BRITISH COLUMBIA
The Best Place on Earth

Freshwater Atlas

WaterBug Watershed Boundary Generator

Technical Design

Prepared for

Base Mapping & Geomatic Services
Integrated Land Management Bureau
Ministry of Agriculture and Lands

Prepared by

Refractions Research Inc.
#300 – 1207 Douglas Street
Victoria, BC
www.refractions.net

Document Tracking

Version: 1

Last Updated: July 16, 2007

Refractions Research
THE GEOSPATIAL EXPERTS

REVISION HISTORY

Date	Version	Author	Description of Changes
Jul 16, 2007	1.0	Martin Davis (Refractions Research)	Final Version

CONTENTS

1 INTRODUCTION

This document describes the technical design for the FWA Watershed Boundary Generator software. This software is called WaterBuG. It serves as a processing engine which computes the watershed boundary linework for a given block (or area) of input data.

1.1 RELATED DOCUMENTS

- *FWA Watershed Boundary Requirements*
- FWA *Watershed Boundary Process* – describes the complete process of generating boundaries for the entire province

2 SYSTEM OVERVIEW

The following dataflow diagram shows the inputs, outputs, and main processes of the WaterBuG application.

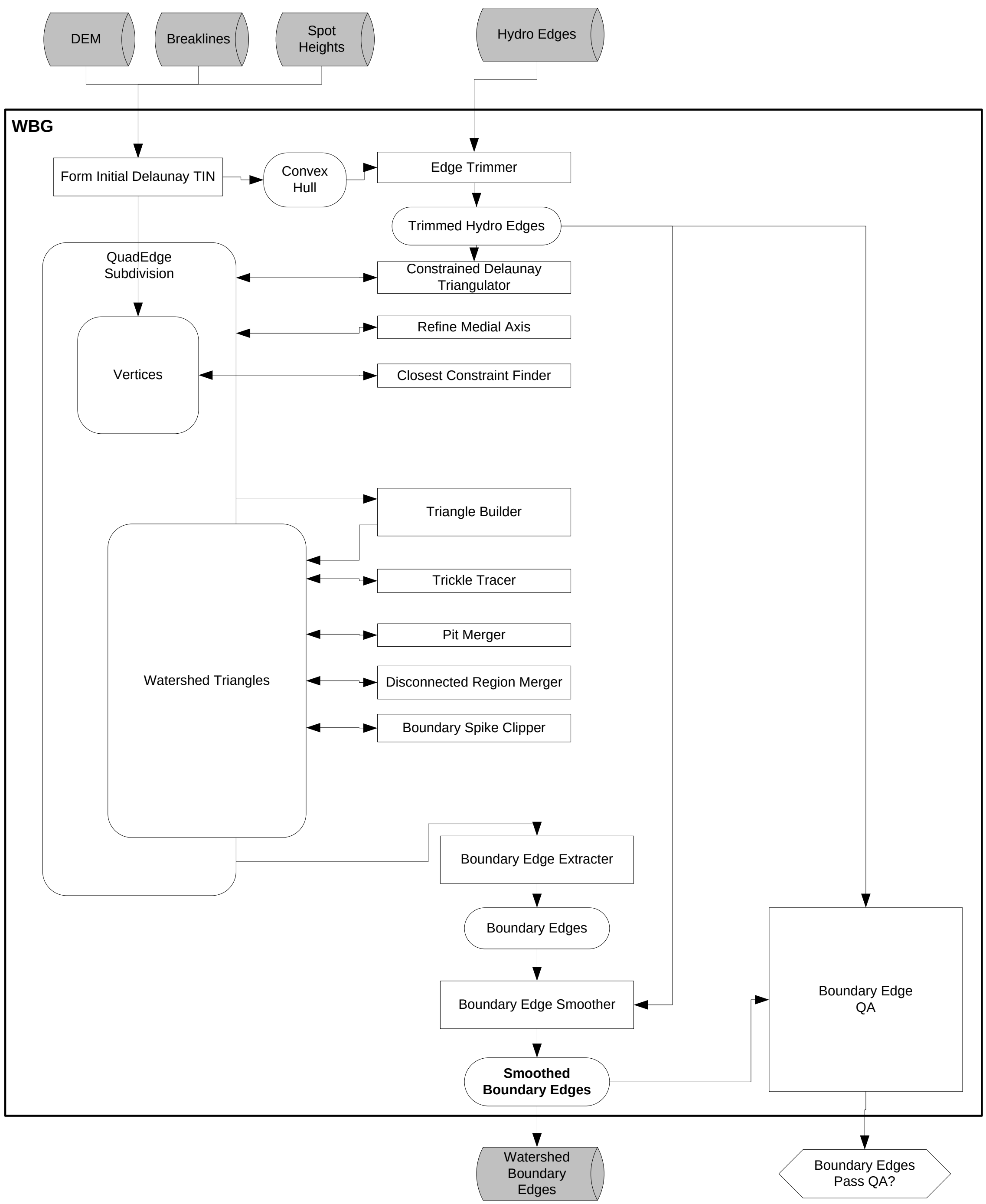


Figure 1: Inputs, Outputs, and Main Processes

3 INPUT DATA SPECIFICATION

This section details the data which is provided as input to WaterBug.

3.1 DEM MASS POINTS

The DEM mass points are obtained from the TRIM DEM dataset. The point locations are non-uniform. Each point has an associated height in metres. The TRIM DEM datasets are stored as a set of points per 1:20,000 mapsheet. The points usually extend a small distance beyond the mapsheet edges (thus there is some duplication of points between adjacent datasets).

3.2 SPOT HEIGHTS

TRIM contains a dataset of spot heights. Spot heights are included as extra DEM elevation points.

Only certain types of spot heights are used, as specified in the following table:

Class	**Included?**
mountainPeak	Y
spotHeight	Y
waterLevel	N

Spot heights may be coincident or nearly coincident with DEM points. In this case the spot height will be use in preference to the DEM point. No consistency check will be made to test if the elevations of the points are identical.

The WBG will ensure that when spot height locations are present along an unsmoothed watershed boundary line, they are respected (maintained) as a vertex of the smoothed boundary line.

3.3 BREAKLINES

TRIM provides a dataset of breaklines as linear features. These breaklines contains a substantial number of additional elevation observations. In order to improve the fidelity of the generated elevation surface to the actual landscape, breakline information is included as additional vertices in the surface triangulation.

Only certain classes of breakline are included, as specified in the following table.

Class	Included?
Area of Exclusion	N
Breakline Hydrographic	N
Breakline Hypsographic	Y
Breakline Manmade	N
Breakline Round	Y
Breakline Sharp	Y

Although the linear topology of the breaklines is not incorporated explicitly, the breakline vertices are dense enough that in most cases the triangulation is faithful to the breakline edges.

3.4 HYDROGRAPHIC NETWORK

The hydrographic network is obtained from the dataset prepared in a previous phase of the FWA project.

- Fully noded and connected
- Includes face information for edges of polygonal waterbodies (indicating which side of edge is wet.
- Only relevant edges should be supplied (this may require removal or flagging of unwanted edges such as isolated lakes, wetlands, etc. This process may require the graph structure to be known)
- Includes Drainage Area ID for each edge.

Each Hydro Edge has a type code associated with it. Some edge types are not relevant to watershed boundary generation and should be eliminated from the input dataset. The following codes should not be present in the input data:

Unused Edge Type Codes
1550
1325
1375
150

3.5 DRAINAGE AREA IDS

A Drainage Area ID (DAID) is a unique identifier for each drainage polygon to be generated. There is no necessary relationship between identifiers (in particular, they are not necessarily hierarchical). Each hydrographic edge must carry the identifier for the drainage area which drains into it. If two edges drain the same drainage, they carry the same DAID.

The DAIDs will determine the boundaries on the TIN which will form the final drainage area polygon coverage edges.

The assignment of the DAIDs controls how hydro segments are assigned to watersheds. With the appropriate DAID/segment assignment, this supports meeting the following FWA-HoL requirements:

RDCS2: Where a first order stream flows into a higher order stream represented in TRIM as a single line, HoL linework will be generated on the opposite bank to the stream confluence. Where a first order stream flows into either the left or right bank of a stream represented as a double edged river, HoL linework will **not** be generated on the opposite bank.

RDCS3: Watershed boundaries will be generated for all lakes with a single inflow and single outflow.

RDCS4: Watershed boundaries will not be generated for a lake with a single outflow.

RDCS5: Streams that widens into a double side river and then back into a single line representation with no inflows on either bank will not generate watershed boundaries.

This information can also be used to implement the "implicit" HoL linework required by Requirement **R5** (although note that this conflicts with RDCS2. This conflict does not affect the software).

In addition, DAIDs solve the problem of how to deal with hydro segments which meet in 2-nodes. These segments are normally be considered to be part of the same drainage. This can be explicitly controlled via assigning them the same DAID.

The diagram below shows examples of how DAIDs are used to meet these requirements.

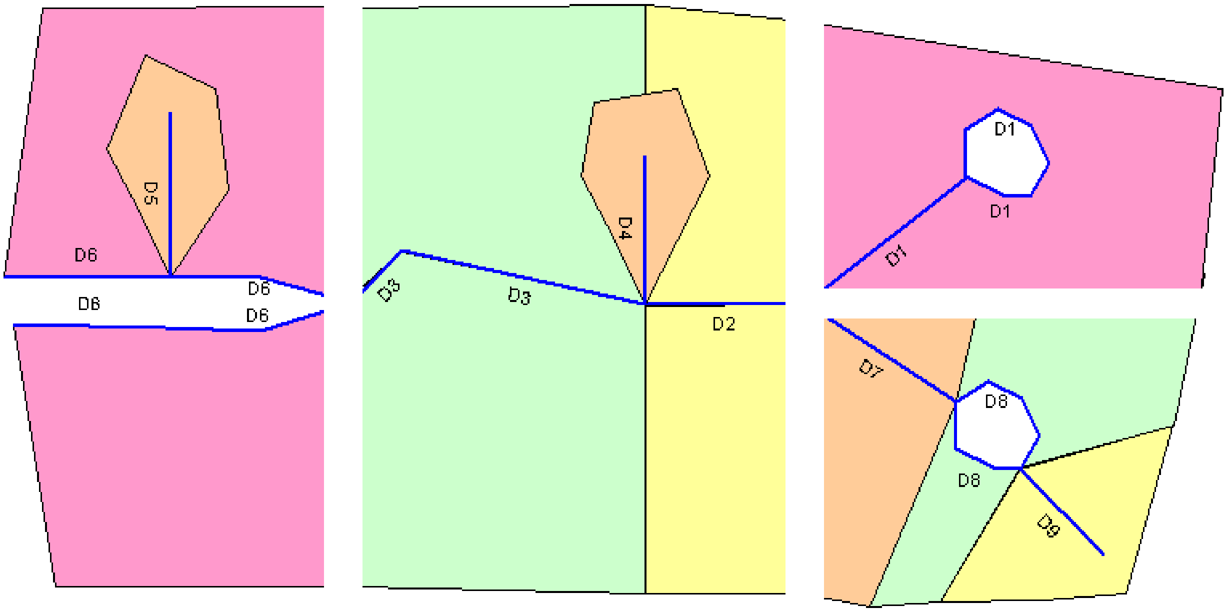


Figure 2: DAID Example

Note that the DAIDs could be used to determine disjoint watersheds, but this is likely not desirable.

The DAID assignment will likely have to rely on topological structure of the hydro edge network (e.g. in order to identify terminal lakes, implicit confluence boundaries, etc.) This process can be carried out in a separate module which generates the required IDs and saves them directly in the database.

4 INPUT CONDITIONING

4.1 MASS POINT THINNING

Mass points are thinned (removed) where they are too close to a Hydro Edge line. This improves the stability and quality of the constrained triangulation.

4.2 HYDRO EDGE CLIPPING

The actual area of processing is determined by the provided DEM points (including mass points, breaklines and spot heights). Hydro edges are clipped to the convex hull of the supplied DEM points.

5 TRIANGULATION

The triangulation of the input data forms the basis of all further processing. It may also be a useful product in its own right.

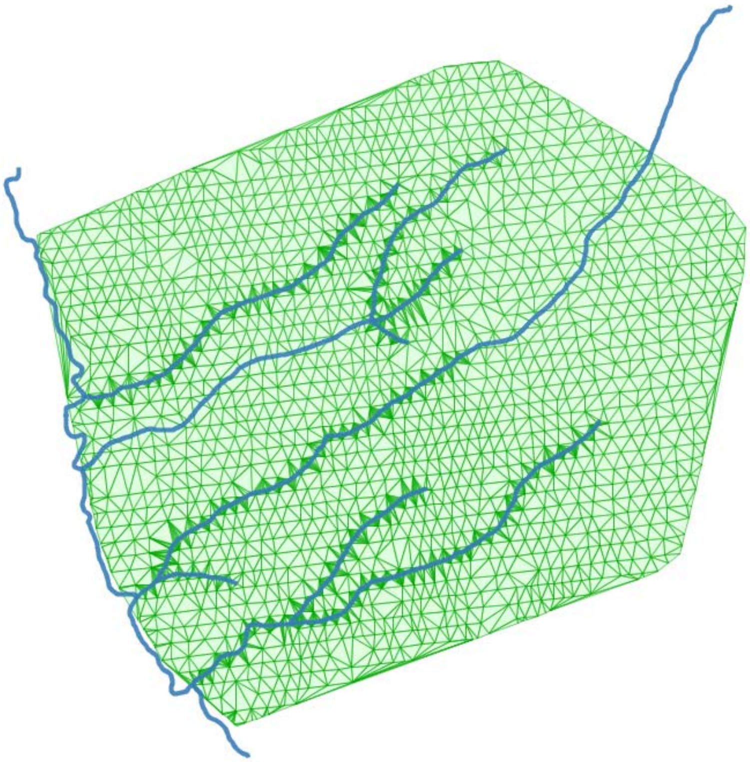


Figure 3 – Example of Triangulated DEM Incorporating Hydrographic Edges as Constraints

5.1 CONVEX HULL

Most triangulation algorithms require a bounded convex area in which to work. This is usually provided by the convex hull of some set of input data. It is essential to have a performant convex hull API available. This is provided by JTS. Some work was required to enhance the performance of the existing JTS algorithm. This includes:

- Make the radial sort for the Graham scan efficient and robust
- Use a heuristic method to reduce the number of points considered for the convex hull. The heuristic used is to construct the Maximal Interior Octilateral of the input point set, and eliminate all points which are contained in it.

5.2 BASIC TRIANGULATION

The basic triangulation (TIN) is constructed using an incremental insertion algorithm. The TIN data structure used is the Quad-edge Subdivision of Guibas and Stolfi. This structure forms the basis of all further processing performed on the TIN surface.

To enhance performance, the subdivision uses a modified locate method which starts from the most recently located edge

5.3 CONSTRAINT ENFORCEMENT

The basic Delaunay triangulation algorithm provides no control over what line segments appear in the triangulation. In order to ensure that the hydrographic edges appear as edges in the triangulation, further processing is required.

The approach taken is to use the constraint segment splitting method of ([Ruppert95]). This refines the triangulation until all of the original constraint segments are represented by one or more triangle edges. The Delaunay property is maintained during the splitting process, which allows the basic Delaunay incremental insertion algorithm to be used unchanged.

The robustness of the refinement process and quality of the resulting triangulation are heavily dependent on how the splits are performed. As described below, heuristics are used to optimize split locations, resulting in fewer and better triangles and preventing infinite loops in the refinement process.

5.3.1 Close DEM Point Elimination

DEM points which are very close to constraint segments are deleted from the input, in order to avoid splitting which would force very narrow triangles. This has no perceptible effect on the resulting surface and flow pattern, since these points are so close to the hydrographic edges that they would be expected to flow to that edge in any case.

5.3.2 Handling Narrow Input Angles

The original segment splitting of Ruppert causes repeated splitting on narrow input angles.

Also, it is desirable to maintain triangle quality as much as possible during splitting (i.e. it is desirable to create triangles with as large angles as possible). These goals can be achieved by the careful choice of two things: which encroaching vertex to split on, and where to place the split location. This is accomplished by using heuristics to control these factors. The heuristics used can take advantage of the nature of the geometry of the hydrographic constraint linework, including aspects such as

- the angle size between constraints is unlikely to be extremely small,

- there are unlikely to be many narrow angles very close to one another

The heuristics used are:

Nearest Encroaching Vertex Heuristic: Choose the encroaching vertex closest to the midpoint of the encroached segment (The original strategy did not use any special heuristics to choose an encroaching vertex to split on)

Longest Safe Split Segment Heuristic: Choose the split location to maximise the lengths of the split segments, but ensure that the split segments are not themselves encroached by the split vertex. (The original strategy of splitting at the midpoint of the segment did not ensure this). There are three candidate split locations:

- The point which is the same distance along the segment as the encroaching vertex is from an endpoint of the segment. If the encroaching vertex has an edge linking it to a segment endpoint, this results in creating an isosceles triangle, which guarantees that the encroaching vertex will not encroach on either split segment.

- The projection of the encroaching vertex on the segment

- A point along the segment which maximises the minimum length of the split segments, but which will not result in themselves being encroached by the original encroaching vertex.

To choose between these options, a further heuristic is used:

Furthest From Confluence Heuristic: The main culprit for causing the problem of repeated splits is narrow angles between constraints. These almost always happen at confluence points (equivalently, constraint endpoints). In order to avoid repeated encroachment, it is best to choose split points which are as far as possible from the apex of the narrow angle. This criteria is used to choose between the candidate split points computed by the previous heuristic.

Narrow Interior Angle Heuristic: Another place narrow angles can happen is interior to a constraint linestring. This heuristic detects this situation and attempts to choose a split point which does not encroach on the segment adjacent to the narrow angle.

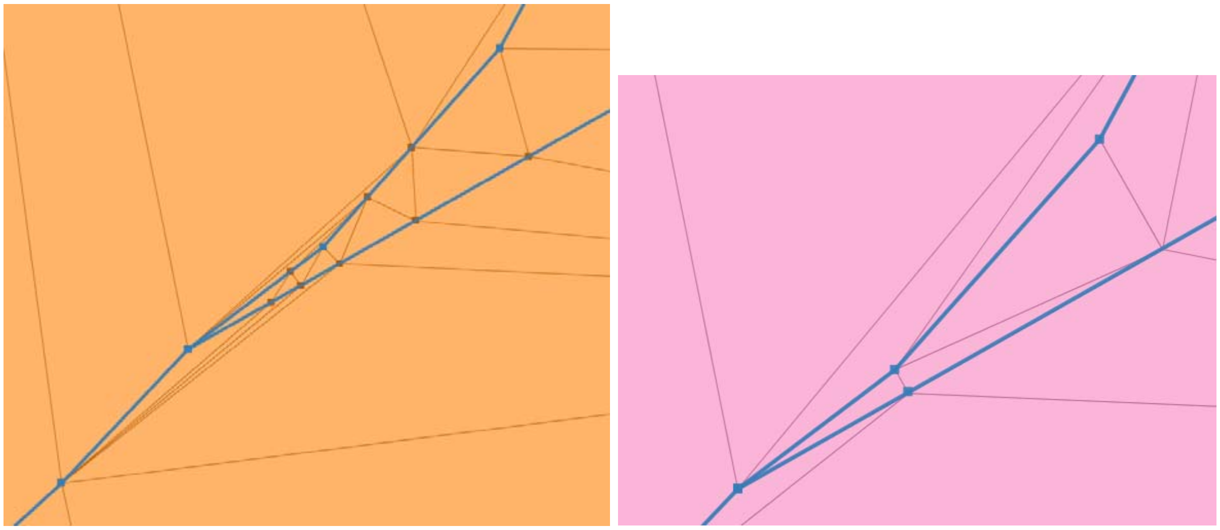


Figure 4 – Narrow Input Angles Causing Split Looping (L); Optimized Split Location Preventing Looping (R)

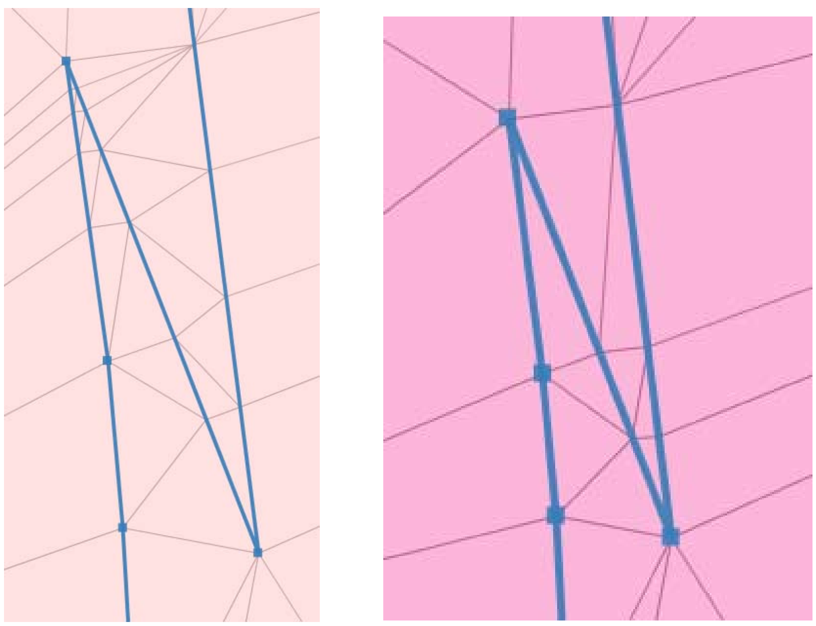


Figure 5 – Narrow Input Angles Causing Split Looping (L); Optimized Split Location Preventing Looping (R)

With all these heuristics implemented, a substantial increase in performance of the constraint enforcement phase was obtained. Both the time for enforcement and the number of splits created were reduced by about 40%.

5.4 REFINEMENT FOR TRIANGLE QUALITY

A further possibility when computing a triangulation is to refine the shapes of the triangles to maximise the minimum angle. This is the goal of several well-known triangulation refinement algorithms such as Ruppert and Shewchuk.

However, in practice it has turned out that the WBG algorithm is not adversely affected by the presence of narrow triangles. For this reason, no refinement for the purpose of improving triangle shape is implemented.

6 MEDIAL AXIS

In order that watershed boundaries can be traced between every constraint edge, constraint edges must be separated by at least two triangles. In areas where DEM points exist this occurs naturally, but since the DEM density is (much) less than the density of the constraints there are many areas where this does not occur. In particular, most confluences do not have enough DEM points nearby to ensure separation of the constraint linework. To provide this separation in a logical way, the medial axis of the constraint edges is introduced wherever it is required.

To introduce the medial axis into the TIN edges, it is sufficient to introduce new TIN vertices into facets and edges wherever the TIN does not currently separate constraint vertices. The new vertices are created to lie on the medial axis (within a reasonable distance approximation). The new vertices split the edge or facet into which they are introduced, along with any adjacent facets.

6.1 ALL-NODE TRIANGLE HANDLING

It can happen that a TIN triangle has all of its edges crossing the medial axis (which is also the watershed boundaries). This can happen when a triangle has inter-region nodes at every vertex. This in turn can happen in various situations:

- A triangle at the confluence of three double-line streams (see below)
- A triangle formed by a small estuary of a single line river, where the river forks into two single segments flowing into another water body (see below)

In this situation the medial axis contains the line segments connecting the vertices of the triangle to a point in its interior. This point is artificially introduced by splitting the triangle at an interior point. The split location used is the **incentre** point, which is the point at which the angle bisectors of the triangle meet.

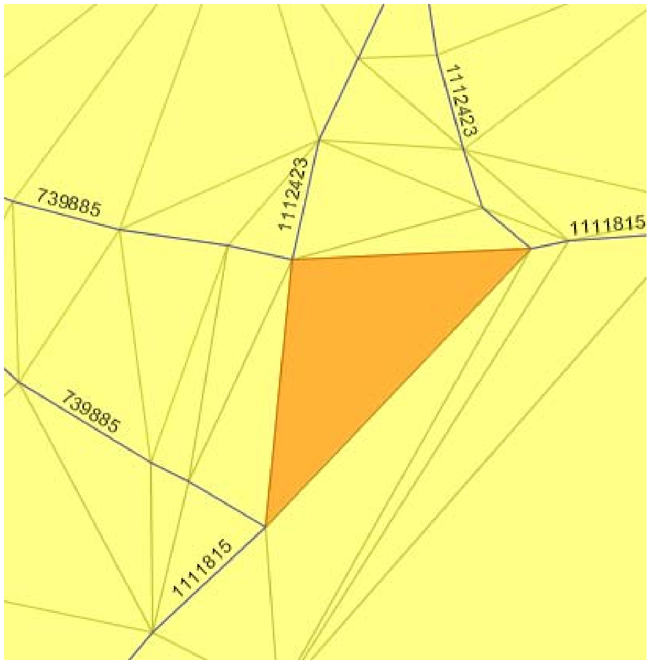


Figure 6 – A Triangle Where All Vertices are Nodes on Different Hydro Edges

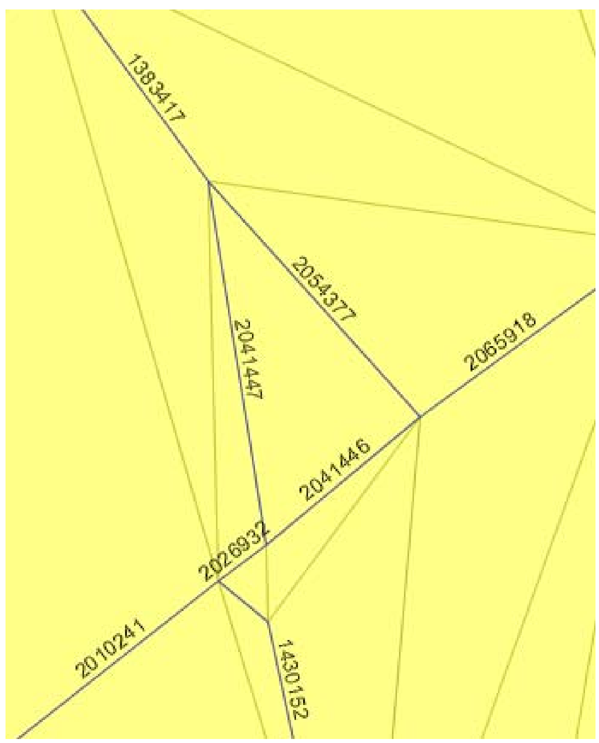


Figure 7 – A Triangle Where All Edges are Constraint Segments

7 LOW RELIEF HANDLING

Areas of low relief present problems in determining flow across the surface. Several problems are caused by low relief:

- Perfectly flat triangles have an undefined flow vector. They must be assigned a reasonable flow vector in order to allow flow modelling to be carried out.

- Local depressions in the surface (here called "pits") become more prevalent. These require special handling to ensure that they ultimately flow into a hydrographic edge.

Another way of stating this problem is that in areas of low relief there is no obvious ridgeline that divides watersheds. The solution to this is to introduce an artificial ridgeline in areas where it is required. There are various ways in which to do this.

7.1 TWA MEDIAL AXIS APPROACH

In the previous TWA project the approach to solving these problems was to compute the medial axis of the hydrographic network. This determines a "midline" linear network. This set of lines is added to the TIN as a further set of linear constraints. The TIN sites along the midline network are biased upwards by a small amount. The biased midline provides e artificial ridgeline, which only becomes significant if there is no higher ridgeline implied by the surface.

The bias offset needs to be chosen to be more significant that the noise in the DEM, but small enough not to obscure natural ridgelines. Heuristic experiments by TWA determined that the noise level in the DEM was about 1.3 m.

The medial axis approach has the disadvantage that the computation of the medial axis is required, which is a complex process in itself. In addition, it requires the addition of a large number of extra constraint edges to the TIN, which is the most expensive component of the TIN building.

7.2 BUBBLE BIASING

WBG incorporates an alternative approach to defining an artificial ridgeline, which is called "bubble biasing". Each non-hydrographic site in the DEM is biased upwards by an function which depends on the distance to the closest hydrographic constraint site. This has the effect of "bubbling" the surface upwards between the hydrographic constraints. The apex of the bubble determines the artificial ridgeline. The bias function is chosen to be small enough that it will not affect the flow pattern in areas of larger relief.

This strategy solves the low-relief problems:

- The presence of the bias ensures that perfectly flat triangles will not occur.

- The bias ensures that in the absence of relief, pits will drain towards the closest hydrographic edge, provided a suitable spilling algorithm is used (see below).

The images below show the bubbling bias field on its own, and as applied to the original surface. It can be seen that bubble biasing is a very subtle effect, which is a good thing, since it allows as much as possible of the original surface to be reflected in the derived watershed boundaries.

Note that if the bubble bias value is used as the TIN elevation value, the resulting watershed boundaries are a close approximation to the actual medial axis of the constraint network.

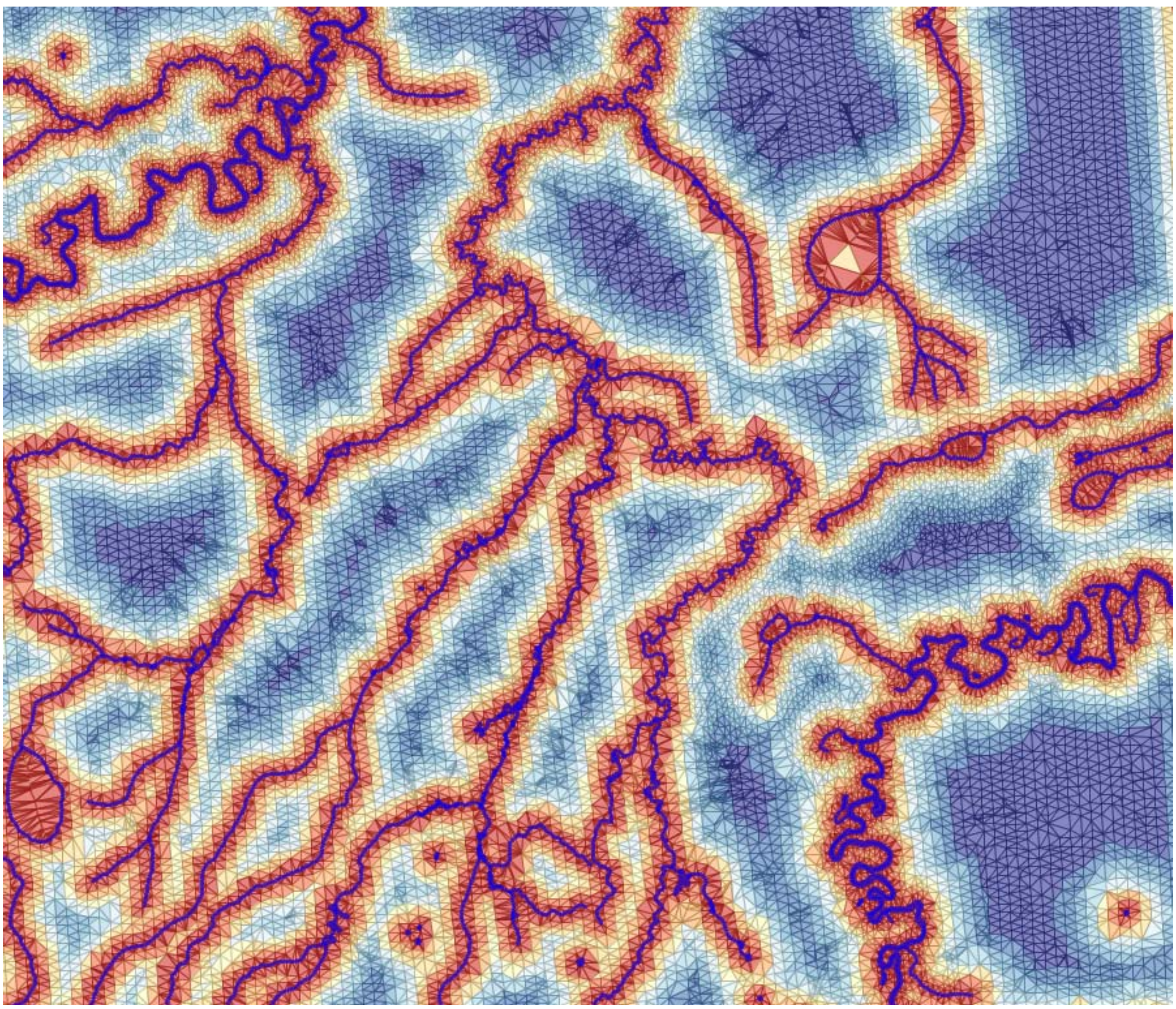


Figure 8 – TIN Surface Showing Elevation Determined by Bubble Bias Only

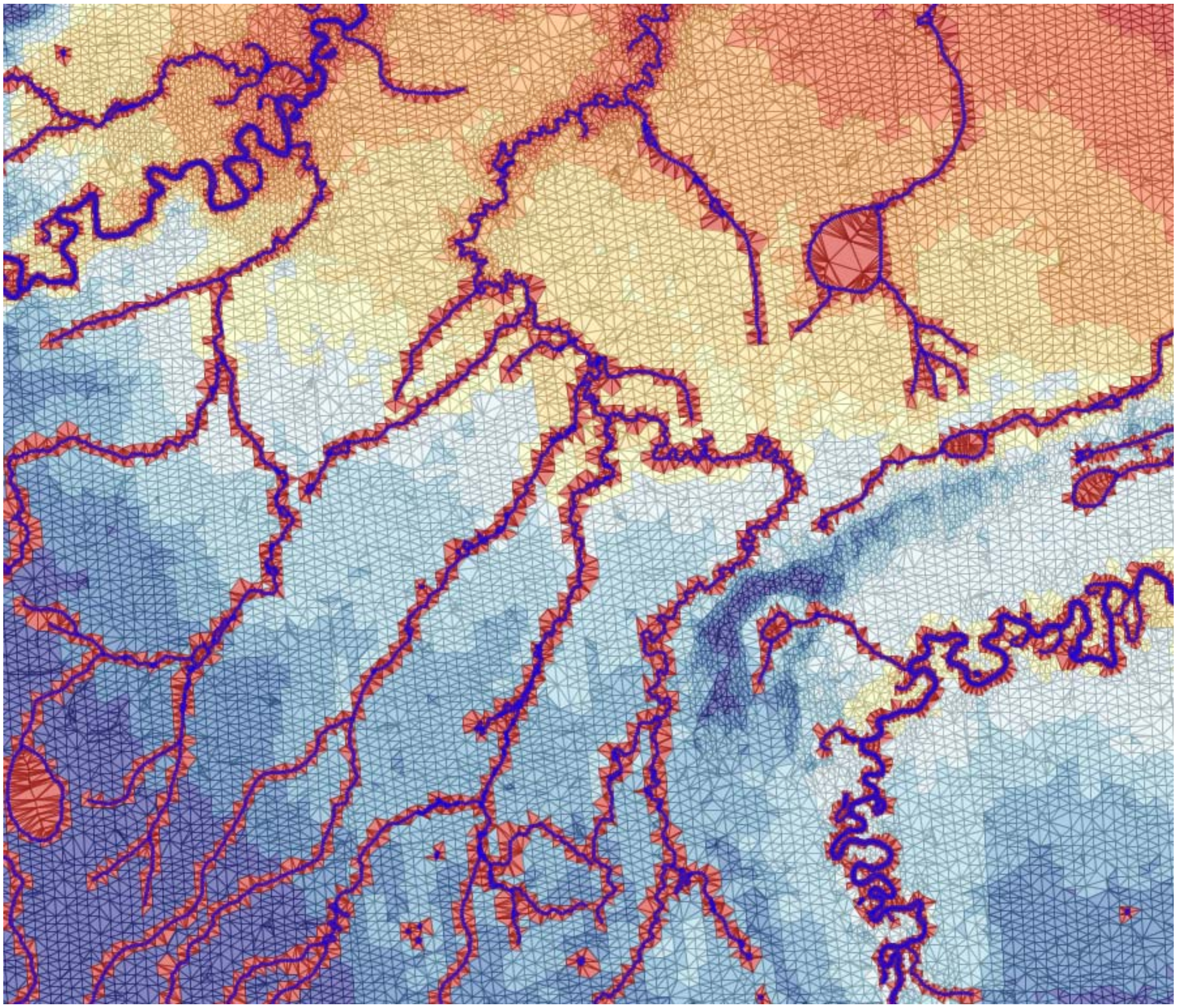


Figure 9 – TIN Surface Showing Original DEM Elevation with Bubble Bias Added

7.2.1 Closest Constraint Determination

In order to determine the bubble bias function value for each non-constraint site, the closest constraint site must be identified. This is performed by a breadth first traversal of the TIN graph, starting at triangles which are adjacent to constraint sites. The closest constraint site is propagated to the vertices of each triangle as they are encountered during the traversal.

8 FLOW MODELLING

The basic method of assigning drainage areas to triangles is to use the drainage field defined by the TIN to associate triangles with the hydro segment they drain into.

There are at least three methods of doing this which have been presented in the literature.

[McAllister99] uses a graph-based method. This method seems complex and has not been explored in this project.

[Skea98] in the original TWA software used a "backwards" or "flooding" approach. This method starts from hydro segments and follows lines of steepest ascent through triangles. This method was explored in detail in this project, but ultimately not used due to the reasons given below.

The method finally chosen for flow modelling is a "forward" or "trickling" approach. This is described in more detail below.

8.1 FLOODING APPROACH

Flooding has some advantages:

- Drainage areas are always connected
- Since flooding proceeds via a "front" through the TIN, most triangles are traversed once only

Flooding has one significant drawback. In order to provide a more accurate assignment along drainage area boundaries, it is necessary to split triangles. Due to the way flow propagates through the TIN, small splits can have big impacts on the upflow drainage assignment. The flooding method relies on an accurate downflow assignment in order to compute a correct upflow assignment. Furthermore, splitting triangles is a delicate operation, since it is important for stability that too-narrow triangles are avoided. This requires some heuristic rules. The final assignment is quite sensitive to the choice of these rules and parameters. This sensitivity to initial conditions and inherent instability make the implementation quite complex.

Furthermore, flooding has the relative disadvantage that no visual (geometric) indicators of actual flow paths are produced.

Ultimately these limitations decided against using a flooding algorithm.

8.2 TRICKLING

[Nelson94] used a "forwards" or "trickling" based method. This method starts from triangles and follows lines of steepest descent down to hydro edges. To discretize the problem flow lines are initiated at triangle centroids. This method has the advantage that tracing downflow is quite stable and relatively simple to implement. It never misassigns triangles, in the sense that if a significant proportion of a triangle drains to a single watershed, the triangle will be assigned to that watershed. Triangles on the boundary of one or more drainage areas may be assigned to an area which drains a relatively small percentage of the triangle, but this error is bounded by the size of the triangle.

Triangle splitting is still required in the trickling method, but because the overall drainage area assignment is not dependent on accurate splitting, fewer and simpler heuristics can be used to ensure that splitting is done in a stable way.

To put this another way, in flooding the assignment of upflow triangles depends critically on the assignment and splitting carried out downflow (often a long way downflow). In trickling the assignment of each triangle is independent of all the others,

Trickling can result in disconnected drainage areas. Disconnected upflow areas indicate TIN flow of a potential pathological nature, which may not represent the true flow pattern. (E.g. flow along very narrow "channels" in the landscape. This was recognized as a problem in the original TWA work). Further rules and processing are required to handle discontinuities. These rules may provide a reasonable way of eliminating or at least limiting channels as well.

Trickling has an advantage of producing "trickle traces", which are lines showing the flow across the surface. This provides an effective way of visually inspecting the results of the trickling algorithm, and confirming the correctness of the derived watershed boundaries.

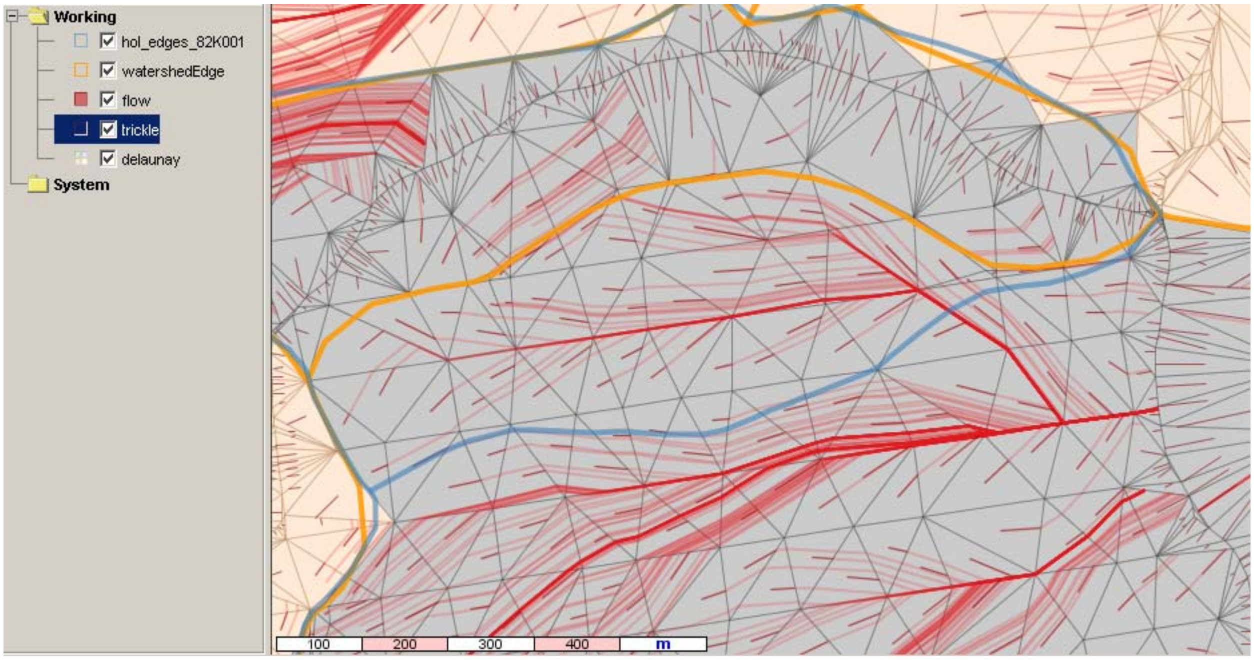
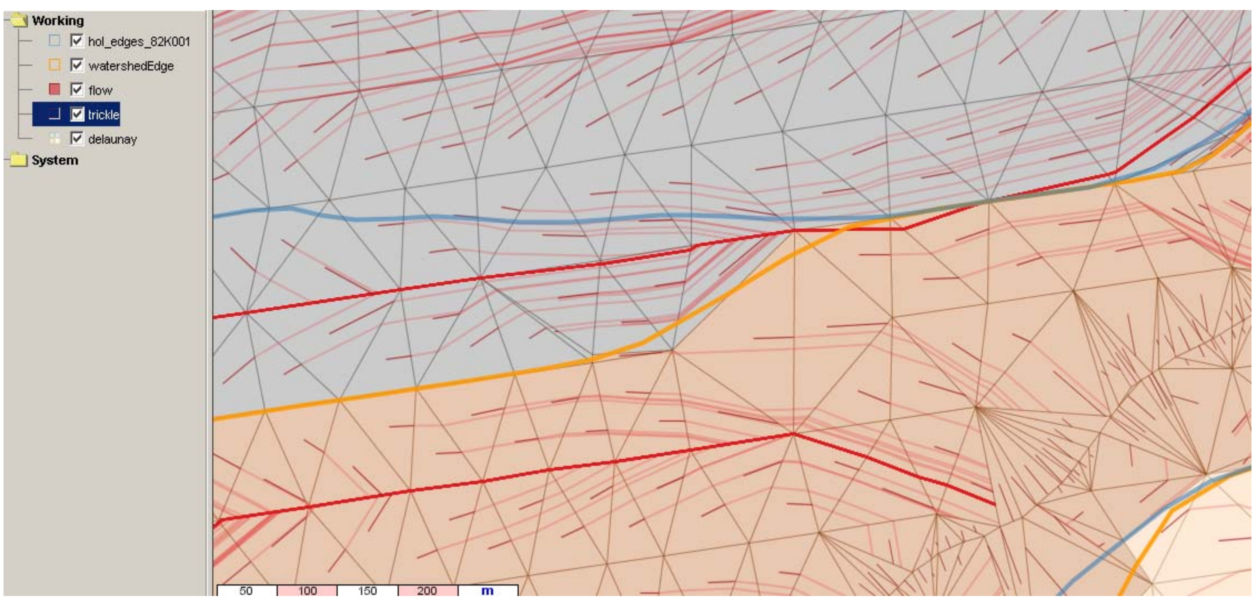


Figure 10 – Example of Difference Between Flooding and Trickling.

In this case trickling is clearly more correct as indicated by the flow lines (Orange is HOL from trickline, blue is HOL from TWA flooding)



8.3 TRIANGLE SPLITTING

The initial triangulation has no connection to the actual flow lines across the induced surface. It is not uncommon for flow to "bisect" a triangle, in the sense that the triangle has two of its edges as outflow edges, and the flow across each edge terminates in different drainage areas. Since the triangle can ultimately be assigned to only a single drainage area, this tends to have the effect of making boundaries between drainage regions "jagged".

In order to refine the triangulation to more closely follow the flow across the TIN, these "bisected" triangles can be split along the flow line. This ensures that the TIN follows the flow exactly for the original triangle which is split. The triangle splitting is accomplished within the TIN data structure. Since the splitting must maintain the triangulation, the neighbour triangle across the split edge must also be split. It is usually the case that this split does not follow the flow lines very closely.

Splitting is iterated to improve the TIN relative to the flow lines as much as possible. No attempt is made to maintain the Delaunay property during splitting. Some heuristic constraints based on triangle minimum angle and width are required to ensure that splitting terminates and does not cause very narrow triangles.

8.4 DETERMINISTIC TRICKLING

There are two possibilities for trickling across a triangle: the flow runs across the face of the triangle, or the flow runs along the edge of a triangle in a channel. Channel flow results in the trickle ending at a vertex of the triangle. At this point there are several possibilities:

1. If the vertex is a pit (i.e. all adjacent triangles flow into the vertex) the flow terminates

2. The flow may proceed along a face of an adjacent triangle

3. The flow may proceed along a channel between two adjacent triangles

It is desirable that the flow pattern be locally deterministic. This means that a given local surface configuration always produces the same local flow pattern. In order to enforce this, cases 2 and 3 above are chosen between by choosing the case which maximises the slope of the flow vector.

9 DRAINAGE AREA ASSIGNMENT

Once the flow across the terrain has been computed for every triangle, triangles can be grouped into regions labelled by the drainage area ID they flow into. The general case is determined by two rules:

- Triangles adjacent to a hydrographic edge are assigned to the region for that edge
- All other triangles are assigned to the region for the edge or vertex which they drain to

However, because of the complexity of flow patterns across a TIN, these two rules are usually not sufficient to ensure that a well-connected and high quality region coverage is generated. The following sections discuss situations which require special handling.

9.1 SPILL HEIGHT

A key concept for processing the terrain surface is the concept of **spill height**. The spill height of a triangle is a measure of how "low" the lowest point on the boundary of the triangle is. The **spill height** of an edge of a triangle is a pair <lowest vertex elevation, slope>. The spill edge is the edge with the lowest spill height. Physically, the spill edge corresponds to the edge that a trickle directly above the lowest vertex of the triangle would flow out of.

Spill heights are ordered lexicographically. The key aspect is that this permits defining a total ordering on a set of triangles or edges which corresponds to their order relative to the flow over the terrain. This concept is used in the following algorithms to ensure that the terrain is respected.

9.2 PIT MERGING

"Pits" are regions on the TIN where the flow lines do not terminate on a constraint segment (hydrographic edge or boundary edge). This prevents them from being assigned to a Drainage Area by the Assignment algorithm. A further process is required to assign pits to the most appropriate neighbour region.

The standard way of handling pits is to identify the lowest point on the pit region boundary, and assign the entire pit region to the region adjacent to it. Since pits are merged across edges rather than vertices, the merge edge is chosen to be the edge with lowest spill height out of all the border edges of the pit.

Pits may be disconnected. Merging the pit may not resolve the disconnectivity - it may simply transfer it to another region. Any remaining disconnected triangles are handled by the following phase of processing.

9.3 DISCONNECTED REGION MERGING

Due to the complexity of flow over the surface, trickling can result in disconnected regions. Both absolutely and edge-wise disconnectivity can occur. In order to guarantee a fully connected drainage area coverage, the drainage area regions must be forced to be edge connected. This is accomplished by merging disconnected areas into edge-adjacent regions. The merging must be performed in a way that respects the terrain as far as possible (although it necessarily will not be absolutely conformant with the drainage pattern).

To ensure that every region is edge-connected, a traversal of edge-connected triangles is performed, marking connected triangles. Triangles which are left unmarked are disconnected. The disconnected triangles are visited and each edge-connected subset from the same region is merged into a connected neighbour region.

In order to ensure that the merging respects the terrain, disconnected areas are merged in order of their spill height. A disconnected area is merged into the connected region adjacent to the triangle with lowest spill edge.

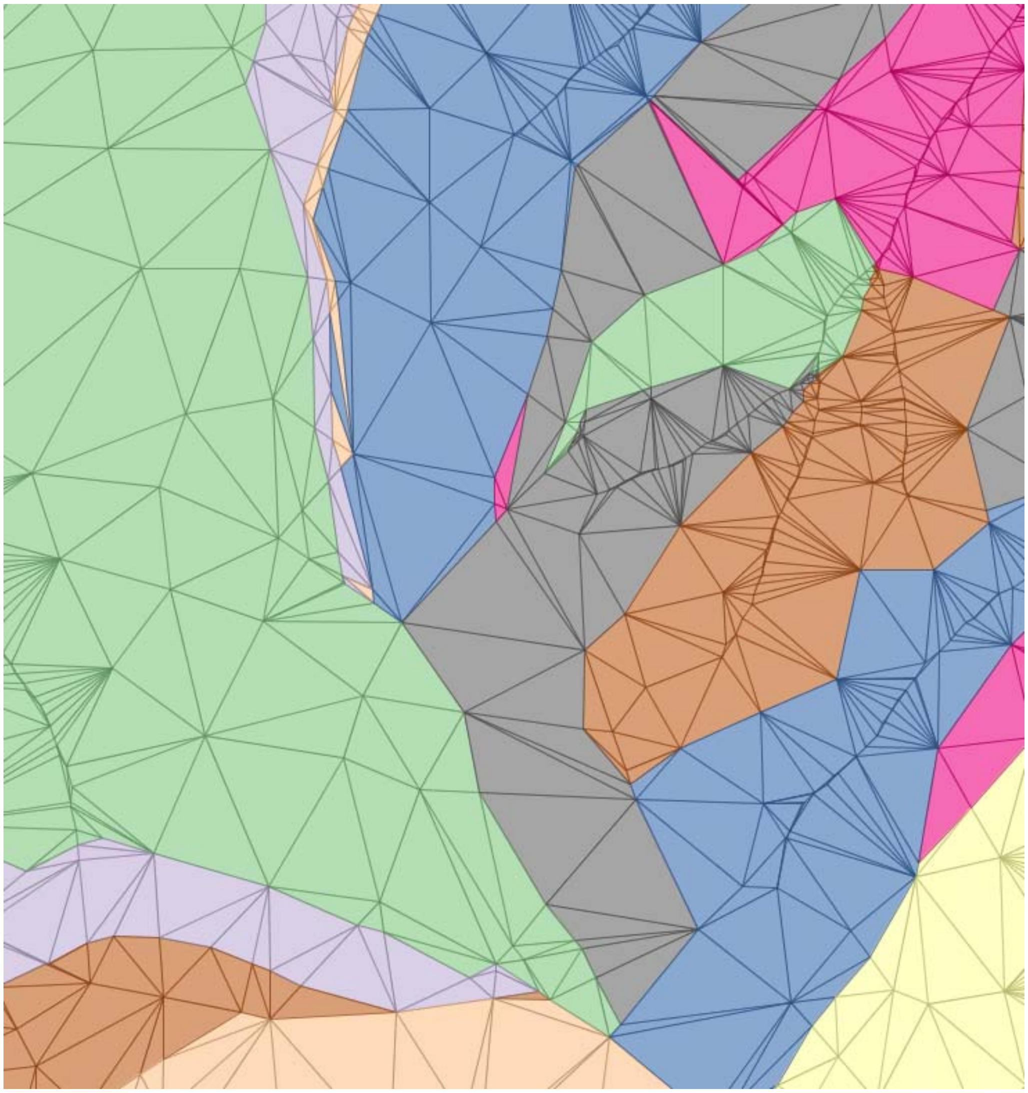


Figure 11 – Example of Disconnected Regions - several are visible in this image

9.4 SPIKE CLIPPING

"Spikes" are triangles which protrude along an otherwise relatively smooth boundary between two drainage regions. They cause narrow angles to appear in the raw watershed boundary linework. They occur because of the fact that the triangulation can sometimes contain narrow triangles, and because of the particular flow pattern in that area. (Spikes can be thought of as "single-triangle corridors", and they occur for similar reasons.

Spikes are identified by two criteria:

1. The base of the spike is smaller than the **Maximum Spike Base Length**
2. The angle at the end of the spike is less than the **Maximum Spike Angle**

The following images show some examples of spikes:

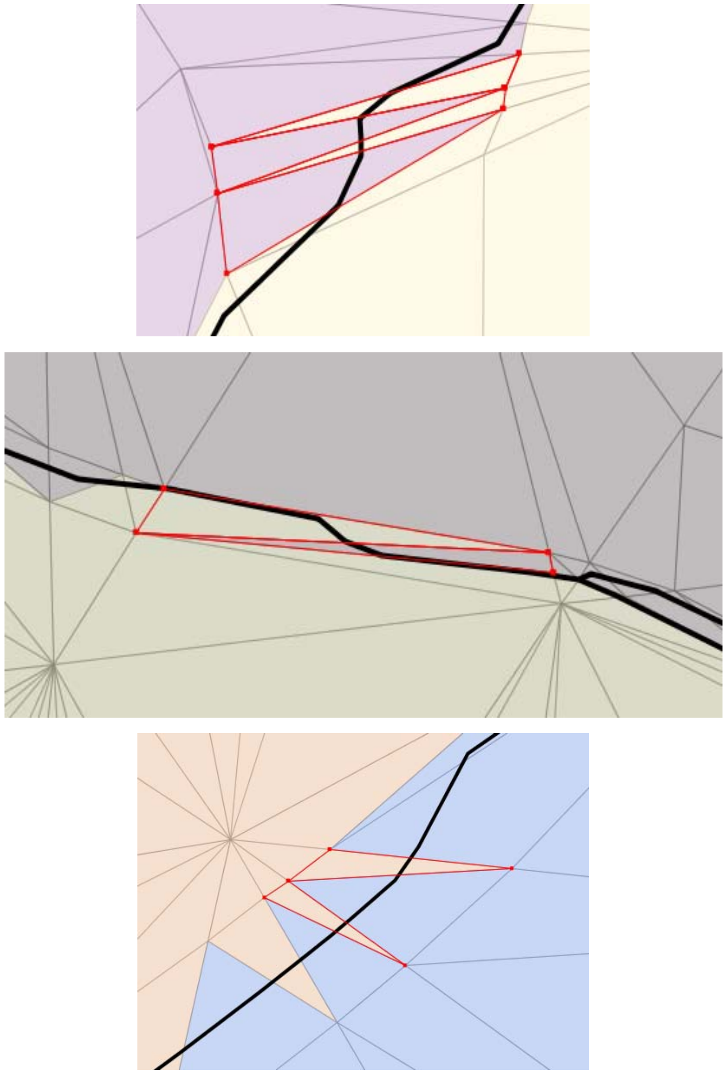


Figure 12 – Examples of Spikes.

Spikes have a tendency to distort the extracted watershed boundary linework, even when smoothing is performed. In order to improve the linework, the spikes are removed. This is done by changing the region of the spike triangle to be the region of one of its neighbours. As can be seen from the images, several spikes can occur adjacent to one another. In order to provide a reasonable merging of the spikes, the merging must be done on a global basis, rather than on a per-spike basis. This is accomplished by first nulling the regions of all spikes, and then merging the null areas (which will often be just a single triangle) left in the TIN.

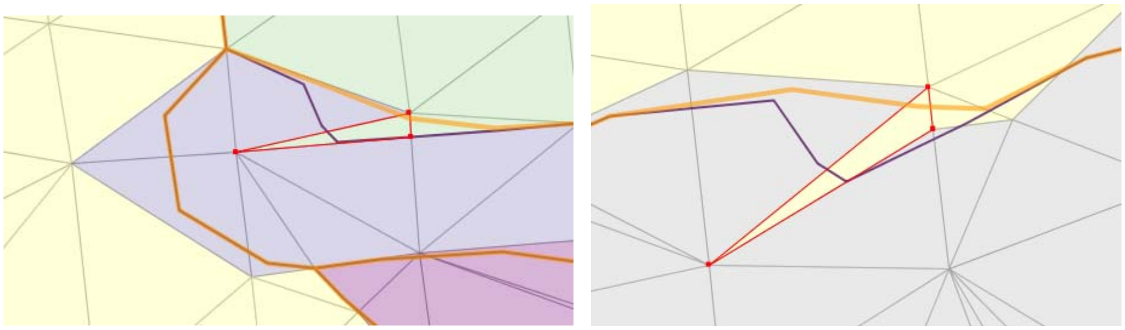


Figure 13 – Examples of how Spike Clipping Improves the Boundary Linework (purple = before)

10 WATERSHED BOUNDARY EDGE PROCESSING

The final stage in the process is to extract the edges defining the boundary between each adjacent watershed, and form the final watershed polygons.

10.1 BOUNDARY EDGE EXTRACTION

Boundary edge lines are determined by TIN edges which separate triangles in different regions. The boundary edges form a polygonal coverage (i.e. they do not touch except at endpoints). Boundary edge nodes are detected by inspecting each TIN vertex to see if 3 or more drainage regions are incident on it. Given the set of boundary nodes, the boundary edges are extracted by "walking" the dividing line between each pair of regions incident on the node.

10.2 BOUNDARY EDGE SMOOTHING

Since the triangulation tends to produce "spiky" paths through its edges, the watershed boundaries need to be smoothed. There are many standard algorithms to do this. The original TWA work used a midpoint smoothing algorithm. Others include Slide-Average Smoothing.

WBG uses a Centroid Smoothing algorithm, iterated twice. This provides a good balance between providing visually pleasing smoothing, while preserving the character of the underlying raw linework.

10.2.1 Preserving Topology Under Smoothing

The conventional Centroid Smoothing algorithm has only a local awareness of data topology. The problem with a purely local approach to smoothing edges is that it can result in the following violations of topology rules:

- An edge may self-intersect
- Two edges may intersect at a non-endpoint
- An edge may intersect a hydrographic edge at a point which is not an endpoint of valence >= 2

To prevent these violations, the WBG smoothing algorithm tests smoothed linework against all other linework for the occurrence of the above topological failure conditions. If a failure is found, the smoothing is "retracted" to avoid the condition.

It is not sufficient to simply avoid smoothing an entire edge, since this would result in visually unacceptable edge linework. Luckily, smoothing failures are usually very localized along the edge. The basic smoothing algorithm is enhanced to operate in a “progressive” way, so that even if smoothing is retracted at one point along an edge, smoothing at other points can still be carried out.

Closed linestrings of length <= 4 cannot be smoothed via centroid smoothing, since this causes the smoothed line to collapse. These lines are simply left in their raw state.

10.2.2 Respected Vertices

In some cases it is necessary prevent vertices from being adjusted by the smoothing process. These cases are:

- Where a vertex is a DEM spot height
- Where a vertex is a local maximum in the TIN
- Where an adjustment would cause the resulting boundary to intersect the hydrographic network

The smoothing process accepts a list of vertices which should not be adjusted, and ensures that they are not changed.

10.3 WATERSHED POLYGON FORMATION

Watershed Polygons are formed from the collection of Watershed Edges. Since the edges are constructed to be a interior-disjoint, correctly noded set of linestrings, this is easily accomplished by a standard polygonization algorithm. The Drainage Area IDs are propagated through the polygonization and attached to the appropriate resultant polygon.

The WaterBuG software does not actually perform the polygonization of the boundary linework. This is carried out in a separate step after generated blocks have been merged.

11 EXTERNAL MEMORY COMPUTATION

Due to the large size of the datasets involved, it is infeasible to compute the entire province in single run in the memory space of a single machine. It is thus necessary to partition the input into blocks, and process each block independently. In addition to making the processing feasible, this also has the advantage of allowing fast reprocessing of blocks which failed to process in some way. It also allows easily recomputing specific areas of the province where either the input data or the nature of the processing has changed.

Each block consists of a single BCGS 1:20,000 mapsheet, in addition to a buffer around the block to prevent "edge effects" from compromising the linework actually lying in the mapsheet. The buffer size can be varied from block to block to allow for variations in the input data (e.g. such as low-density DEM points in glacial areas).

12 QUALITY ASSURANCE

The following QA checks are performed after the linework has been generated for a block.

12.1.1 Topological Checks

- WB edges do not self-intersect
- WB edges do not cross (these two are equivalent to testing that the watershed boundary edges form a clean coverage)
- WB edges do not cross Hydro edges
- every Drainage Area ID is contained in a WB polygon with that ID on the appropriate edge side
- every WB polygon contains a hydrographic edge, with the same Drainage Area ID as the appropriate side of the polygon edges

12.1.2 Linework Quality Checks

- minimum angle between WB edge and Hydro edge at confluence
- minimum angle of bend along WB edge
- minimum angle between WB edges at node

13 BIBLIOGRAPHY

Dillabaugh, C. (2002). *Drainage basin delineation from vector drainage networks*. Symposium on Geospatial Theory, Processing and Applications, Ottawa.

Guibas, L. and J. Stolfi (1985). *Primitives for the manipulation of general subdivisions and the computation of Voroinoi diagrams*. ACM Transactions on Graphics, 4(2), pp. 74-123

McAllister, M. and Snoeyink, J. (1999). *Extracting Consistent Watersheds from Digitial River and Elevation Data*

Nelson, E. J., N. L. Jones, and A. W. Miller (1994). *Algorithm for precise drainage-basin delineation*. Journal of Hydraulic Engineering 120 (3), 298-312.

Ruppert, J. (1995). *A Delaunay refinement algorithm for quality 2-dimensional mesh generation*. Journal of Algorithms, Vol. 18, No. 3, pp. 548-585.

Shewchuck, J.R. (2002). *Delaunay Refinement Algorithms for Triangular Mesh Generation*, Computational Geometry: Theory and Applications 22(1-3):21-74.

Skea, D., P. Friesen, J. Carr, A. Gallagher, I. Barrodale, E. Davies, F. Milanzzo, B. Corrie, and B. Noureddin (1998). *GeoData BC|TRIM watershed atlas*. http://ssbux2.env.gov.bc.ca/srmb/twa home.htm.

Alper Üngör (2004). *Off-centers: A New Type of Steiner Points for Computing Size-Optimal Quality-Guaranteed Delaunay Triangulations*, Proceedings of LATIN 2004 (Buenos Aires, Argentina).