Morphologically-Aware Elimination of Flat Edges from a TIN

Paola Magillo
DIBRIS, University of Genova
Via Dodecaneso 35
16146 Genova, Italy
paola.magillo@unige.it

Leila De Floriani
DIBRIS, University of Genova
Via Dodecaneso 35
16146 Genova, Italy
leila.defloriani@unige.it

Federico Iuricich
DIBRIS, University of Genova
Via Dodecaneso 35
16146 Genova, Italy
federico.iuricich@gmail.com

ABSTRACT

We propose a new technique for eliminating flat edges from a Triangulated Irregular Network (TIN) in a morphologically consistent way. The algorithm is meant to be a preprocessing step for performing morphological computations on a terrain. Terrain morphology is rooted in Morse theory for smooth functions. Segmentation algorithms have been defined for TINs, mostly based on discrete versions of Morse theory, and under the assumption that the terrain model does not include flat edges. On the other hand, flat edges often occur in real data, and thus either they are eliminated through data perturbation, or the segmentation algorithms must be able to deal with them. In both cases, the resulting Morse segmentations are highly affected by the presence of flat edges. The new technique we propose provides a better solution, as it preserves the set of maxima and minima of the original terrain, and improves consistency among the terrain decompositions produced by different segmentation algorithms.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; I.4.10 [**Image Processing and Computer Vision**]: Image Representation—*Morphological*

General Terms

Algorithms

Keywords

Terrain Modeling, Morphological Analysis, Triangulated Irregular Networks, Segmentation

1. INTRODUCTION

Modeling the morphology of a terrain is a relevant issue in several applications, including terrain analysis and understanding, knowledge-based reasoning, and hydrological simulation, just to mention a few. Morphology consists of feature points (pits, peaks and passes), feature lines (like ridges and ravines), or segmentation of the terrain into regions of influence of minima and maxima.

Morphological models are rooted in Morse theory [13], which is defined for smooth functions. In real applications, discrete terrain models are used, such as *Triangulated Irregular Networks (TINs)*, i.e., linear terrain approximations described by piece-wise functions defined over a triangulation of the domain with vertices at the sample points. Concepts of Morse theory have been transposed into a discrete setting [1, 9], usually under the assumption that that no two adjacent vertices have the same elevation, i.e., the terrain has no flat edges. The watershed transform is another concept used to define terrain segmentations, which are equivalent to Morse ones in the smooth case. The watershed transform is used in image processing to segment raster terrains.

Flat edges (and triangles) occur in real terrains. Some configurations correspond to real flat features (e.g.. lakes), but many are due to the limited precision in elevation sampling. Our work addresses the problems caused by the latter type of flat edges. Algorithms deriving from discrete versions of Morse theory need to remove flat edges in a preprocessing step, and just few of them can be internally modified to deal with flat edges as special cases. Algorithms based on the watershed transform deal with flat edges. However, the presence of flat edges is a serious problem, because preliminary perturbation of elevations leads to over-segmentation, and even methods handling flat edges produce much more different results on terrains with flat edges than on terrains without flat edges.

Here, we propose a preprocessing method to eliminate flat edges from a TIN in a morphologically consistent way. We modify the elevation of the vertices of flat edges in such a way that segmentations of the modified terrain do not have missing or extra basins and mountains, when transferred back to the original terrain.

The reminder of this paper is organized as follows. Section 2 reviews some concepts from Morse theory. Section 3 analyzes plateaus in a TIN. Section 4 presents rules to remove them in a morphologically consistent way, while Section 5 summarizes our algorithm. Section 6 presents experimental results. Finally, Section 7 contains some concluding remarks.

ACM SIGSPATIAL GIS '13, November 5-8, 2013. Orlando, Florida, USA

http://dx.doi.org/10.1145/2525314.2525341 ...$15.00.

2. MORSE DECOMPOSITIONS

2.1 Morse theory and Morse complexes

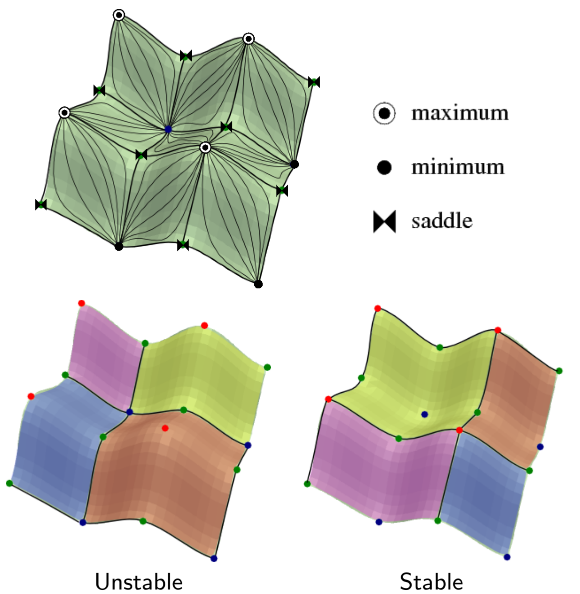


Figure 1: A function with integral lines, its stable and unstable Morse complexes, and Morse-Smale complex.

Let D be a domain in the two-dimensional Euclidean space and f be a smooth real-valued function defined over D. A point p of D is *critical* if the gradient of f at p is null. Otherwise, p is *regular*. Critical points of f are *minima*, *saddles* and *maxima*. A critical point p is *degenerate* if the determinant of the Hessian matrix of f at p is zero. A function f is called a *Morse function* if all its critical points are not degenerate [13]. Note that a Morse function admits only finitely many isolated critical points.

An *integral line* of f is a path which is everywhere tangent to the gradient of f (see Figure 1), and is maximal. The union of all integral lines that converge to a minimum [saddle, maximum] p cover a 2-cell [1-cell, 0-cell], called the *stable cell* of p. Similarly, integral lines that originate from a minimum [saddle, maximum] p cover a 0-cell [1-cell, 2-cell], called the *unstable cell* of p. The stable (unstable) cells decompose D into a Euclidean cell complex, called a *stable* (*unstable*) *Morse complex* (see Figure 1). On a terrain, the stable cell of a minimum corresponds to a basin, the unstable cell of a maximum corresponds to a mountain, stable (unstable) 1-cells are lines connecting two maxima (minima) and passing through saddles, they correspond to ridges (valleys). In the following, stable and unstable cells will also be called *basins* and *mountains*, respectively, and both of them will be also referred to generically as *regions*.

In the literature, there are two extensions of Morse theory to a discrete domain. Banchoff's *piece-wise linear Morse theory* [1] extends Morse theory to piece-wise linear functions defined on TINs, by introducing a definition of critical point for discrete scalar fields. The basic assumption is that every pair of points of the TIN have distinct elevation values or, at least, that each pair of adjacent vertices (i.e., connected by an edge) have different elevation values. Such condition ensures a decomposition of the domain into cell complexes having the same properties as the Morse complexes. Unfortunately, real data often contain flat edges.

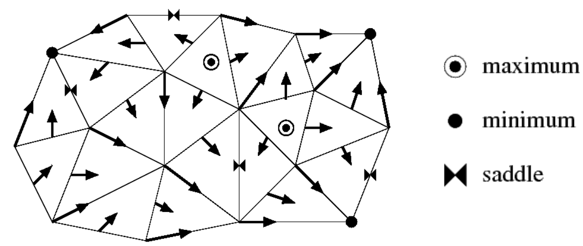


Figure 2: A discrete gradient vector field on a triangulation.

Forman's discrete Morse theory [9] simulates Morse theory in the discrete. This goal is achieved by defining a function F on all the cells (of any dimension) in a cell complex; in a triangulation, F is defined on vertices, edges, and triangles (generally called *simplices*). Function F is a *discrete Morse function* if it satisfies the following property: every i-dimensional simplex σ has a larger value than all $(i-1)$-dimensional simplices bounding σ, and σ has a lower value than all $(i+1)$-dimensional simplices bounded by σ, with at most one exception. If there is such an exception, the pair of simplices defines a *discrete gradient vector field* (and we draw an arrow from the simplex with higher value to the one with lower value). Otherwise, σ is a critical simplex. Note that all minima are at vertices and all maxima are at triangles.

It is not easy to construct a discrete Morse function, it is simpler to construct a discrete gradient vector field. A discrete vector field can be viewed as a collection of *arrows*, each connecting a i-dimensional simplex to an incident $(i+1)$-dimensional simplex (arrows are vertex$\rightarrow$edge or edge$\rightarrow$triangle) where each simplex is a head or a tail of at most one arrow. On the vector field, we can move from an edge to an adjacent edge or from a triangle to an adjacent triangle, if there is an arrow from their common vertex [edge] to the second edge [triangle]. A discrete vector field is a discrete gradient vector field if there are no cyclic paths (see Figure 2).

2.2 Segmentation algorithms

Terrain segmentation in a GIS is a complex problem that involves many different issues. Real GISs include a number of accessory tools to segmentation, such as filling spurious basins, completing broken valley lines, assigning a water flow direction to plateaus (see, for instance, TerraStream at `http://madalgo.au.dk/Trac-TerraSTREAM`). Here, we want to highlight the effect of our preprocessing method on terrain segmentation. Therefore, we consider a "raw" application of segmentation algorithms, i.e., without accessory data treatments.

Algorithms for computing Morse decompositions can be classified into four categories: boundary-based [18, 8, 16, 7], region-growing [11, 5, 6], watershed-based [15, 19, 17, 12], and Forman-based approaches [10, 14]. An extended survey of such methods can be found in [2, 4]. Here, we provide a short description of the basic ideas behind them. In the description, we focus on the computation of the stable segmentation. The unstable segmentation is built as the stable segmentation by simply changing the sign of all field values.

Boundary-based algorithms build the 1-skeleton of the stable segmentation by starting from saddle vertices and following edges which correspond to the maximum ascent, until a maximum is reached. From the 1-skeleton, we can easily obtain the triangles forming the 2-cells of the Morse complex.

Region-based algorithms builds the stable 2-cells by starting from each minimum v and progressively adding triangles.

Watershed algorithms are rooted in image processing, and their extensions to a TIN approximate 2-cells of the Morse complexes as sets of TIN vertices. The idea of watershed by *simulated immersion* [19, 17] is to drill holes at local minima, and insert the terrain surface in a pool of water. Where water coming from different minima would merge, we build separations between basins. The watershed by *rainfall* algorithm [12] starts from each vertex and moves to its lowest neighbor vertex, until it reaches a minimum. Watershed algorithms accept flat edges. When the simulated immersion algorithm reaches a certain elevation, it floods all vertices at that elevation. The rainfall algorithm identifies plateaus in a preprocessing step, and then treats each of them as a single vertex.

The algorithms based on **Forman** theory [10, 14] first build a discrete gradient field over the triangulation, then extract the basins. The discrete gradient field is built by processing each vertex v independently. Unless v is a local minimum, we draw an arrow from v to the edge corresponding to its lowest adjacent vertex. Then, a homotopic expansion process will draw all necessary arrows from an edge to a triangle among those having their highest vertex in v. When arrows cannot be drawn for a simplex, we recognize it as a critical simplex. Basins are built by starting from each minimum, and moving from a vertex v to an adjacent vertex w whenever there is an arrow from v to edge vw.

Algorithms which assume an input TIN without flat edges can be used on real data only after a preprocessing on the TIN to remove flat edges. This is commonly done through perturbation of elevations values (see, for instance, [14]).

3. PLATEAUS IN A TIN

Given a set S of points sampled on the surface of a terrain, a *Triangulated Irregular Network (TIN)* consists of a plane triangulation Σ, having its vertices at the projections of points of S on the plane, and of a set of linear interpolant functions defined on each triangle of Σ. In a TIN, we can define a *primal graph*, where nodes are the vertices of Σ and arcs describe vertex-vertex adjacency relations, and a *dual graph*, where nodes are the triangles of Σ and arcs describe triangle-triangle adjacency relations.

An edge of a TIN is *flat* if its two extreme vertices have the same elevation. A triangle of a TIN is *flat* if its three vertices have the same elevation, or (equivalently) it its three edges are flat.

Let v be a vertex of a TIN. Among the neighbors of v, we distinguish vertices having higher, lower, or equal elevation with respect to v, and label them with symbols $+$, $-$, or $=$, respectively. Any vertex labeled $=$ forms a flat edge with v.

In the sorted sequence of neighbors of v, we identify maximal sub-sequences of vertices having the same label (see Figure 3). Sequences labelled $=$ are called *flat sequences*. Other sequences are called *signed sequences* (where the sign of the sequence can be $+$ or $-$).

If all sequences around a vertex v are signed, the vertex

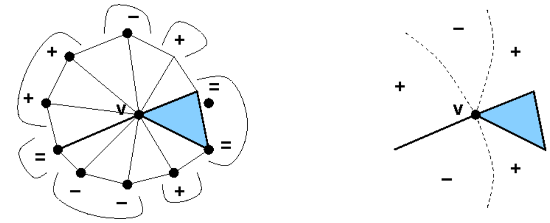


Figure 3: Neighbors of a vertex v labeled depending on their elevation with respect to v. Thick edges and colored triangles are flat, dashed thin lines denote sign changes.

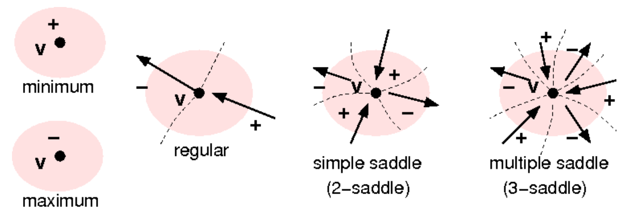


Figure 4: Classification of an internal vertex v with no flat incident edges. Dashed lines represent sign changes, thick arrows represent integral lines.

classification defined in [1] for a TIN without flat edges applies to v. Let h be the number of signed sequences around v (see Figure 4):

- If $h = 1$, then v is a local *extremum*, more precisely v is a local *maximum* (or *minimum*) if the sign of the sequence is $-$ (or $+$).
- If $h = 2$, then v is a *regular* vertex.
- If $h > 2$, then v is a *saddle* vertex.

If a saddle vertex v is internal, then h is even i.e., $h = 2k$ with $k \geq 2$. Saddle v is *simple* if $k = 2$ and *multiple* otherwise. We say that k is the *multiplicity* of a saddle v, and v is a k-saddle. If a saddle vertex v is on the boundary, then h can be even (as above) or odd, i.e., $h = 2k - 1$ with $k \geq 2$. In this case, we say that the *multiplicity* of the saddle is $k = \lceil h/2 \rceil$.

Around a vertex v with $h > 0$ signed sequences, there are h sign changes if v is internal, and $h - 1$ sign changes if v is on the boundary. We prefer to classify vertices based on the number of signed sequences rather than on the number of sign changes. However, the two approaches are equivalent.

Let v be a vertex in which flat edges are incident. We can count signed sequences around v separately in each section of the neighborhood of v between two flat sequences.

A *plateau* C is a maximal vertex-connected set of flat edges and flat triangles. For simplicity, we consider the case in which all vertices of C are internal. We consider the set T_C of triangles which have an edge or a vertex in C and are not flat. In the dual graph, we consider arcs connecting pairs of triangles in T_C which are mutually adjacent along a non-flat edge having one endpoint in C (see Figure 5). Such arcs form one or more closed chains, which correspond to rings of edge-adjacent triangles in the TIN. Given a ring within T_C, sorted counterclockwise, we extract, in sorted order, the edges that its triangles share with C. Such edges form a *contour* of C,

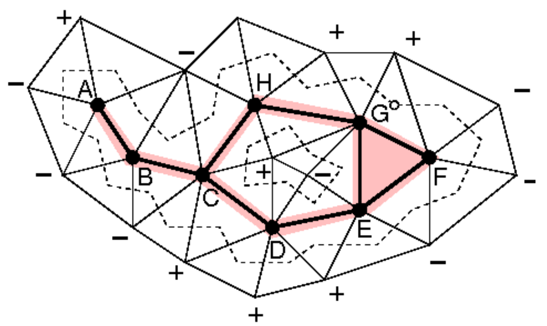


Figure 5: A plateau C (colored edges and triangle are flat), and triangles of T_C (which form two rings of edge-adjacent triangles).

also called a *flat contour*. Note that a flat contour is oriented, and it forms a closed line, which is not necessarily simple. If a plateau C has $h+1$ flat contours (with $h \geq 0$), we say that plateau C has h holes (no holes for $h=0$). In Figure 5, we have two flat contours: $A, B, C, D, E, F, G, H, C, B$ and C, H, G, E, D.

Let v be a vertex of C where both flat and non-flat edges are incident. We say that v is an *exposed vertex* of C. Exposed vertices lie on the contours of C. If an exposed vertex v has one flat sequence, then v appears once in one flat contour; otherwise v may appear in different contours of C, or it may appear more times in a single contour.

Given a flat contour, we list vertices that are adjacent to the right side of its vertices in the order induced by the contour, we give them a sign in $\{+, -\}$ with respect to the elevation of C, and define *signed sequences* of C in a similar way as done for vertices. In Figure 5, the flat contours have six and two signed sequences, respectively. The total number of signed sequences of a plateau C (for all contours) is also obtained in the following way. For each exposed vertex of C, we count the number of signed sequences of v minus the number of flat sequences of v, and we sum up such quantities.

A *plateau C without holes* is classified based on the number h of signed sequences, in the same way as a single vertex. Thus, C may be a local *extremum* (*maximum* or *minimum*), it may be *regular*, or it may be a *saddle*.

A *plateau C with holes* may act as just one critical point, or as different types of critical points simultaneously. For instance, let a plateau C have one hole, and just one signed sequence on each side. Let the signs of such two sequences be opposite. Then, there is one minimum inside and one maximum outside C (or vice versa). Each point of C has one ascending and one descending integral line (see Figure 6, left). C is regular. Now, let the signs of such two sequences be the same. Then, there are two extrema of the same type (two minima or two maxima), one lying inside and one outside C. C acts as a local extremum as the terrain in the immediate neighborhood of C is entirely higher (or entirely lower). But C also acts like a simple saddle because there are two local extrema of the same type, connected through C. Therefore, some of its points have two ascending or two descending integral lines (see Figure 6, right).

4. SIMPLIFICATION OF PLATEAUS

We first illustrate the rules used in our algorithm to eliminate flat edges by detaching a vertex at a time from a

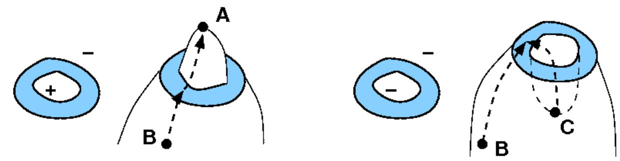


Figure 6: Plateaus with one hole. Arrows denote (ascending) integral lines, A is a maximum, B and C are minima.

plateau. Then, we present the overall algorithm. For brevity, we describe rules for internal vertices only. We use different rules to simplify plateaus without holes and plateaus with holes.

4.1 Simplification of plateaus without holes

The purpose of the following rules is to eliminate all plateaus without holes, and delete all flat triangles from plateau with holes, thus reducing each of them to one or more rings of flat edges.

Let v be an internal vertex belonging to a plateau C, such that v is exposed for C, and v has one flat sequence. As v is internal, then all signed sequence of v are consecutive around v.

Let α be the number of signed sequences around v, and let γ be the total number of signed sequences of plateau C. Suppose that we raise or lower v sligthly (i.e., increase / decrease its elevation by an offset $\Delta > 0$ which is smaller than the elevation difference between v and all its higher/lower neighbors).

Then, vertex v and all its incident flat edges and flat triangles are removed from plateau C. We call C' the remaining part of C (note that C' may be a single point; in this case, we consider it as a degenerate plateau). After such modification, we have α' signed sequences at v and γ' signed sequences at C'. We will show that $\gamma' + \alpha' = \gamma + 2$. In other words, moving vertex v is equivalent, in terms of signed sequences, to the creation of a new regular vertex in the TIN.

Let the two signed sequences adjacent at the two sides of the unique flat sequence of v have the same sign (i.e., as v is internal, α is odd), and let it be $+$ $(-)$. This situation is illustrated in Figure 7, where $\alpha = 5$. If we raise (lower) v, then the number of signed sequences around v becomes $\alpha' = \alpha + 1$, because the sign of the old flat sequence is now $-$ $(+)$ for v, and the number of signed sequences around the modified flat contour becomes $\gamma' = \gamma - \alpha + 1$, because all signed sequences existing at v are replaced by just one positive (negative) sequence consisting of v. In the same conditions, if $\alpha > 1$ and we lower (raise) v, the number of signed sequences around v becomes $\alpha' = \alpha - 1$, because the sign of the old flat sequence is now $+$ $(-)$ for v and thus two positive (negative) sequences of v are merged, and the number of signed sequences around C' is $\gamma' = \gamma - \alpha + 3$, because C' retains the first and last positive (negative) sequences of v and gains a new negative (positive) sequence consisting of v. If $\alpha = 1$, then after lowering (raising) v we have $\alpha' = \alpha$, because no merging occurs as the sequence is already one, and $\gamma' = \gamma - \alpha + 2$, because the unique positive (negative) sequence of v is split into two, with the insertion of v as a new negative (positive) sequence.

Let the two signed sequences adjacent at the two sides of the flat sequence have opposite signs (i.e., as v is internal,

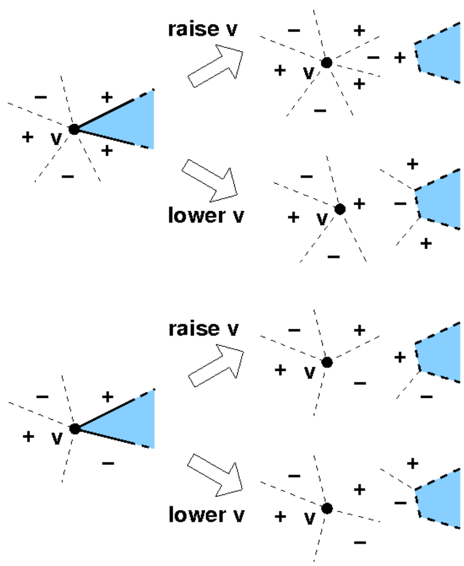


Figure 7: Raising or lowering vertex v when the first and last signed sequence of v have the same sign (here $+$), and when they have opposite signs.

α is even). This situations is illustrated in Figure 7, where $\alpha = 4$. If we lower or raise v, then the number of signed sequences around v remains α, and the number of signed sequences around the modified flat contour becomes $\gamma' = \gamma - \alpha + 2$.

Depending on the value of α, vertex v may become an extremum, a regular vertex, or a saddle (see Table 1). In order to preserve morphology, we do not execute moves in the first case.

4.2 Simplification of plateaus with holes

The purpose of the following rules is to open plateaus with holes, in such a way that they can later be treated with the rules presented in the previous subsection.

Let v be an internal vertex belonging to a plateau C, such that v is exposed for C, and v has two flat sequences, each of them corresponding to just one flat edge. As v is an internal vertex, it follows that its signed sequences form exactly two sections around v.

Let a and b be the other endpoints of the two flat edges incident in v. Let α_1 and α_2 be the numbers of signed sequences in the two non-flat sections of the neighborhood of v. Let γ be the total number of signed sequences of C. Suppose that we change the elevation of v slightly. Such operation removes v and the two flat edges from C. If v belonged to two different holes of C, then such two holes are merged. Otherwise, C is split into two plateaus. In any case, let C' be what remains of plateau C (C' may degenerate into two vertices).

At each flat edge incident in v, the effect of changing the elevation of v is the one described in Section 4.1. Simply, here v we perform the same movement at both flat edges va and vb simultaneously.

Let α' be the final number of signed sequences around v after the movement. Let γ' be the number of signed sequences of C'. Then, $\alpha' + \gamma' = \alpha + \gamma + 4$. In other words, the operation creates a number of signed sequences which is equivalent to a new simple saddle.

This is not difficult to check, if we consider all possible configurations of signs across the two flat edges va and vb. The description of individual cases is omitted for brevity. The resulting cases are summarized in Table 2. Again, we do not execute moves causing v to become an extremum.

5. SIMPLIFICATION ALGORITHM

Now, we present a summary of our method. In the simplification, rules are applied with the following priorities:

1. rules for plateaus without holes, in situations in which v becomes a regular vertex;

2. rules for plateaus with holes, in situations in which v becomes a regular vertex;

3. rules for plateaus with holes, in situations in which v becomes a simple saddle;

4. rules for plateaus without or with holes, in other situations (sorted by increasing multiplicity of the saddle created in v)

The rationale is that we prefer not to create saddles. Secondly, we prefer to create simple saddles from plateaus with holes (as, in any case, their solution will add four signed sequences, equivalently to a 2-saddle). Finally, we prefer to create saddles with small multiplicity.

Our rules allow the complete elimination of flat edges from a terrain, with limited and controlled modification of morphology. It is not difficult to verify the validity of the following properties.

- Extrema are not created or deleted. In particular, a plateau without holes, which is a maximum (minimum), gives rise to one maximum (minimum) vertex and a number of regular vertices. A plateau with a hole, which is a maximum (minimum), gives rise to one maximum (minimum) vertex, one simple saddle, and a number of regular vertices.

- A regular plateau (with or without holes) gives rise to a set of regular vertices.

- A plateau C without holes, which is a saddle, gives rise to a number of saddles and regular vertices. If $2\,\beta$ is the initial number of signed sequences of C, and $s+1$ is the number of final saddles, then the total multiplicity of such saddles is $\beta + s$.

- A plateau with holes, with initially $2\,\beta$ signed sequences, generates a number of saddles whose total multiplicity is $\beta + h + s$, where h is the number of holes and $s + 1$ is the number of final saddles.

Raising and lowering a vertex deserves attention. If a plateau has a large number of vertices, after many successive simplifications, the elevation difference between two adjacent vertices may become zero in machine precision. Therefore, we raise and lower vertices in a symbolic way. At the end of the process, we scan vertices in increasing order of their (symbolically) modified elevations, and assign them new elevations.

Note that the modified TINs differ from the original ones just in vertex elevations. The underlying triangulation is unchanged. Therefore, segmentations computed on the modified TIN can be transferred back to the original TIN.

Before			move	After					
v	C	signs at sides		v	C'	total	extremum -NO-	regular -OK-	2-saddle -OK-
odd α	γ	$++$ $(--)$	raise (lower)	$\alpha+1$	$\gamma-\alpha+1$	$\gamma+2$	–	$\alpha=1$	$\alpha=3$
odd $\alpha>1$	γ	$++$ $(--)$	lower (raise)	$\alpha-1$	$\gamma-\alpha+3$	$\gamma+2$	–	$\alpha=3$	$\alpha=5$
$\alpha=1$	γ	$++$ $(--)$	lower (raise)	α	$\gamma-\alpha+2$	$\gamma+2$	$\alpha=1$	–	–
even α	γ	$+-$ $-+$	raise or lower	α	$\gamma-\alpha+2$	$\gamma+2$	–	$\alpha=2$	$\alpha=4$

Table 1: Simplification of a plateau without holes. For larger values of α, v becomes a multiple saddle.

Before			move	After					
v	C	signs at va, vb		v	C'	total	extremum -NO-	regular -OK-	2-saddle -OK-
odd α_1, odd α_2	γ	$++, ++$ $(--, --)$	raise (lower)	$\alpha_1+\alpha_2+2$	$\gamma-\alpha_1-\alpha_2+2$	$\gamma+4$	–	–	$\alpha_1=\alpha_2$ $=1$
odd α_1, odd α_2 $\alpha_1, \alpha 2>1$	γ	$++, ++$ $(--, --)$	lower (raise)	$\alpha_1+\alpha_2-2$	$\gamma-\alpha_1-\alpha_2+6$	$\gamma+4$	–	$\alpha_1+\alpha_2$ $=4$	$\alpha_1+\alpha_2$ $=6$
$\alpha_1=1$, $\alpha_2=1$	γ	$++, ++$ $(--, --)$	lower (raise)	1	$\gamma+4$	$\gamma+4$	$\alpha_1=1$ $\alpha_2=1$	–	–
α_1, α_2	γ	$+-, +-$ or $+-, -+$	raise or lower	$\alpha_1+\alpha_2$	$\gamma-\alpha_1-\alpha_2+4$	$\gamma+4$	–	$\alpha_1=1$ $\alpha_2=1$	$\alpha_1+\alpha_2$ $=4$
even α_1, even α_2	γ	$++, --$	raise or lower	$\alpha_1+\alpha_2$	$\gamma-\alpha_1-\alpha_2+4$	$\gamma+4$	–	–	$\alpha_1=\alpha_2$ $=2$
odd α_1, even α_2	γ	$++, +-$	raise	$\alpha_1+\alpha_2+1$	$\gamma-\alpha_1-\alpha_2+3$	$\gamma+4$	–	–	$\alpha_1+\alpha_2$ $=3$
odd α_1, even α_2	γ	$++, -+$	lower	$\alpha_1+\alpha_2-1$	$\gamma-\alpha_1-\alpha_2+5$	$\gamma+4$	–	$\alpha_1+\alpha_2$ $=3$	$\alpha_1+\alpha_2$ $=5$
odd α_1, even α_2	γ	$--, -+$	lower	$\alpha_1+\alpha_2+1$	$\gamma-\alpha_1-\alpha_2+3$	$\gamma+4$	–	–	$\alpha_1+\alpha_2$ $=3$
odd α_1, even α_2	γ	$--, +-$	raise	$\alpha_1+\alpha_2-1$	$\gamma-\alpha_1-\alpha_2+5$	$\gamma+4$	–	$\alpha_1+\alpha_2$ $=3$	$\alpha_1+\alpha_2$ $=5$

Table 2: Simplification of a plateau with holes. The last six rows are intended to include also the symmetric configurations. For larger values of α, v becomes a multiple saddle.

6. EXPERIMENTS

Table 3 shows the behavior of our algorithm on a number of terrains. The algorithm creates no new minima or maxima, and a limited number of new saddles.

An alternative solution to remove flat edges from a terrain consists of perturbing elevations. A typical formula for a grid of size $M \times N$ is adding to the elevation of each point (i, j) a value equal to $e((i + M j)/(3 MN))$, where e is the smallest difference (in absolute value) between two adjacent, not equal, vertices (see, for instance, [14]). We adapted this formula to a TIN, by summing to the i-th vertex the quantity $e(i/(3 n_v))$ where n_v is the number of vertices.

In Table 4, we compare the number of critical points in the resulting TINs after removing flat edges by perturbation, or by our method. Perturbation creates new maxima, new minima, and a larger number of new saddles. The number of extrema in the output TIN, in case of perturbation, is from 120% to 150% than with our method, and the number of saddles is from 120% to more than 200%. Saddles created by our method are no more than 5% of the final saddles (see Tables 3 and 4).

Table 4 also compares execution times. Times of perturbation are roughly proportional to the number of vertices in the TIN. Note that the execution times of our method also depend on the number of flat edges in the initial TIN. On **Maggiore**, where more than 11% of the edges are flat, the time taken by our method is about 500% of the time required for perturbation. On **Baia**, which has less than 1% of flat edges, our method is even slightly faster than perturbation. However, the order of magnitude of running times of the two methods is the same.

In the following subsections, we investigate the properties of our algorithm. In Subsection 6.1, we compare our method against perturbation in a case in which a ground truth is known. In Subsection 6.2, we compare the direct computation of Morse segmentations on a TIN with flat edges (by handling flat edges inside the segmentation algorithm) with the computation of Morse segmentations after removing flat edges with our method. We present results on small-size terrains, in order to better highlight problems and features.

Terrain	type	vertices	triangles	flat edges	flat triangles	applied rules	new saddles
Ustica	gridded	1128	2111	182	4	178	0
Marcy	irreg.	3590	6898	466	68	398	0
Elba	gridded	1335	2483	390	63	324	1
Genova	irreg.	433174	863059	60192	8971	50750	129
Maggiore	gridded	810000	1616402	273817	109149	163661	1384
Maui	irreg.	2000812	4001507	228331	75465	152801	58
Baia	irreg.	4166490	8324898	86470	14145	72261	113
Puget	irreg.	9734926	19459233	684021	195408	487577	1880

Table 3: Results of our algorithm to test terrains. The last column counts the times in which a rule made v become a saddle.

Terrain	vert.	triang.	flat	perturbation				our method			
			edges	min.	sad.	max.	time	min.	sad.	max.	time
Genova	433174	863059	60192	9367	17777	8412	2 sec	6214	12160	5948	5 sec
Maggiore	810000	1616402	273817	16707	32202	15496	5 sec	12811	25252	12442	23 sec
Maui	2000812	4001507	228331	17011	30721	13711	16 sec	11843	18846	7004	18 sec
Baia	4166490	8324898	86470	9878	18884	9007	37 sec	8091	15763	7673	35 sec
Puget	9734926	19459233	684021	135835	227153	91319	70 sec	83230	163786	80557	138 sec

Table 4: Number of critical points and execution times with data perturbation, and with our algorithm.

6.1 Comparison with a ground truth

Eggs is a synthetic TIN obtained by sampling a combination of Gaussian functions on a grid. It has 5751 vertices and 11200 triangles, and no flat edges. We rounded vertex elevations by truncating them to 2, 1, and 0 decimal digits, thus obtaining terrains with flat edges, that we call $T_{dec2}, T_{dec1}, T_{dec0}$, respectively (see Figure 8). Later, we eliminated flat edges by using our method. The resulting terrains will be called *restored terrains* in the following.

The original **Eggs** TIN has 25 minima and 26 maxima. The restored terrain from T_{dec2} has 25 minima and 26 maxima. All minima are at the same vertices as in the original terrain, while one maximum moved to a neighboring vertex. This happened because the maximum in T_{dec2} was a plateau (in particular, it was a flat edge), and our method made the other endpoint of such edge a maximum. The restored terrain from T_{dec1} has 25 minima and 24 maxima. Two maxima actually disappeared in the smoothing process due to coordinate rounding (see Figure 9). One maximum and five minima moved to neighboring vertices. The restored terrain from T_{dec0} has 21 minima and 24 maxima. The same two maxima as in T_{dec1}, and four minima are lost. About half of surviving extrema moved to neighboring vertices.

These results confirm that our algorithm eliminates flat edges from a terrain in a way which is consistent with morphology. Differences between the original TIN and the restored TINs are limited and correspond to morphological changes (extrema which disappeared) happened in coordinate rounding rather than during the elimination of flat edges.

Now, we eliminate flat edges through perturbation. The number of minima and maxima on perturbed terrains are 25 and 26 (i.e., the same number as on the original terrain) from T_{dec2}; 26 and 27 from T_{dec1}; 69 and 64 (i.e., more than 200% than the original terrain) from T_{dec0}.

By increasing the degree of smoothing, the number of extrema decreases in restored terrains with our method, while it increases in perturbed terrains. If the former behavior is intuitively correct (coordinate rounding produces a terrain with fewer peaks and pits), the latter one is not (reducing information contained in data cannot lead to new morphological features), and it is due to the noise introduced with perturbation. As we noticed before, our method exhibits higher computation times than perturbation, but not much higher (as shown in Table 4 for large TINs), and they are compensated by a better quality in the result.

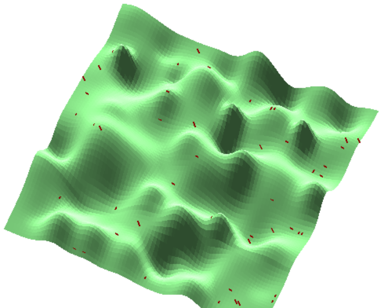
6.2 Impact of flat edges on segmentation

We consider the first three terrains in Table 3. The original TINs, with flat edges, are shown in Figure 10. TINs without flat edges have been obtained by applying our method.

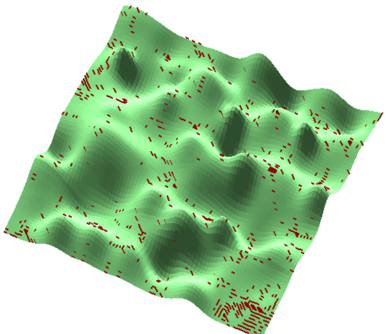
In our experiments, we have used representative algorithms for each class, namely a boundary-based algorithm [18, 16], a region-growing algorithm [11], two watershed algorithms (based on simulated immersion [19, 17] and on rainfall [12], respectively), and a Forman-based algorithm [14]. We have implemented all algorithms for TINs.

Watershed-based and Forman-based algorithms label each vertex with a basin index, while boundary-based and region-growing algorithms label triangles. For comparison purposes, we obtain triangle basins from vertex basins. The simulated immersion method produces boundary vertices labeled as watershed. We use edges defined by pairs of such vertices to divide basins, and assign a basin to triangles with three watershed vertices based on their slope. For the other algorithms, we use the following process. If all three vertices of a triangle t have the same basin, then t has that basin. For each triangle t, if its adjacent triangle t_1 along the lowest edge (edge opposite to highest vertex of t) has a basin, then t has the same basin as t_1. If t_1 has not a basin, and (t, t_1) form a diamond of mutually adjacent triangles along their lowest edge, and triangle t_2, externally adjacent to the lowest edge of the diamond, has a basin, then t and t_1 have the same basin as t_2.

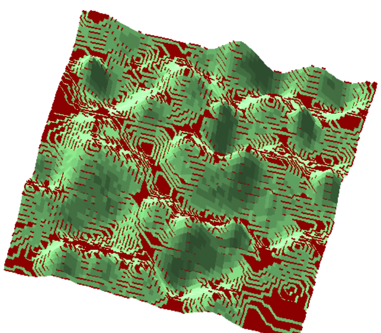
On TINs without flat edges, we run all five representative segmentation algorithms. On TINs with flat edges, we run



T_{dec2} (50 flat edges, no flat triangles)



T_{dec1} (548 flat edges, 18 flat triangles)



T_{dec0} (5090 flat edges, 1249 flat triangles)

Figure 8: Flat edges in Eggs terrains after rounding of z values to 2, 1, and 0 decimal digits.

original

restored

Figure 9: One of the two maxima disappeared in T_{dec1} due to elevation rounding. The image shows an unstable segmentation of the terrain, balls denote maxima.

the two watershed algorithms, and the region-growing algorithm (which is equipped with ad-hoc solutions for flat edges [11]). The boundary-based and Forman-based algorithms cannot be applied on TINs having flat edges. The region-growing algorithm identifies plateau minima and grows 2-cells of the segmentation from them. It also uses specific rules to enlarge 2-cells when it encounters flat triangles.

For all algorithms, the number of 2-cells is 5 in the stable segmentation, and 16 in the unstable segmentation, but boundaries between 2-cells may be differently drawn by each algorithm.

We compare the segmentations pairwise by using two metrics: *Rand Index (RI)*, and *Hamming Distance (HD)* [3]. RI counts the fraction of pairs of triangles both assigned to the same region, or both assigned to different regions, in the two segmentations. HD finds, for each region of one segmentation, a corresponding region in the other segmentation, and checks the fraction of triangles assigned to corresponding regions. Table 5 shows the values of HD and RI metrics on TINs with and without flat edges. Boundary- and Forman-based algorithms are not shown in the comparisons because they cannot be applied on TINs with flat edges, and their results on TINs without flat edges are always identical to the region-growing algorithm and to the rainfall algorithm, respectively.

Values of RI and HD metrics are higher on TINs without flat edges, than on TINs with flat edges, and they are (almost) equal to 100%. In presence of flat edges, values are sometimes below 85%, while they are always above 93% after removal of flat edges. Figure 11 shows unstable segmentations for the **Ustica** terrain, computed on the original TIN and on the TIN from which flat edges have been removed. This is a case in which the improvement in similarity is small (see green histograms for **Ustica** in Table 5, unstable case), but we can still see it.

Table 6 shows differently classified triangles on TINs without flat edges. On all TINs without flat edges, the boundary-based and region-growing algorithms provide the same stable and unstable segmentations. Segmentations output by Forman's and rainfall algorithms are also equal. The output segmentation of the simulated immersion algorithm is usually different.

Triangles, which are classified differently between any two stable segmentations, or unstable segmentations, are generally fewer than 2.5%. An exception happens for **Ustica**, where rainfall and Forman's algorithms give an unstable segmentation which is different from others on more than 6% of the triangles (shown by a circle in Figure 11).

All algorithms generally give equal or very similar segmentations on terrains satisfying the theoretical condition of no flat edges. This is quite remarkable since they are based on very different approaches. The similarity in the output segmentations of different algorithms suggests that each of them, in its basic idea, is theoretically well founded.

7. CONCLUDING REMARKS

We have proposed a new method to eliminate flat edges from a TIN while preserving its morphological structure. The conditions for applying our simplification rules can be evaluated locally. The final TIN has the same maxima and minima, and a limited number of new saddle points with respect to the original one. Our method performs better than classical perturbation in removing flat edges, since it does not introduce new maxima or minima. Also, even if some segmentation algorithms can manage flat edges internally, the degree of similarity among the results of different

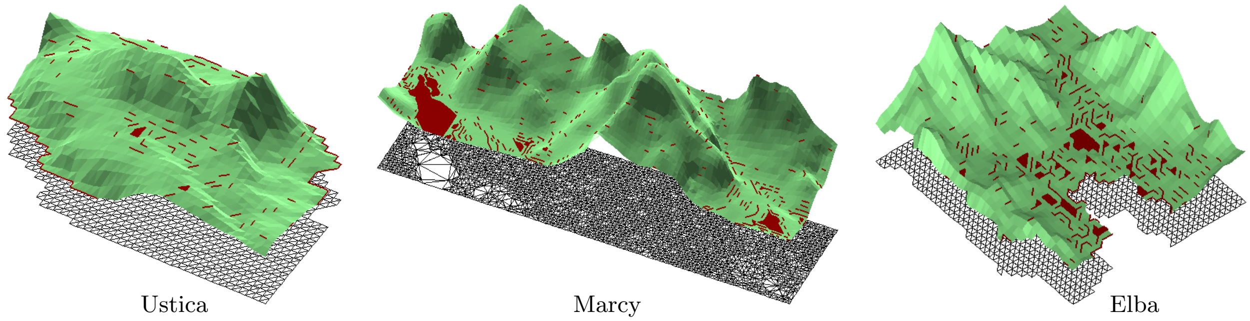


Figure 10: The original terrains with their flat edges and flat triangles (in red).

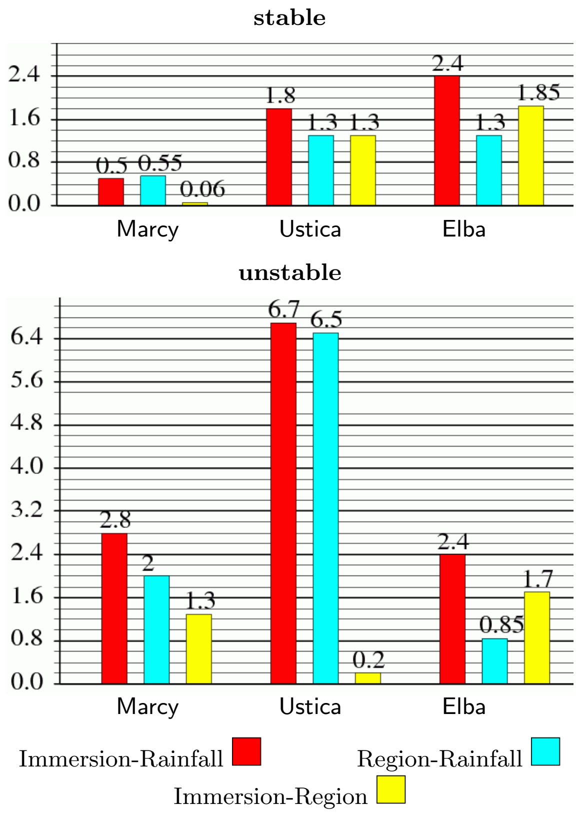


Table 6: Percentage of differently classified triangles in segmentations produced by the various algorithms on terrains with no flat edges.

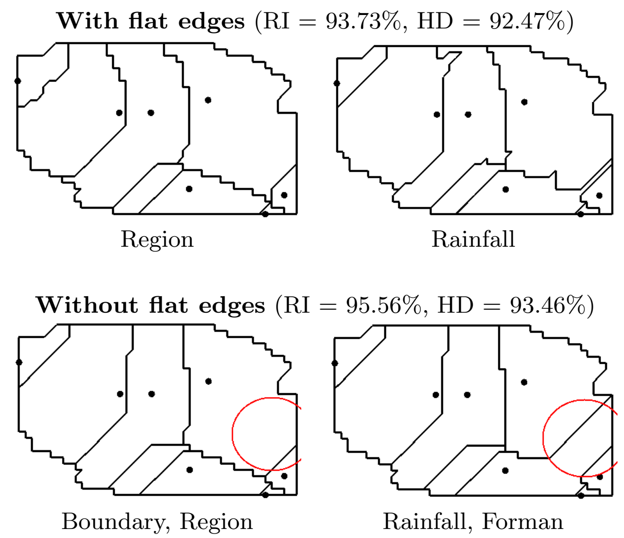


Figure 11: Unstable segmentations found by the various algorithms for Ustica on the TINs with and without flat edges.

approaches is much higher (often equal to 100%) on TINs without flat edges.

As a side contribution, our work showed that a comparison of the different existing approaches to TIN segmentation, not biased by the presence of flat edges, highlights similar behaviors from inherently different approaches. This is an interesting basis for further investigations.

An extension of our method to 3D scalar fields can be considered. Rules for eliminating dangling parts of plateaus (Section 4.1) can be easily extended to 3D for a vertex v whose incident elements include just either (i) one flat edge and no flat triangles or tetrahedra, (ii) one connected component of flat triangles and no isolated flat edges or flat tetrahedra, or (iii) one connected component of flat tetrahedra and no lower-dimensional flat parts. But in 3D we also need to deal with vertices whose configuration of incident flat edges, triangles and tetrahedra is non-manifold. Rules for plateaus with holes (Section 4.2) extend to open flat 2D shells that are hollow inside, but in 3D we also need to open closed chains of one-dimensional flat edges.

8. ACKNOWLEDGMENTS

This work has been partially supported by the Italian Ministry of Education and Research under the PRIN 2009 program and by the National Science Foundation under grant number IIS-1116747.

9. REFERENCES

[1] T. Banchoff. Critical points and curvature for embedded polyhedral surfaces. *American Mathematical Monthly*, 77(5):475–485, 1970.

[2] S. Biasotti, L. De Floriani, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, and M. Spagnuolo. Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv.*, 40(4), 2008.

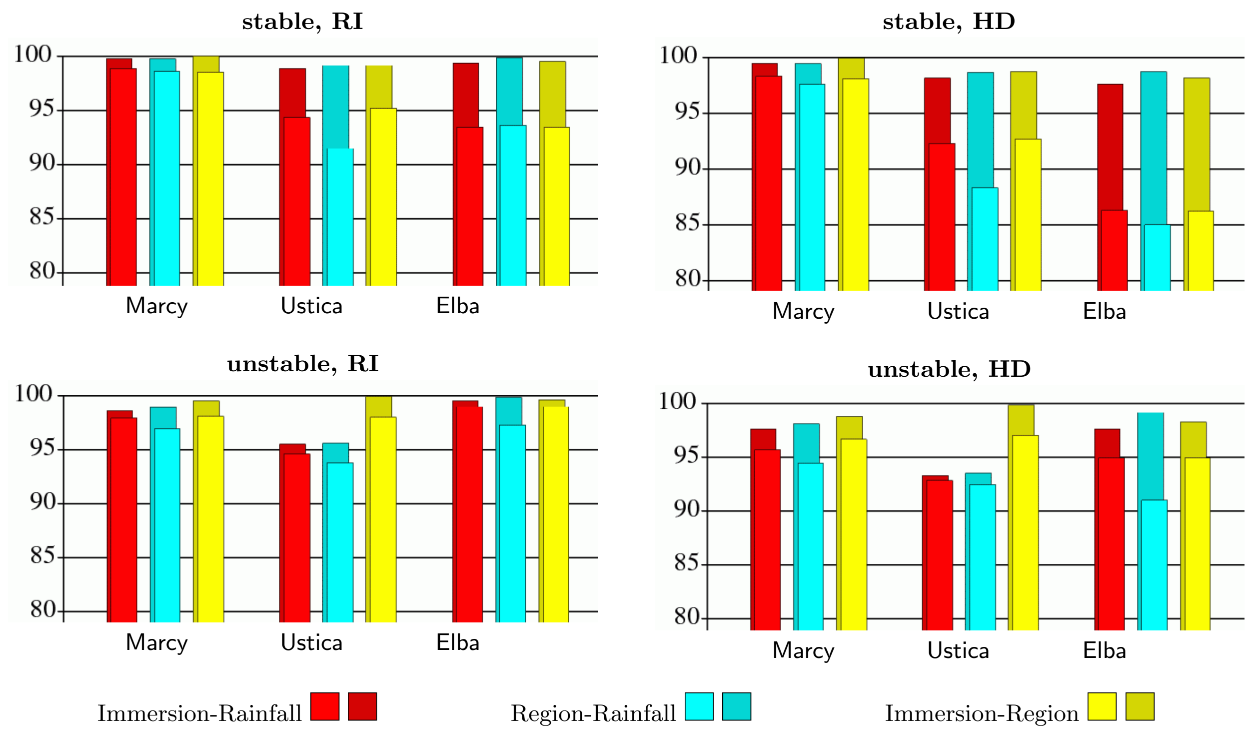


Table 5: Values of Rand Index (RI) and Hamming Distance (HD) metrics on pairs of segmentations computed on TINs with flat edges (lighter colors) and without flat edges (darker colors).

[3] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Trans. on Graphics (Proc SIGGRAPH)*, 28(3), 2009.

[4] L. Comic, L. De Floriani, and L. Papaleo. Morse-Smale decompositions for modeling terrain knowledge. In *Spatial Information Theory*, pages 426–444, 2005.

[5] E. Danovaro, L. De Floriani, P. Magillo, M. M. Mesmoudi, and E. Puppo. Morphology-driven simplification and multi-resolution modeling of terrains. In *Proc. ACM-GIS 2003*, pages 63–70, 2003.

[6] E. Danovaro, L. De Floriani, and M. Mesmoudi. Topological analysis and characterization of discrete scalar fields. *Theoretical Foundations of Computer Vision, Geometry, Morphology, and Computational Imaging, Springer Verlag, Lecture Notes on Computer Science*, 2616:386–402, 2003.

[7] M. de Berg and C. Tsirogiannis. Exact and approximate computations of watersheds on triangulated terrains. In *Proc. ACM-GIS 2011*, pages 74–83, 2011.

[8] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse complexes for piecewise linear 2-manifolds. In *Proc. 17th ACM Symposium on Computational Geometry*, pages 70–79, 2001.

[9] R. Forman. Morse theory for cell complexes. *Adv in Math*, 134:90–145, 1998.

[10] H. King, K. Knudson, and M. Neza. Generating discrete Morse functions from point data. *Experimental Mathematics*, 14(4):435–444, 2005.

[11] P. Magillo, E. Danovaro, L. De Floriani, L. Papaleo, and M. Vitali. A discrete approach to compute terrain morphology. *Computer Vision and Computer Graphics Theory and Applications*, 21:13–26, 2009.

[12] A. Mangan and R. Withaker. Surface segmentation using morphological watersheds. In *Proc. IEEE Visualization 1998, Late Breaking Hot Topics*, 1998.

[13] J. Milnor. *Morse Theory*. Princeton Univ. Press, 1963.

[14] V. Robins, P. Wood, and A. Sheppard. Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(8):1646–1658, 2011.

[15] J. Roerdink and A. Andmeijester. The watershed transform: Definitions, algorithms, and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2000.

[16] B. Schneider. Extraction of hierarchical surface networks from bilinear surface patches. *Geographical Analysis*, 37:244–263, 2005.

[17] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, Berlin and New York, 2004.

[18] S. Takahashi, T. Ikeda, T. L. Kunii, and M. Ueda. Algorithms for extracting correct critical points and constructing topological graphs from discrete geographic elevation data. *Computer Graphics Forum*, 14(3):181–192, 1995.

[19] L. Vincent and P. Soille. Watershed in digital spaces: an efficient algorithm based on immersion simulation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.