TERRAIN DRAINAGE FEATURES AND QUERIES

by

Sidi Yu

B.S., Lock Haven University of Pennsylvania, 1993

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
Master of Science
in
THE FACULTY OF GRADUATE STUDIES
(Department of Computer Science)

We accept this thesis as conforming
to the required standard

The University of British Columbia
June 1996
© Sidi Yu, 1996
In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Computer Science
The University of British Columbia
2366 Main Mall
Vancouver, Canada
V6T 1Z4

Date: ____ Sept 5, 1996
Abstract

Terrain drainage characteristics are of interest to a number of fields such as hydrology, hydraulic engineering, natural resources management, flood control, environmental sciences, geographic information systems, etc. The traditional way of obtaining terrain drainage characteristics information is to have human operators visually identify and delineate terrain features of interest from aerial photography or contour maps. The disadvantages of this mode of operation are obviously the tediousness, the inefficiency of the process, and the inaccuracy of the information thus obtained.

Over the past twenty years, considerable research effort has been devoted to automatic generation of terrain drainage information using digital computers. Significant progress has been observed in this field. Yet most of the research done in this area does not share common definitions of terms. Some publications have attempted to address this problem by proposing general definitions that are not tied to a particular data representation or algorithm. We perceive this as the direction to follow for all future research in this area.

This thesis sets out to further develop the general definitions of terrain drainage characteristics. Particular terrain features such as pits, peaks, water-
courses, ridges are defined, and methods for identifying them are presented. Suggestions for data structures and algorithms are also given. The methods proposed not only solve the standard problems, but answer additional interesting queries as well. Proofs of computational complexity are given for the more important algorithms and problems.
Contents

Abstract ii

Contents iv

List of Tables viii

List of Figures ix

Acknowledgements xi

Dedication xiii

1 Introduction 1

1.1 Models and Approaches ............................................ 2

1.2 What Problems Can Be Addressed Using Spatial Analysis ...... 3

1.3 Digital Terrain Representations .................................... 4

1.3.1 Arguments for TINs .............................................. 6

1.4 Characteristics of Our Approach ................................... 8

1.4.1 Implicit vs. Explicit Surface ............................... 8

1.4.2 Global vs. Local Processing ................................. 9

1.5 How Real is the Model ............................................. 10
5.2 Local Terrain Features .............................................. 34
5.3 Global Terrain Features ........................................... 35
   5.3.1 Processing Complexity Is the Same as Output Complexity .. 35
   5.3.2 Upper Bound on Drainage Network Complexity ............... 36
   5.3.3 A Worst-case Example ........................................ 38
   5.3.4 Average-case Complexity .................................... 39
   5.3.5 Terrain Planar Graph ......................................... 40

6 Drainage Queries .................................................... 41
   6.1 Simple Queries .................................................. 41
      6.1.1 Finding Basin Boundaries ................................ 41
   6.2 Finer Partition .................................................. 42
   6.3 Querying the Watershed of a Point ............................ 42
   6.4 Querying the Size of the Watershed of a Point ............... 44
      6.4.1 Definitions .................................................. 44
      6.4.2 Lemma and Corollaries .................................... 45

7 Conclusions ........................................................ 48
   7.1 Level of Abstraction ............................................ 48
      7.1.1 Advantages .................................................. 48
      7.1.2 Disadvantages .............................................. 49
   7.2 Reflection of Reality ............................................ 50
   7.3 Future Research ................................................ 50
      7.3.1 Making the Model More Realistic ......................... 51
      7.3.2 Data Structures ............................................ 51
      7.3.3 Handling Special Cases .................................... 51
7.3.4 Adding Other Factors ........................................ 52
7.3.5 Merging Data Sources ........................................ 52

Bibliography ....................................................... 53

Appendix A Terrain Function .................................... 58

Appendix B Horizontal Slicing Plane Method ............... 60
List of Tables

5.1 Observed Drainage Network Complexity 40
List of Figures

3.1 Drainage Forest .............................................. 20
3.2 Three Types of General Separating Lines within a Basin .... 21
3.3 A Non-pass Point ............................................. 22
3.4 Terrain Graph ................................................. 25

4.1 Terrain Feature Around a Vertex ............................ 27
4.2 Drainage and Non-drainage Edges ........................ 28
4.3 Local Channel .................................................. 28
4.4 Three Types of Edges ......................................... 29
4.5 Two Types of Watercourse Merge Points .................... 30
4.6 Dividing Line Coincides with Watercourse ................. 30
4.7 Breaking Down of Dualities .................................. 32

5.1 Repeating a Triangle Repeats the Subsequent Sequence .... 37
5.2 Worst Case Example for Drainage Network Complexity: the Pyramid 38

6.1 New Trace-up Lines ............................................ 43
6.2 Query Watershed of a Point .................................. 43
6.3 Strips ............................................................. 45
6.4 Sweeping Trace-up Line ........................................ 46
6.5 Interpolation Points ............................................. 47
A.1 Sinusoidal Directional Derivative ............................. 59
B.1 Horizontal Plane of a Pit ....................................... 61
Acknowledgements

I would like to express my special thanks to my thesis supervisor Dr. Jack Snoeyink. I am very grateful for the fruitful discussions with him, the constructive guidance from him, and just the fun of working with him. I can always get my drafts back, marked with lots of comments, from him sooner than I dare to hope. He has offered materials directly to this thesis: Theorem 1, its proof, and the worst case example; the formal presentation of the concept “strip” in querying sizes of watersheds of drainage network points in Chapter 6; proof of terrain partition; Figure 5.1, Figure 5.2, Figure 6.3, Table 5.1, etc.

Many thanks to Dr. Dave Kirkpatrick for squeezing time to be the second reader of this thesis.

I would also like to thank Dr. Kelly Booth for giving me much valuable advice and guidance in the Virtual Hand Project.

Many friends and mentors have made my pursuit easier and possible. Here is my salute to them all for their generous help and support.

SIDI YU

xi
This thesis is dedicated to my mom, who's the best.
Where does water flow on a terrain? If we know the underlying dry surface of the terrain, are we able to know where the streams and rivers lie?

Terrain drainage characteristics include watercourse network, ridges, basins, lakes, peaks, hills, valleys, pits, watersheds, river volume, basin outlet (lowest point on the basin boundary), etc. They provide important information on water resources, possible flood areas, erosion and other natural processes. This information can be used in a number of fields such as hydrology, hydraulic engineering, natural resource management, etc.
1.1 Models and Approaches

Let’s revisit our opening question: where does water flow on a terrain? The answer, in the physical world, depends on a variety of factors: gravity, viscosity of water, sediment concentration, undersurface water, permeability of the soil, vegetation, physical laws such as dynamics, and, of course, the terrain surface geometry.

Thus, to answer the question “correctly” requires an all encompassing model of the entire physical process. First of all, that model has yet to be proposed; secondly, with the computational resources we have, we can only settle for a less ambitious approach, namely to ignore some of the factors, or to make assumptions about them and proceed with what we believe important. We have to choose the right kind and level of abstraction for the problem.

Besides the mathematical model we use, we need to decide on the means of processing. Traditionally, human operators either visually inspect aerial photography to decide where rivers, ridges and basin boundaries lie; or they have a contour map at hand, and estimate terrain features from contour information; or they combine the two pieces of information to get better results. Manual quantification of terrain drainage characteristics is tedious and time consuming, and inevitably subject to inaccuracies.

A lot of research has been dedicated to automatic generation of terrain drainage information using computers. Most algorithms use terrain surface geometry as the main data source in their processing. This is due to a number of factors:

1. Availability of data: many government geographic survey agencies, such as USGS (The United States Geological Survey), have been surveying terrain elevations at densely placed sampling locations (for instance, on a grid with 30-
meter inter-grid spacing). Such information has become available in the public domain in large volumes. Other types of information, such as vegetation, soil absorption, are harder to obtain.

2. *Importance of geometric data:* among all factors that determine terrain drainage, surface geometry is probably the most important.

3. *Computational cost:* a model that incorporates other factors, such as erosion, can be expensive computationally. The model itself will be complex, and the computational cost will be prohibitive.

Considering all the above factors, we will stick with the geometric approach in this thesis. Our raw data will be the widely available digital elevation data. We call our approach spatial analysis of digital representations of topographic surfaces.

By “spatial analysis” we mean the analysis is based solely on the geometric information contained in the digital terrain representations. One of the goals of this thesis is to outline certain sound choices in carrying out such spatial analysis which, we believe, help generate more consistent and accurate determination of drainage features.

1.2 What Problems Can Be Addressed Using Spatial Analysis

Without formally defining drainage network features (they are defined in later chapters), we claim that the following questions can be answered by spatial analysis:

- *Standard Problems:* We can identify the drainage network (network of watercourses), network of ridges, pits, peaks, passes (saddle points). We can delin-
eate basins and hills (the duals of basins). With formal definitions, we can obtain a consistent graph representation of the overall structure of drainage features.

- **Non-standard Problems**: With a richer representation, an additional set of problems can be solved correctly and efficiently. The watershed of a terrain surface point is the set of terrain surface points that contribute to the flow of that point. Watersheds of watercourses completely partition the terrain. By precomputing watersheds of certain critical points on the terrain, we can answer many queries efficiently, such as: what is the watershed of any point on the terrain; what is the watershed of a finite area; how much is the flux into a point on the terrain (or the size of that points watershed); where is the outlet of a basin, what is the volume of a basin, thus how much flood will come out from the basin if we know the level of precipitation.

### 1.3 Digital Terrain Representations

To perform spatial analysis correctly and accurately, it is essential to use an appropriate digital terrain representation, because the representation's properties have a bearing on what kind of information we can get and how structured the information is. Thus to describe the characteristics of our approach, we need to discuss the different forms of digital terrain representations.

"Digital" here refers mainly to the fact that the sample points on the terrain surface are discrete as opposed to continuous. Digital terrain representations contain only surface points.

There are three main forms of digital terrain representations. In these representa-
tions, each surface sample point has an $x$-$y$ location and an elevation $z$.

- **digital contour**: represents the surface as a set of contours. Like traditional contour maps, each contour is of a particular elevation. There are sample points along the contour in the form of a sequence of $x$-$y$ locations. There is an implied adjacency relation between neighboring $x$-$y$ locations.

- **DEM** (digital elevation model or regular grid): the sample points are regularly spaced in their $x$-$y$ locations, like a regular grid. When storing them in a computer, the memory adjacency relations can be used to imply the spatial adjacency relations and orientation relations. Thus, we only need to explicitly store the elevations.

- **TIN** (triangulated irregular network): uses a triangulation to represent the terrain surface. The $x$-$y$ locations of the sample points are irregularly spaced. A triangulation of the $x$-$y$ locations is performed, and that triangulation defines the adjacency relations between points. The most popular scheme for the triangulation is Delaunay Triangulation. Delaunay Triangulation maximizes the smallest angle of the triangles, and thus avoids long and skinny triangles. The resulting triangulation with triangles all having finite and local extent gives better approximations to the actual terrain surface. The triangulation surface with all the points elevated to their original heights becomes the TIN surface, expressed as a function $z = f(x, y)$. It is continuous, but not differentiable at the edges and vertices of the triangulation. To store a TIN, we need to store the $x$-$y$ locations, elevations and adjacency relations.

Due to processing efficiency and low complexity, the latter two representations are more popular. Virtually all current GIS systems support DEMs; more and more are
supporting TINs and the conversion between the two.

A critical study by M.P. Kumler [19] compares the relative strengths of the two representations. The author maintains that for the selected representative terrains in the United States, DEMs are more accurate, given the same amount of storage. TINs, because of its need for storing x-y locations and adjacency relations, are less space efficient. In general, for the same amount of storage, a DEM stores 10 times as many points as a TIN. The study uses a particular set of procedures in acquiring TIN vertices (there are many approaches to acquiring TIN vertices), thus the author contends that the study is not a conclusive statement on the relative merit of the two representations. The author also indicates further study is necessary to determine which of the two gives better estimates on the derivative statistics such as slope and aspect. These derivatives are important in the context of determining terrain drainage.

1.3.1 Arguments for TINs

- **Data Acquisition** From data acquisition point of view, since raw data generally are not collected at regular grid points, the grid point elevations in DEMs must be obtained by some kind of interpolation scheme. In TINs, the irregularly acquired points can be used as vertices directly. Furthermore, locations of certain important points such as pits, peaks and saddle points on the terrain can be represented exactly in TINs, but not in DEMs.

- **Storage. Efficiency** Traditionally, an argument for TINs is that they adapt to the surface complexity better than DEMs: for instance, at flat areas, few triangles are needed in TINs, while, because of the uniformity, a large number of grid points are used in DEMs (the traditional fixed grid DEMs) anyway. By the
same token, a rugged area, an area of high relief can be represented accurately by a large number of TIN vertices, which is not possible for DEMs. This is referred as variable resolution of the TINs. Nevertheless, the disadvantages of TINs lie mainly in the inefficiency of storage (of the x-y coordinates and the adjacency relations), and the consequent processing inefficiency (for example, having to calculate orientation between triangles in certain queries).

- Richness of Information Kumler’s paper[19] compares the two representations mainly by the criterion of efficiency defined as the elevation accuracy of the representation per storage space used. By accuracy he means the absolute deviations of the model surface and the actual measured elevations at many test points on the studied terrains.

- Simplification Possibilities For DEMs, simplification entails adjusting the uniform grid size and resampling with some interpolation scheme. There is little we can do to exploit the local characteristics of the terrain in that process. For TINs, however, since we can control the x-y locations of the vertices, we can sparsely sample terrain regions where little information is lost in so doing, and densely sample regions where many vertices are necessary to represent the complexity of the surface. In essence, simplification amounts to choosing another set of parameters (in this case, vertices) to represent the surface. This is the classic resampling problem. TINs’ superiority over DEMs in this respect arises from the non-uniformity of vertex selection, which affords them a more flexible, powerful and localized resampling scheme.

The comparison between DEMs and TINs is similar to that between raster and vector representations in image processing and graphics. DEMs are simple yet at
times imprecise; TINs can be complex yet exact.

1.4 Characteristics of Our Approach

Our approach to the drainage problems has a number of characteristics.

1.4.1 Implicit vs. Explicit Surface

We call our approach an *explicit surface* approach, or defined surface approach as opposed to *implicit surface* or undefined surface approach. We use an explicitly defined continuous surface (the TIN triangulation surface) on which water flows. Many proposed methods [14, 33, 29, 7, 41] do not have an underlying surface, and that makes them mathematically unsound. For example, a number of methods based on DEMs treat each DEM grid point as a cell, and do not have the notion of a surface. In our method, with an explicit surface, many terrain features and processes become well defined and easily extracted.

In a large number of previous approaches, a terrain feature is simply defined as the output of a particular algorithm. This makes it impossible for different researchers to reach the same result, and there is no objective measure of correctness. Several researchers have realized this difficulty, and tried to remedy it[10, 17, 46]. We follow the suggestions made by Frank et al. [10] that formal definitions should be used to define terrain-specific features so that properties of the structure can be established mathematically and contradictory definitions (or at least those in disagreement with other related research) can be avoided.

Our definitions apply to any piecewise planar surface. This includes TIN surfaces and DEM surfaces if a piecewise planar interpolated surface is specified. This mathematical approach provides a common ground for discussions of terrain features.
Furthermore, by following a mathematical approach, we are able to generate consistent global structures for the terrain.

1.4.2 Global vs. Local Processing

Many approaches, particularly the ones based on DEMs, generate drainage information by performing only local operations. They are similar to raster image processing, where you apply local operators on an image and then use line-connection or region-growing algorithms to generate global structures. The hope is that global structures will manifest themselves locally everywhere they lie, and turn into local structures, so they can be detected locally.

Unfortunately, many global drainage features do not always reveal themselves as local structures; furthermore, the same type of local features may belong to different global structures. Thus, the local approach often results in broken or ambiguous structures.

We prefer to generate global structures, such as watercourse network, basin boundaries and watersheds, by performing global operations, and hope that the global operations are not too expensive computationally.

Fortunately, with our definitions, it turns out that the operations to generate global structures do no extra work, because linear-complexity local operations give us all and only the necessary starts for global structures, and the tracing of the global structures is a linear operation to their own size. In natural terrains, these global features typically have linear complexity. This fact makes our approach both "correct" and efficient.
1.5 How Real is the Model

Let's return to our discussion of level of abstraction and representativeness of the model. Our approach assumes that water flows on the terrain surface; certain water bodies, such as streams and rivers, have no depths or widths, thus they are one-dimensional mathematical objects; streams merge, but never bifurcate. There are three assumptions in this model that are at variance with the physical world:

1. The sample points we get from the raw data are not on the dry surface underneath water bodies. If the sample points are on a river or a lake, they are sampled on the surface of these water bodies. Thus to say the given surface is the dry surface water runs on is not accurate. But this discrepancy is largely due to the limitation of our data source, and we have to live with it.

2. Real water bodies have widths and depths, so they are three-dimensional rather than one-dimensional. Our model currently treats rivers as one dimensional objects. A 3D model will be much more costly computationally, and hard to define. And the benefit of this complexity may not be relevant in the context of huge amount of data, which is the usual case for our purposes.

3. Streams and rivers, as we know them, in the real world, do bifurcate. We do not allow that in our model, because it will turn the river network from rooted forest, in graph theoretic sense, into network. That will make the structure and the algorithm more complex, and could potentially make the computational cost too high. Furthermore, such features are more prone to inaccuracies and artifacts of the input data, thus may be better left out, at least for the scope of this thesis.
Artifacts in the data can cause problems in processing. For example, many small pits in the data are the result of inaccurate measurement of elevations. Given the precision of the survey data, such artifacts are inevitable. Our model assumes that the source data have already been put through some filtering operation, and most artifacts are removed. We will have more discussion on this in the next chapter.
Chapter 2

Previous Research

We divide previous works in extracting terrain drainage features into two categories based on the digital representations they use, namely DEMs and TINs.

2.1 In DEMs

Methods based on DEMs do not have an explicit surface that water runs on, so typical models of how water flows bear little resemblance to the physical process. For example, an early approach proposed by Peucker [33] has no concept of surface. Many approaches [14, 15, 33, 29, 7, 36, 20, 41] based on DEMs extract features use techniques similar to those in raster image processing: local operators and region growing, etc. Typically, local operators are used to detect possible pits, peaks, channel cells (cells that have the potential to be channels), etc., then the channel cells are linked according to some heuristics to form the drainage network. Douglas [7] describes several methods for detecting channel (ridge) cells. Basin boundaries are identified in a similar fashion[14].

As an example, a channel is usually detected with a local operator similar to a
line detector in image processing. The idea is if a grid point and its 8 neighbor grid points form a V shaped local surface, we say the grid point is on a channel. One then attempts to use some heuristic line-connection algorithm to connect these channel grid points into a linked channel line.

As with image processing, there are inherent ambiguities with this raster approach, which emphasizes local characteristics. The global nature of the terrain drainage is not adequately reflected, and consistent global structures are hard to obtain. Different heuristics result in different sets of extracted terrain features.

Mark [22] proposed a drainage network detection method that does account for the global nature of drainage. Each grid point represents a cell, and the flow behavior of the grid point represents the flow behavior of the entire cell. The water on a cell goes to one of the eight neighbors: the one that forms the steepest down slope with the current grid point. This last assumption is not so unreasonable per se, it is similar to our assumption of no bifurcation, which keeps the complexity tractable.

But the combination of the these assumptions is unrealistic.

The algorithm goes as follows: sort all the cells in order of decreasing elevations. Each cell has water quantity one. For each cell in that order, decide to which of its eight neighboring cells the water will flow (the one that forms the steepest down slope with the current cell). Add the current cell’s water quantity to that cell. In the end, cells with water quantity over certain level is claimed to be on the water course. These cells are linked naturally to form the drainage network.

This method takes into account the fact that drainage process has a global nature. Because of that, it is more costly: the sorting process is of order $\Theta(n \log n)$, which is higher than the local methods’ $O(n)$, where $n$ is the number of grid points.

These two types of DEM methods have the same complexity during their second
phase of processing: the line-connection phase. The complexity for the second phase is proportional to the size of the drainage network. In our approach the complexity is proportional to the resulting drainage network too. The difference lies in the fact that in these approaches, the drainage network complexity is bounded by the number of grid points, because they do not introduce new points. But in our approach, we introduce new points, thus the complexity can be potentially high. Fortunately, as we will see later, in natural terrains, the complexity is more likely to be linear or sub-linear in the input size.

Takahashi, et al. [41] proposed a method that first extracts critical points (pits, peaks and passes) that satisfy Euler’s relation, then traces channels and ridges from those critical points. This method takes advantage of the fact that there is a global structure in drainage features.

2.2 In TINs

There are fewer papers dealing with extracting terrain drainage features from TINs [30, 16]. In TINs, we do have an explicit surface. In [30], what later we define as local channels are extracted, then joined to form the drainage network, although how to do the joining is not explained. To partition the terrain into tributaries of channel segments, the terrain is subdivided into cells (not dissimilar to Mark’s grid cells), all the water in a single cell flows into a single watercourse.

In this paper, we will outline explicitly how to do the joining, and discard the notion of cells, because in reality, water on different parts of a cell can flow into different watercourses.

In [16], the drainage network is formed by tracing up local channels, which, as we will see in detail later, can not account for all the watercourses, because watercourses
include not only local channels. Tracing up local channels doesn’t work when local channels do not connect. In the paper, when delineating tributaries, tracing up along the steepest accent is employed, but details are lacking.

We will show that because water flows down, its more natural and complete to trace down instead of up. Tracing down gives us the complete set of watercourses. We will also explain explicitly how the trace-downs and trace-ups are performed.

2.3 Topological Structure

In [7], a new type of digital elevation model is proposed which extracts important ("information rich") terrain features such as ridges and channels, and use them to form a "richline" representation of the terrain.

This is in the same spirit as the works on graphs of surface networks [34, 37, 41, 46]. Pfaltz [34] defines a graph structure with vertices at critical points (pits, peaks and passes) and edges at ravine and ridge lines. Takahashi et al. [37] give a different graph structure.

We are interested in the quantitative aspects of terrain drainage, yet these graphs represent more of the qualitative ones. Nevertheless they provide an economical way of representing the overall terrain structure. The construction of these graphs will be addressed in this thesis.
Chapter 3

Definitions

We have given justifications to some of the assumptions and choices we made in our approach. In this chapter, we present these assumptions and definitions in detail. These definitions are applicable to any kind of surfaces, not limited to TIN surfaces or piecewise planar surfaces. Their realization in TINs will be discussed in next chapter.

3.1 Assumptions: How the Water Flows

3.1.1 Assumptions by Other Researchers

Defining how water flows is essential to computing terrain drainage.

In raster DEMs, as proposed by Douglas [7] and Mark [22], a network of discrete point elements is used. In [22], each element represents an entire cell of the grid. Then it is decided to which of its four or eight neighbor elements the water on this element flows. Thus there are two levels of discretization: one in the use of cells, the other in limiting the possible directions water can flow. Douglas discarded the
second discretization by allowing water to flow along the steepest descent instead of to the eight neighbor elements.

To have an explicit surface for the steepest descent calculation, Douglas inserts a virtual grid point at the center of each square cornered at four neighboring grid points. The elevation of the center point is the average of the four corner points. Then four triangle surfaces each consisting of two points of an edge of the square and the center point are used in the steepest descent calculation. Nevertheless, Douglas's method still keeps the concept of cell.

3.1.2 Our Assumptions

Assume we have a continuous surface which can be modeled as a function \( z = f(x, y) \) over the \( x-y \) plane, and that:

1. Water flows on the surface.

2. At any point on the surface, the water follows the steepest descent at that point\(^1\).

3. At any point on the surface, the steepest descent is either unique or nonexistent \(^2\).

Because the steepest descent is unique, we obtain a tree structure of watercourses. The uniqueness of steepest descent forbids bifurcation. As pointed out by Poiker in Chapter 39 of the NCGIA Core Curriculum, this assumption is also the basis of the traditional definition of watershed in hydrology.

\(^1\)For the precise definition of "steepest descent", see the Appendix.

\(^2\)We leave special cases such as flat surfaces out of our present discussion. They can be accounted for by assigning a steepest descent by some criteria.
To account for some of the special cases, in our processing, we may desire that local depressions be filled to form lakes and that their outlet points be identified, so that water can pass through the depression and flow down from its outlet, and find its final destination. Smoothing operations can be performed to remove small depressions. O'Callaghan and Mark [29] reported 90% removal rate for one pass of a simple binomial smoothing operator. For larger depressions, explicit methods have to be used to fill them.

3.2 Land Forms

In this section, we give definitions of important land forms.

3.2.1 General Drainage Definitions

- **Trickle path** of a point $p$: the path that a drop of water follows starting at $p$. The destination point is either a pit, which is a local minimum, or a point on the boundary of the terrain.

- **Watershed** of a point $p$: the area formed by all the points whose trickle paths go through or end at $p$. All and only the water falling on the watershed of $p$ flows through or into $p$.

- **Basin**: the watershed of a pit.

3.2.2 Local Drainage Features

- **Pit**: local minimum of the surface.

- **Peak**: local maximum of the surface.
3.2.3 Global Drainage Features

Definition of Drainage Network (or Watercourses)

Drainage network is the network of streams and rivers in a terrain. Formally,

- **Drainage network**: the set of all points whose watersheds have nonzero projected area onto the x-y plane. In other words, any point that has a measurable amount of water flowing through it is on the drainage network.

A **watercourse** is a monotonically descending branch of the drainage network.

- **Watercourse origin**: the limit of drainage network points whose watershed sizes approach zero.

Because any point's trickle path is unique and trickle paths do not cross, the drainage network is a forest in graph theoretic sense, see Figure 3.1. Each tree in the forest is rooted at a pit. The **nodes** (vertices in graph theory) in the forest are watercourse origins (called leaves) and merge points of watercourses (called interior nodes). The portion of the watercourse connecting two nodes is a **link** (edge in graph theory). Because the interior nodes are defined as merge points, they all have degrees greater than 2 (except for possibly pits). A parent node's watershed includes the watersheds of all its descendents' watersheds.

Compliment of Watercourses

In a terrain, there are features that form the boundaries of watersheds. These are: dividing lines, crests, and watercourse trace-ups.
Dividing Lines

If an interior node $A$ on the drainage network is not a pit, then one and only one of its adjacent nodes has elevation lower than $A$ (it is the parent of $A$). Now we examine the projections (on the $x$-$y$ plane) of all the incident links of $A$. If two consecutive (counting clockwise or counterclockwise) links lead to two nodes of higher elevations, i.e. if two watercourses merge at $A$, we define a separating line between the two watercourses by tracing up along the maximum ascent between them starting from the node and ending at a peak. We call this separating line a *dividing line* (see Figure 3.2).

- **Dividing line**: a separating line between two merging watercourses. It starts at the merge point of the watercourses, traces up along the steepest ascent and ends at a peak.
A dividing line has the property that no measurable amount of water from one side flows to the other side by crossing any point on the dividing line.

Figure 3.2: Three Types of General Separating Lines within a Basin

Crests

A crest is what is generally called a ridge line between two peaks. It starts at a pass (saddle point), and ascends in two distinct directions following steepest ascents, and reaches two peaks, which need not be distinct.

- **Pass**: any non-pit point \( p \) that has at least two maximum ascents\(^3\) \( A \) and \( B \), and there is no measurable amount of water that flows from one side of \( A-B \) to the other through \( p \).

It can be proved that the latter condition is satisfied if and only if the following is false: at \( p \), there is an incoming watercourse on one side of \( A-B \), and the direction of steepest descent is on the other side of \( A-B \), see Figure 3.3.

At a pass, it's also true that there must be at least one minimum descent on each side of \( A-B \), otherwise the water falling on one or both sides of \( A-B \) at

\(^3\)For the precise definition of maximum ascent, see Appendix.
Figure 3.3: A Non-pass Point

the vicinity of \( p \) will have to flow across \( A-B \).

- **Crest**: a line consisting of two trace-ups that each originate from the same pass, ascend up along the steepest ascent, and end at a peak.

  No measurable amount of water from one side of the crest flows to the other side by crossing any point on the crest. A pass is the lowest point on a crest.

  We soon will see that crests are important division lines between basins.

**Watercourse Trace-ups**

- **Watercourse trace-up**: is a line that starts at a watercourse origin, traces up along the steepest ascent, ends at a peak.

  Watercourses trace-ups are used to connect watercourse origins to crests and eventually to peaks. They complete the partition of the terrain.

  Because dividing lines, crests, and watercourse trace-ups all have the function of separating watercourses, they can be classified into the category of *general separating line*.

- **General separating line**: a dividing line, a crest or a watercourse trace-up.
3.3 Terrain Partition

Watercourses and general separating lines form the boundaries of a complete partition of the terrain.

- **source area**: a contiguous area of the terrain surface bounded by a portion of a watercourse and a portion of a general separating line which has no portions of other watercourses or general separating lines lying inside it.

**Lemma 1** *Source areas partition the terrain. All and only the water falling on a source area flows into its bounding watercourse portion directly. Furthermore, the bounding watercourse portion descends monotonously from one end to the other.*

When we say the water flows "directly" into a watercourse, we mean the water makes its initial entry into the drainage network at that watercourse.

A pair of source areas arise when two watercourses merge: the dividing line together with each of the two watercourses bounds one source area. The upper end of the watercourse is a watercourse origin. The lower end is a watercourse merge point. Therefore, each source area is related to a leaf node and an interior node of the drainage network forest. Each leaf is related to two source areas. Each interior node is related to two or more source areas. Each source area is part of the watershed of its related interior node. This fact is essential to using partition to solve drainage problems. It's possible that a small basin can lie completely within a source area spatially, but it is not part of the source area. While using the partition to solve drainage problems, we should not make the opposite assumption.

We observe that when two dividing lines that bound the two sides of a watercourse trace up to a single peak, the boundaries for the two source areas are closed. But
when the two dividing lines end up at two distinct peaks, we do not have closed boundaries. This is where crests' role comes into play: there exists some crest that completes the boundaries. Conversely, if there is a crest, some watercourse will need it to complete its two source areas' boundaries. Therefore the work of finding crests are part of the work of finding the boundaries of source areas.

3.4 Graph Structure of Global Features

There is a planar graph structure coming out of the partition. The crests partition the terrain. A basin is bounded by crests. A crest connects two summits. Each summit is a vertex in graph theoretic sense. A crest is an edge that connects two vertices. A basin is a face in the graph.

We can obtain a dual graph of the above described graph. Each basin has a pit in it, which becomes a vertex of the dual graph; watercourses are the dual graph's edges, they connect pits; hills are faces, which are bounded by watercourses. See Figure 3.4.

Two basins are adjacent if they share a common crest. A watercourse and a crest meet at a pass.

This graph is a compact representation of the overall structure of the terrain. Douglas's "richline" representation [7] is precisely an attempt on using this structure to represent the most important features of the terrain. This graph structure arises as a result of our definitions. Generating this graph takes little extra processing after we obtain all-the watercourses and general separating lines.
Figure 3.4: Terrain Graph
Chapter 4

Extracting Terrain Features in TINs

We now apply the definitions we presented in the previous chapter to TINs. We start by outlining extraction methods for local features in TINs, then we will introduce new definitions that only apply to TINs. Most of these definitions and methods apply to any piecewise planar surface.

We also assume that there are no degenerate cases such as flat areas or horizontal edges. These are special cases that can be handled by assigning steepest descent direction on the basis of other information such as flow direction, medial axes of a lake, etc.

4.1 Local Features

Local features such as pits, peaks, passes can be easily extracted by examining vertices and their adjacent vertices, edges and faces. For instance, we can redefine the following:
- **Pit**: any vertex that has lower elevation than all its adjacent vertices.

- **Peak**: any vertex that has higher elevation than all its adjacent vertices.

In TIN, certain local features can only occur at certain types of triangulation objects (vertices, edges, faces). This can be decided by enumerating all TIN objects, and examining what features can lie on what objects. For instance, as we have seen, pits can only be on vertices. Another important observation is that according to the definition in the previous chapter, a “pass” in TIN can only lie on some vertex of the TIN. The identification of passes is a simple local test at TIN vertices.

Figure 4.1 gives some idea about what types of objects can occur around a vertex.

![Figure 4.1: Terrain Feature Around a Vertex](image)

### 4.2 Definitions Particular to TINs

We define the following terms in preparation for discussion on global feature extraction. In TIN,
• **Drainage edge** of a face $A$: any incident edge of $A$ that intersects the steepest descent line of some point on $A$. Some water on $A$ drains out into that edge. See Figure 4.2.

![Drainage and Non-drainage Edges](image)

Figure 4.2: Drainage and Non-drainage Edges

• **Non-drainage edge** of a face $A$: any incident edge of $A$ that intersects the steepest ascent line of some point on $A$.

An edge can then be classified into one of the following three types (see Figure 4.4):

1. **Local channel**: An edge $E$ that is a drainage edge of both its incident faces. Some water on both those faces drains into $E$, and flows down along $E$ to its lower vertex. Local channels collect water by themselves. See Figure 4.3\(^1\).

![Local Channel](image)

Figure 4.3: Local Channel

\(^1\)h is higher than $l$, $\overline{ma}$ and $\overline{na}$ are slope lines down to $\overline{hl}$. All water in the shaded area flows through $a$. 

28
2. **Local ridge**: An edge that is a non-drainage edge of both its incident faces. Water never drains into it. It is the dual of a local channel.

3. **Plain edge**: An edge that is a drainage edge of one incident face, and a non-drainage edge of the other.

![Diagram of local channel, local ridge, and plain edge](image)

**Figure 4.4: Three Types of Edges**

### 4.3 Global Drainage Features

Now we are ready to present algorithms for global feature extractions.

#### 4.3.1 Drainage Network in TINs

According to our definition of drainage network, in a TIN, every local channel is on the drainage network, because they collect water by themselves. For a point $p$ on a triangular face to be on the network, since $p$ does not collect water by itself, $p$ has to lie on the trickle path of some local channel.

This fact makes it easy to identify watercourses: they can start only at local channels, whose identification can be achieved by a local test. Specifically, the origin
of a watercourse is a local channel $C$ whose upper vertex $v$ has no incoming local channels or $v$ doesn't drain into $C$ (ie. $C$ is not the steepest descent for $v$).

By tracing down the steepest descents from watercourse origins, we obtain the complete drainage network.

An interior node (upon which two or more watercourses merge) of the drainage network can lie on either a vertex or a local channel in TIN (see Figure 4.5). If it's on a local channel, the dividing line that separates the two merging watercourses will coincide with one of them, but we can consider the dividing line as being shifted up along the face by an infinitesimal offset (see Figure 4.6).

---

**Figure 4.5: Two Types of Watercourse Merge Points**

**Figure 4.6: Dividing Line Coincides with Watercourse**
4.3.2 General Separating Lines in TINs

Since crests start at passes, which can be identified with local operations, we can trace up along the steepest ascent from passes to obtain all the crests.

Watercourse Trace-ups can be traced starting at watercourse origins, which can be identified with local operations.

Watercourse dividing lines start at watercourse merge points, which are a by-product of identifying drainage networks.

We can identify the starting points of these general separating lines easily. We then perform tracing-ups from these points to obtain all general separating lines.

4.4 Dubious Dualities

4.4.1 Dualities in a Terrain

There are many dual pairs of features in a terrain. For instance, pits and peaks, basins and hills, channels and ridges. A strict duality exists if by flipping the terrain up-side-down (by, say, negating all elevations), the definition we use for one feature can now identify the dual of that feature.

In reality, if we insist on maintaining strict duality, we end up with incomplete features or infinite regress[44, 48].

For instance, in TIN, if we keep the definition of drainage network, and modify the definition of general separating lines to meet the strict duality requirement, then drainage network and general separating lines will not interlock to form a complete partition of the terrain. Instead, it can be shown, some needed separating lines will be missing, and some will be superfluous for the purpose of separating catchments.

If, on the other hand, we keep our definition of general separating lines, and modify
the definition of drainage network by insisting that there must be a watercourse between two merging general separating lines (to form a dual of dividing lines, which separate two merging watercourses), we can get into troubles within even a single triangle. Say we have a pyramid triangle that has two local ridges and a local channel at the base, as in Figure 4.7. To separate the ridges, we have to introduce a watercourse, which, unfortunately inevitably, merges with the local channel. Now we need a general separating line to divide the two merging watercourses, *ad infinitum*.

![Figure 4.7: Breaking Down of Dualities](image)

If we use differentiable surfaces instead of piecewise planar ones as an attempt to preserve duality, we end up facing a bigger problem: we even may not find a watercourse at the first place. If the terrain is shaped like a big bowl, then only the bottom point of the bowl is on the drainage network. This is undesirable. The fact that we are using a piecewise planar surface actually gives our definitions meanings, and makes them useful.
Chapter 5

Data Structure and Complexity of Drainage Processing

In this chapter, we present the data structures used in and the computational complexities of the algorithms in terrain drainage information processing. The complexity is composed of several parts: complexity of building the TIN data structure; complexity of identifying local features; complexity of delineating global features, etc. If we are to handle queries, there are also the complexity of preprocessing and the complexity of queries themselves. We will address queries and their complexities in next chapter.

Let \( n \) be the number of vertices in a TIN. By Euler's relation, the number of edges is at most \( 3n \), thus \( O(n) \). The number of triangles is at most \( 2n \), thus \( O(n) \). For each vertex, there are on average less than 6 adjacent vertices.
5.1 TIN Data Structure

Traditionally, a TIN is stored as a list of triangles, or as a graph, as indicated by Poiker in Chapter 39 of NCGIA Core Curriculum. Neither of the two has an explicit representation for the spatial ordering of adjacent edges of a vertex.

The winged edge and quad edge[12] data structures, however, do explicitly represent the spatial ordering information. The operations on them are simple. There are only two primitive topology operations on quad edge data structure: “makeedge”, which initializes a single edge, and “splice”, which modifies the topology. All the complex operations can be implemented by using the two primitives. Quad edge data structure also has the additional advantage of representing the dual graph along with the original graph. For its simplicity and expressiveness, we choose to use the quad edge data structure to represent TINs.

Given the input in the form of a list of 3-D surface points, we use a Delaunay Triangulation algorithm presented by Guibas et al. [12] to triangulate the x-y locations of the points. The algorithm uses quad edge data structure. The complexity of building a TIN is the same as that of the Delaunay Triangulation algorithm, which is \( O(n \log n) \).

5.2 Local Terrain Features

To extract local terrain features (pits, peaks, passes), each vertex along with its adjacent vertices is examined. Let \( \text{degree}(v_i) \) be the degree (in graph theoretic sense) of the \( i \)th vertex \( v_i \), then the numbers of adjacent vertices, edges, and faces the extraction operation has to examine are all \( \text{degree}(v_i) \). Furthermore, the complexity of the operation on a vertex is linear in the number of vertices, edges, and faces it has.
to examine. Therefore the complexity of the operation on \( v_i \) is \( O(\text{degree}(v_i)) \). The total complexity of the operation on all the vertices is thus linear in the total degree of the graph: \( O(\sum_{i=1}^{n} \text{degree}(v_i)) \). By Euler's relation, we know the total degree of the graph is at most \( 6n \), in other words, \( \sum_{i=1}^{n} \text{degree}(v_i) = O(n) \). Therefore the complexity of the local feature extraction operation is \( O(n) \).

The lower bound of the computational complexity on local feature extraction is also \( O(n) \), therefore it is tight.

Since each type of features is a distinct subset of the original vertices, the storage complexity for them is at most linear, \( O(n) \).

5.3 Global Terrain Features

5.3.1 Processing Complexity Is the Same as Output Complexity

All global features (drainage network, dividing lines, crests and watercourse trace-ups) starting points can be locally identified at TIN vertices in \( O(n) \) time. The only exception are dividing lines which start at merge points of the drainage network. Since the drainage network is a forest, the number of merge points are linear in the number of watercourse origins, which is linear in the number of vertices \( n \). Therefore, the number of starting points of dividing lines is \( O(n) \). As we already know, finding starting points of dividing lines is just a by-product of generating drainage network.

Watercourses and general separating lines each consist of connected line segments. Each line segment lies within a triangle. Its two end points lie on two edges of that triangle. To "extract" or "traverse" a watercourse or a general separating line, we have to go through all of its consisting line segments. At each line segment, it takes constant \( O(1) \) amount of processing. Also, it takes constant amount of space.
to store a line segment. Thus, processing and storing a watercourse or a general separating line are linear in the number of its consisting segments.

Let $S_i$ be a watercourse or a general separating line, $|S_i|$ be the number of line segments in $S_i$,

**Lemma 2** Extracting a watercourse or a general separating line $S_i$ takes $\Theta(|S_i|)$. Storing it also takes $\Theta(|S_i|)$.

We thus establish the following: after identifying all the local features, the time and space complexity of delineating drainage network and general separating lines is linear in the number of line segments comprising them. In other words, the processing complexity is the same as output complexity.

We now set out to find the worst case and average case complexities of drainage networks and general separating lines, where "complexity" means the number of line segments.

### 5.3.2 Upper Bound on Drainage Network Complexity

We start with drainage network complexity. The same analysis applies to general separating lines as well.

We need to determine how many triangles (or, equivalently, triangulation edges) the drainage network can cross, noticing that watercourses are trickle paths of watercourse origins. We do this by showing that, first, there are at most $O(n)$ watercourse origins; secondly, a drop of water trickling down along the surface can cross a particular triangulation edge at most $O(n)$ times; thirdly, there are at most $O(n)$ triangulation edges, therefore the total complexity of all watercourses is $O(n^3)$. The first and the third facts are already established. We now show the second one.
Lemma 3 In a terrain represented by a TIN with n vertices, a trickle path meets any edge of the TIN at most $O(n)$ times.

Proof: Break the trickle path into n subpaths by cutting at the elevation of each vertex of the TIN; let $p$ denote one of these subpaths. We will show that $p$ hits each edge of the TIN at most twice.

Subpath $p$ visits a sequence of triangles and edges. Within each triangle, $p$’s direction is determined by the steepest descent. Thus if $p$ visits a triangle a second time, then it repeats the sequence that it visited before until it ends at the elevation of a vertex, as illustrated in Figure 5.1.

![Figure 5.1: Repeating a Triangle Repeats the Subsequent Sequence](image)

Consider now the edge in the repeating sequence that has the shallowest slope, where the slope of a segment in 3-D is the ratio of the positive difference in its endpoints’ z coordinates to the length of its projection in the x-y plane. If $p$ visits this edge first at point $p$ and lower at $q$, then we can derive a contradiction: The segment $pq$ is shorter then the portion of $p$ from $p$ to $q$, denoted as $p \rightarrow q$. $p \rightarrow q$, however is steeper than $pq$, because $p$ follows the steepest descent, thus the direction $p$ follows on each triangle must be steeper than the preceding edge (so that $p$ follows the face
of the triangle instead of the edge), which, in turn, is steeper than $\overline{pq}$ ($\overline{pq}$ being on the edge of the shallowest slope). Thus each segment on $p \rightarrow q$ is steeper than $\overline{pq}$. A longer path (in projected length) from $p$ having a steeper slope than $\overline{pq}$, however, cannot end at $q$.

Within a subpath, since the shallowest edge cannot be visited twice, and the sequence of visited triangles repeat itself, no other edge can be visited more than twice. Because there are $n$ vertices, a single edge in TIN can be cut into at most $n - 1$ subpaths, thus a single edge in TIN can be visited at most $2n - 2$, which is $O(n)$, times by a single trickle path. QED.

5.3.3 A Worst-case Example

The above stated complexity $O(n^3)$ of the drainage network can actually be achieved, as shown in Figure 5.2.

![Diagram of a pyramid showing the worst-case example for drainage network complexity.](image)

Figure 5.2: Worst Case Example for Drainage Network Complexity: the Pyramid

Along one edge of the pyramid, form a scree ("rocky slope") of $n$ triangles. On the top, carve $n$ local channels that will catch water and start $n$ rivers near the top of the scree.
Form n gently-sloping helical ducts that meet the rivers after the rivers meet the scree. Each duct begins with a nearly-flat top whose steepest descent angles slightly away from the pyramid face (preventing itself from forming a local channel with the pyramid face), so that rivers that join the duct separately remain separate. At a turn, the two top triangles of the duct share a vertex at the outside corner and a small, near-vertical triangle interpolates between them, forming separate waterfalls for separate rivers.

The resulting terrain has $O(n)$ vertices and $n$ initial rivers each cross the $n$ scree triangles $n$ times. Thus,

**Theorem 1** *In a TIN with n vertices, the worst-case complexity of a watercourse is $\theta(n^2)$, and of all watercourses is $\theta(n^3)$*

This upper bound is tight.

The worse-case complexity of storing global features is also $O(N^3)$.

Theorem 1 applies to arbitrary triangulations. We suspect Delaunay Triangulation with its empty circle property and maximization of smallest angles will give us a less expensive upper-bound.

### 5.3.4 Average-case Complexity

Analysis of average-case complexity based on randomized surface elevations gives unrealistically high numbers of pits, peaks, and watercourses. For example, great number of pits and peaks are introduced by randomization, while in nature, the numbers of pits and peaks are small relative to the number of sample points.

Table 5.1 shows experimental results we have obtained from sample data on St. Mary's Lake, B.C. We can see that the network has slightly fewer line segments than there are vertices.
Table 5.1: Observed Drainage Network Complexity

<table>
<thead>
<tr>
<th>No. Vertices</th>
<th>450</th>
<th>1800</th>
<th>4956</th>
<th>11125</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Size</td>
<td>353</td>
<td>1427</td>
<td>4134</td>
<td>10579</td>
</tr>
</tbody>
</table>

5.3.5 Terrain Planar Graph

To obtain the planar graph we described in the previous chapter, we need to perform some extra processing. The information we need can all be collected while we generate the global features, without substantially increasing the running time of the algorithm. From there, we can generate the graph itself.

The complexity is a function of the size of the graph: how many pits there are, how many crests there are, etc. Storage complexity is linear in the size of the graph. To indicate which crest belongs to an edge in the graph, we simply need a pointer to the crest.
Chapter 6

Drainage Queries

Identifying local and global drainage features is useful in its own right, furthermore, it helps answer many interesting terrain drainage queries. In this chapter, we will examine what types of queries can be answered with the help of our data structure; what preprocessing steps we can take to reduce the complexity of the queries; and what the complexities of preprocessing and the queries themselves are. Some of the queries can be answered with little extra processing beyond generating the global features. We will discuss them first.

6.1 Simple Queries

6.1.1 Finding Basin Boundaries

Theorem 2 *A basin's boundary is formed by a closed chain of crests.*

With a planar graph of the terrain, finding basin boundaries amounts to finding all the incident edges of a face in the graph. We can do this by following the "right-hand" rule: walk along the edges while keeping the right hand touching the wall.
Using quad edge data structure to store the graph makes following the “right-hand” rule even more trivial.

With known basin boundaries, we can compute the area and volume of basins. The area is defined as the projected area of the basin onto the $x$-$y$ plane. The volume is defined as the the space bounded from below by the terrain surface and from above by a plane at the level of the basin’s outlet (lowest point on the basin’s boundary).

### 6.2 Finer Partition

In order to reduce the cost of watershed related queries, we need to generate more trace-up lines in addition to our global features. We add a new type of trace-ups called “interior node trace-ups” starting at drainage forest’s interior nodes. For each interior node, we add two trace-ups; each divides an incoming watercourse and an out going watercourse of the node, see Figure 6.1. With these additions, we have obtained watershed boundaries of all drainage forest nodes. These additions along with drainage network and general separating lines form a finer partition of the terrain. We call each area bounded by these lines a **partition unit**.

### 6.3 Querying the Watershed of a Point

To find out the boundary of the watershed of a query point $p$, trace-ups have to be performed. The query operation takes several steps: first locate the triangle that contains $p$. This takes $O(\sqrt{n})$ expected time if no other assisting data structure is used along with the quad edge (we need to walk across the triangles). If other data structures are used, this may be improved to $O(\log n)$.

The second step is to perform trace-ups, see Figure 6.2. Finally, the watershed
boundary of $p$ can be delineated by using these trace-ups and the finer terrain partition boundaries.

Figure 6.1: New Trace-up Lines

Figure 6.2: Query Watershed of a Point
6.4 Querying the Size of the Watershed of a Point

If the query point is not on the drainage network, its watershed is zero in size. Otherwise, we need more preprocessing to achieve constant time query (excluding point location) using an interpolation scheme.

Storing watershed sizes of drainage merge points does not provide enough information for performing accurate interpolation, since the interpolation scheme is not well defined between merge points.

What we need is to store watershed sizes of as few selected points along the drainage network as possible, yet achieving constant time interpolation. Now we describe the criteria for selecting these points. We start by making a few more definitions.

6.4.1 Definitions

- A watercourse segment is a maximal portion of the drainage network that doesn’t contain watercourse merge points, watercourse origins, or pits. Drainage network consisting of watercourse segments are called refined drainage network.

- The drain of a point $p$, denoted $\text{drain}(p)$, is the point where $p$’s trickle path first meets the drainage network. If $p$ is on the network, then $\text{drain}(p) = p$.

- A catchment area of a watercourse segment $s$ consists of all points $p$ for which $\text{drain}(p)$ is on $s$. It is the same as a partition unit of the finer partition in preceding discussions.

- A strip is formed as follows: for every vertex $v$ of the TIN, we introduce a node at $\text{drain}(v)$. These nodes cut the watercourse segments into yet smaller
segments that we call the base segments. It is easily verified that all these nodes lie on local channels, see Figure 6.3.

- For a base segment s, define the right strip to be the set of points p, such that drain(p) is on s, and p's trickle path comes from the right, where the front is in line with the direction of the flow. Define the left strip similarly.

Figure 6.3: Strips

6.4.2 Lemma and Corollaries

**Lemma 4** A strip is a region bounded by a base segment from a local channel, a segment of a local ridge, and two steepest descent paths. No vertices of the TIN lie inside a strip.

We have two corollaries following directly from the lemma.

**Corollary 1** Basins, source areas, and catchment areas are bounded by local ridges and steepest descent paths.

**Corollary 2** Watershed area, projected area, or steady-state flow rate under uniform rainfall assumptions can be summarized by piecewise quadratic functions on the base segments of the refined drainage network.

45
**Proof:** For a query point $q$ on the drainage network, the left and right strips of the base segment $s$ containing $q$ are *partially* inside the watershed of $q$. Every other strip is either *entirely* inside or outside of the watershed of $q$, and contributes a constant amount of area, projected area, or steady-state flow.

The contribution from the left and right strips of base segment $s$, as shown before, can be summarized as a quadratic function $at^2 + bt + c$, where $t$ indicates the position of $q$ on $s$.

We now give an argument for the quadratic behavior of the function:

Imagine we move along the drainage network, the trace-ups starting at the current position would thus sweep the terrain. Within a triangle, the sweeping trace-ups are all parallel, since the steepest ascent direction is unique on a plane. The area bounded by the two parallel lines and the triangle’s boundary is quadratic in the distance between the two parallel lines, provided none of the triangle’s vertices lie between the two parallel lines (see Figure 6.4). The distance between the two parallel lines is linear in the distance between the intersections of an edge with the two parallel lines.

The size of the swept area is quadratic in the distance swept. The interpolation parameters for the area change only when the trace-ups sweep across a vertex.

![Sweeping Trace-up Line](image)

**Figure 6.4: Sweeping Trace-up Line**
This interpolation scheme can be extended to the whole trace-ups across multiple triangles: we can find the watershed sizes of drainage network points whose trace-ups sweep a vertex. We then pick a point on the drainage network that lies between each pair of consecutive points above described, and find its watershed area. These in-between points are necessary since we need three points to establish a quadratic interpolation. See Figure 6.5.

![Figure 6.5: Interpolation Points](image)

The following algorithm for preprocessing proves to be efficient:

Trace down from each vertex until hitting the drainage network. This takes $\sum_{i=1}^{n} |S_i|$, where $S_i$ is a trace-up, and $|S_i|$ is the size (number of line segments) of $S_i$. For each hit point and a point between each pair of consecutive hit points, we perform a trace-up to calculate the point's watershed size. The work needed for tracing up and calculating the area of the strip between trace-ups is $O(|S_i|)$. Therefore, the total complexity is $\sum_{i=1}^{n} |S_i|$. Storing the interpolation data takes $O(n)$ storage.

**Lemma 5.** Preprocessing for constant time (excluding point location operation) watershed size queries takes $\sum_{i=1}^{n} |S_i|$ time, and $O(n)$ space in the above algorithm.
Chapter 7

Conclusions

We have delved into the particulars of our models and methods in the preceding chapters. We conclude with an overview of the whole problem.

7.1 Level of Abstraction

Like anything else, to solving drainage problems, we need to keep our goal and resources in mind. We claim that richer and important information is obtained through extra processing in our method.

7.1.1 Advantages

We recount the benefits of our approach as follows:

- The model we have provides a common ground for discussions on terrain drainage characteristics. The definitions are general rather than particular to a specific representation such as TIN or DEM. They are mathematical, thus precise. It is a step closer to reality compared with previous researches.
• We can generate all the standard terrain features such as pits, peaks, watercourses, ridge lines. We give more consistent answers to these problems, because we have a well defined model and generally applicable definitions, rather than define these features as a product of a particular algorithm.

• We can easily extract the planar graph structure of terrain, which characterizes terrain drainage at the highest level. This graph structure is an inherent feature of the terrain. It is known to researchers for decades.

• We can accurately and efficiently answer watershed and catchment queries that are of interest to hydrology, hydraulic engineering and natural resource management. This is an important function that previous research has not fully addressed.

7.1.2 Disadvantages

The main disadvantages are:

• **Complex.** Our methods are more complex than cell based methods. However, we note that the local-operator/line-connection approach of finding V shaped grid points and connecting them into drainage network is complex to specify completely, especially since the line-connection part is not well defined and may not work properly. Approaches like Mark’s are simple in concept, but costly in computation (O(n log n)). Our approach uses a more complete model than Mark’s, but wins out in computation.

• **Computational Cost.** The computational cost is quite high in the theoretical worst-case. Because unlike DEM approaches, which never introduce new nodes or points besides the existing grid points, in our method we do need to add
nodes, intersection points between edges and trickle paths, etc. Thus our approach could potentially be very expensive, as indicated in the worst-case analysis, reaching $\Theta(n^3)$. In tests of actual terrain data, the complexity of global features in our method is linear $O(n)$ (excluding the TIN construction complexity).

7.2 Reflection of Reality

How well does our model reflect the physical process?

It bears more resemblance to the processes of the physical world than previous methods. It makes water follow along the steepest descent of the surface. Even though it cannot account for dynamics, it does reflect the law of gravity to certain extent.

Do the features we extract closely correspond to the features in the actual terrain?

As mentioned in the introduction, the assumption that the surface, computed out of sample points possibly lying on the surface of water bodies, is the underlying dry surface water flows on is at variance with the nature of the sample data. But that's as good as we can do, since those are the only data we have.

Furthermore, there are factors such as erosion, weather, sediment concentration, vegetation that affect the drainage process. We are not looking into any of those, although it is possible to incorporate them.

7.3 Future Research

Future research in terrain drainage will try to address some of the data structure and algorithm issues, make the model capable of handling special cases, and incorporate
different factors into our model.

7.3.1 Making the Model More Realistic

Currently, a pit is a true sink: water that goes in never comes out. Instead, we may want to make water fill to certain level, and have an absorption/evaporation model to keep the water level, or make water drain from the outlet.

7.3.2 Data Structures

Efficient data structures and algorithms need to be devised to process and store watercourses, general separating lines, planar terrain graph, and drainage forest.

7.3.3 Handling Special Cases

In our model, we haven’t explicitly addressed the problem of special cases. Some of the special cases are:

- At data level, there can be flat areas on the surface, where the direction of steepest descent is not well defined; there can be horizontal edges; there can be artifacts of sampling errors such as shallow pits, etc. We need to have a strategy to handle these cases. For example, consider flat areas as pits or lakes, and stop trickling down at such; use a filter to fill in small pits, make small pits overflow at the outlet, etc.

- At structure level, watercourses can go off the terrain boundary or come into the terrain boundary—from outside, and there can be partial basins. In such cases, we can never correctly trace all the drainage features because the necessary connections are outside of our data region. We need to devise methods to handle these cases.
7.3.4 Adding Other Factors

We can add factors such as variable precipitation, water absorption of soil, accumulation of water at basins, sub-surface water flow.

7.3.5 Merging Data Sources

So far, we only used surface geometry data of the terrain. One possible additional data source is a river/stream map of the terrain. If we can get both the digital elevation data and a river/stream map, one way of merging the two pieces of information would be to overlay them, and adjust the elevations and/or x-y locations of the sample points with certain constraint, so that wherever there is a river in the river map, there will be one in the elevation model conforming to our definition of a river.

Although there may be artifacts in adjusting the points, it is a first step in merging drainage information.
Bibliography


Appendix A

Terrain Function

The surface of a terrain can be represented as the graph in $\mathbb{R}^3$ of a continuous real function $z = f(x, y)$ defined on a closed, compact, connected subset of the xy plane. For $r \in \mathbb{R}$ and $r > 0$, the forward directional derivative $g(\theta)$ of $f(x, y)$ at point $(x_0, y_0)$ in the direction $\theta \in [0, 2\pi]$ (measured from the positive x direction) is:

$$g(\theta) = \lim_{r \to 0} \frac{f(x_0 + r \cos(\theta), y_0 + r \sin(\theta)) - f(x_0, y_0)}{r}$$

$g(0) = g(2\pi)$.

If the part of the surface $(x_0, y_0, f(x_0, y_0))$ lies on is a plane, we derive:

$$g(\theta) = k \cos(\theta - \theta_0)$$

where $k$ is the greatest slope value of the plane, and $\theta_0$ is the direction of the greatest ascent.

In a TIN, for a point $(x_0, y_0)$, $g(\theta)$ is defined on a partition of $[0, 2\pi]$, where the number of intervals (wrapped around intervals are considered as one interval) in the partition is the same as the number of triangle(s) incident upon the point. On each
interval, \( g(\theta) \) is a sinusoid with a possibly different coefficient \( k \) and phase angle \( \theta_0 \).

See Figure A.1.

![Diagram of sinusoidal directional derivative](image)

Figure A.1: Sinusoidal Directional Derivative

The analysis of drainage characteristics can be strictly based on the function \( g(\theta) \). A graph of \( g(\theta) \) will be very helpful in the analysis. For instance, the steepest descent direction is the direction \( \theta \) for which \( g(\theta) \) has the least negative value. A pit's \( g(\theta) \) is all positive. In TIN, \( g(\theta) \) of a point on a local ridge has two negative minima, and two maxima that lie on the division points of the intervals. A maximum ascent lies in the direction \( \theta \), where \( g(\theta) \) is a positive maximum. A maximum descent lies in the direction \( \theta \), where \( g(\theta) \) is a negative minimum.
Appendix B

Horizontal Slicing Plane Method

Many of the relations between ascents and descents can be analyzed by a horizontal slicing plane method. In analyzing a point, imagine cutting the terrain surface with a horizontal plane which lies slightly above or below the point. The intersection of the terrain surface and the plane is polylines. We then project the edges that are below (or above) this plane onto the plane, we get a set of new polygons. If we project the steepest descents of faces onto this plane, we know that these projections are perpendicular to their corresponding faces' intersections with the plane.

As an example, if we cut above a pit, we get a polygon, where the pit sees each vertex of the polygon. See Figure B.1. Because all the points on the polygon are of the same elevation, the further away the point is from the pit, the less $g(\theta)$ is in the direction pointing from the pit to that point, i.e. the smaller the slope is in that direction. Therefore the analysis of maximum and minimum of $g(\theta)$ can be reduced to analyzing the distance between the intersection point and the point of interest.
By using these facts, we can easily show that a pit has at least and possibly only one local channel; between two incoming watercourses that are not separated by any outgoing watercourse, there can be only one maximum ascent.

Figure B.1: Horizontal Plane of a Pit