BEG90]. Hence there is a tradeoff: more slivery triangles versus more surface patches and hence more data to process.

Sliveriness of a polygon may be parameterized by the expression

$$\frac{\text{Perimeter}^2}{\text{Area}}$$

The larger this value, the more slivery (i.e. long and thin) the polygon. For example, this formula yields a sliveriness of $4\pi \approx 12.5$ for a circle, 16 for a square, and approximately 20.8 for an equilateral triangle. Once this parameter is found, overall sliveriness of the model may be expressed as a maximum, average, or percent of polygons with angles less than some threshold such as 30˚.

2.3.3. Size of the model

Size of the model is a significant factor for several reasons. First, it affects how rapidly the model can be displayed or analyzed. This is especially critical for real-time applications. For models of very large objects — such as terrain models — another factor is how much disk space the model occupies, and how quickly it may be retrieved from the disk when paging is required. I discuss both aspects of model size here.

2.3.3.1. Number of surface patches

The number of polygons (surface patches) in a model directly impacts the performance of most spatial applications. In analysis of the surface, examining n surface patches takes time proportional to n. Rendering time, on the other hand depends on both the number of polygons and the number of pixels in each polygon. Typically, larger polygons take longer to render. Yet there is a constant overhead associated with each polygon, so that the savings incurred by reducing polygon sizes levels off at some point. Therefore in real-time systems such as flight simulators, it is important to minimize the number of surface patches in the model. Ideally this should be accomplished without seriously impacting accuracy.

Shape of the polygons also impacts performance. In scan conversion algorithms, the edges of a polygon are sorted as a first step. With triangles, sorting is not necessary. Therefore rendering a triangulation is computationally less complex than rendering other polygonalizations.

2.3.3.2. Storage space

One primary advantage of regular tessellations is that for surfaces of the form $z = f(x,y)$ the (x,y) values for each z are implied. Irregular tessellations require all three coordinates to be stored explicitly. This raises the following question: what level of compression is required to make the irregular structures more compact than the regular ones?

Suppose we have a single triangle that fits a set of n data points on the surface. If the height can be stored using h bits and a ground position (x,y) can be stored in g bits, then the triangular representation will occupy the same amount of space as the original grid if

$$nh = 3(g + h).$$

If each coordinate occupies the same number of bits, i.e. $g = 2h$, then the triangle must fit 9 points from the original grid. This means that the triangulation must provide better than a 3:1 compression ratio if it is to occupy less space than the original grid.

However, we can do better than this. If the surface is divided into tiles, each with a given spatial origin (x_0,y_0) and sample rate s — like the grid representations — then the (x,y) component of a point in an ms x ms tile will occupy $(\log_2 m)^2$ bits. Therefore it is conceivable that $g \le h$ giving us

$$g + h \le 2h \;\; |= \;\; n \ge 6.$$

This means that if the triangulation can eliminate at least every other point, it can be as — or more — space efficient as the grid representation. If no such triangulation exists for a given surface, then it should not be triangulated.

2.3.4. Processing speed

Size is not the only factor affecting how quickly an application can do what it must with the surface model. Model representation and organization also impact the ability of an application to operate in real time. For example, some representations may be rendered in real time using readily available graphics hardware. Organization determines how quickly an application can access the data it needs, either in main memory or while paging from the disk. Both of these considerations are discussed below.

2.3.4.1. Model representation

Some models inherently support rapid processing better than others. For example, most computer graphics hardware provides the ability to render polygonal surfaces in real time. Models that lack this inherent ease of use must be transformed to another format for processing. For example, curved surfaces are frequently fit with polygonal surface patches for rendering [SB87, VB87]. Quadtree representations are also frequently triangulated [SB87, TD89, BEG90]. Here the tradeoff is between having to do complex triangulations or maintaining balance conditions that introduce points that would otherwise be irrelevant to the model. No matter how rapid this polygonization occurs, it is always faster to start with a triangulated model.

2.3.4.2. Model organization

Model organization affects the speed with which applications can find the pieces of data required for a specific application. For irregular distributions of data, the related point location problem can be solved in O(log n) time [Pre90, LL87]. Regularly sampled data may be accessed in constant time.

Multiple levels of detail can significantly improve the performance of spatial operations by customizing the model to meet the needs of the moment. For example, multiple levels of detail may be used to filter data so that processing time is concentrated primarily on portions of the model that require it. Coarser levels of detail may also be used to rapidly produce a general picture of or approximate answers about the object. For real-time applications relying on three-dimensional surface data, these levels of detail are critical to performance. As noted by Devarajan and McArthur [DM93], such filtering operations are best supported by levels of detail in a tree structure where pruning may be done without affecting the continuity of the surface. It is also important that the coarser levels of detail omit only the least salient features [TSDB88].

Model organization may also impact data retrieval times. When paging in pieces of a large surface model — such as terrain for flight simulators — it is important to limit the number of disk accesses as well as the volume of data that must be retrieved [Nie90]. Both disk accesses and volume may be limited by considering spatial coherence, so that points that are close on the model are also close on the storage media.

2.4. Surface tessellations

Surface tessellations or tilings approximate the surface with a mosaic of planar patches. In the world of digital spatial data, polygonal tilings are especially popular because they are so simple to render. Several commercial computer platforms provide graphics hardware that will rapidly render polygonal surface meshes. Polygonal surface tessellations are commonly used for a wide variety of applications including scientific numerical simulation calculations [Coo90], civil engineering analysis in geographic information systems [ESRI90, EH91, WH91], simulations [ISWG88, FEKR90, Sou91], finite element analysis [Ban90], fluid flow analysis [HH90], and animation [VB87].

This section surveys surface tessellations, especially triangular tessellations. Triangulations are the three-dimensional representation of choice for many of the applications that rely on surface tessellations. Simplicity of form contributes to the popularity of triangulation models, making them easy to define and manipulate. Removing regularity constraints compounds the benefits of this representation: because any three points on a surface define a triangular patch, triangle vertices and edges may be positioned such that they correspond to the critical points and edges on the surface. Thus data points may appear sparsely in smooth regions and densely clustered over rough or irregular terrain. This permits greatest accuracy with the least data. Irregular triangulations are affine transformation invariant for the affine transformations: translation, rotation, and scaling. Irregular triangulations may also be developed from a greater variety of data sources including contour maps, stereo imagery, and gridded digital elevation models. Important features such as ridge lines, drainage, coastlines, and political boundaries may be strategically added to the model with ease.

2.4.1. Planar tessellations

Planar tessellations partition a finite region into a series of smaller polygons. Planar tessellations are generally created to solve some geometric problem involving the plane. Some examples are Voronoi diagrams and other regions for efficient point searching and nearest-neighbor calculations. Other tessellations are generated to simplify the visibility problem within a polygon. Raster images may be thought of as regular planar tessellations, where each polygon (pixel) represents some color or value. Quadtrees operate on a similar principle, clustering similar regions together for more efficient storage.

Many papers on triangulation tend to focus on the two-dimensional partitioning problem in the area of computational geometry. Most of these algorithms seek to either reduce sliveriness of the triangles, or reduce computational complexity. A Delaunay triangulation, the straight-line dual of the Voronoi diagram [PS85] produces non-slivery triangulations by connecting nearest neighbors in a planar graph. Reference [BGR88] describes an algorithm for triangulating a polygon with the constraint that no obtuse angles are allowed. A survey of greedy, Delaunay, and optimal triangulations of isolated points (the last of which is still an open problem) is found in [WP84][1]. A greedy triangulation algorithm for polygons using dynamic programming is outlined in [AHU74], while others such as [GJPT78, FM84, CTV89] continue to work on algorithms with ever lower computational complexity.

2.4.2. Triangulations of 3D objects

Many of the earliest attempts to tessellate three-dimensional surfaces just extended the idea of the planar tessellations. By simply attributing the vertices with a height value, these tessellations readily model surfaces represented by the form $z = f(x,y)$. One of the most common examples is the regular grid tessellation of the digital elevation models (DEM). Here, elevations are measured at regular intervals also known as posts. Regular tessellations are attractive because they are simple to generate, produce regular polygons with consistent convex shape, and may be searched in constant time. Digital elevation grids or matrices are the most commonly available and used terrain models. However the grid structure does not

1. The optimal triangulation is that which has the minimal sum of edge lengths [MZ79].

reflect the behavior of the surface being modeled, and therefore tends to contain too much or too little information. For example, high resolution grids violate the requirement for economy of information, characterized by unnecessarily large quantities of data which can grind processing to a halt. In low resolution grids, accuracy suffers as critical features change shape, shift, or disappear altogether [Sca89b]. Rigid grid structures also do not accommodate addition of irregular critical lines or elevation matrices at different resolutions with different orientations.

Recognizing these drawbacks, many developers have chosen instead to model surfaces with triangulated irregular networks. The triangulated irregular network or TIN was first introduced as a digital terrain model by Peucker et al. [PFLM77] to remedy the problems inherent to regular grids, i.e. that terrain is highly variant and no one resolution is ideal for all. Irregular triangulations are also able to accommodate input of multi-resolution data and local updates for dynamic terrain [WPZ92].

Numerous algorithms have been proposed for triangulating a set of irregularly spaced points on the surface being modeled. Aside from some notable exceptions [MW82, MZ79] most are variations of Delaunay triangulation. References [Wat81, Dwy87] describe only a few of these. DeFloriani's more recent paper on hierarchical triangulations [DeF89] — based on a search structure for triangles [Kir83] — proposes a stepwise refinement of the levels of detail using Delaunay triangulation. The trouble with Delaunay triangulation for surfaces is that it solves geometric locality problems, ignoring basic terrain characteristics. It may therefore introduce errors by producing lines that contradict the critical ridge, valley, and break lines along the terrain [Chr87]. This problem is illustrated in figure 2.1. In this picture, different shadings (with light outlines) represent different contour levels, with darker values representing lower elevations. Results of Delaunay triangulation are drawn with heavy lines. As shown, this technique of connecting nearest neighbors forms an artificial bridge over the ravine. A truly accurate data base must respect the interconnections of points along critical lines.

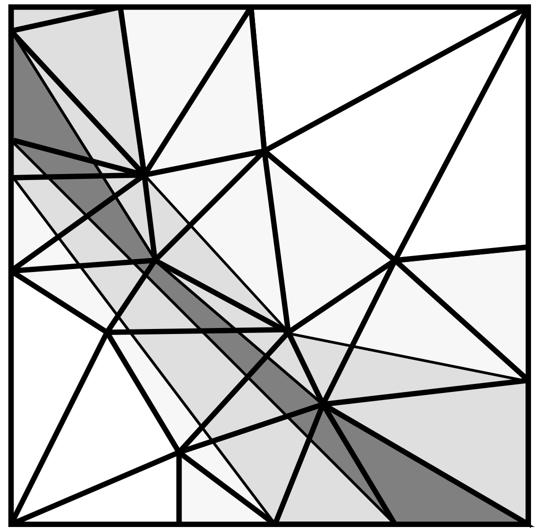

Figure 2.1. Delaunay triangulation can introduce artificial features, like this artificial bridge over a ravine

Triangulations of critical line graphs — i.e. surface points with some initial connections — are more likely to produce accurate models because the lines describing surface topology are included in the final triangulation. Some papers such as [CS78, DG82] deal with triangulating cross-sections from tomographic scans, although the methods of both of these papers require human intervention when the contours get complex. Christensen [Chr87] proposed a fully automated technique which produces a good triangulation but doubles the number of points and polygons in the model. Other triangulations of cartographic critical lines have also been proposed [Sca89a, Che89].

2.4.3. Hierarchical Tessellations

Hierarchical models compound the advantages of surface tessellations in several ways. Hierarchical levels corresponding to specified levels of detail represent generalizations of a surface. Using these generalizations can improve performance of operations such as retrieval, spatial operations, and display. If the hierarchy takes the form of a tree, it may also be used to improve search times and support multi-precision views and zoom.

2.4.3.1. Regular Hierarchical Triangulation

Recent papers [Goo89, Fek90] propose to represent the entire planetary surface with a quadtree-like hierarchy of regular triangular tessellations. This is an excellent scheme for dividing huge data bases into manageable areas of interest (AOIs) which may be geo-referenced in constant time. However, the placement of points in a regular tessellation is independent of the surface topology. Hence coarser levels of detail can distort or entirely miss important terrain features, and finer levels of detail can cause unnecessary bottlenecks by producing large numbers of triangles where a few would do as well. Figure 2.2 illustrates the former problem. This picture shows a perspective view of a surface which has been modeled with a set of regularly spaced points selected at three different sampling rates. The upper-left frame shows the finest sampling rate, while the lower-right frame shows the coarsest. Here, changing the sample rate changes the character of the surface so much, the hill in the foreground appears to actually move.

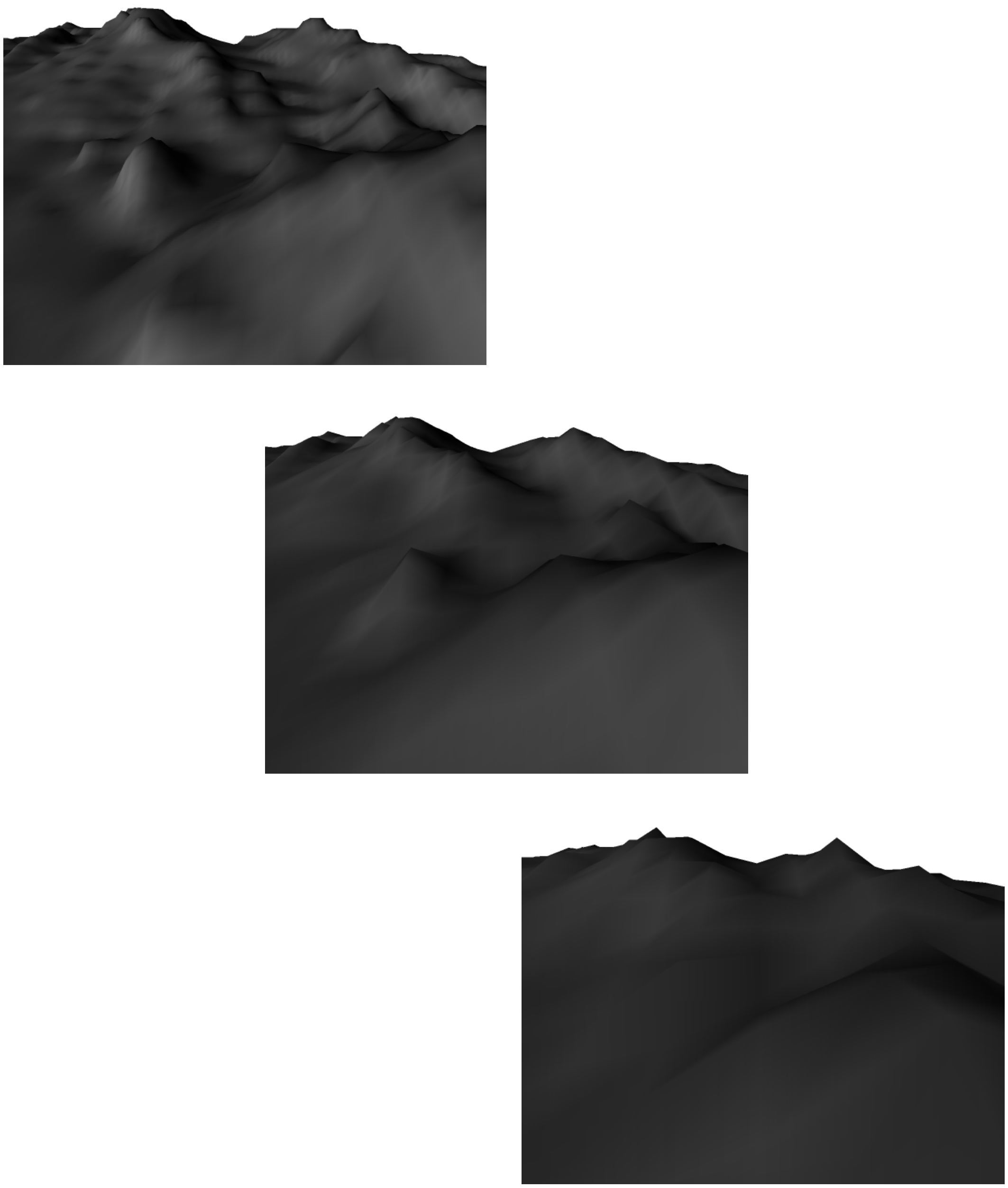

Figure 2.2. Coarse sampling rates can cause features to shift or disappear

2.4.3.2. Regular Hierarchical Triangulation

Several projects have attempted to collect irregular triangulations in hierarchical levels of detail. The U.S. Government's Project 2851, recognizing the advantages of TIN models over grid models, has adopted the surface triangulation technique described in Fowler and Little's paper [FL79]. This technique uses a clever algorithm for finding critical points along ridge and channel lines, but then triangulates these points with Delaunay's method. Because the resulting triangulation does not produce a good surface fit, the paper then suggests iteratively adding points where the surface deviates farthest from the actual data. However, a Project 2851 report [Luf89] indicated that in some iterations this actually generates more errors in the model with more points. Another shortcoming of the Project 2851 model is that level of detail TINs are generated independently of one another. Hence these TINs will not support irregular level-of-detail islands, and provide no guarantee of continuity between two different resolution TINs for the same area. Finally, spatial search on these isolated levels of detail take O(n) time.

In recognition of these issues, DeFloriani et al. have suggested two hierarchical triangulations [DFNP84, DeF89] that retain their triangular nature while providing the advantages of a hierarchy. The earlier paper [DFNP84] is important because it was one of the first to propose triangulation refinement for approximating three-dimensional surfaces. The algorithm builds a hierarchy from the top-down, splitting each triangle at the one central surface point that deviates farthest from the plane of the triangle. A triangle is split by connecting this point to its

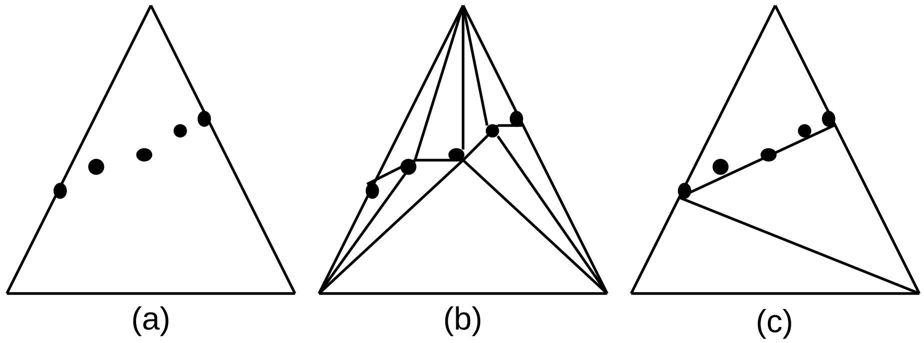

Figure 2.3. Simply splitting a triangle at its point of greatest error — and ignoring surface coherence — can produce poor approximations

corners. Yet these triangles rapidly degenerate into a mass of thin sliver-like triangles, producing very distracting artifacts. As with other triangulations of isolated points, this algorithm ignores the coherence of cartographic features such as valleys or ridges. Figure 2.3 shows the results of ignoring such coherence. Assuming that the high frequency points define a ridge (a), iteratively splitting at the point of greatest error produces a mass of slivery triangles (b). With this solution, integrity of the data suffers. A better solution is to approximate the ridge with a single edge (c).

Acknowledging this severe shortcoming, DeFloriani suggested a second model [DeF89] in which a single point subdivides several triangles instead of just one. Although this alleviates the problem of slivers, it doesn't address the integrity issue. Re-triangulating a cluster of triangles, based solely on the error of a single point and without concern for the critical lines, may correct small errors at the expense of introducing more serious errors in the form of false critical lines. These must then be corrected by adding more points.

2.4.3.3. Hierarchical Triangulation vs. Quadtrees

Certainly triangulations are not the only data representation available for modeling three-dimensional objects. Another hierarchical representation proposed for this use is the quadtree [Sam90a, TD89]. Its regularity allows us to infer ground coordinates rather than store them explicitly, thereby reducing storage space. Yet quadtrees are far more adaptive than grids, retaining more critical features at all levels of detail. Encoded with quadcodes, neighboring nodes may also be inferred [LL87]. However, this regularity also introduces problems that irregular triangulations are free of. First, quadtrees do not meet invariance constraints. Because placement of the vertices in a quadtree is dependent on the frame defining the area of interest, moving that frame can result in a very different model. Consider, for example, figure 2.4. In the leftmost model only one split is required to produce homogeneous regions. When the frame is shifted so that the box is centered within it, all quadrants must be split again to produce the homogeneous regions. With the frame shifted even further, many more splits are required to make the regions homogeneous.

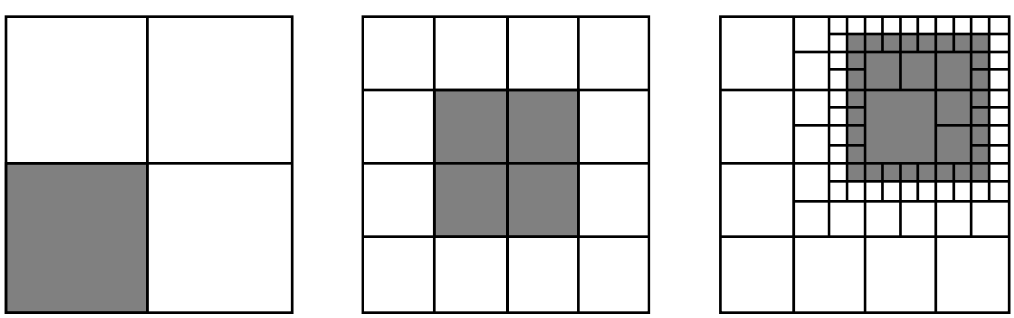

Figure 2.4. Quadtree models are dependent on the orientation of the frame

A second problem caused by this frame dependency is the fact that vertices and edges in a quadtree model do not necessarily correspond to critical points and edges of the surface. Patches that do not fit the surface adequately are always split in the center, regardless of whether that represents a critical point on the surface or not. The result can be a model that contains more surface patches than necessary. This is also evident in figure 2.4.

Quadtrees are generally triangulated anyway to form a continuous mesh for rendering and analysis [SB87, VB87, TD89]. This is a non-trivial task unless either balancing conditions are imposed [BEG90] or incomplete triangulations are allowed. With the latter option, vertices are forced to lie on any edge where high-resolution nodes adjoin low-resolution nodes.