Notes on the Simplification of the Morse-Smale Complex

David Günther, Jan Reininghaus, Hans-Peter Seidel, and Tino Weinkauf

Abstract The Morse-Smale complex can be either explicitly or implicitly represented. Depending on the type of representation, the simplification of the Morse-Smale complex works differently. In the explicit representation, the Morse-Smale complex is directly simplified by explicitly reconnecting the critical points during the simplification. In the implicit representation, on the other hand, the Morse-Smale complex is given by a combinatorial gradient field. In this setting, the simplification changes the combinatorial flow, which yields an indirect simplification of the Morse-Smale complex. The topological complexity of the Morse-Smale complex is reduced in both representations. However, the simplifications generally yield different results. In this chapter, we emphasize properties of the two representations that cause these differences. We also provide a complexity analysis of the two schemes with respect to running time and memory consumption.

1 Introduction

The Morse-Smale (MS) complex [17, 21] has been proven useful in many applications due to its compact representation of the input data. However, a simplification of this complex is mandatory to determine the dominant features within the data.

Over the last few years, two concepts have been established to represent the MS-complex. The first concept goes back to Edelsbrunner et al. [4] and was originally proposed for piecewise linear input data given on a triangulated domain. In this setting, the MS-complex is explicitly represented as a graph called the 1-skeleton. The

David Günther
Institut Mines-Télécom; Télécom ParisTech; CNRS LTCI, e-mail: gunther@telecom-paristech.fr

Jan Reininghaus
IST Austria, e-mail: jan.reininghaus@ist.ac.at

Hans-Peter Seidel, and Tino Weinkauf
MPI for Informatics, e-mail: {hpseidel,weinkauf}@mpi-inf.mpg.de

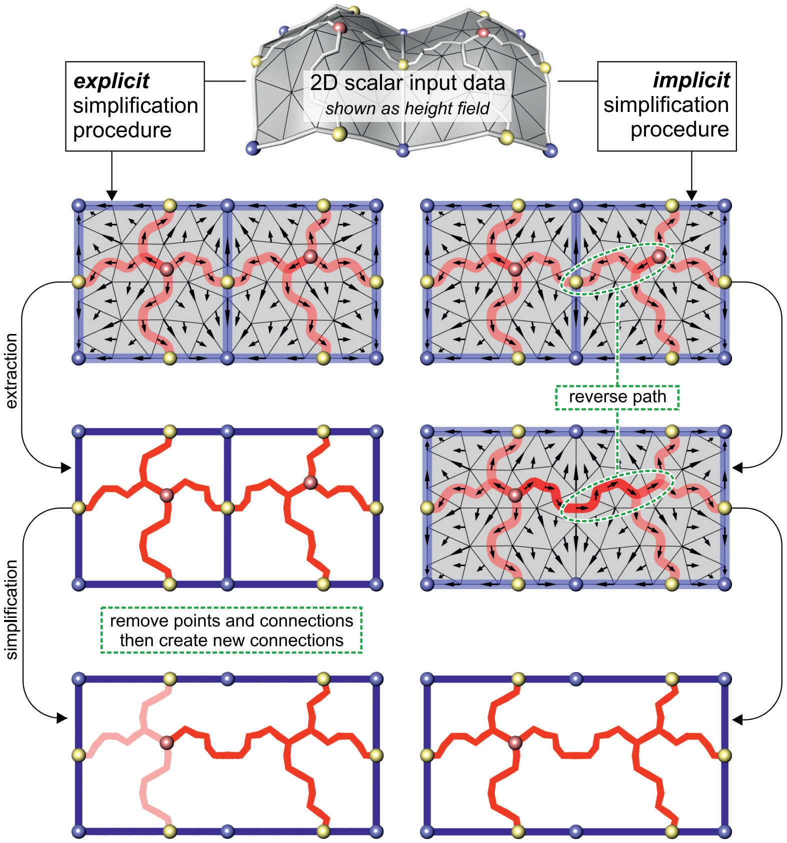

explicit simplification procedure
2D scalar input data
shown as height field
implicit simplification procedure
extraction
simplification
reverse path
simplification
extraction
remove points and connections then create new connections

Fig. 1 Illustration of the explicit and implicit simplification of the MS-complex. The scalar input data is given on a discretization. From this input, the explicit and implicit simplification procedure computes a combinatorial gradient field that represents the MS-complex (first row). The left column shows the explicit procedure. From the gradient field, the MS-complex is extracted and represented as a graph (second row). The simplification now removes pairs of critical points in the graph (third row). This involves a local change of the connectivity of the affected critical points: incident connections are deleted and new connections are added. The right column shows the implicit procedure. The MS-complex is implicitly simplified by reversing the direction of the flow along a combinatorial gradient line that connects two critical points (second row). An explicit representation of the MS-complex is computed after the gradient field was simplified (third row).

graph consists of nodes and edges – called *links* in this chapter to avoid confusion with geometric edges. The nodes are the critical points of the data and the links describe their neighborhood relation given by the separatrices. The simplification of the MS-complex works iteratively on this graph. Two critical points can be removed from the graph if they are connected by a single link. After a removal, the neighborhood information of their respective adjacent critical points needs to be updated, i.e., the nodes of the graph are newly connected.

The second concept follows a formulation proposed by Forman [6] to describe the MS-complex. In contrast to the first approach, the MS-complex is only implicitly represented by a combinatorial gradient field. A simplification is now applied to the gradient field by changing the direction of the combinatorial flow along a separatrix; see right column of Figure 1. This change in direction implicitly simplifies the MS-complex. Its explicit representation is computed from a simplified gradient field.

The motivation behind these two simplification strategies is different. While the explicit procedure refers to an explicit computation of a true simplified scalar field, the implicit case directly addresses the (combinatorial) gradient flow. However, both simplification procedures follow the same rules to determine which simplification step is admissible and they may also follow both the same principle to determine the next removal. But the results of the explicit and implicit simplification procedures differ, in general.

For 2D data sets, the difference is limited to the geometric embedding of the separation lines. For 3D data sets, on the other hand, the simplification procedures may even lead to different hierarchies: the order of the removal as well as the connectivity of critical points might differ. In this work, we investigate specific properties of the two representations and show that these properties can cause these differences. The different representations also affect the algorithmic side. Therefore, we also provide a complexity analysis of the two simplification schemes.

In Section 2, we provide a brief overview of prior work. Both simplification strategies are presented in Section 3, and we discuss specific properties in Section 4. We conclude this chapter in Section 5 by discussing the effects of these properties.

2 Related Work

In the following, we give a brief overview about the computation and simplification of the MS-complex in the discrete setting. The MS-complex [17, 21] consists of critical points (minima, saddles, maxima) and separatrices – the gradient lines that connect the critical points.

The MS-complex for 2D or 3D piecewise linear data can be computed using an approach proposed by Edelsbrunner et al. [4, 3]. The critical points are given by an analysis of the lower star of each vertex. The separatrices are approximated as a sequence of steepest edges in the triangulation. An extension to more general input was proposed by Gyulassy et al. [11] using a region-growing approach based on the definition [21] of the MS-complex. These approaches result in an explicit graph representation of the MS-complex. The nodes of this graph are the critical points and its links represent the separatrices that connect the critical points.

A simplification of this representation of the MS-complex is obtained by eliminating pairs of critical points in the graph and a subsequent update of the neighborhood relationship of adjacent critical points [4, 9]; see the left column of Figure 1.

In contrast to the above techniques, the MS-complex can also be computed and simplified using the approach proposed by Forman [6]. The complex is implicitly given by a combinatorial gradient field. The critical points represent the topological changes of the sub-level sets of the data [16]. The separatrices are computed by starting at the (combinatorial) saddle points and following the grid along the combinatorial gradient field. A simplification of the MS-complex is implicitly done by changing the combinatorial flow in the gradient field.

The first computational realization of Forman's theory was presented by Lewiner et al. [14, 15]. Robins et al. [19] presented the first algorithm to compute a combinatorial gradient field which is provably correct in up to three dimensions in the sense that its critical points correspond one-to-one to the topological changes in the sub-level sets of the input. Based on this method, Günther et al. [7] presented an optimal algorithm to compute an explicit representation of the MS-complex from a combinatorial gradient field for the 2D and 3D case. This algorithm can be run in parallel and has a complexity of $O(cn)$ with c denoting the number of critical points and n denoting the size of the input. A memory and running time efficient hierarchy of this complex is obtained by simplifying the combinatorial gradient field [7, 18] and extracting the MS-complex afterward as shown in the right column of Figure 1.

In the next section, we present the technical details of the explicit and implicit simplification. While other strategies are conceivable, we concentrate on the techniques proposed by Gyulassy et al. [9] and Günther et al. [7] as representatives for the explicit and implicit case, respectively.

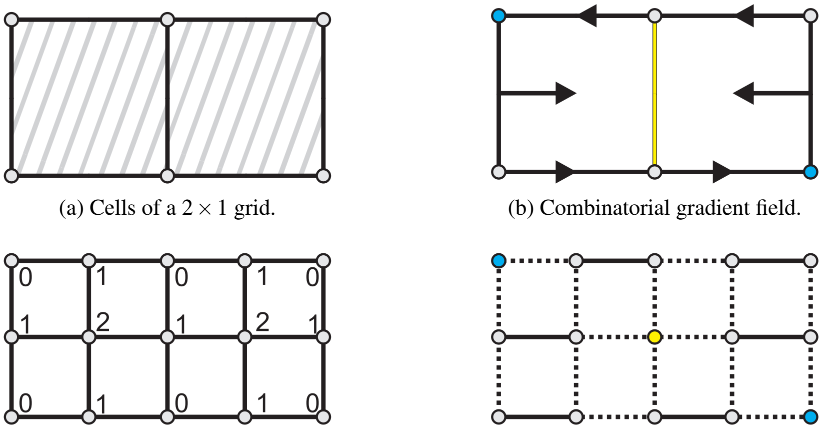
3 Simplification of the Morse-Smale Complex

In this section, we briefly present the explicit and implicit simplification scheme. The basis for both approaches is an initial combinatorial gradient field that describes the combinatorial flow of the input data. Although other settings are conceivable, we concentrate on the setting of discrete Morse theory [6].

3.1 Combinatorial Gradient Field and Morse-Smale Complex

A combinatorial gradient field represents the gradient flow of a given input but restricted to the discretization of the underlying domain. The discretization decomposes into cells of different dimensions. Assuming that the domain is given as a 2D grid as shown in Figure 2a, the cells are the k-dimensional entities of the grid: vertices, edges, quads. Similarily, higher dimensional grids or triangulations also decomposes into cells. A combinatorial gradient field now pairs cells of consecutive dimension yielding a combinatorial flow restricted to the entities of the discretization, see Figure 2b. Unpaired cells are the combinatorial analogue of the continuous critical points [6].

Recently, several approaches [9, 13, 20] to construct such a combinatorial gradient field based on an input field are proposed. The first provably correct algorithm,



(a) Cells of a 2 × 1 grid.

(b) Combinatorial gradient field.

(c) Graph representation of the grid.

(d) Flow represented by a matching.

Fig. 2 Illustration of a combinatorial flow. (a) shows the cells of 2D grid: vertices (circles), edges (black lines), quads (shaded area). (b) shows a combinatorial gradient field: cells of consecutive dimension are paired, unpaired cells (blue and yellow) are critical cells. (c) shows a graph representation of the grid shown in (a): each node (sphere) is labeled by the dimension of the cell it represents, the adjacency of the cells is represented by links (black lines). (d) shows a matching representation of the combinatorial flow shown in (b): paired cells are represented by links (solid black lines), nodes with no incident matched links are critical (blue and yellow).

however, is proposed by Robins et al. [19]. While all of the approaches work in practice, it is beneficial to use this algorithm since the topological structures in the gradient field are reduced to a minimum, i.e., all structures correspond to evolution of the sub-level sets and no falsely identified structures are present.

Given a combinatorial gradient field, an explicit representation of the MS-complex C^{MS} can be computed using the intersection of the underlying ascending and descending manifolds [21]. The MS-complex consists of the critical points and separatrices. The critical points are the unpaired cells in the combinatorial flow (Figure 2b). The dimension k of the cells defines the type of the critical point. For a 2D input, we call a critical point a minimum ($k = 0$), a saddle ($k = 1$), or a maximum ($k = 2$). In case of a 3D input, the critical points are called: minimum ($k = 0$), 1-saddle ($k = 1$), 2-saddle ($k = 2$), or maximum ($k = 3$).

The separatrices, on the other hand, are special combinatorial gradient lines that connect critical points of different type. The type of the critical points define the dimensions of the cells a separatrix is allowed to cover in the discretization. The smaller dimension of the cells defines then the type of the separatrix. For example, a separatrix that connects a saddle with a minimum only covers cells of dimension zero and one; we call it a 0-separatrix.

In contrast to their continuous counterpart, separatrices can merge or split in 2D. In 3D, a single separatrix can even merge *and* split. As we will see in the following, this property causes the differences in the simplification of the explicit and implicit representation of the MS-complex.

3.2 Explicit Simplification of the Morse-Smale Complex

In the following, we assume that an explicit representation of the initial MS-complex C_0^{MS} is given, i.e., the complex is given as a graph with nodes representing the critical points and links representing the separatrices. Topologically simplifying this graph means reducing the number of nodes and consequently the topological complexity of the MS-complex. Based on a simplification guideline, pairs of critical points are iteratively removed from C^{MS} which yields a hierarchy $\mathscr{C}$ of MS-complexes

$$\mathscr{C} = (C_k^{MS})_{k=0,\ldots,m}. \tag{1}$$

A pair of critical points is a valid candidate for a removal if it is connected by a single link. Let p and q denote two critical points of index ℓ and $\ell+1$, respectively. We denote the ℓ-neighborhood of a critical point q by N_q^ℓ, i.e., N_q^ℓ contains all critical points of index ℓ that are connected to q in the MS-complex. Let the pair (p,q) be connected by a single link, i.e., $q \in N_p^{\ell+1}$ and $p \in N_q^\ell$.

The removal of the pair (p,q) now changes the neighborhood of all their adjacent critical points. The removal in the graph consists of two operations:

1. The nodes p and q and all their incident links are deleted from the graph.
2. New connections between each node in $N_p^{\ell+1} \setminus \{q\}$ and $N_q^\ell \setminus \{p\}$ are created.

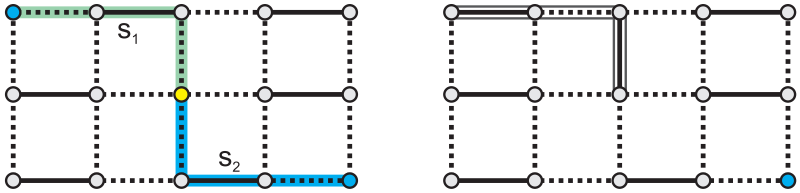
The simplification process terminates if no valid pair of critical points can be found in the MS-complex. For more details on this simplification, we refer to [9].

3.3 Implicit Simplification of the Morse-Smale Complex

In the following, we assume that an initial combinatorial gradient field V_0 is given. In contrast to the explicit representation, the implicit simplification directly addresses the combinatorial flow. Let a pair of critical points which is uniquely connected by a separatrix be given. A simplification is now done by reversing the combinatorial flow along the separatrix. Applying the simplification iteratively yields a sequence $\mathscr{V}$ of combinatorial gradient fields

$$\mathscr{V} = (V_k)_{k=0,\ldots,m} \,. \tag{2}$$

From an algorithmic point of view, the simplification can be best expressed in a graph-theoretical notation. A combinatorial gradient field works on the cells of a given discretization, i.e., the vertices, edges, triangles, quads, cubes, etc.. Each discretization can be represented as a graph $G = (N,E)$. The nodes N of the graphs are the k-dimensional cells and its links E describe the adjacency of these cells, see Figure 2c for a 2D example. The combinatorial flow is expressed as a pairing of cells. This pairing can be represented by a subset of links $V \subset E$. Since each cell can only occur in one pair [6], none of the links in V are adjacent; hence, V is a matching. Figure 2d shows a 2D example.



(a) Two separatrices in V. (b) $V \triangle s_1$.

Fig. 3 Illustration of the implicit simplification. (a) shows two separatrices (blue and green lines) that begin at a saddle (yellow circle) and end in two minima (blue circles). (b) shows the change of the combinatorial flow along s_1 (bounded region). The incident saddle and minimum of s_1 are matched and no longer critical. The symbol $\triangle$ denotes the symmetric difference.

We now interpret the combinatorial gradient field V_0 as a matching. In graph theoretical terms, an combinatorial separatrix $s \subset E$ connecting two critical points is an augmenting path since it is alternating and its start- and end-node are not matched (Figure 3a). Hence, we can produce a larger matching $V_{k+1} \subset E$ by taking the symmetric difference

$$V_{k+1} = V_k \triangle s. \tag{3}$$

Equation (3) is called *augmenting* the matching. Since the incident critical nodes of s are matched after the augmentation, the number of critical nodes is decreased by two. Note that Equation (3) does not depend on the dimension of G, i.e., we can apply the augmentation to 2D as well as 3D data.

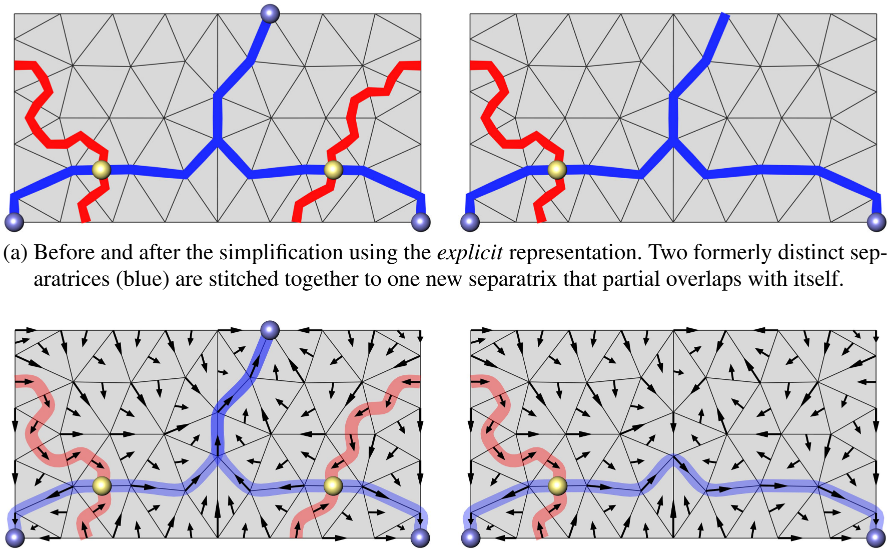
The symmetric difference in Equation (3) reverses the direction of the combinatorial gradient flow. Loosely speaking, the augmentation flips the direction of the arrows representing the combinatorial flow. An illustration of the augmentation is given in Figure 3b. The simplification stops if the matching cannot be augmented anymore, i.e., there are no critical points that are connected by a unique separatrix.

The sequence $\mathscr{V}$ is completely defined by the final matching V_m and the (typically rather short) augmenting paths (p_i). An arbitrary element $V_k \in \mathscr{V}$ can be restored by iteratively applying Equation (3) with respect to (p_i). For a more detailed description of this simplification, we refer to [7, 18][1].

4 Differences in the Simplifications of the Morse-Smale complex

In this section, we elucidate the properties of the explicit and the implicit simplification schemes. Both schemes are based on different representations of the MS-complex. The representations influence not only the geometric embedding of the separatrices but also the order of simplifications. In general, both schemes yield different hierarchies of MS-complexes.

[1] Note that the idea of reversing the flow along a separation line was also used to modify a scalar field based on a simplified contour tree [23].



(a) Before and after the simplification using the *explicit* representation. Two formerly distinct separatrices (blue) are stitched together to one new separatrix that partial overlaps with itself.

(b) Before and after the simplification using the *implicit* representation. Reversing the flow between the two canceled critical points leads to a new embedding, which does not overlap itself.

Fig. 4 The explicit and implicit simplification may create different embeddings of the separatrices.

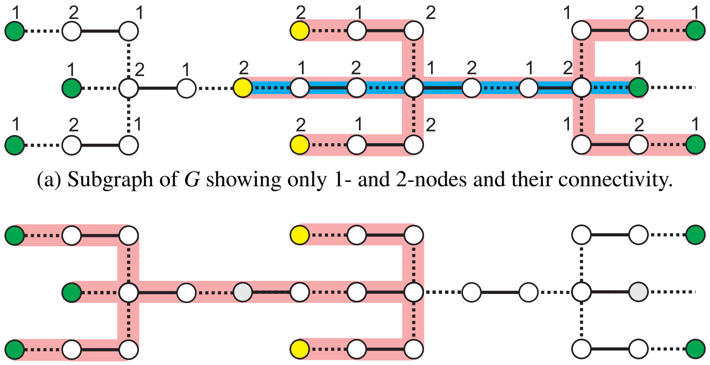
4.1 Geometric Embedding of Separatrices

The geometric embedding of the separatrices in the initial MS-complex coincides in the explicit and implicit representation. They are defined by the combinatorial gradient field, and their embedding is given by the cells of the discretization which they cover, see Figure 4 (left). However, differences occur if the complex is simplified.

In the explicit representation, critical points are newly connected after a simplification. Those new connections involve a merge of separatrices. Figure 4a shows an example. The separatrices (blue lines) emerging at the two saddles (yellow circles) meet each other at a non-critical point. Both separatrices share the same cells from this point on to the central minimum (blue circle). If the saddle and the central minimum are now removed from the MS-complex, the two separatrices merge, which yields a partly overlap of the separatrix with itself.

In the implicit representation, on the other hand, the situation is different, see Figure 4b. The removal of the right saddle and the central minimum is done by changing the combinatorial flow along the separatrix connecting these two points. The *new* separatrix that emanates from the left saddle directly goes to the right minimum without any self-overlapping.

This situation also occurs in 3D. However, it only affects the geometric embedding of the separatrices and not the simplification process. In the following, we discuss two properties in the explicit and implicit representation which can yield different hierarchies.



(a) Subgraph of *G* showing only 1- and 2-nodes and their connectivity.

(b) Changed connectivity.

Fig. 5 Illustration of the connectivity change due to an implicit saddle-saddle simplification (3D). Shown is a subgraph of *G* connecting 2-saddles (yellow) and 1-saddles (green). The links of the matching are depicted as black solid lines. The blue line in (a) depicts an augmenting path. After the augmentation, the connectivity of the saddles (red lines) completely changed (b).

4.2 Connectivity of Critical Points

The following situation only occurs when a single separatrix can merge *and* split. Hence, we assume 3D data as input and concentrate on a saddle-saddle simplification, i.e., the simplification of an 1- and 2-saddle and the involved 1-separatrices. For explanatory reasons, we use the above graph-theoretical notation in the following. This allows us to embed the saddle-saddle connections in the plane, see Figure 5.

A simplification of a saddle-saddle pair can completely change the connectivity of the adjacent saddles in the implicit representation. Figure 5 depicts such a situation. It may happen that the 1-separatrices of multiple 2-saddles merge and share several links before they split again and end in 1-saddles, see Figure 5a. The shared links describe in some sense a narrow one-way street. All 1-separatrices on the left side of this one-way street must cross it to enter the right side.

However, simplifying the gradient field along one of the 1-separatrices (the blue line in Figure 5a) changes the connectivity of the saddles. Before the simplification, all 2-saddles (yellow) were connected to the 1-saddles (green) on the right side. Only the central 2-saddle was also connected to the 1-saddles on the left side. After the simplification, none of the remaining 2-saddles is connected to the right 1-saddles anymore, see Figure 5b. There is no 1-separation line connecting them. All of them are now connected to the 1-saddles on the left side, which was not the case before.

In the explicit representation, on the other hand, the above situation does not occur. Each saddle-saddle pair and its connectivity is separately stored. Although the geometric embedding of the connections may partly coincide, each connection is independently treated. A simplification only removes the current pair and its connections from the MS-complex, and new connections between all neighboring critical points are created [9]. Therefore, the green 1-saddles on the right are still connected to the two remaining 2-saddles, and they form valid candidates for further simplification in the explicit representation.

Since the neighborhood of a saddle changes differently, the above situation causes different hierarchies in the explicit and implicit representation, in general. However, the differences are caused by the fact that 1-separation lines partly overlap in the discrete setting. Given an initial combinatorial gradient field with non-overlapping separation lines, both schemes would yield the same result.

4.3 Iterative Simplification

A common way to guide the simplification is the use of the height difference of adjacent critical points [9, 14, 7]. This heuristic assumes that unimportant and noise-induced critical points differ only slightly in their scalar value. Applying an iterative simplification based on this heuristic yields a hierarchy of MS-complexes. The initial MS-complex contains all fine-grained structures in the data. The last level, on the other hand, only contains the large-scale structures of the data.

Pairs of critical points are iteratively removed from the MS-complex such that each pair represents the currently smallest height difference of all possible pairs. This height difference guided removal assigns an importance value to each critical points allowing to distinguish spurious and dominant critical points.

This importance measure is closely related to the concept of persistent homology [5]. In 2D, it was shown that persistent homology can also be used to simplify the MS-complex [4]. The *adjacency lemma* guarantees that two critical points are connected by a unique separatrix at the moment when they should be canceled.

Recently, it was also shown that the pairing generated by the height difference produces the same pairing as by persistent homology [2] for data given on a smooth 2D manifold. However, this is no longer true in 3D. It was shown [1] that there are pairs of critical points generated by persistent homology that can not be obtained by a sequential removal of critical points as described in Section 3.2 and 3.3. Therefore, the assessment of importance of critical points based on the height difference differs from persistence, in general.

4.4 Monotonicity of the Simplification

As discussed in Section 4.3, the height difference is typically used to guide the simplification assuming that it represents the currently smallest fluctuation. Therefore, a natural assumption is that the height difference ω is monotonically increasing over the simplification process. While this is the case in 2D for the explicit/implicit representation and also for the explicit representation in 3D, it is no longer monotone in the 3D implicit simplification. This stems from the fact that 1-separatices in 3D can merge *and* split. Figure 6 shows such an example together with the combinatorial

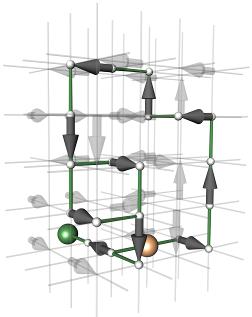
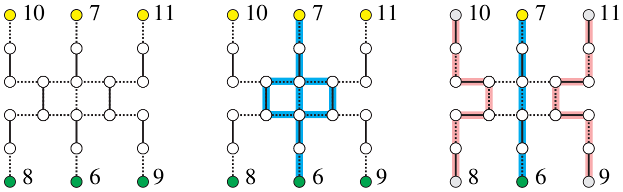


Fig. 6 Illustration of the 3D flow (grey arrows) along a split-merge of a 1-separation line (green).



(a) Subgraph of G containing only 1- and 2-nodes.

(b) Multiple paths (blue) between two saddles.

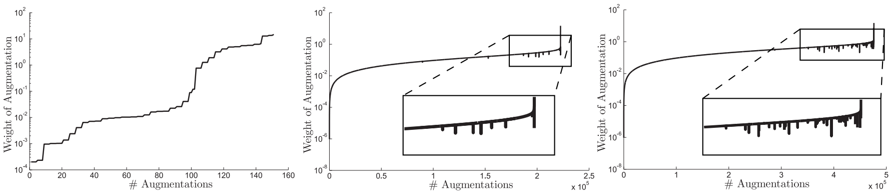
(c) Unique path (blue) after several augmentations (red).

Fig. 7 Sketch of a combinatorial gradient field in a subgraph of G (3D) connecting 2-saddles (yellow) and 1-saddles (green). The critical nodes are labeled with their assigned scalar value. The links of the matching are depicted as black solid lines.

flow. In the following, we concentrate on saddle-saddle simplifications in an implicit representation and investigate this non-monotonic behavior in detail. For simplicity, we use the above graph-theoretical notation in the following.

Due to the degree of the 1- and 2-nodes, the 1-separation lines can merge *and* split. Consider a subgraph with different saddle-saddle pairs as shown in Figure 7a. Assume that the central pair represents the smallest height difference ($\omega = 1$) and that this pair is next in line for a simplification. However, this pair cannot be removed since there are three paths connecting them, as shown in Figure 7b. Since the saddles could still be canceled with their adjacent extrema, they are deferred with a weight defined by these extrema. In the next steps, other saddle pairs are removed with a greater weight ($\omega = 2$), see Figure 7c. Parts of the corresponding augmenting paths, however, share links with the paths connecting the central pair. Due to the augmentations, the orientation of the links is changed, and the central pair is suddenly uniquely connected. If the central yellow saddle is now next in line, its connectivity is recomputed and the central green saddle is determined as the partner with the smallest height difference ($\omega = 1$). Since the connection between them is now unique, the path allows for an augmentation. However, the weight of the augmentation is smaller than in the prior operations.

In practice, the breaks of monotony in the weight sequence are caused by noise. The perturbations introduced by it yield to short split-merge sequences in the 1-separation lines as depicted in Figure 7b. To analyze this behavior, we sampled the artificial function g defined in [8] on an 128^3 grid and added two levels of uniform noise to it. Figure 8 shows the results. The 1-separation lines in the pure function g are well distributed. No deferring of saddles in the above sense can be observed. The weights of the augmenting paths are monotonically increasing, see Figure 8a. However, adding a small amount of noise in the range of $[-0.5, 0.5]$ results in 12 monotony breaks, see Figure 8b. If we add noise in the range of $[-1, 1]$, the number of breaks increases further to 59, see Figure 8c. This indicates that the level of noise heavily influences the number of splits in the 1-separation lines.



(a) g (b) Small amount of noise. (c) Large amount of noise.

Fig. 8 Graph of the weights of the augmenting paths. (a) shows the weights of augmentations for the artificial function g [8] over the number of augmentations. The weights behave monotonically increasing. The monotony is broken in (b) where a small amount of noise is added to g. The number of monotony breaks further increases if the level of noise is increased (c).

In contrast, the height difference is monotonically increasing in the explicit representation. A single simplification only considers the critical points. It does not consider if the connection between a pair partially overlaps with other separation lines (since all connections are treated independently). Hence, a removal of a critical point pair does not affect other connections. A saddle pair which is connected by multiple separation lines can never be removed from the MS-complex, in contrast to the implicit representation. The height difference during the simplification process is therefore monotonically increasing in the explicit representation.

4.5 Computational Complexity and Memory Consumption

In the following, we give a brief discussion about the computational complexity and the memory consumption of the explicit and implicit simplification in case of a 3D input. We denote the number of vertices by n and the number of saddles by m. Note that in the worst-case scenario there holds $m \approx n$.

We begin with the explicit representation. Let (p,q) be a critical point pair with indices ℓ and $\ell+1$, respectively. Let N_q^ℓ be the set of adjacent critical points of index ℓ in the graph representation. A removal of (p,q) deletes all incident links of p and q and creates new links between the nodes $N_q^\ell \setminus \{p\}$ and $N_p^{\ell+1} \setminus \{q\}$. Using efficient data structures with constant *insert* and *delete* operations [9], the complexity of a removal is therefore $O(|N_q^\ell| \times |N_p^{\ell+1}|)$. Since a saddle can be connected to m other saddles [22], the worst-case complexity of a single simplification step is $O(m^2)$. Hence, the overall complexity of performing all simplifications is $O(m^3)$.

The adjacency of the critical points is explicitly stored. Each critical point contains a list with its neighboring critical points. Since a saddle can be connected to m saddles, the total memory consumption is $O(m^2)$ in the explicit scheme.

In the implicit representation, the connectivity of a critical point needs to be recomputed since previous simplification steps may have affected its incident separatrices. The neighborhood of a critical point can be computed with a complexity of

$O(n)$ using a restricted breadth-first search [7]. Since the simplification procedure is guided by an importance measure such as the height difference of critical points, it needs to be checked if the current pair represents the *best* pair. In the worst case, this check involves the comparison to m other saddles. Hence, a single simplification step has a worst-case complexity of $O(nm)$. The overall complexity of performing all simplifications is therefore $O(nm^2)$.

The adjacency of the critical points is implicitly represented by the combinatorial gradient field. The representation of such a field can be realized by a Boolean vector representing all cells of the underlying discretization. The size of this vector is therefore given by the size of the input. In contrast to the explicit representation, the memory consumption is of order $O(n)$.

5 Discussions and Conclusions

We investigated the properties of the explicit and implicit simplification procedures and showed that the corresponding results may be different in terms of geometry and topology. In particular, the simplification procedures generally lead to different hierarchies of MS-complexes in 3D data sets. Nevertheless, it should be stressed that both simplification procedures yield valid MS-complexes.

Theoretically, the simplification procedures would give the same results in 3D if separation lines would not partially overlap. However, this is rarely the case for real-world data. The different results, as discussed in this chapter, are caused by saddle-saddle simplifications. Saddles correspond to tunnels and cavities of the level sets and our practical observations are that qualitative differences regarding these features are often negligible. As shown in several examples [7, 9], both simplification procedures are able to distill the essential features of the input.

Algorithmically, the main difference between the two simplification procedures is their memory consumption. In the worst case, the implicit method has a linear memory consumption while the explicit method has a quadratic memory consumption. Additional heuristics controlling the order of simplifications can mitigate this issue for the explicit case [10]. However, this also introduces a computational parameter, which depends on the input data and needs to be adjusted by the user. In the implicit case, contrarily, no computational parameter is needed.

In 2D and 3D, the geometric embedding of the separation lines can differ between the two schemes. In the explicit scheme, two lines can merge into a single partially overlapping line, i.e., cells in the underlying domain are visited multiple times. In the implicit representation cells can only be visited once allowing no overlap. This might be useful for noisy data since spurious lines are automatically removed.

We want to stress that both simplification schemes are based on the height difference heuristic. In 3D, the initial MS-complex cannot be perfectly simplified. As shown by Joswig and Pfetch [12] and illustrated by Bauer [1], this is an NP-complete task. In practice, this results in the situation that some critical points – which represent spurious features – cannot be removed.

In the both schemes critical points are always removed in pairs. However, it would be interesting how clusters of critical points could be consistently canceled. Such a removal might also allow for the removal of spurious critical points in 3D. However, this needs to be done without creating closed cominatorial streamlines.

Acknowledgements. This research is supported and funded by the Digiteo *unTopoVis* project, the *TOPOSYS* project FP7-ICT-318493-STREP, and MPC-VCC.

References

1. U. Bauer. *Persistence in discrete Morse Theory*. PhD thesis, University of Göttingen, 2011.
2. T. Dey, K. Li, C. Luo, P. Ranjan, I. Safa, and Y. Wang. Persistent heat signature for pose-oblivious matching of incomplete models. *CGF*, 29(5):1545–1554, 2010.
3. H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *19th annual Proc. SoCG*, pages 361–370, NY, USA, 2003. ACM.
4. H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse complexes for piecewise linear 2-manifolds. *Discrete Computational Geometry*, 30:87–107, 2003.
5. H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Computational Geometry*, 28:511–533, 2002.
6. R. Forman. A user's guide to discrete Morse theory. In *Proc. of the 2001 Int. Conf. on Formal Power Series and Algebraic Combinatorics*, Advances in Applied Mathematics, 2001.
7. D. Günther. *Topological Analysis of Discrete Scalar Data*. PhD thesis, Saarland University, Saarbrücken, Germany, 2012.
8. D. Günther, J. Reininghaus, H. Wagner, and I. Hotz. Efficient computation of 3D Morse-Smale complexes and persistent homology using discrete Morse theory. *The Visual Computer*, 28:959–969, 2012.
9. A. Gyulassy. *Combinatorial Construction of Morse-Smale Complexes for Data Analysis and Visualization.* PhD thesis, University of California, Davis, 2008.
10. A. Gyulassy, P.-T. Bremer, V. Pascucci, and B. Hamann. Practical considerations in Morse-Smale complex computation. In *Proc. TopoInVis*, pages 67–78. Springer, 2011.
11. A. Gyulassy, V. Natarajan, V. Pascucci, and B. Hamann. Efficient computation of Morse-Smale complexes for three-dimensional scalar functions. *TVCG*, 13:1440–1447, 2007.
12. M. Joswig and M. E. Pfetsch. Computing optimal Morse matchings. *SIAM Journal on Discrete Mathematics*, 20(1):11–25, 2006.
13. H. King, K. Knudson, and N. Mramor. Generating discrete Morse functions from point data. *Experimental Mathematics*, 14(4):435–444, 2005.
14. T. Lewiner. *Geometric discrete Morse complexes*. PhD thesis, PUC-Rio, 2005.
15. T. Lewiner, H. Lopes, and G. Tavares. Optimal discrete Morse functions for 2-manifolds. *Computational Geometry*, 26(3):221 – 233, 2003.
16. J. Milnor. *Morse Theory*. Princeton University Press, 1963.
17. M. Morse. *The Calculus of Variations in the Large*. Number v. 18 in Coll. Publ. - AMS. 1934.
18. J. Reininghaus. *Computational Discrete Morse Theory*. PhD thesis, Freie Universität, 2012.
19. V. Robins, P. J. Wood, and A. P. Sheppard. Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE PAMI*, 33(8):1646–1658, 2011.
20. N. Shivashankar and V. Natarajan. Parallel computation of 3D Morse-Smale complexes. *Computer Graphics Forum*, 31(3pt1):965–974, 2012.
21. S. Smale. On gradient dynamical systems. *The Annals of Mathematics*, 74:199–206, 1961.
22. H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. On the applicability of topological methods for complex flow data. In *Proc. TopoInVis*, pages 105–120. Springer, 2007.
23. G. Weber, S. Dillard, H. Carr, V. Pascucci, and B. Hamann. Topology-controlled volume rendering. *Visualization and Computer Graphics, IEEE Transactions on*, 13(2):330–341, 2007.