Right-Triangulated Irregular Networks

William Evans*
Department of Computer Science
University of Arizona

David Kirkpatrick†
Department of Computer Science
University of British Columbia

Gregg Townsend
Department of Computer Science
University of Arizona

Abstract

We describe a hierarchical data structure for representing a digital terrain (height field) which contains approximations of the terrain at different levels of detail. The approximations are based on triangulations of the underlying two-dimensional space using right-angled triangles. The methods we discuss permit a single approximation to have a varying level of approximation accuracy across the surface. Thus, for example, the area close to an observer may be represented with greater detail than areas which lie outside their field of view.

We discuss the application of this hierarchical data structure to the problem of interactive terrain visualization. We point out some of the advantages of this method in terms of memory usage and speed.

1 Introduction

In this paper, we describe a method for approximating a surface which is presented to us as a two-dimensional array of height values. We assume that the height value in location i, j of the array is the true height of the surface at location x_i, y_j where x_i and y_j are easily derived from the indices i and j. Thus we ignore questions of error in the input. For our purpose, the surface is defined by the values in the height array. Of course, this leaves most of the surface, those points with x, y coordinate not equal to x_i, y_j for some i, j, undefined. We make no assumptions about these intermediate points. We measure the goodness of our approximations only against the elevations in the array.

Such surface representation are common and can be quite large. For example, a 10 meter resolution 7.5 minute digital elevation model (DEM) of Mount Rainier, Washington [1] has $977 \times 1405 = 1,372,685$ data points. Combining several of these map sheets, which is necessary, for example, in accurate watershed calculation, can result in arbitrarily large data sets.

Computations involving such data sets are very expensive. Both memory and time used by the analysis algorithm can be prohibitively large. In such a case, we cannot use the entire surface representation. Either the surface must be subdivided into manageable pieces, which only works if the analysis can be localized to these pieces, or we must use an approximation to the surface in place of the original. Of course, the approximation should be constructed so that the answers we obtain in analyzing it approximate the answers we would obtain for the true surface.

Surface approximation of regular gridded DEMs typically possesses one of two forms. Either the approximation is itself a regular gridded DEM, a regular *sub-grid* of the original data, or it is a

*Research supported by NSERC Canada International Fellowship and NSF Grant CDA-9500991.
†Research supported by NSERC Research Grant A3583.

triangulated irregular network (TIN), a possibly irregular subset of the original data points whose projection into the x, y-plane is triangulated.

The approximation form we describe in this paper lies somewhere between the regular sub-grid and the irregular TIN. It is more regular than the TIN and more flexible than the sub-grid. Like the TIN, it is a triangulated subset of the original data points, but, unlike the TIN, the subset is not arbitrary. The subset is such that all triangles are right angled and isosceles. Consequently, we call this form of approximation a right-triangulated irregular network (RTIN).

The main benefit of the RTIN approximation is not in providing a single approximation to a surface, but rather in providing a framework of many approximations at varying levels of detail.

A given approximation may be well suited to one type of approximation but not another. An approximation that represents ridge lines very accurately may not be the best approximation for determining wildlife habitat. In order to form an appropriate approximation to the surface, we need to know for what purpose the approximation is intended. It is time consuming and often impractical to create and store an approximation for each intended application. Often a better solution is to devise a general framework for approximation that can yield approximations tailored to particular applications. The problem then becomes calculating the appropriate approximation from the framework. Extracting an approximation from an existing framework has the potential advantage of providing tailor-made approximations much more rapidly than constructing a suitable approximation from scratch. Of course, this works well only if the framework can provide an appropriate approximation quickly. This is critically important when many different approximations are required in rapid succession.

The framework we propose in this paper is a hierarchy of approximations. It is a hierarchy in that the framework contains a range of representations, from coarse to fine. However, the representations are not segregated. The hierarchy can provide a single approximation that mixes coarse- and fine-level representations. This allows us to obtain high levels of detail in certain regions of the surface without forcing us to represent the entire surface at this high resolution. For example, in order to accurately approximate elevation, mountainous regions may require much finer detail than flat plains.

An additional benefit to having the entire hierarchy of approximations is that it can make tasks such as point location faster than they would be if one had only a single approximation.

The disadvantage to keeping a hierarchy of approximations is that it requires more storage than a single approximation. One of the main goals of this work is to demonstrate a practical and space efficient approximation hierarchy. Our implementation of the RTIN hierarchy requires less than 14 bytes per original data point. This includes two bytes for the height value itself[1]. Though seven times more expensive than the original data set, the hierarchy provides an extensive range of approximations.

To illustrate the utility of the RTIN hierarchy, we describe its implementation for interactive visualization of a surface. The goal of interactive visualization is to show what a user would see as they "fly" or "walk" over the surface. The problem is that the surface is too detailed to render at full resolution at a reasonable speed. Thus, the surface must be approximated in order to reduce the time needed to update the display. A single approximation, independent of the eye position, could be used, but the resulting image may be unacceptably coarse in areas close to the eye. Our approach is to adapt the approximation to the location of the eye so that regions close to the viewer are at high resolution while distant or non-visible regions are more coarsely represented.

The demand for varying levels of detail within one approximation must be answered carefully. Areas of high detail must match with adjacent areas of low detail in such a way that the surface

[1]Our input is U.S. Geological Survey DEM data which uses two bytes to represent an elevation.

remains "continuous". Discontinuities are very noticeable in interactive applications.

Interactive visualization is a good test of our hierarchy since it requires many of the desired properties: we must be able to choose an appropriate approximation rapidly; the approximation must allow different levels of detail at different locations on the surface; and the surface must be a continuous surface after interpolation (gaps in the surface can be glaringly obvious).

The focus of this paper will be on describing the RTIN hierarchy and the data structure used to implement it. Later sections of the paper will detail the performance of the hierarchy both as a solution to the interactive visualization problem (section 7.2) and as a way to represent a single (non-hierarchical) surface approximation (section 7.1).

We start by reviewing a portion of the large body of work that has been done on surface approximation. We then review the two popular surface representations, the sub-grid and the TIN, and describe hierarchies based on these representations. This sets the stage for our discussion of the RTIN hierarchy of approximations (section 5).

2 Related Work

A great deal of work has been done on the approximation of surfaces, and, in fact, on the particular problem of *multi-resolution surface modeling* of which hierarchies are one type. A recent survey by De Floriani, Marzano, and Puppo discusses many of the advances in this field [2]. This survey does not refer to a large body of more recent work that has appeared in SIGGRAPH96 [3, 4] and EuroGraphics96, both of which had sessions on multi-resolution surface modeling. A recent tutorial by Puppo and Scopigno [5] covers this more recent work as well as a wide range of surface simplification techniques.

The most relevant work to our approach that has appeared in the surface approximation literature is the work by Lindstrom et al. [6] and Puppo [7, 8]. Both independently discovered hierarchies similar to the one discussed in this paper. We discuss and compare their approaches to ours in section 8.

Another source of related work comes from the study of general lattice structures. Bell et al. [9] cite work on crystal lattice structures by Šubnikov in 1916 [10] and Laves [11] in 1931 that defined various planar partitions including the one upon which our hierarchy is based. These partitions have been studied as a means of addressing spatial regions with application to image encoding [12, 13] and adaptive mesh generation for finite element solvers in two and higher dimensions [14, 15]. Hebert appears to be the first to describe non-uniform versions of the particular partition discussed in this paper, though the concept is closely related to the quad-tree data structure discussed by Samet [16] and the the *restricted quadtree* of Von Herzen and Barr [17].

3 Sub-grid and TIN

Approximations of height data typically come in one of two types: sub-grid or TIN. A sub-grid is of the same form as the input array of height values. It is an array of height values at regularly spaced x and y coordinates. Since the sample points are regularly spaced, it is possible to calculate the true x, y coordinates of a height value from its array indices. Thus, the only storage needed is for the height or z values of the sample points. To obtain the height at x, y coordinates not in the sample set, one uses some form of interpolation. In fact, what is often done is that the four sample points forming the smallest square containing the x, y position are used to determine the height at x, y via bilinear interpolation. The regular spacing of the sample points makes it easy to

determine the closest sample points to a given x, y position. Alternatively, one might choose to use linear interpolation based on a triangulation of this smallest square (arbitrarily picking one of the square's two diagonals to split it into two triangles).

The sample points of a TIN (Triangulated Irregular Network) are an arbitrary subset of the original data points. In order to specify this subset, the x, y coordinates (i.e. the indices in the input array) of each sample point in the TIN must be stored explicitly, in addition to the z value. The sample points, without their height component, are triangulated (typically using a Delaunay triangulation) and linear interpolation is used to obtain the height at an arbitrary point (x, y) from the heights of the vertices of the triangle which contains the point.

The advantage of the TIN is that the sampling can be nonuniform: large featureless regions can be sampled at a coarser resolution than regions with a great deal of variation. The disadvantage is that a TIN requires more storage for the same number of sample points; a 3:1 ratio if x, y, and z values require the same precision. The storage disadvantage becomes worse if one requires adjacency information for the triangulation; if, for example, one needs the ability to traverse the surface approximation from triangle to adjacent triangle.

The factor of three difference in storage requirements has led to an unfavorable view of TINs. Kumler [18] argues that to achieve a certain degree of accuracy in the representation of a height field, sub-grids require less storage space than TINs. However, he then states, "I did not expect these results. My intuition led me to believe, and I continue to believe, that the irregular TIN model is, in general, a more efficient method for representing an irregular natural surface." He then goes on to explain his conviction by claiming that methods to construct more efficient TINs will be developed in the future. Our results comparing error measures to space usage for sub-grids and TINs may shed some light on this issue (section 7.1).

4 Sub-grid and TIN hierarchies

Both sub-grid and TIN approximations may be organized into hierarchies. A hierarchy of sub-grid approximations is obtained by varying the spacing of the regularly sampled grid points. For example, one level of the hierarchy may contain every other sample point from the input array while a coarser level may contain every fourth sample point. Typically, each level of the hierarchy is a regularly subsampled approximation of the next finer level. Thus, the amount of storage needed for the entire hierarchy is just the amount needed for the finest level of approximation. The price of such small storage is the rigidity and uniformity of the sampling pattern.

A TIN hierarchy allows varying levels of detail within an approximation at the expense of more storage. There are several hierarchies based on irregular triangulations [19, 20]. One may roughly divide them into those that preserve the boundaries of coarse triangulations in going to finer triangulations, and those that do not. In the first case, the hierarchy is organized in a tree structure; each node represents a region, and the children of a node represent the regions into which the node's region is partitioned in going from one level to the next finer. Refinement is typically performed by choosing within each region a point from the original input data, adding it to the set of sampled points, and retriangulating (maintaining the previous boundaries). Such a scheme tends to create long, thin triangles which may provide a less accurate approximation than shorter-edged, larger-angled triangles. On the other hand, any level of detail can be achieved in one area of the triangulation without affecting other areas. In terms of the tree structure, any subtree containing the root represents a valid surface approximation.

In the second case, one can choose to re-triangulate at each level using, for example, Delaunay triangulation. In this case, it is not in general possible to refine a particular region independent of

other regions, i.e. combining pieces from different levels of the hierarchy is difficult. DeBerg and Dobrindt recently showed how a hierarchy of Delaunay triangulations which allows local refinement can be constructed. This is accomplished using a more complicated structure than the simple tree used in the first case. DeBerg and Dobrindt cite memory usage of 132 bytes per data point[2].

Our goal is to provide a hierarchy of surface approximations which is a compromise between the sub-grid and the TIN approximations. We hope that a more regular "grid-like" TIN will combine the sub-grid's advantage of smaller space per vertex with the TIN's advantage of local adaptability to terrain.

5 RTIN Hierarchy

Our hierarchical surface representation contains partitions, called RTIN partitions, of the two dimensional square into right-angled, isosceles triangles. Such partitions have been studied in relation to crystal lattices [9], image encoding [12], and, in fact, geographic visualization systems [6, 7, 8]. The partitions contained in the hierarchy are defined inductively as follows. The partition composed of two triangles formed by dividing the square from northwest to southeast corner is the coarsest partition in the hierarchy (i.e. the one containing the largest triangles). From a partition in the hierarchy, a new, more detailed partition may be formed by *splitting* any one of the *non-terminal* triangles in the partition, where a non-terminal triangle is one which is larger than some fixed threshold depending on the resolution of the underlying height field. A triangle T is split by adding an edge from its right-angled vertex to the midpoint of its hypotenuse. The resulting partition may be *non-triangular* meaning that the new vertex introduced at the midpoint of the hypotenuse may lie on the border of a triangle R (T's neighbor across the hypotenuse) turning R into a quadrilateral. This represents a problem when using the partition to create a continuous surface via linear interpolation. Each vertex in the partition has a height value defined by the height field. Since R is a quadrilateral, linear interpolation over R is not well-defined, and performing linear interpolation using R's original three vertices may cause gaps in the surface. To avoid this problem, we continue or *propagate* the split into R. If R is larger than T then the hypotenuse of T forms a leg of R and we first split R (perhaps propagating this split) and then split the resulting triangle which shares its hypotenuse with T. If R is the same size as T we need only split R (no further propagation is necessary). See figure 1.

Propagation means that a single split operation can cause more than one triangle to split. However, splitting a triangle T cannot cause a triangle smaller than T to be split. Thus, only a finite number of triangles will be split in a single split operation, and the operation is guaranteed to terminate. In fact, at most two triangles of each size equal to or larger than T and no triangles smaller than T will split as a result of splitting T.

The most detailed partition in the hierarchy is one in which no triangle may be split, i.e. all triangles are terminal. The most detailed partition is *uniform*: every triangle is the same size. The vertices of the partition form a grid of size $(2^k + 1) \times (2^k + 1)$ for some integer k. The height of the (i, j)th grid point is the height of the (i, j)th sample point in the height array. The integer k (or, equivalently, the threshold defining the terminal triangle) is chosen so that the grid is large enough to contain the height array.

If extended to cover the plane, such a uniform partition is called a $[4.8^2]$ Laves net named by Bell, Diaz, Holroyd, and Jackson [9] after F. Laves. The name lists the degree of the vertices of

[2]Memory usage can be reduced by increasing the difference in detail between levels, but the hierarchy is then less capable of matching the exact resolution requirements of a desired approximation.