Prototype Variable Resolution Bathymetry Toolbox and Right TIN Implementation

Eric Mixon, Paul Elmore*, Fred Petry, Kevin Duvieilh

Naval Research Laboratory, Marine Geosciences Division, Stennis Space Center, MS 39529

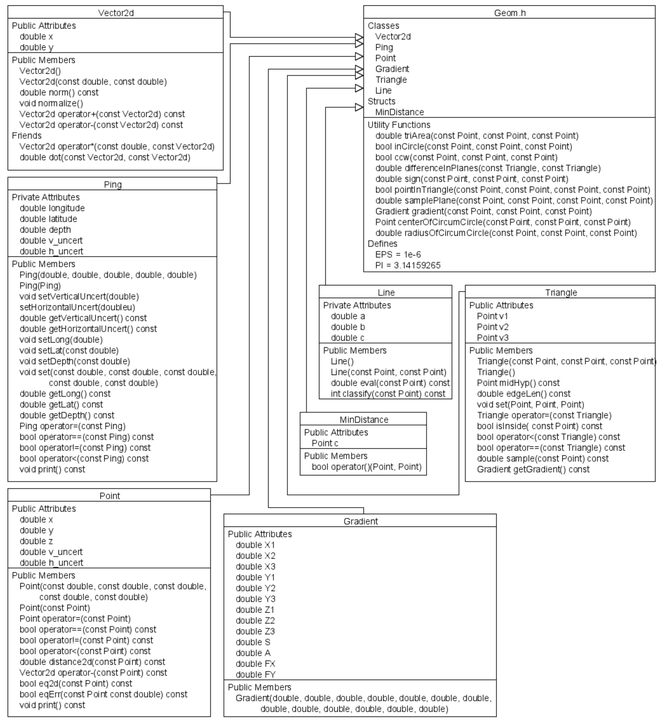
Abstract Implementation of object-oriented software for bathymetry data analysis and visualization is described. Our efforts focus on adaptation, implementation, and integration of differing techniques for visualization and analysis. The toolbox brings together separate algorithms into a common toolbox, giving users the ability to use different forms of approximation, interpolation, and storage techniques to analyze and visualize bathymetry surfaces. The strength of the toolbox is that it combines irregular mesh structures with regular grid spaced structures as well as irregular grid structures. These structures have the capability to thin dense areas of data where little data is needed while maintaining fidelity in areas of more detail. The toolbox allows for interpolation and analysis of all structures as well as the ability to convert all structures to a mesh that can be exported and visualized in other software such as Matlab.

I. Description

The software provides integrated object-oriented methods to use irregular, regular, and variable resolution data structures such as Triangular Irregular Networks (TIN), Right-TINs (RTIN), grids, and quad-trees for analyzing, interpolating, and storing bathymetric data. Tools are provided for projecting sets of bathymetric data to a plane in meters and then performing linear interpolations of the data to populate variable resolution structures. Methods are also provided for analyzing resulting surfaces for depth, error, gradient, density, and other characteristics that can help in classifying a set of data. The library is designed to be easy to use and modify for future developments and also works for all platforms with a C++ compiler.

A. Geometry Classes: Provides the capability to store, analyze and represent point data. A UML diagram of these classes and utility functions are shown in the figure to the right.

Figure 1: UML Diagram Geometry Classes of Bathymetry Toolbox



B. Containers: Provides container classes for sets of Ping and Point objects

PingList and PointList: Holds large sets of Ping and Point objects, respectively. Offers IO functionality for reading in complete data files, to locate minimum and maximum values, and to allow conversion from PingList to PointList, which results in the projection of a complete dataset to a plane in meters, and vice-versa.

C. Structure Classes: Provides classes and methods for creation of grids, triangular irregular networks (TIN) and right TINs.

QuadEdge: Implementation of a data structure by Guibas and Stolfi (1985) for representing graph structures of polygons; here, it is used for containing TIN structures and performing Delaunay Triangulation on sets of Points using either an incremental insertion algorithm (Lischinski, 1994) or sweep-hull algorithm (Sinclair 2010, s-hull.org, see figures to the right).

EdgeList: A QuadEdge support object, edgeList is created so that when edges are created and entered into the structure, they are also populated into the edgeList so that access to each triangle edge can be accomplished in a linear fashion.

Delaunay: Utilizes the QuadEdge class and methods to create a TIN of the bathymetric surface data. The flow chart to the right briefly lists the steps of the sweep-hull Delaunay Triangulation algorithm (Sinclair 2010, s-hull.org, see figures to the right).

DynamicRTIN: An implementation of the quad-tree structure by Pajarola (2002) that works for any dimension of dataset. Instead of using an actual quadtree, the RTIN accomplishes the same result with 2 binary trees of right Triangles, implemented using two linked list binary trees.

Grid: Using a simple array structure, implement a regular spaced surface from irregular surfaces using sampling class. The grid structure can be populated directly from an XYZ or GeoTIFF image file or can be a generated approximation of a Delaunay surface by sampling the depth of the provided Delaunay surface at each position.

Mesh: Converts surface data structures into a generic mesh that can be input into external visualization software. The Mesh can be exported to file, which will create two files containing the list of positions and indices respectively. The structure can also be traversed by triangles to retrieve gradient data for Triangle objects. A Mesh generated from a grid will simply be pairs of Triangles made from each Grid square.

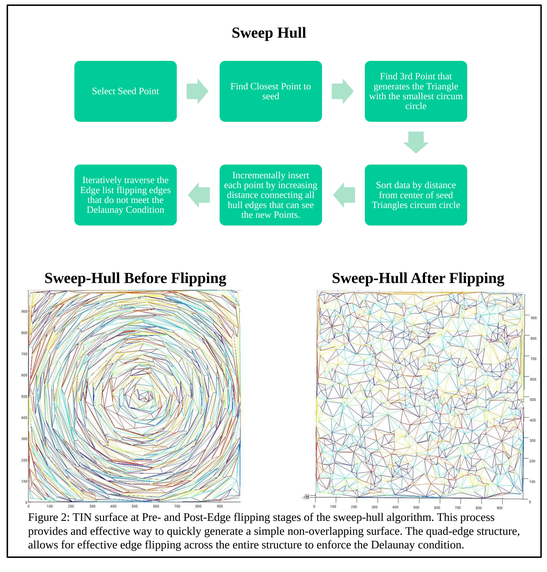


Figure 2: TIN surface at Pre- and Post-Edge flipping stages of the sweep-hull algorithm. This process provides and effective way to quickly generate a simple non-overlapping surface. The quad-edge structure, allows for effective edge flipping across the entire structure to enforce the Delaunay condition.

II. Right TIN Implementation

A. Background: The current focus of our exercises with the toolbox is directed toward the RTIN surface. The RTIN structure provides a set of points that are interconnected by right triangles and variably spaced by thinning an initially dense rectilinear grid of points through thresholding criteria. RTINs support the creation of surfaces whose resolution adapts with seafloor depth and morphology. A rationale for use of this technique is to provide variable resolution bathymetry surfaces for ocean models. High resolution is needed to properly render shallow water areas and significant in seafloor morphologies for numerical accuracy, but low resolution is used in deep flat areas to ease computational overhead.

Right TINs have a mature theoretical foundation (Evans et al., 2001). The RTIN approach is equivalent a quad-tree approach while providing smoother surface visualization, faster search performance of the point space, and simpler programming of APIs (Pajarola, 2002). Instead of using an actual quad-tree, the RTIN gives the same result with 2 binary trees of right triangles. It is this implementation that makes RTINs faster and easier than quad-tree methods since it eliminates a final computation step when generating a triangulation of the structure for visualization. Specifically, the structure is implemented using two binary trees, where the root of each tree represents the lowest resolution mesh that the structure can support. The structure is always going to be in square dimensions; however, where data does not exist for the approximation, the structure will be ignored upon extraction, leaving only the relevant surface.

B. Thinning: To initialize the structure, the data space is completely filled at every data point by sampling a Delaunay surface of the initial dense data set. After initial filling, the RTIN is the equivalent of a grid of equal dimensions, but with connected hypotenuses. To change this dense RTIN into a variable resolution grid, the data is reduced or thinned based on threshold criteria provided by the user. NRL is presently implementing three types of reduction techniques on the RTINs, where each point is analyzed for the effect its removal has on inducing error in the thinned surface when compared to the initial dense grid. Figures 3a-c show results of thinning exercises on grids extracted from the Naval Oceanographic Office Digital Bathymetry Database (DBDB), Version 5.2. A grid at 0.5 minutes of resolution occupies the northeast section. The rest is altimetry data at 2-minute resolution (see arrows in Fig. 3a). The first reduction technique (Fig. 3a) uses a depth criterion (Suarez and Plaza, 2009), where a vertex is removed if the resultant error would result in a change of depth by less than the threshold. This type of reduction allows for deep sea morphologies to be maintained, but causes sparseness in the shallow water regions. The second type of reduction (Fig. 3b) constrains the induced error as a percentage of water depth to be less than a specified threshold level. This type of reduction maintains a denser grid in shallow water, but results in loss of detail for deeper sea morphologies. To balance these two behaviors in thinning, a combined threshold approach is used (Fig. 3c) where both criteria have to be maintained before vertex removal. This approach better maintains point density for both shallow water and morphological regions. Table I gives the number of vertices retained after thinning, percentage reduction in data points, and residual error generated for different thresholds of the various techniques. Also shown is the time in seconds to perform the reductions on a standalone desktop PC with an Athlon-II X6 T1090 processor. The combined technique takes the most time as two criteria must check, but the production time for the surface in this example is reasonable. Other criteria, such as slope, maximum distance between points, and so forth also could be implemented along with the capability to retain key grid points.

C. Discussion: In all three thinned RTINs, the mismatch between 2-minute and 0.5 minute data remains. Thus, it is important to note that data mismatches between differing resolutions of gridded data will remain. Also of note is that the only real information that is stored in the RTIN structure are data points in Points objects, all the edges of the structure are abstracted from its organization for the purposes of visualization; however , for storage the only requirement is essentially an array of vertices along with the parameters of the RTIN. In addition, the RTIN is a variable resolution structure, but it is not a multiresolution structure. One cannot equate two or more grids of differing resolutions with an RTIN without extremely unlikely conditions. Data at differing resolutions would have to be interpolated into a new surface with constant grid spacing, and then placed into an RTIN. To implement this structure in operational environments, storage of the data into a file system such as Hierarchical Data Format 5 (HDF5) file system would be required.

Further research efforts are needed for the following points. Uncertainty estimates that apply to the thinned RTIN structure also need development. Then, for application of RTIN to ocean models, assessments are needed regarding the compatibility and effectiveness of the variable resolution structure with operational ocean models. Furthermore, research is needed to determine the effect variable bathymetric grid uncertainties have on models in which the RTIN is used.

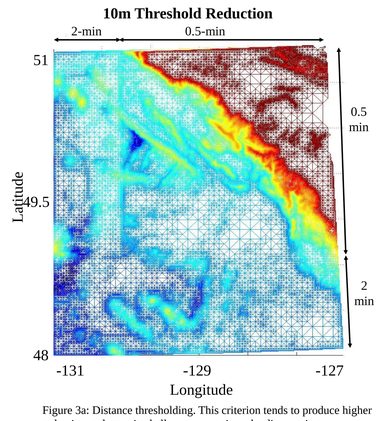


Figure 3a: Distance thresholding. This criterion tends to produce higher reduction and error in shallower water since the distance is a constant, and the percentage of change in the depth is higher when the value of the depth is smaller. Arrows: 0.5 and 2 minute grid boundaries. Victoria Island, Canada located in upper right.

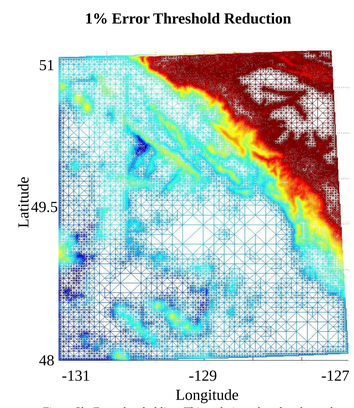


Figure 3b: Error thresholding. This technique throttles change based on a percentage of change in the depth. This gives a threshold whose literal distance varies based on the depth of the water. This can be useful in preserving shallow water data while being more lenient on deeper portions of the data.

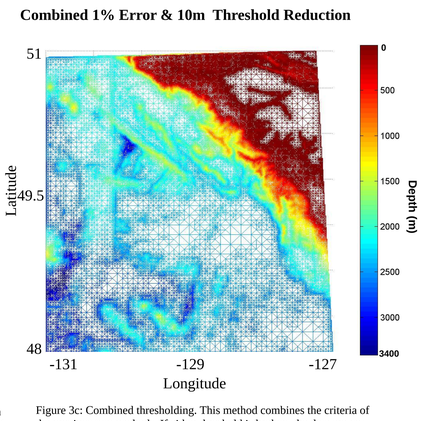


Figure 3c: Combined thresholding. This method combines the criteria of the previous two methods. If either threshold is broken, the data are not relaxed. This prevents the case were deeper water can experience large distances of change though the error is still a small percentage. This method, however, is slower to produce than using one criterion alone.

Table I: DBDBV RTIN Reduction Statistics

Percentage reduction of vertices versus percent residual error for different thresholds of the various techniques and computation times on a standalone desktop PC (Athlon-II X6 T1090 processor)

DBDBV Data							
Number of points	92462						
Min Depth (m)	3						
Max Depth (m)	3421						
Spacing (m)	926						
Δheight threshold (m)	0	1	3	10	30	100	300
Number of Vertices	218406	136532	95085	49117	20853	6873	3576
% Error	0	0.0299	0.181	1.0574	3.0324	6.5092	14.6356
Percent Reduction	0	37.4871	56.4641	77.5111	90.4522	96.8531	98.3627
Computation time (sec)		4.18	4.15	4.18	4.16	4.14	4.1
% Error Threshold		0.05	0.1	0.5	1	2	5
Number of Vertices		137085	115103	63551	46637	32886	20785
% Error		0.0045	0.0121	0.0925	0.1948	0.388	0.8823
Percent Reduction		37.2339	47.2986	70.9024	78.6466	84.9427	90.4833
Computation time (sec)		14.92	14.83	14.92	15.02	14.88	15
Combined (% error & Δheight (m))			1, 10				
Number of Vertices			62659				
% Error			0.0505				
Percent Reduction			29.2205				
Computation time (sec)			40.54				

***Corresponding Author:** Paul Elmore, email: paul.elmore@nrlssc.navy.mil

Acknowledgements: This work was sponsored by the Office of Naval Research through the "Irregular Multiresolution Database Algorithm" Base Program Project at the Naval Research Laboratory under program element 62435N.

References:

Evans WD, et al. (2001) "Right-triangulated irregular networks." Algorithmica **30:** 264-286.

Guibas L and Stolfi J (1985) "Primitives for the manipulation of general subdivisions and the computation of Voronoi." ACM Transactions on Graphics **4**: 74-123.

Lischinski D (1994) Incremental Delaunay Triangulation, in Graphic Gems IV., P. S. Heckbert, ed. Morgan Kaufmann: San Francisco, 47-59.

Pajarola R (2002) Overview of Quadtree-based Terrain Triangulation and Visualization, UCI-ICS Technical Report No. 02-01, Department of Information & Computer Science, University of California, Irvine.

Sinclair D (2010) "S-hull: a fast sweep-hull routine for Delaunay triangulation." Retrieved August 24, 2011, from http://www.s-hull.org.

Suarez JP and Plaza A (2009) "Four-triangles adaptive algorithms for RTIN terrain meshes." Mathematical and Computer Modelling, **49**, 1012-1020.

Vincenty T (1975) Geodetic inverse solution between antipodal points. Technical Report DMAAC Geodetic Survey Squadron. Geocentric Datum of Australia (GDA) Reference Manual. Onlize at http://geographiclib.sourceforge.net/ geodesic-papers/vincenty75b.pdf