

EXTRACTING CONSISTENT WATERSHEDS FROM DIGITAL RIVER AND ELEVATION DATA

Michael McAllister Jack Snoeyink
Department of Computer Science
University of British Columbia
mcallist@cs.ubc.ca snoeyink@cs.ubc.ca

Abstract

The Terrain Resources Inventory Mapping (TRIM) data standard in BC, Canada, includes specifications for river data and elevation data that are typically met by interpretation of stereo orthophotos. It does not specify that breaklines for watersheds be interpreted from the photos, thus we must extract them from the data. We seek a system of watersheds that, while not exact due to errors in the data, is at least self-consistent – having no overlapping watersheds and no unclassified points.

We build a TIN terrain model using a Delaunay triangulation that combines both river and elevation data. Based on the standard definition of water flow along steepest-descent paths, we create an algorithm that identifies ridges and channels in the TIN, and extracts watersheds. Several unexpected geometric configurations, which we have not seen in the literature, follow from the standard definitions. These must be correctly computed to obtain a consistent system of watersheds.

1 INTRODUCTION

The old data processing phrase “garbage in, garbage out” has never implied that the input of good data will necessarily produce good information. In analysis of data in a Geographic Information System, one also needs good models for the underlying reality that the data represents and algorithms that faithfully implement computation on the model. With consistent models and algorithms, even less-than-ideal data can produce useful information. In this paper, we illustrate this with the computation of systems of watersheds that, while not exact due to errors in the data, is at least self-consistent – having no overlapping watersheds and no unclassified points.

In particular, we use data that is collected according to the Terrain Resources Inventory Mapping (TRIM) specifications (Ministry of Environment, Lands, and Parks, Province of British Columbia 1992). Two types of data in the TRIM specification are relevant for watershed computations: point elevation data collected at 50 to 75 meter spacing, and digitized river segments, collected at greater detail. Both are specified in UTM coordinates with accuracy of 10 meters in x and y and 5 meters in elevation; typically the data comes from stereo digitizing of orthophotos. Ridges and other hydrographic breaklines are, unfortunately, not uniformly collected.

While the lack of hydrographic breaklines limits the accuracy when we compute a system of watersheds, we can still aim for a systems that is consistent with the two types of data, is consistent with standard assumptions of steepest-descent flow, and is self-consistent in assigning each point to exactly one watershed. We achieve this by building a terrain surface model that combines the

two types of data, and computing watershed boundaries by careful analysis of the geometry of steepest-descent flow on this surface, including the handling of degenerate cases.

We are all familiar with data inconsistencies within one data set, when the quality of the data fails to meet its encoding or digitizing standard. Nevertheless, the standard or expectation on the data set gives us a target to which we can correct the data: polygons can be closed, edges can be reversed, features can be relabeled, points can be added or moved. While sometimes a tedious task, individual data sets can be made self-consistent.

In watersheds, inconsistencies also occur between different types of data sets that represent the same geographic area. Figure 1 illustrates a river that is not completely contained in its watershed. (In this case, the watershed was digitized from a 1:50 000 scale map while the river was digitized from a 1:20 000 scale map.) Section 4 describes a method for deriving watershed boundaries from elevation and river data that tries to avoid the inconsistencies of figure 1.

A major source of inconsistency in watersheds, however, is model inconsistency. Frank et al. (Frank, Palmer, and Robinson 1986) point out that formal definitions should be used to define terrain-specific features so that properties of the structure can be established mathematically and inconsistent definitions can be avoided. We believe that the best way to obtain consistency is to adopt standard models for terrain representation and water flow, but then look carefully at the mathematics of the interaction of these models and be prepared to handle outputs that are initially unintuitive. For a specific example, we may expect that the watershed of a point is a single polygon, is connected, and either encloses an area or is a single linear path along a slope. As we describe in Section 3, a consistent implementation of the earlier modeling assumptions allows a watershed to have a disconnected interior or to have no area yet not be a single line on the TIN. These degeneracies can be classified as inconsistencies in our expectations of the model or of the actual terrain.

In the next section, we survey prior work on the computation of watersheds. In Section 3, we define notation and briefly discuss how we combine river and elevation data into a TIN terrain model. In Section 4, we describe our algorithm for computing watershed boundaries, paying particular attention to data structure elements that support the degenerate cases that are inherent in combining the standard models for TIN terrains and steepest-descent flow. In Section 5, we give some sample results from our implementation inside the Cause & Effect decision management system.

2 ALGORITHMS FOR WATERSHEDS

Several algorithms have been proposed for finding watersheds or watershed boundaries under the standard assumptions of steepest-descent flow.

Many algorithms begin with raster elevation data and use filter techniques that are common in raster image processing to identify terrain features (Peucker and Douglas 1975; Douglas 1986; Peucker and Chrisman 1975; Seemuller 1989; Takahashi, Ikeda, Shinagawa, Kunii, and Ueda 1995) or to find watersheds themselves (Jensen 1985; Collins 1975; Jenson and Domingue 1988). Because it can be difficult to join the fragments of watershed boundary detected by local filters into a consistent whole, Mark's global approach of discretizing flow on a raster also has adherents (Mark 1978; Mark 1984; O'Callaghan and Mark 1984).

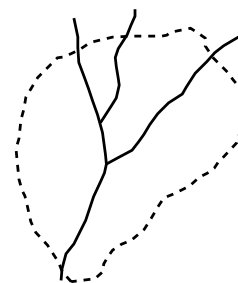


Figure 1:
Inconsistent river &
watershed

Some algorithms work with contour data (Hutchinson 1988; Kweon and Kanade 1994; Morris and Flavin 1990; Tang 1992), although this is not mathematically correct (Koenderink and van Doorn 1993).

One algorithm approximates watersheds using river data only—(Skea et al. 1998) computes the Voronoi cells of each river in the river network and treat the cells as the watershed of the river. Although self-consistent, this approach ignores the landscape around the river. Subsequently, Skea et al. used a hybrid approach that accounted for steep terrain.

Because our data is not gridded and combines lower density elevation data with river data, we prefer terrain models structured as TINs (triangulated irregular networks).

Among algorithms that compute flow on TINs, some consider the triangles (Palacios-Velez and Cuevas-Renaud 1986) or the edges (Frank, Palmer, and Robinson 1986; Silfer, Kinn, and Hassett 1987; Theobald and Goodchild 1990) as the basic units for discretizing flow and forming watershed boundaries. To obtain consistency properties, however, it is necessary to subdivide triangles.

Nelson and others (Jones, Wright, and Maidment 1990; Environmental Modeling Research Laboratory, Brigham Young University 1998; Nelson, Jones, and Miller 1994) create a triangulated irregular network (TIN) of the landscape, subdivide each triangle of the TIN into regions whose points drain to the same pit in the terrain, and group the regions that drain to the same pit p as the watershed of p . The TIN can embed river data as breaklines to improve the accuracy of the TIN. Nelson et al.'s watershed for a point p is consistent with the terrain, but we show in Section 3 that it may not be a single polygon.

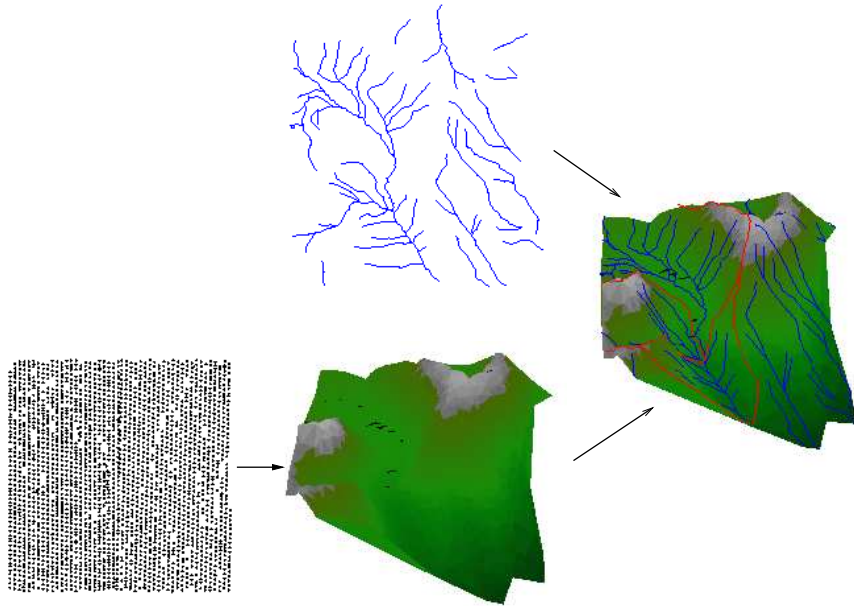


Figure 2: Combining river and elevation information to compute watershed boundaries

In our approach, we apply the same constraints as Nelson et al. to find watersheds. As illustrated in figure 2, we create a TIN that is a suitable fit to the terrain elevation data, embed that river data to correct the TIN, and assume that water flows along the path of steepest descent on the TIN. Rather than subdivide TIN triangles, we create an embedded planar graph from the structure of the TIN in which each face of the graph corresponds to the watershed of one pit in the TIN. From

this graph, any face-tracing algorithm extracts the watershed boundaries. Our graph for identifying watershed boundaries is consistent with the TIN and the water-flow assumption. Moreover, every watershed appears as a single polygon, with the possibility of having a degenerate boundary.

3 BASIC MODELS AND WATERSHED STRUCTURE

The steepest-descent flow model is commonly used for surface water. Let $trickle(p)$ be the path of steepest descent out of a point p . Then the watershed of a point q on the terrain is $watershed(q) = \{p \mid q \in trickle(p)\}$. The advantage of the steepest-descent flow model is that paths of steepest ascent out of p delimit the areas that drain into p in a neighborhood of p .

We find the TIN to be the natural terrain model for our work, not only because we have been using BC TRIM data (Ministry of Environment, Lands, and Parks, Province of British Columbia 1992), which has irregularly sampled elevation points, but also because it can more easily capture the non-uniform distribution of channels that we are trying to model.

We create the TIN with an algorithm from Garland and Heckbert (Garland and Heckbert 1995). We begin with a (mostly) regular grid of elevation data. In our case, the elevation data comes from BC TRIM data, which is manually digitized from stereo orthophotos. Other forms of remotely-sensed data, such as radar plots, are also suitable as input. The user supplies a desired error tolerance for the TIN along with the data. The TIN is then constructed incrementally. We begin with a two triangle decomposition of the 2D bounding box for the terrain points. Points are successively added to the triangulation by finding the raster point whose vertical distance to the TIN is greatest, adding the point as a vertex of the TIN, and retriangulating the TIN vertices with a 2D Delaunay triangulation algorithm. We continue to add points to the TIN until the vertical distance between every raster point and the TIN is bounded by the error tolerance.

The TIN construction ignores the terrain features when it triangulates its vertices. As a consequence, a TIN edge can inadvertently force a blockage across a narrow valley. Known rivers, valleys, and ridges are often embedded in the TIN as breaklines to remedy these types of errors. We add the points of these features as TIN vertices and force the edges that trace these features to be TIN edges. In general, breaklines help the TIN to have similar drainage characteristics to the terrain. However, the combination of data still leaves us open to inconsistencies between data sets: river breaklines may not match the elevation data perfectly, whether in elevation or in xy position.

To combine the steepest-descent flow model with a TIN, we define features of the TIN that affect the watersheds. An edge e of the TIN is a *ridge* if neither of the two triangles bounded by e send flow across e . In this case, the triangles bounded by e drain away from one another at e and e acts as a watershed boundary. A vertex v of the TIN is a *pit* if all the land in a small neighborhood of v drains into v ; all the land around v has a higher elevation than v . A vertex v is a *peak* if all the land in a small neighborhood of v drains away from v ; all the land around v has a lower elevation than v . Finally, a vertex v is a *saddle* if the land in a small neighborhood around v that have a lower elevation than v do not form a half-open ball around v . Saddles form passes in the terrain. More mathematical definitions for these terms are provided by Yu, van Kreveld, and Snoeyink (Yu, van Kreveld, and Snoeyink 1996).

We expect that watersheds of pits have certain characteristics: connected boundaries that have no self-intersections, connected interiors, and interiors that are disjoint from all other watersheds. Only the last characteristic, disjoint watersheds for different pits, is completely consistent with the combination of TIN and steepest-descent flow models.

Since the direction from which water flows out of a point is independent of how the water enters the point, the TIN and steepest-descent flow models can have unexpected interactions at saddles. For example, suppose that TIN vertex v is incident on a TIN triangle t and that a path of steepest descent π on t leads directly into v . Then water on t to either side of π drains to either side of v while water along π drains into v . If v is a saddle point, it is consistent with the models to have the water at vertex v drain to a pit p_1 while the water to each side of π on t drain to two different pits p_2 and p_3 as in figure 3. Consequently, path π belongs to the watershed of p_1 but there is no area along π ; the boundary of the watershed for p_1 is self-intersecting. In a more extreme case, water from some area of land can be funneled by a pair of ridges to drain along the path π . This extreme case creates a watershed whose interior is not connected, as we will illustrate in figure 5.

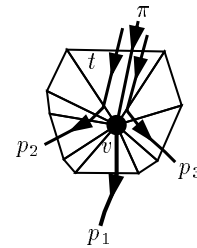


Figure 3: Flow to different pits

4 BOUNDARY-BASED ALGORITHM

Previous algorithms match areas of land, whether grid cells or TIN triangles, with the pit to which the land drains. When a watershed has a disjoint interior as in Section 3, these algorithms either report the watershed of a point as more than one polygon or require additional post-processing to connect the disjoint boundaries of the watershed. We focus on finding the connected boundary of the watershed instead of the area.

Our algorithm creates a planar graph, which we call the watershed graph, whose edges are potential watershed boundary edges and whose faces are exactly the watersheds of the pits in the TIN. The TIN model defines the edges of the watershed graph while the steepest-descent flow model dictates how these edges are connected to one another so that the graph faces correspond to watersheds. Once we have the watershed graph, any algorithm that follows the faces of a graph will extract the watershed boundaries as single polygons. The difficulty lies in creating the correct watershed graph.

Watershed Graph Edges Three features of the TIN define watershed graph edges: ridges, ridge endpoints, and saddles. The ridges of the TIN are already line segments so they are added, as edges, to the watershed graph. The remaining two features, ridge endpoints and saddles, are TIN vertices around which water may flow in different directions.

For a TIN vertex v , whether a ridge endpoint or a saddle, we add edges to the watershed graph that are paths of steepest ascent out of v . The circle of radius ϵ centered at v and placed on the surface of the TIN has a set of local maxima in the elevation of the points along the circle. Let these local maxima be at angles $\{\theta_1, \theta_2, \dots, \theta_n\}$ around v . For each angle θ_i , let π_i be the path of steepest ascent out of v that starts off in the direction θ_i and that stops as soon as it reaches either a peak, a saddle, a ridge endpoint, or a ridge edge. Add to the watershed graph two copies of each path π_i , designating the paths as *left* and *right* copies of π_i . The paths π_i account for degenerate boundaries in the watershed. The two copies allow two disjoint open sets to be connected as a single face through the space between the left and right copies.

The ridges and the paths of steepest ascent are the only edges that can be boundary edges of watersheds.

Watershed Graph Topology Although the watershed graph edges can have common endpoints, their adjacency in the watershed graph is based on the drainage characteristics of the TIN according to the steepest-descent flow model. Let e_1 and e_2 be two watershed graph edges that have a common point. There are two possibilities for the common point:

- edge e_1 is a ridge of the TIN and edge e_2 meets e_1 in its interior, or
- edges e_1 and e_2 share a common geometric endpoint that is also a TIN vertex.

In the first case, where e_1 is a ridge and e_2 meets e_1 in its interior, edges e_1 and e_2 are topologically adjacent in the watershed graph at the common point, essentially dividing edge e_1 into two graph edges. The order of the graph edges around the common point matches the geometric order of the edges on the TIN. When two edges e_2 and e_3 meet ridge e_1 at the same point, e_2 and e_3 are copies of the same path of steepest ascent; assume that e_2 is the “left” copy and e_3 is the “right” copy. The order of the edges around the common point has e_2 immediately clockwise of e_3 .

In the second case, where e_1 and e_2 share a common geometric endpoint v , we rely on the steepest-descent flow model to dictate the topological adjacencies of the edges. Edges e_1 and e_2 are topologically connected at v if and only if

- there is a sector bounded by e_1 and e_2 that has no other watershed graph edges in it,
- the land in the sector between e_1 and e_2 drains away from v , and
- point v does not drain into the sector.

These adjacency rules define the face closures in a small neighborhood around v (sectors a and d in figure 4). Every sector around v that has a point lower than v is closed at v except for the sector that contains the direction in which v drains. Every other sector, namely sectors b, c, and f that are higher than v , are open at v and are in a common face of the watershed graph with the drain of v .

When edges e_1 and e_2 are the left and right copies respectively of a path of steepest ascent π , the adjacency definition needs some clarification. First, the “land” between the two edges is the path of steepest ascent π that defines e_1 and e_2 . We imagine that e_1 and e_2 are separated by an ϵ -width corridor. Second, if v is the lower endpoint of e_1 and e_2 then e_1 is counterclockwise of e_2 , otherwise v is the upper endpoint and e_1 is clockwise of e_2 . Third, if the path π has v as its upper endpoint and, according to the steepest-descent flow model, v drains along π then v drains between e_1 and e_2 . Sectors b and f of figure 4 are bounded by steepest ascent paths with v at the lower end while sectors e and d contain dotted steepest ascent paths with v at the upper end.

The adjacency conditions for copies of a path of steepest ascent capture watersheds with unconnected interiors. When a path of steepest ascent π out of TIN vertex u is stopped by a TIN saddle v and the path approaches v along its drain direction then the water that accumulates at point v drains out of u , regardless of where the water to either side of π drains. Figure 5 shows a TIN where the ϵ -width corridor joins two unconnected watershed interiors as one face of the graph.

Although only local information on the TIN defines the topology of the watershed graph, the graph expresses some global properties:

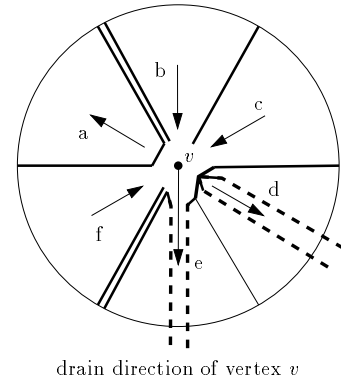


Figure 4: watershed graph at saddle v

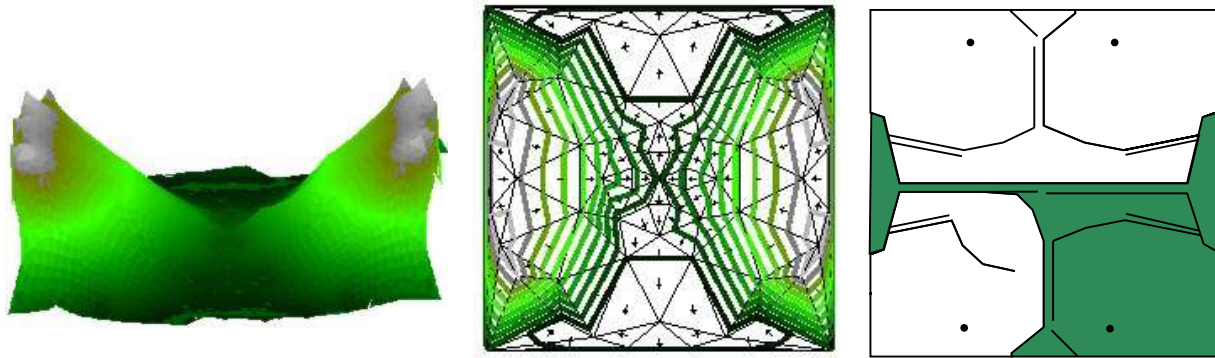


Figure 5: A terrain with an “unconnected” watershed. Two catchment basins in the mountains drain into a pit at the lower right; one through an ε corridor

- The watershed graph has a planar embedding.
- Every face of the watershed graph contains exactly one pit of the TIN.
- A point on the TIN and the pit to which it drains are in the same face of the watershed graph.
- With a boundary of ridges around the TIN, the watershed graph is connected.
- The faces of the watershed graph are consistent with both the TIN representation of the terrain and the steepest-descent flow model.

Watersheds of Arbitrary Points While the previous paragraphs focus on the watersheds of pits in the terrain, we also want to identify the watersheds of rivers. If we want the watershed of a river r and r drains to a pit p then the watershed of r is contained in the watershed of p . Consequently, to obtain the watershed of r , we subdivide the face of the watershed graph that corresponds to p starting at the most-downstream point of r (figure 6).

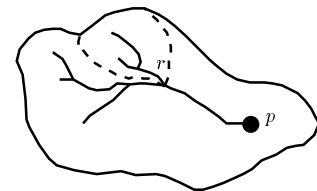


Figure 6: r 's watershed in p 's

5 SAMPLE WATERSHEDS

We have implemented our watershed algorithm inside the Cause & Effect decision management system by Facet Decision Systems of Vancouver. Figure 7 shows the watersheds that our algorithm detects in the mountains north of Vancouver. These outlines can be compared to the colored watersheds from the Watershed Atlas of British Columbia shown in the same figure.

The TIN for Vancouver covers a 60 km by 35 km area and uses 30 500 points of the original 677 000 data points. The original data has 1 meter accuracy in the xy -plane and 5 meter accuracy in the elevation and was converted into a TIN as described in Section 3. In this figure, the TIN has a 20 meter error tolerance relative to the original data. The watershed graph for this terrain has 46 500 edges and a total of 88 000 points; the longest edge has only 18 points. The small size

of the graph edges in practice is encouraging since de Berg et al. indicate that the edge complexity can, in the worst case, be quadratic in the number of TIN vertices (de Berg et al. 1996)

Each face of the graph has an internal structure that is partially shown in figure 7. The internal structure encodes a description of how water reaches the pit of a each face from any point in the face. With the internal structure, we can answer queries about the watershed of a single point in the terrain without recomputing the entire watershed graph. If we want only the watersheds of pits then the internal structure can be eliminated by post-processing the watershed graph.

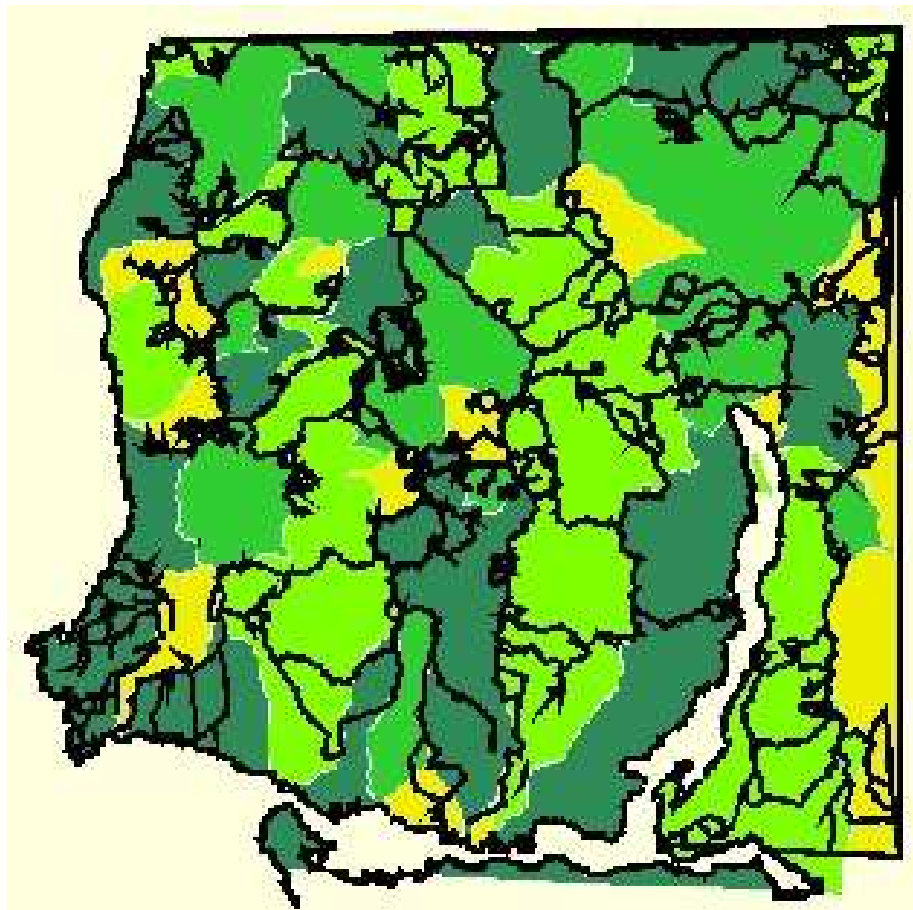


Figure 7: Computed watershed boundaries (lines) and watersheds from the Watershed Atlas of British Columbia for the mountains north of Vancouver.

Although our goal is to eliminate inconsistencies in the watershed boundaries, the edges of figure 7 have some obvious differences with the watershed boundaries as reported by the Watershed Atlas of British Columbia. First, there are many more watersheds in the interior of the mountains than reported by the Watershed Atlas. Remember that our algorithm finds the watersheds of the pits in the TIN. In some cases, the TIN has pits in the interior of the mountains caused by the error in the TIN relative to the terrain; our algorithm finds watersheds for these pits. Smoothing the terrain, reducing the error tolerance of the TIN, and grouping the watersheds of pits that lie along a common river would improve the correspondence between the watersheds. In other cases, edges of the watershed graph converge to a point and seem to close off a face, but the edges

are not topologically connected so two faces in figure 7 may be one face in the graph. These conditions represent places where part of the watershed is funneled into a narrow pass. Second, the computed watersheds contain lines that extend from the watershed boundaries into the watershed interiors. These lines capture the flow patterns inside the watersheds and are necessary for finding the watersheds of arbitrary points; many of these flow pattern lines have been omitted from the figure. Third, the computed watershed boundaries of figure 7 do not always extend to the coast. In the TIN, all the water along the coast drains to the ocean so our algorithm classifies that land as one watershed. This can be improved by adding the watershed boundaries of all points where a river meets a coast. Finally, some edges from the Watershed Atlas are missing, most notably to the left of the center of the figure. These edges do not appear in the watershed graph because the error in the TIN relative to the actual terrain has created an artificial pass between two mountains that, in accordance with the steepest-descent flow model, lets part of the watershed drain through the artificial pass.

As a final example, figure 8 shows the contour lines for a Big Beef Creek in Washington State, USA. The terrain has the Hood canal in the upper left corner of the picture, Big Beef Creek flowing from the center of the figure to the the upper right corner, and two mountains in the lower right corner. The river starts in relatively flat marsh and drains towards the Hood canal; the terrain near the mouth of the river is steeper than near the marsh. The TIN from which the contours were derived has a 12 meter error tolerance relative to the original data. Since the canyon is narrow, the river was embedded in the TIN as a breakline to better define the canyon. The TIN has 5 400 of the original 16 900 elevation points. As with the Vancouver data, the TIN was not preprocessed to eliminate or orient flat triangles or horizontal edges.

The figure also shows the faces of the watershed graph that are generated for the terrain. Since the river has relatively flat sections and was embedded in the terrain, the TIN has many pits along the river. Still, the watershed polygons track the course and the branches of the river.

The watershed graph has 8 816 edges with a total of 17 900 points. Since this TIN has a lower error tolerance than the Vancouver TIN, we expect it to have more points per path of steepest ascent: the longest graph edge has 32 vertices.

6 CONCLUSION

We combine two basic models to identify the watersheds of pits on a terrain: a TIN representation of the terrain, and a steepest-descent flow model for surface water. While these models have been combined before to identify watersheds, they admit degenerate polygons as watersheds, such as watersheds whose interiors are unconnected.

We described an algorithm for identifying watersheds that is provably consistent with both models. The algorithm identifies the watersheds of pits on the terrain, but can also identify the watersheds of arbitrary points on the terrain. We implemented our algorithm inside the Cause & Effect decision support system with the straight-forward approach used to describe the watershed graph: identify the graph edges, sort the edges around the common points, and then construct the graph topology around each common point.

As part of this work, we characterized watersheds under the two models (TINs and steepest-descent flow). The watersheds that are consistent with the models are not necessarily the watersheds that we want. Further work could use a more sophisticated model for steepest-descent flow that overflows minor pits in the terrain or modifies the TIN so that the valleys in the TIN more closely match the river network of the terrain. The watershed extraction algorithm can also be changed

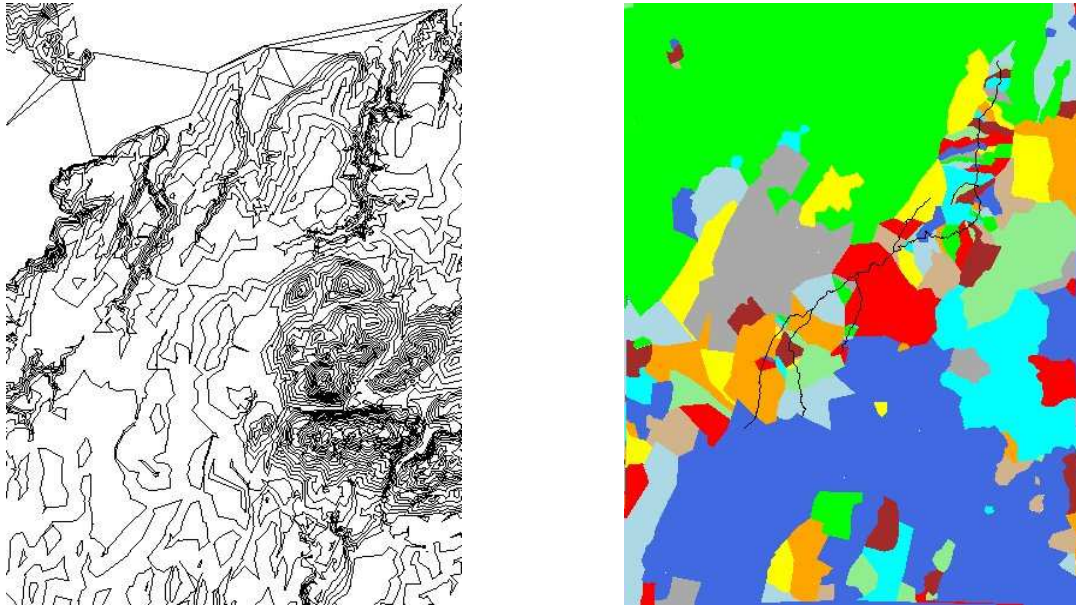


Figure 8: Contours and derived watersheds of Big Beef Creek.

to take better advantage of the hierarchical structure of river networks when tracing watershed boundaries.

7 ACKNOWLEDGEMENTS

We especially thank Facet Decision Systems for many discussions, data, and support. Thanks also to David Skea at MELP, BC. This research was partially supported by NSERC and the IRIS NCE.

References

- Collins, S. (1975). Terrain parameters directly from a digital terrain model. *Canadian Surveyor* 29(5), 507–518.
- de Berg, M., P. Bose, K. Dobrint, M. van Kreveld, M. Overmars, M. de Groot, T. Roos, J. Snoeyink, and S. Yu (1996). The complexity of rivers in triangulated terrains. In *Proc. 8th Canad. Conf. Comput. Geom.*, pp. 325–330.
- Douglas, D. H. (1986). Experiments to locate ridges and channels to create a new type of digital elevation model. *The Canadian Surveyor* 41(3), 373–406.
- Environmental Modeling Research Laboratory, Brigham Young University (1998). WMS. <http://www.ecgl.byu.edu/wms.htm>.
- Frank, A. U., B. Palmer, and V. B. Robinson (1986). Formal methods for the accurate definition of some fundamental terms in physical geography. In *Proc. 2nd Intl. Symp. Spatial Data Handling*, pp. 585–599.
- Garland, M. and P. S. Heckbert (1995). Fast polygonal approximation of terrains and height fields. Technical report, CS Dept., Carnegie Mellon U. CMU-CS-95-181, URL=<http://www.cs.cmu.edu/garland/scape>.
- Hutchinson, M. F. (1988). Calculation of hydrologically sound digital elevation models. In *Proc. 3rd Intl. Symp. Spatial Data Handling*, pp. 117–133.

- Jensen, S. (1985). Automated derivation of hydrologic basin characteristics from digital elevation model data. In *AutoCarto 7*, pp. 301–310.
- Jenson, S. K. and J. O. Domingue (1988, November). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing* 54(11), 1593–1600.
- Jones, N. L., S. G. Wright, and D. R. Maidment (1990, October). Watershed delineation with triangle-based terrain models. *Journal of Hydraulic Engineering* 116(10), 1232–1251.
- Koenderink, J. J. and A. J. van Doorn (1993). Local features of smooth shapes: Ridges and courses. In *Geometric Methods in Computer Vision II*, Volume 2031, pp. 2–13. SPIE.
- Kweon, I. S. and T. Kanade (1994, March). Extracting topological terrain features from elevation maps. *Comp. Vis. Graph. Image Proc.* 59(2), 171–182.
- Mark, D. M. (1978). Topological properties of geographic surfaces: Applications in computer cartography. In G. Dutton (Ed.), *First Int'l Adv. Study Symp. on Topological Data Structures for Geo. Info. Sys.*, Volume 5. Harvard.
- Mark, D. M. (1984). Automated detection of drainage networks from digital elevation models. *Cartographica* 21, 168–178.
- Ministry of Environment, Lands, and Parks, Province of British Columbia (1992, January). British Columbia specifications and guidelines for geomatics, content series, volume 3, digital baseline mapping at 1:20 000. Release 2.0.
- Morris, D. G. and R. W. Flavin (1990). A digital terrain model for hydrology. In *Proc. 4th Intl. Symp. Spatial Data Handling*, pp. 250–262.
- Nelson, E. J., N. L. Jones, and A. W. Miller (1994, March). Algorithm for precise drainage-basin delineation. *Journal of Hydraulic Engineering* 120(3), 298–312.
- O'Callaghan, J. F. and D. M. Mark (1984). The extraction of drainage networks from digital elevation data. *Comp. Vis. Graph. Image Proc.* 28, 323–344.
- Palacios-Velez, O. L. and B. Cuevas-Renaud (1986). Automated river-course, ridge and basin delineation from digital elevation data. *Journal of Hydrology* 86, 299–314.
- Peucker, T. K. and N. Chrisman (1975). Cartographic data structures. *Amer. Cartog.* 2(1), 55–69.
- Peucker, T. K. and D. H. Douglas (1975). Detection of surface specific points by local parallel processing of discrete terrain elevation data. *Computer Graphics and Image Processing* 4, 375–387.
- Seemuller, W. W. (1989). The extraction of ordered vector drainage networks from elevation data. *Comp. Vis. Graph. Image Proc.* 47, 45–58.
- Silfer, A. T., G. J. Kinn, and J. M. Hassett (1987). A geographic information system utilizing the triangulated irregular network as a basis for hydrologic modeling. In *Proc. Auto-Carto 8*, pp. 129–136.
- Skea, D., P. Friesen, J. Carr, A. Gallagher, I. Barrodale, E. Davies, F. Milanzzo, B. Corrie, and B. Nouredin (1998). GeoData BC—TRIM watershed atlas. http://ssbux2.env.gov.bc.ca/srmb/twa_home.htm.
- Takahashi, S., T. Ikeda, Y. Shinagawa, T. L. Kunii, and M. Ueda (1995). Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. In F. Post and M. Göbel (Eds.), *Eurographics '95*, Volume 14, pp. C–181–C–192. Blackwell Publishers.
- Tang, L. (1992). Automatic extraction of specific geomorphological elements from contours. In *Proc. 5th Intl. Symp. Spatial Data Handling*, pp. 554–566.
- Theobald, D. M. and M. F. Goodchild (1990). Artifacts of TIN-based surface flow modeling. In *Proc. GIS/LIS'90*, pp. 955–964.
- Yu, S., M. van Kreveld, and J. Snoeyink (1996). Drainage queries in TINs: From local to global and back again. In *Proc. 7th Intl. Symp. Spatial Data Handling*, pp. 13A.1–13A.14. Intl. Geographical Union.