Incremental Delaunay Triangulation Algorithm for Digital Terrain Modelling

Sulak Soommart and Kitti Paitoonwattanakij
Image and Graphics Research Room, Department of Electronics, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand.

Abstract

One of the problems when storing and computing an elevation data is that the amount of data can be enormous. A Triangulated Irregular Network (TIN) is an alternative to represent an elevation by which a TIN can be extracted from a dense grid of data. We present an incremental Delaunay triangulation algorithm for constructing a TIN. The algorithm starts with an initial triangular network covering entire area of an elevation grid. More data points are inserted into a network sequentially by selecting a subset of the grid points; this subset is the most significant contribution to the terrain model. The Delaunay triangulation of this subset is the TIN that approximates the elevation at all grid points within a distant criterion. As a result, the conversion of spatial data structure from an elevation grid to a TIN significantly reduces the number of data points to represent the terrain surface.

1. Introduction

Digital Terrain Modelling (DTM) [10] usually uses a rectangular grid (or elevation matrix) as a major data structure to represent the terrain surface. This is because the handling of elevation matrices is simple, and grid-based terrain modelling algorithms tend to be relatively straightforward. But on the other hand, the point density of regular grids cannot be adapted to the complexity of the relief. Thus, an excessive number of data points are needed to represent the terrain to a required level of accuracy. Also, rectangular grids cannot describe the structural features in comparison with the topographic features.

Triangulated Irregular Network (TIN) [2] is another way to represent an elevation data. TIN is based on triangular elements, with vertices at the sample points. The great advantage of using triangles in terrain modelling is the possibility of adapting the triangle shape to fit variation in the terrain surface; more data points can be used in regions where there is much elevation change, and fewer points in regions where the elevation hardly changes. Consequently, a TIN can approximate any surface at any desired tolerance with a minimal number of triangles.

Although TIN has not been used in DTM for a long although time, many algorithms for constructing the TIN to represent the terrain have been proposed. Section 2 explains the reason why we adapt the incremental algorithm to calculate the TIN from the dense grid source. After the four steps of incremental algorithm are briefly presented, we describe the detailed theory behind the algorithm. Finally, Sections 3, 4, and 5 give the implementation, the results, and the conclusions of the experiment.

2. Incremental Algorithm

Without doubt the most popular triangulation of a point set is the Delaunay triangulation [6], [2]. There are several well-known triangulation methods, but from a generalization point of view, a dynamic method for Delaunay triangulation is very promising. One dynamic triangulation method is the incremental algorithm. When a point is included in the network, the network is rearranged until the max-min angle criterion is met [5]. Consequently, during the triangulation, it is possible to select the points that make the most significant contribution to the model (this technique is described in Section 2.5). The four steps of the incremental algorithm are:

1. The initial triangular network is created covering the entire area of all data points. In this research, we use a circumscribing rectangle that is divided into two triangles as an initial network (Figure 1.a),

2. The first point of interior area is included into the network. The point is connected to its enclosing triangle by three new triangle edges

between the point and the vertices of the triangle (Figure 1.b),

3. The quadrilaterals, which have the old edges of the enclosing triangle as a diagonal, have to be tested by max-min angle rule [5]. If they do not meet the criterion, their diagonals are swapped and the new opposite edges to the recently inserted point will be examined as diagonals in their quadrilaterals (Figure 1.c),

4. The Delaunay network now contains one more point. All the remaining points will be repeatedly included in the network in the same way as step 2 and 3 (Figure 1.d - 1.f).

2.1 Data structure

The data structure for the incremental algorithm contains at least two pieces of information: geometry and topology. The geometrical data structure contains the information of coordinates: point, edge, vertex, and triangle. The topological data structure is the information that describes how each geometrical data connects to each other in the network to represent the aspect of the terrain model. This topological network structure is a kind of planar graph subdivision where the geometrical data is a node and the topological data is an edge that links nodes together. Thus one can use the Dept-First Search [7] operation of the Graph algorithm to traverse all elements inside the TIN. This operation is very useful for extracting the information from the TIN (described in Section 3.3).

In this research we use the Twin-Edge [3] as a topological data structure. The Twin-Edge is an edge based data structure in which all information about the triangle is implicitly stored in an edge network. A pointer to the endpoint and a pointer to the next edge in the triangle are stored for each edge. Further there is a pointer to the twin edge, but it points at the opposite endpoint. In addition there is a pointer from each edge to the attribute, which is the point set belonging to the triangle. A triangle is formed by three separate edges. The pointers of the Twin-Edge structure are outlined in Figure 2.

2.2 Point insertion

Point insertion is a fundamental operation of the incremental algorithm. This section describes how insertion of a point, $n+1$, interior to triangle T_n, influences the edges in the triangular network. The point insertion results in T_{n+1}. When point D is inserted into ΔABC (Figure 3), three new edges are constructed (Figure 1.b). These edges are all valid in T_{n+1}. Some of the surrounding edges have to be swapped (Figure 1.c) and located with one end in D. A swapped edge is a member of T_{n+1}, and is only swapped once, consequently.

2.3 Recursive process for edge reorganization

The insertion of a new point usually results in some swapped edges. Every quadrilateral that is adjacent to the inserted point has to be tested by the max-min angle rule. If the diagonal is swapped, two new adjacent quadrilaterals must be tested as well. Figure 3 shows the result after the edges surrounding the triangle are recursively swapped. The problem can be formulated concisely and precisely by a recursive procedure:

```
FUNCTION  Reorganize(diagonal)
 VARIABLE:
  *diagonal;
  # The diagonal of the quadrilateral
  # to check

  *edge1, *edge2;
  # These two edges are the diagonals
  # of the quadrilateral to check if
  # if diagonal is swapped

 IF checkMaxMinAngle(diagonal) = TRUE
  # Check the max-min angle criterion
 THEN
  edge1 = diagonal->twin->next;
  edge2 = diagonal->twin->next->next;
  SwapDiagonal(diagonal);
  Reorganize(edge1);
  Reorganize(edge2);
 ENDIF
END
```

2.4 Reorganization of the point handling structure

Point set, stored as attributes to the triangles during the triangulation process, is frequently updated. It is necessary to optimize the reorganization. When the diagonal in a quadrilateral is swapped, it is necessary to determine which triangle that the points should be interior to (Figure 4). The task is to find out which side of the diagonal each point is situated. By using Equation (1) the distance point-line is calculated.

$$d = \frac{ax + by + c}{\sqrt{a^2 + b^2}} \quad (1)$$

where

$$a = y_A - y_B$$
$$b = x_B - x_A$$
$$c = x_A y_B - y_A x_B$$

Practically, when we only need to determine on which side of the line the point is, it is adequate to examine the numerator of Equation (1). More precisely we have to find the sign of Equation (2). A point lies to the right of the line *AB* if s_i is positive. Further, the point lies on the line if $s_i = 0$, and to the left if s_i is negative.

$$s_i = ax_i + by_i + c, \ (i = 1,2,..,n) \quad (2)$$

where n is the number of points interior to the quadrilateral of the swapped diagonal.

2.5 Qualified selection of points

Douglas and Peucker [1] present the arc simplification algorithm, which approximates a curve by a polyline in 2 dimensions. We consequently applied this technique for the qualified selection of points in 3 dimensions, which is based on the triangulation paradigm. During the insertion process in the incremental algorithm, a split point is chosen from points that are enclosed by the present triangle. The most distant point in each triangle makes the most significant contribution to the model, and is consequently chosen as the split point.

Figure 5 shows a threshold λ_p. Every point that is closer to its enclosing triangle than λ_p will not be included in the mesh. The threshold in Figure 5 is designed for the perpendicular distance from point $Q(x_Q, y_Q, z_Q)$ to the triangular plane, ΔABC, and calculated by Equation (3)

$$dist_p = \frac{ax_Q + by_Q + cz_Q + d}{\sqrt{a^2 + b^2 + c^2}} \quad (3)$$

where

$$a = y_A(z_B - z_C) + y_B(z_C - z_A) + y_C(z_A - z_B)$$
$$b = x_A(z_C - z_B) + x_B(z_A - z_C) + x_C(z_B - z_A)$$
$$c = x_A(y_B - y_C) + x_B(y_C - y_A) + x_C(y_A - y_B)$$
$$d = x_A(y_C z_B - y_B z_C) + x_B(y_A z_C - y_C z_A) + x_C(y_B z_A - y_A z_B)$$

When we are searching for the most distant point from a triangle, it consumes less time by comparing the numerator of Equation (3). The denominator is equal for all points enclosed by a triangle. Consequently the split point is concisely determined by Equation (4)

$$s_p = \max|ax_i + by_i + cz_i + d| \quad (4)$$

where $i = 1,2,..n$, $c_i \neq 0$, and n is the number of points enclosed by the horizontal projection of ΔABC.

2.6 Automatic determination of the most distant point

The point set inside the triangles is represented by the Binary Search Tree (BSTree) data structure [4]. Points are stored in the nodes of the BSTree (Figure 6) sorted on a distance. After the computation to find which triangle that the point (Section 2.4) should belong to is achieved, the point is then inserted into the BSTree. The points in the nodes of BSTree are sorted automatically while the point insertion of BSTree is performing. Therefore, the most distant point of that point set (Section 2.5) is simply found at the rightmost node of the BSTree (Figure 6). Figure 7 shows the process of point handle during the edge reorganization.

2.7 Succession of the inserted triangles

Before any points are inserted in the mesh, the edges of the two initial triangles are stored in the edge list if they have the qualified points to be inserted. After the process of point insertion and edge reorganization are completed, the edge list is rearranged by removing edges from the list if their triangles are influenced by the triangulation process, and adding more edges of the new triangles recently created if they have the qualified points to insert. The edge in the list is orderly selected for the next triangulation process.

The triangles to be split can be chosen in randomized succession. This is simple, efficient, and from the experiment shows that the results are quite good. However, the surface models do not necessarily become identical when one data set is processed two times with altering the point succession. But the variation is small and the precision of the surface model will still be the same.

3. Implementation

The incremental algorithm has been implemented as a Grid-to-TIN converter in our DTM project. Usually we apply the algorithm repeatedly to eliminate vertices from a regular mesh until a specified reduction threshold is achieved. The degree of elimination is controlled by adjusting the distant threshold. The steps of implementation started as follows: data points are extracted from the rectangular grids. Then, the data points are projected onto the xy-plane by using only x and y coordinates of the points. Next, the 2-dimensional incremental triangulation method is used on these points to compute the TIN. Finally, the TIN is visualized by re-introducing the z-coordinate (elevation values) to the TIN.

3.1 Digital Elevation Models

In order to work with real terrain data, we apply the incremental algorithm to two Digital Elevation Models (DEM) [9] data sets: Mount St. Helen's, Washington, and the Grand Canyon, Arizona. The DEM data set, produced by the U. S. Geological Survey (USGS), consists of a sampled array of elevations for ground positions that are usually, but not always, at regularly spaced intervals. The 7.5-minute DEM format is the densest format available consisting of elevation values at 30 meters spacing in the north-south and east-west directions. The data are ordered from south to north in profiles that are ordered from west to east.

Since the regular array of elevations are referenced horizontally in the Universal Transverse Mercator (UTM) coordinate system, the profiles do not always have the same number of elevations due to the variable angle between true north and grid north of UTM coordinate system. Figure 9 shows the irregular boundary around the array. We used only the maximal rectangular array of elevation values, which were extracted from the DEM file, in a simple form of the 2D array.

3.2 Data structures and Delaunay Triangulations

We implement classes in the Java code to represent both geometrical and topological information. Each class contains two groups of information: data member and operation member. The Point class, for instance, contains coordinates as the data member and geometrical functions as operation member, which support the operations as classifying this point relative to a given line segment and computing the point's distance from a given triangular plane.

The Twin-Edge class is simple, only has necessary members, which be used to connect itself to the Point object, two Twin-Edge objects (next and twin edges), and Info object by another object. The class that actually operates the Twin-Edge object and the geometrical objects to form the network is the TINGraph class. This class contains various data members and operation functions to deal with the triangulating calculation.

After the DEM file is extracted as a 2D array of the data points; the TINGraph take this array as an input data points, using only x and y coordinates. The initial triangular network covering these points is constructed. The process of the 2-dimensional Delaunay triangulation then started the incremental algorithm. When the triangulation is successful, the Depth-First Search operation of the Graph algorithm traversed the TIN. Finally, the output of this operation presents the list of triangle objects that are ready to be visualized in the next step.

3.3 Visualization

In this step, the z-coordinate is re-obtained. The terrain map can be enhanced by hill shading technique [11], where an imaginary light source is placed in 3-dimensional space, and parts of the terrain that do not receive much light are shaded. The degree of shades is represented by the gray values. Where the gray values of the triangles will be proportional to the angle between the normal vector of the surface and the vector of the light source (the northwest corner of the model (upper left) or (1, -1, -1)). Figure 10 and 11-14 (b) show the results of this applied technique for the visual representation of TIN.

4. Results

The experiment was performed on the workstation computer: Pentium II - 333 with 64 Mbytes of RAM and 256 gray-level display of image. To demonstrate the operations of the algorithm, the operative variables are introduced (Table 1 and 2) as follows: the number of Split points, Reorganized edges and Swapped diagonals are the complexity of the terrain model as the frequency of operations. The Subset points and Irregular triangles are the result of remaining points and triangles after the triangulation completed. The Point's ratio and Triangle's ratio express the percentage of approximation where the number of points and irregular triangles as an output of the triangulation are compared to the input, the number of points and triangles of regular grids.

As the results in Table 1 and 2 where the threshold of distance is varied, the time consumption of the triangulation is shown by the Time usage variable. The timing results show that randomized succession spreads over the triangulated area are quite good, slightly faster than the orderly succession.

If every point from the data set is to be included in the mesh, the succession of the inserted points has little influence on the final result. A unique network will always be equal in spite of the succession of inserted points. The incremental algorithm is tested with random point set and compared to another static Delaunay triangulation algorithm, namely the Step-by-Step algorithm [8]. Table 3 and Figure 8 shows that the performance of the incremental algorithms are close to linear, $O(n \log n)$, and significantly faster than the Step-by-Step algorithm, $O(n^2)$.

Figure 10 shows a screen shot of Mount St. Helen's and the Grand Canyon at the full resolution of DEM(s). After the triangulation of these two data sets is complete, the models are visualized with two techniques, wired-frame and hill shading, as shown in Figure 11 - 14 (a).

5. Conclusions

The conversion of an elevation grid to a TIN significantly reduces the number of data points to represent a terrain surface. Using the novel data structures, the incremental algorithm starts constructing a coarse TIN with only a few vertices covering the entire area of an elevation grid. The algorithm keeps adding more points, by selecting the most contribution points from the grid to the TIN. The recursive examination and swapping procedure let the algorithm dynamically triangulate a set of points; for each point included in the TIN, the network will be rearranged until the criterion of the Delaunay triangulation theorem is satisfied for all triangles in the network. The result of timing performance shows that the incremental algorithm as a dynamic method is faster than the static method when the point set is re-triangulated.

6.References

[1] D. H. Douglas and T. K. Peuker (1973), Algorithms for the Reduction of the Number of Points Required to Represent a Line or Its Caricature, The Canadian Cartographer, Vol. 10, No. 2, pp. 112-122.

[2] L. J. Guibas and J. Stolfi (1985), Primitives for the Manipulation of General Subdivisions and the computation of Voronoi diagrams, ACM Trans. Graph., pp. 4(2):74-123.

[3] M. Heller (1990), Triangulation Algorithms for Adaptive Terrain Modeling, Proceedings of the 4th International Symposium on Spatial Data Handling, pp. 163-174.

[4] M. J. Laszlo (1996), Computational Geometry and Computer Graphics in C++, Prentice Hall Inc., New Jersey.

[5] C. L. Lawson (1977), Software for c^1 Surface Interpolation, Mathematical Software III, Academic Press, pp. 161-194.

[6] F. P. Preparata and M. I. Shamos (1985), Computational Geometry An Introduction, Springer-Verlag Inc., New York, pp. 203-204.

[7] R. Sedgewick (1992), Algorithms in C++, Addison-Wesley Publishing Inc., New York.

[8] S. Soommart and K. Paitoonwattanakij (1998), Digital Terrain Modelling by Using Triangulated Irregular Network (in Thai), Proceedings of the 21st Electrical Engineering Conference, November 12-13, pp. 573-576.

[9] United States Geological Survey (1987), Digital Elevation Models – Data Users Guide 5, Reston, U. S. Department of Interior, Virginia.

[10] R. Weibel and M. Heller (1991), Digital Terrain Modelling, Geographical Information Systems - Principles and Applications, Longman, London, pp. 269-297.

[11] Yoeli P. (1965), Analytical Hill Shading, Surveying and Mapping 25, pp. 273-279.

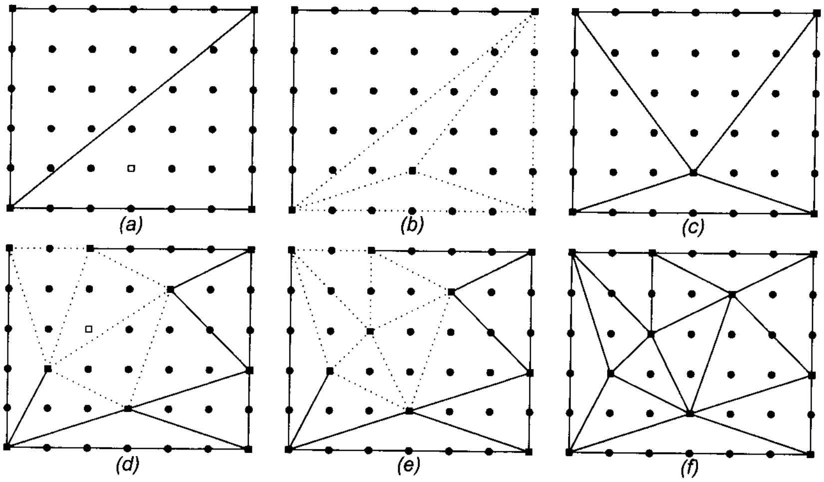


Figure 1: Triangulation process: (a) the initial network, (b) insertion of the first point, (c) the surrounding edge is swapped, (d) insertion of the last point, (e) some edges are swapped, (f) final network after the last point is included.

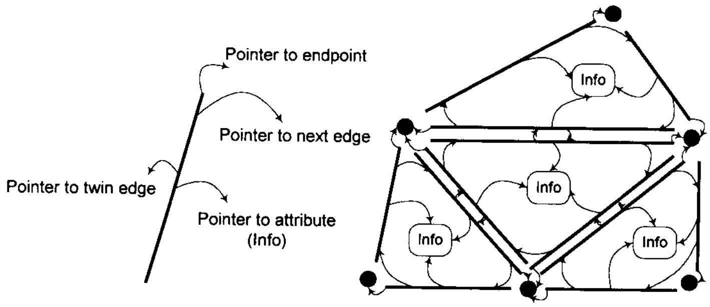


Figure 2: Principles for the Twin-Edge data structure.

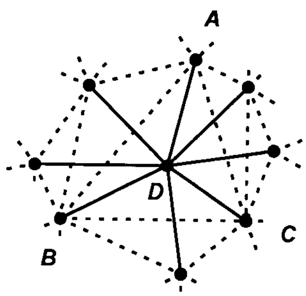


Figure 3: The orientation of new and swapped edges after point D is inserted into ΔABC. (The dotted lines show the triangular network before the point insertion.)

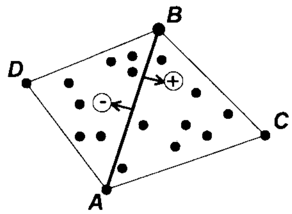


Figure 4: The points are sorted by the diagonal AB.

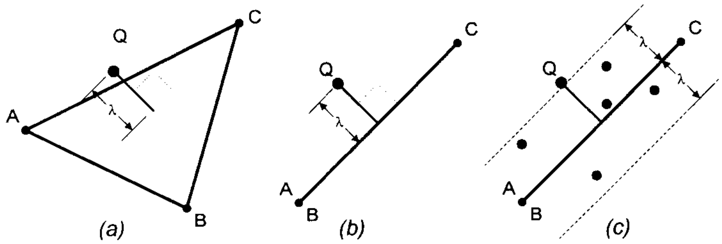


Figure 5: (a) Threshold value, λ_p, for the perpendicular distance. (b) The cut through the triangle. (c) Region inside the threshold value.

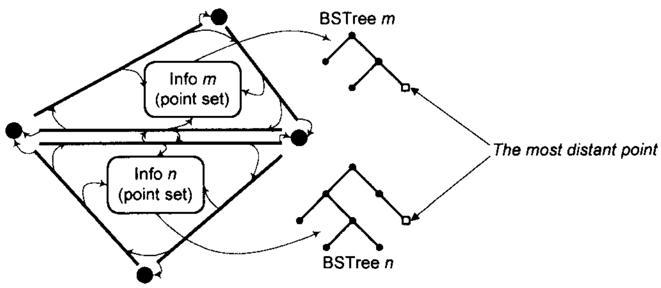


Figure 6: Interior points of each triangle are stored in the nodes of the Binary Search Tree.

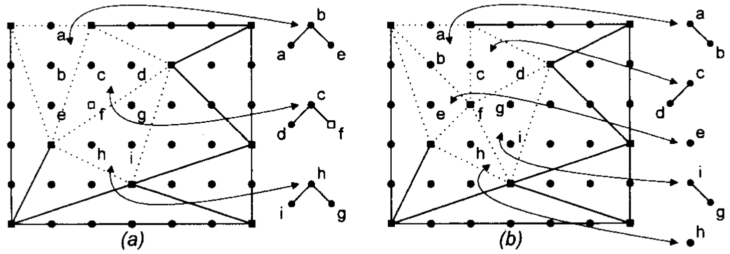


Figure 7: Point handling process by using the BSTree during the edge reorganization:
(a) BSTree(s), before the new point is inserted, (b) Point sets are transferred to the new BSTree(s).

Table 1: Experiment results when the DEM of the Mount St. Helen's are triangulated (Point set = 145820 (317 × 460), Regular triangles = 290088).

Table 1(a): Orderly succession

Distant threshold (m.)	20	30	40	60	80	100
Split points	16676	3907	1936	716	411	270
Reorganized edges	166302	35657	17222	6111	3465	2238
Swapped diagonals	58151	11982	5721	1996	1131	729
Subset points	16680	3911	1940	720	415	274
Irregular triangles	33354	7816	3874	1434	824	542
Time usage (min.)	30.59	5.47	4.29	3.41	3.31	3.22
Point's ratio (%)	11.44	2.68	1.33	0.49	0.29	0.19
Triangle's ratio (%)	11.50	2.69	1.34	0.49	0.28	0.19

Table 1(b): Randomized succession

Distant threshold (m.)	20	30	40	60	80	100
Split points	16463	3786	1894	747	424	274
Reorganized edges	161189	33671	16478	6369	3663	2199
Swapped diagonals	55919	11174	5415	2080	1214	707
Subset points	16467	3790	1898	751	428	278
Irregular triangles	32928	7574	3790	1496	850	550
Time usage (min.)	25.36	4.35	4.15	3.13	3.03	2.53
Point's ratio (%)	11.30	2.60	1.30	0.52	0.29	0.19
Triangle's ratio (%)	11.35	2.61	1.31	0.52	0.29	0.19

Table 2: Experiment results when the DEM of the Grand Canyon are triangulated (Point set = 169002 (369 × 458), Regular triangles = 336352).

Table 2(a): Orderly succession

Distant threshold (m.)	20	30	40	60	80	100
Split points	11911	3512	2216	1210	797	551
Reorganized edges	114051	31587	19610	10540	6869	4688
Swapped diagonals	39183	10550	6500	3470	2252	1530
Subset points	11915	3516	2220	1214	801	555
Irregular triangles	23824	7026	4434	2422	1596	1104
Time usage (min.)	20.37	7.29	7.13	4.41	3.53	3.41
Point's ratio (%)	7.05	2.08	1.31	0.72	0.47	0.32
Triangle's ratio (%)	7.08	2.0	1.32	0.72	0.48	0.32

Table 2(b): Randomized succession

Distant threshold (m.)	20	30	40	60	80	100
Split points	11530	3371	2146	1194	774	565
Reorganized edges	107606	29424	18730	10327	6630	4767
Swapped diagonals	36535	9683	6174	3394	2169	1548
Subset points	11534	3375	2150	1198	778	569
Irregular triangles	23062	6744	4294	2390	1550	1132
Time usage (min.)	20.10	6.46	6.24	4.06	3.10	3.32
Point's ratio (%)	6.82	2.00	1.27	0.71	0.46	0.34
Triangle's ratio (%)	6.86	2.00	1.28	0.71	0.46	0.34

Table 3: Timing results of implemented Delaunay algorithms

Number of random points	Incremental algorithm (second)		Step-by-Step algorithm (second)
	Orderly succession	Randomized succession	
50	0.69	0.62	0.92
100	1.13	1.07	1.83
500	4.96	4.88	21.53
1000	10.10	10.03	126.24
2000	21.42	21.34	494.07

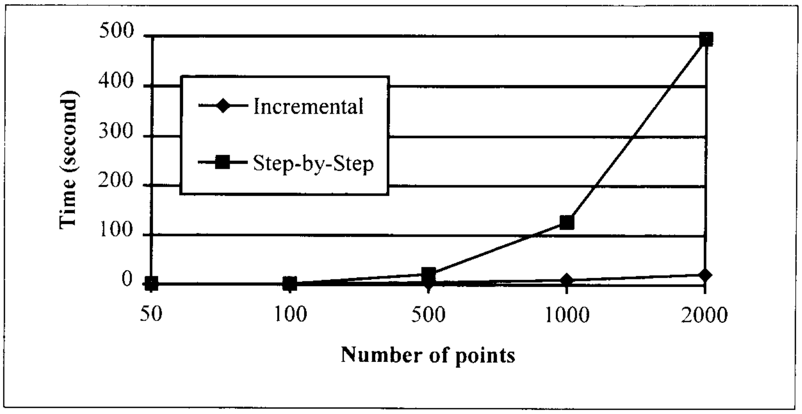
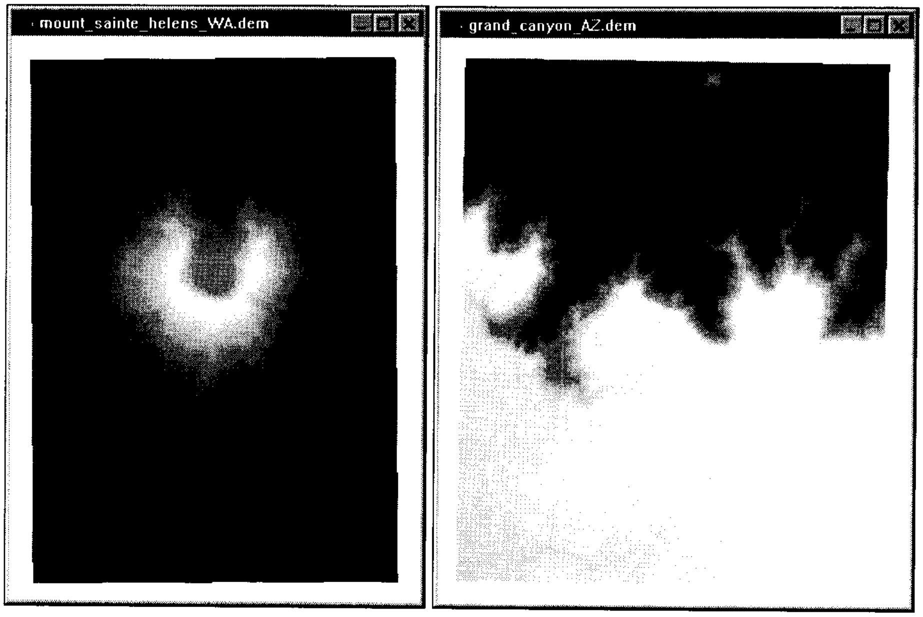
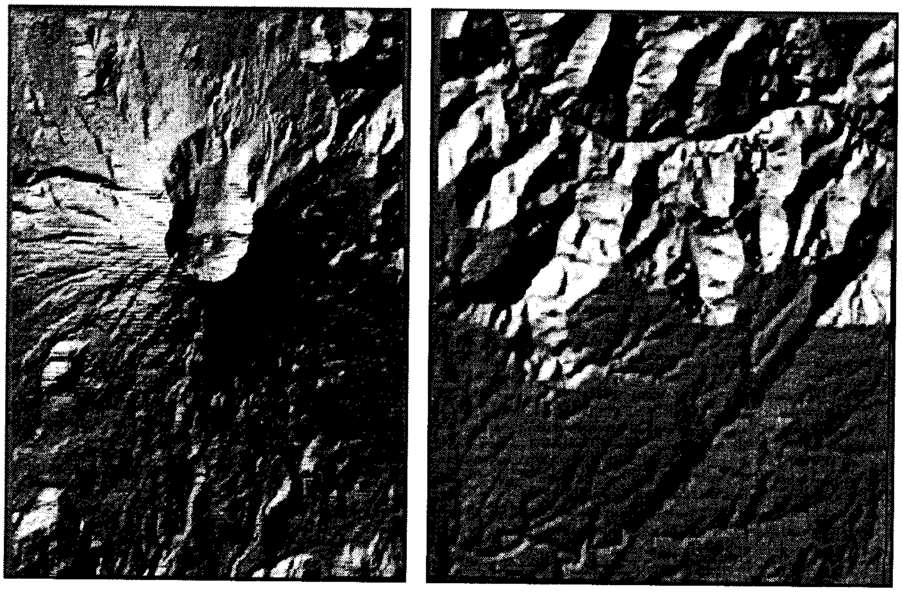


Figure 8: Timing results of implemented Delaunay algorithms



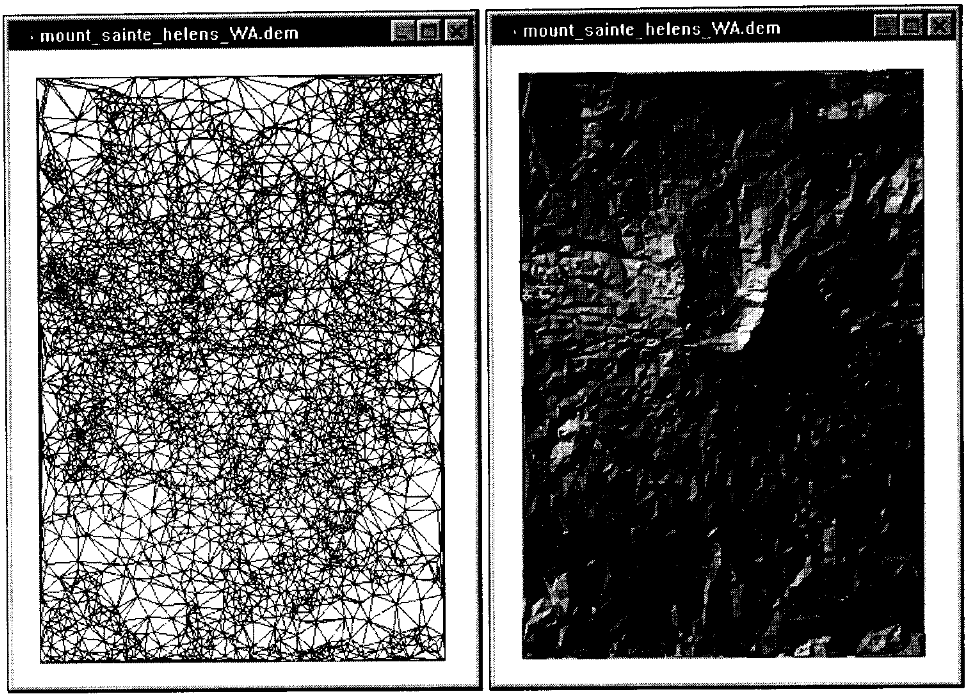
(a) Mount St. Helen's (b) Grand Canyon

Figure 9: Area covered by 7.5-minute DEM (Gray-Depth technique).



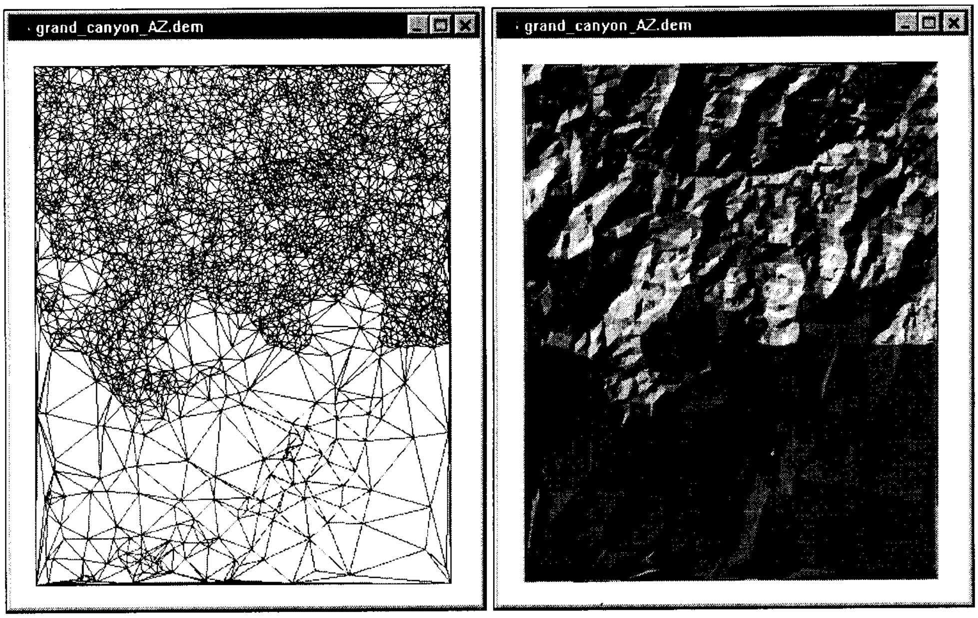
(a) Mount St. Helen's (b) Grand Canyon

Figure 10: Hill shading visualization of DEM(s) (full resolution).



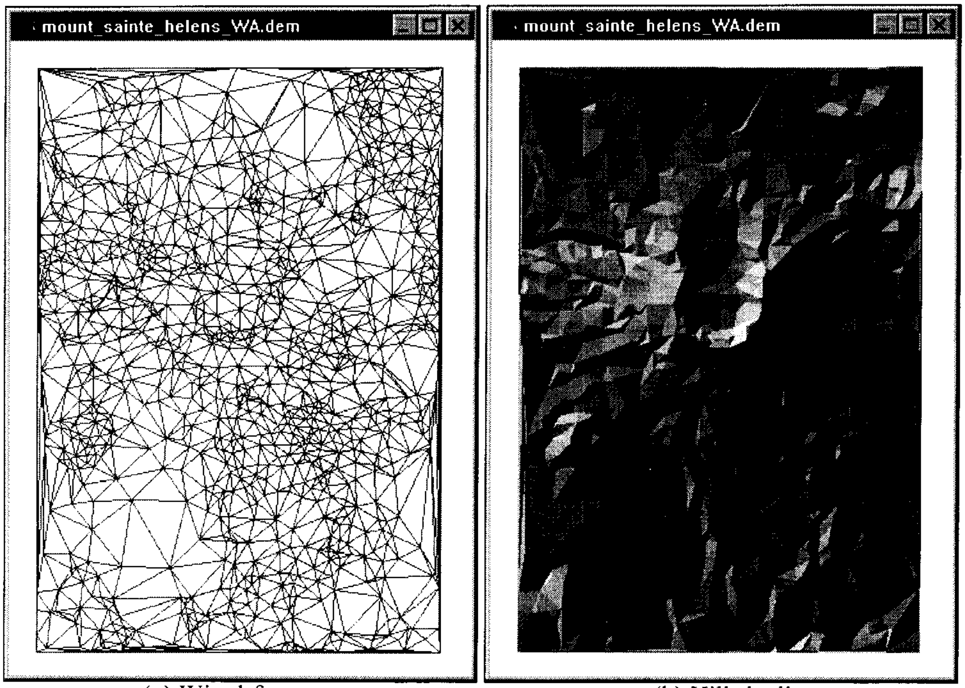
(a) Wired-frame (b) Hill shading

Figure 11: Visualizations of Mount St. Helen's; triangulated at *Th* = 30 m., Using only 3911 points (2.68 %).



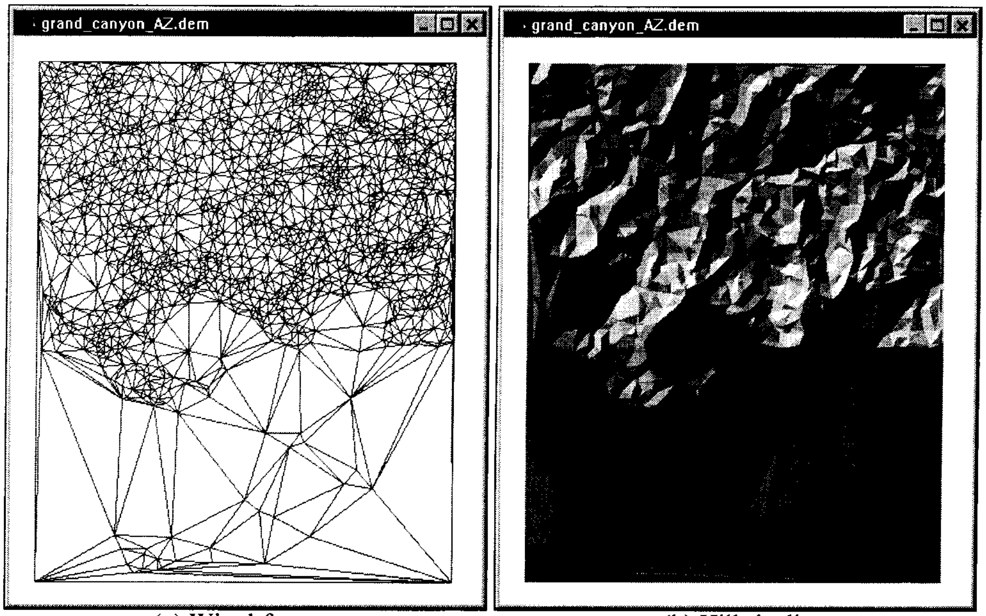
(a) Wired-frame (b) Hill shading

Figure 12: Visualizations of Grand Canyon; triangulated at *Th* = 30 m., Using only 3516 points (2.08 %).



(a) Wired-frame (b) Hill shading

Figure 13: Visualizations of Grand Canyon; triangulated at *Th* = 50 m., Using only 1091 points (0.75 %).



(a) Wired-frame (b) Hill shading

Figure 14: Visualizations of Grand Canyon; triangulated at *Th* = 50 m., Using only 1574 points (0.93 %).