

A primal/dual representation for discrete Morse complexes on tetrahedral meshes

Kenneth Weiss¹ and Federico Iuricich² and Riccardo Fellegara² and Leila De Floriani²

¹Lawrence Livermore National Laboratory, United States

²Department of Computer Science, University of Genova, Italy.

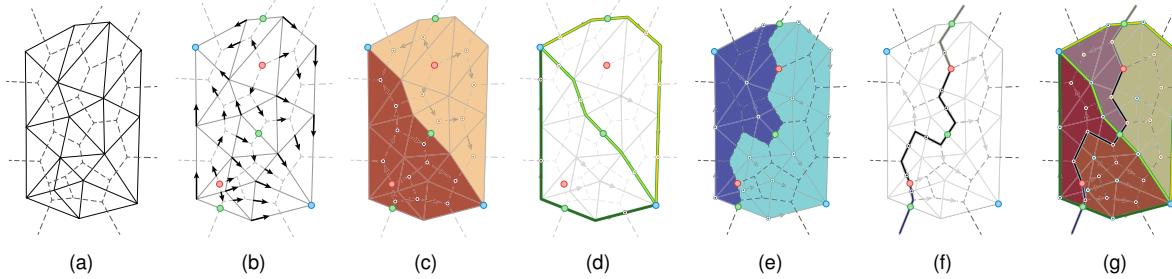


Figure 1: Overview our scheme for tetrahedral meshes (illustrated in 2D). (a) We interpret the Morse complex of a simplicial mesh in terms of the primal mesh Σ (solid lines) and its dual Σ_d (dashed lines). (b) Encoding the Discrete Morse gradient field entirely with the tetrahedra enables the use of compact topological data structures for morphological extraction. We associate the descending Morse complexes with the cells of Σ (c-d), the ascending Morse complexes with the cells of Σ_d (e-f) and the Morse-Smale complex with the dually subdivided tetrahedral mesh Σ_S (g), whose hexahedral cells are defined by a tetrahedron and one of its vertices. All relations are encoded strictly in terms of the vertices and tetrahedra of Σ .

Abstract

We consider the problem of computing discrete Morse and Morse-Smale complexes on an unstructured tetrahedral mesh discretizing the domain of a 3D scalar field. We use a duality argument to define the cells of the descending Morse complex in terms of the supplied (primal) tetrahedral mesh and those of the ascending complex in terms of its dual mesh. The Morse-Smale complex is then described combinatorially as collections of cells from the intersection of the primal and dual meshes. We introduce a simple compact encoding for discrete vector fields attached to the mesh tetrahedra that is suitable for combination with any topological data structure encoding just the vertices and tetrahedra of the mesh. We demonstrate the effectiveness and scalability of our approach over large unstructured tetrahedral meshes by developing algorithms for computing the discrete gradient field and for extracting the cells of the Morse and Morse-Smale complexes. We compare implementations of our approach on an adjacency-based topological data structure and on the PR-star octree, a compact spatio-topological data structure.

1. Introduction

Topological methods rooted in Morse theory [Mil63] have gained increasing importance in the analysis and visualization of scalar fields, due to their ability to extract essential morphological features through decompositions of the field domain into regions of influence of the critical points of the

field, called *Morse* and *Morse-Smale (MS) complexes*. However, Morse theory applies to smooth functions, while in practical applications we often deal with scalar fields that are regularly or irregularly sampled at discrete locations within a domain. Forman [For98] has developed a discrete analogue of Morse theory for cell complexes. Since this approach is com-

pletely combinatorial, it avoids computing derivatives and is beneficial in the presence of noise in the data. However, despite the applicability of Forman theory to very general discretized domains, memory constraints on the encoding of the discrete Morse gradient field have limited its practical use to scalar fields defined over regular square and cubic grids.

Our work concentrates on computing discrete Morse and Morse-Smale complexes on unstructured tetrahedral meshes discretizing the domain of a 3D scalar field. Since Forman's discrete gradient field is defined on all the simplices of the mesh, a natural representation for the mesh would be the *In-cidence graph (IG)* [Ede87] (implicitly encoded for regular grids [GBHP08, GSW12, GRWH12]). When encoding unstructured tetrahedral meshes, the IG can be verbose, since it explicitly encodes all vertices, edges, faces and tetrahedra in the mesh plus several topological connectivity relations. On the contrary, data structures which encode only the vertices and the tetrahedra [PBCF93, GR09] have been shown to be much more compact [CDW11]. Here, we introduce a compact encoding for discrete vector fields which is only based on the tetrahedra, and is therefore suitable for combination with any such mesh representation. Moreover, we propose an implementation for the PR-star octree [WFDV11], which derives the local connectivity of the mesh through a spatial index on the mesh geometry. Thus, by trading a reasonable amount of computation at runtime, all connectivity relations can be compactly encoded. A benefit of this approach is that the spatial index provides a means of understanding the spatial embedding of the mesh and its associated fields.

Another contribution of our work is our expression of the primal/dual relationship between the *descending* and *ascending* Morse complexes in terms of the supplied simplicial mesh Σ , and its dual mesh Σ_d . Furthermore, we express the combinatorial structure of the MS complex (crystals in 3D, quads in 2D) as collection of cells in the mesh obtained by the intersection of Σ and Σ_d , which we refer to as the *dually subdivided* mesh Σ_S . This leads to a compact encoding of these complexes in terms of only the vertices and tetrahedra of the primal mesh. In this way, we can efficiently express morphological structures of the scalar field, such as the regions of influence of critical points (i.e. the maxima, minima, and saddles), the arcs of the extrema graphs (the graphs connecting the maxima with 2-saddles and the minima with 1-saddles [CLB11]) and the 1-skeleton of the Morse-Smale complex (the graph connecting all critical points). See [GKK*12] for a definition of these features in terms of the Morse and MS complexes.

We extend the algorithm of Robins et al. [RWS11] for discrete Morse gradient computation to simplicial meshes with irregular connectivity and we implement it both on the PR-star octree and on a compact tetrahedron-based topological data structure. We develop algorithms for computing the cells of the Morse complexes, and compare the performance of these data structures.

2. Background notions

Let f be a C^2 real-valued function (scalar field) defined over a d -dimensional manifold \mathbb{M} . A point $p \in \mathbb{M}$ is a *critical point* of f if the gradient vanishes at p . Function f is said to be a *Morse function* if the Hessian matrix $H_p f$ of the second derivatives of f at a critical point p is non-singular. The number of negative eigenvalues of $H_p f$ is called the *index* of critical point p . For $d = 3$, there are four types of critical points, namely minima, 1-saddles, 2-saddles and maxima.

An *integral line* of a function f is a maximal path that is everywhere tangent to its gradient. Integral lines that converge to a critical point p of index i cover an i -cell, called the *descending manifold* of p . Dually, integral lines that originate at p cover a $(d-i)$ -cell, called the *ascending manifold* of p . The descending manifolds decompose \mathbb{M} into a cell complex, called the *descending Morse complex* of f on \mathbb{M} , while the ascending manifolds form the *ascending Morse complex* of f on \mathbb{M} . A Morse function f is a *Morse-Smale function* if intersecting descending and ascending manifolds are transversal. The connected components of the intersection of descending and ascending cells of a Morse-Smale function f decompose \mathbb{M} into a *Morse-Smale (MS) complex*.

The discrete Morse theory due to Forman [For98] is an elegant adaptation of classical Morse theory to functions defined over a cell complex in which most of the main results from Morse theory are valid. This goal is achieved by considering a function F defined on all cells of the complex (i.e. not only on its vertices). Here, we define it for a simplicial mesh Σ , i.e. for a cell complex in which all the cells are simplices. A function F is a discrete Morse function if for any p -simplex σ , all its facets (the $(p-1)$ -simplices on its boundary) have a lower F value and all its cofacets (the $(p+1)$ -simplices in its co-boundary) have a higher F value, with at most one exception. If there is such an exception, it defines a pairing of cells called a *discrete gradient vector*. Otherwise, p -simplex σ is a *critical simplex* of index p . The collection of such discrete gradient vectors is referred to as the *discrete Morse gradient field*, which we denote as V .

As noted by Forman, it is simpler to define a discrete Morse gradient vector field than a discrete Morse function. Intuitively, a discrete vector field can be viewed as a collection of *arrows*, connecting a p -simplex of mesh Σ to an incident $(p+1)$ -simplex in such a way that each simplex is a *head*, or a *tail* of at most one arrow and the critical simplices are neither the head nor the tail of any arrow. A discrete vector field V is a *discrete gradient field* if there are no closed V -paths in V , where a V -path is a sequence $\sigma_0, \tau_0, \sigma_1, \tau_1, \dots, \sigma_{r+1}$ of p -simplices σ_i and $(p+1)$ -simplices τ_j , such that σ_i and σ_{i+1} are distinct facets of τ_i , and (σ_i, τ_i) are paired in V .

For each discrete Morse function F , a discrete gradient vector field V can be constructed by pairing any p -simplex σ and $(p+1)$ -simplex τ in its co-boundary (denoted $\tau \succ \sigma$) whenever $F(\tau) \leq F(\sigma)$.

3. Related work

There have been a few approaches to extend the results of Morse theory and to represent Morse and Morse-Smale complexes in the discrete case. One is based on Banchoff's extension [Ban70] of Morse theory to piecewise-linear manifolds and functions [Ban70, EHZ01, EHNP03]. A survey on algorithms based on this approach and on the watershed approach can be found in [BDF^{*}08].

Recently, a lot of attention has been devoted to algorithms rooted in discrete Morse theory [For98] (see Section 2), leading to the formulation of robust discrete algorithms [CCL03, LLT04, GBHP08, GSW12, KKN05, RWS11, SN12, SSN12] for computing Morse and MS complexes. Algorithms based on discrete theory have been developed for regular grids [RWS11, GBHP08] or for triangle and tetrahedral meshes of limited size [KKN05]. Parallel algorithms for computing 2D and 3D Morse-Smale complexes for large 2D and 3D structured meshes are presented in [SN12, SSN12]. In [GSW12, GRWH12], a memory-efficient implementation of [RWS11] is presented and applied to 3D scalar fields defined over regular grids for efficient persistent homology computation and as the basis for extending the estimation of persistence from critical points to extremal lines and surfaces. Algorithms that produce more accurate geometry and, thus, correct connectivity for the MS complex are discussed in [GBP12], also in the context of regular grids. The relationship between the piecewise linear and combinatorial approaches to discrete Morse theory is discussed in [Lew12].

A variety of hierarchical spatial indexes have been proposed for points, polygonal maps, boundary representations of objects and triangle meshes [Sam06]. Such indexes contain the geometrical objects only in their leaf nodes and, thus, the shape of the tree is independent of the order in which the points are inserted. In [DFM10], a family of spatial indexes for tetrahedral meshes, called *tetrahedral trees*, has been introduced which generalizes similar data structures for maps and triangle meshes to 3D. A fundamentally and conceptually different data structure for tetrahedral meshes, called the *PR-star octree* [WFDV11], uses the spatial embedding of the mesh to index its topological connectivity (see Section 7). This data structure was used to extract morphological features from terrain and volume datasets in [DFIW12] using a streaming implementation of Robins et al.'s gradient computation. They used a primal-only approach to extract a subset of the features and did not propose a compact encoding for the gradient vector field.

4. Encoding the discrete Morse gradient vector field

We introduce a new encoding for discrete vector fields defined over irregular tetrahedral meshes in which information is attached only to the tetrahedra. Thus, it is well suited to be used in connection with any topological data structure which explicitly encodes only the vertices and the tetrahedra of the

mesh [PBCF93, GR09, WFDV11]. We use this encoding as a compact representation for the discrete Morse gradient field of a discrete Morse function defined over the mesh.

Let us consider a tetrahedral mesh Σ and a tetrahedron τ in Σ . Our encoding associates with τ a subset of the discrete vector pairs involving its faces. Specifically, it encodes all vector pairs (σ_i, σ_j) where σ_i and σ_j are both faces of τ , as well as the pairs (σ_i, τ') from a triangular facet σ_i of τ to its adjacent tetrahedron τ' along σ_i in Σ .

Recall that a tetrahedron τ has $\binom{4}{i+1}$ faces of dimension i , and each face has $(i+1)$ facets. Since each edge, triangle or tetrahedron of τ can be paired with any of its facets, there are

$$\sum_{i=1}^4 \binom{4}{i+1} \cdot (i+1) = 6 \cdot 2 + 4 \cdot 3 + 1 \cdot 4 = 28$$

possible discrete vector pairs in the restriction of vector field V to τ (see Figure 3a). Adding the four additional vector pairs from a facet σ of τ to an adjacent tetrahedron gives a total of 32 possible discrete vector field pairings. We refer to this collection of pairs from the discrete vector field V in the vicinity of tetrahedron τ as a *local frame* of the discrete vector field. Since each such pairing within a local frame encodes a single bit of information (i.e. the presence or absence of that particular pairing), we can encode each local frame using 32 bit flags per tetrahedron. This bit flag representation simplifies testing for the presence of vector pairings as well as updates to the discrete vector field.

We next observe that the restrictions imposed by discrete vector fields (i.e. that each simplex can be involved in at most one pairing) imply that there are significantly fewer valid local frame configurations than the 2^{32} possibilities provided by the bit flag representation. In fact, for the 28 possible interior pairings, there are only 14,721 valid configurations, and for the full local frame containing 32 possible pairings there are only 51,030 valid cases. Thus we can encode a *compressed* local frame using only two bytes per tetrahedron in the mesh. To facilitate easy conversion between the two representations, we encode two small auxiliary lookup tables: an array EXPANDFRAME[.] of 51,030 bit flag entries (encoded using four bytes) and a map COMPRESSFRAME[.] from the 51,030 valid bit flags to the compressed local frame representation (encoded as unsigned short).

Note, that the local frame representation is redundant. Each pairing involving a triangle face is encoded twice (i.e. once for each tetrahedra in its co-boundary) and the pairings involving a vertex and an edge e are encoded within the local frame of each tetrahedron in the star of e . However, we exploit the fixed size of the simplex boundary relations to achieve our compact representation.

5. Primal/dual representations

In this section, we present an interpretation of the Morse and Morse-Smale (MS) complexes in terms of the provided tetrahedral mesh and its dual mesh. We use this interpretation to

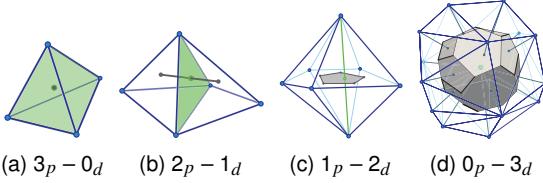


Figure 2: The primal/dual relationships in a tetrahedral mesh. Each dual cell i_d (gray) is an i -polytope contained within the star (blue) of its corresponding primal k -simplex k_p (green), where $i+k=3$.

define an efficient encoding for these complexes and their individual cells. We denote the *primal* tetrahedral mesh as Σ . The *dual* mesh of Σ , which we denote as Σ_d , is a polyhedral mesh in which the 0-cells (vertices) correspond to the tetrahedra of Σ (Figure 2a), the 1-cells (edges) correspond to the triangles of Σ (Figure 2b), the 2-cells to the edges of Σ (Figure 2c) and the 3-cells to the vertices of Σ (Figure 2d).

5.1. Representing ascending and descending manifolds

Using the above correspondences, we observe that the k -saddles of the Morse function correspond to k -simplices in the primal mesh Σ and to $(d-k)$ -cells in the dual mesh Σ_d . See Figure 1b for an example in 2D, where minima (blue) are associated with vertices of the triangle mesh, and hence, 2-cells of the dual mesh; saddles (green) are associated with edges of the mesh, and hence edges of the dual mesh; and maxima (red) are associated with triangles of the mesh, and hence, vertices of the dual mesh. Therefore, the descending Morse complex consists of elements from the primal mesh, while the ascending Morse complex consists of elements from the dual mesh. Specifically:

- A *descending 3-manifold* corresponds to a maximum, and thus to a collection of (primal) tetrahedra. Dually, an *ascending 3-manifold* corresponds to a minimum, and thus to a collection of dual 3-cells (i.e., primal vertices).
- A *descending 2-manifold* corresponds to a 2-saddle, and thus to a collection of primal triangles, each of which can be expressed as a pair of primal tetrahedra (see Figure 2b). An *ascending 2-manifold* corresponds to a 1-saddle, and thus to a collection of dual 2-cells, each of which can be expressed as a pair of dual 3-cells, corresponding to a primal edge (see Figure 2c).
- A *descending 1-manifold* corresponds to a 1-saddle and thus to a sequence of primal edges, or, equivalently, as a sequence of primal vertices (see Figure 2c). An *ascending 1-manifold* corresponds to a 2-saddle and thus to a sequence of dual edges, which can be seen as a pair of dual vertices (i.e., as a sequence of primal tetrahedra).

Figure 1(c-f) illustrates the above observations for a discrete Morse function defined on a triangle mesh, where the

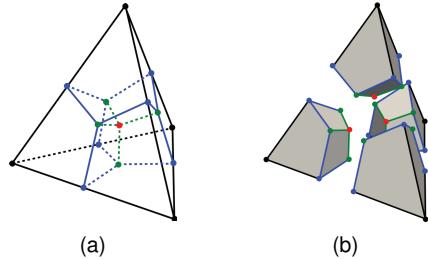


Figure 3: A dually-subdivided tetrahedron τ (a) is decomposed into four hexahedra (b), each defined by a vertex of τ (black) and interior points from each incident face within τ .

descending 2-manifolds (Figure 1c) are collections of triangles from Σ associated with the maxima (red critical points), while the ascending 2-manifolds (Figure 1e) are collections of dual 2-cells (corresponding to vertices from Σ) associated with the minima (blue critical points). Similarly, the descending (Figure 1d) and ascending 1-manifolds (Figure 1f) correspond to collections of primal and dual edges, respectively, associated with the saddles (green critical points).

Note that all the descending and ascending manifolds are expressed entirely in terms of tetrahedra and vertices. This is a relevant issue from an implementation point of view, since there is no need to encode the primal edges and triangles, and of course no need for encoding the dual mesh.

This encoding can be generalized to arbitrary dimension d , where each manifold of the Morse complex can be described in terms of collections of cells of uniform dimension from the primal or dual mesh, each of which can be expressed using at most $\lceil (d+1)/2 \rceil$ vertices or d -simplices.

5.2. Representing the cells of the MS complex

We define the *dually subdivided mesh*, denoted as Σ_S , as the mesh obtained by the intersection of the primal mesh Σ and its dual Σ_d (see Figure 1a for a 2D example). Each 3-cell of Σ_S is the intersection of a tetrahedron τ and of a dual 3-cell (corresponding to a vertex). This defines a hexahedron, which we refer to as a *micro-hex*, whose boundary consists of six quadrilateral *micro-quads* and twelve *micro-edges*. Thus, a tetrahedron τ in Σ is decomposed into four hexahedra in Σ_S , each corresponding to a vertex of τ (see Figure 3).

Similarly, a triangle σ in primal mesh Σ is decomposed into three micro-quads in Σ_S , namely those micro-quads corresponding to the three vertices of σ in the two tetrahedra of Σ sharing σ . An edge e in Σ is the union of two micro-edges in Σ_S belonging to the micro-hexes forming the tetrahedra incident in e and corresponding to its endpoints.

A micro-hex, being the intersection of a tetrahedron τ and a 3-cell in the dual mesh (a vertex v in the primal mesh), is

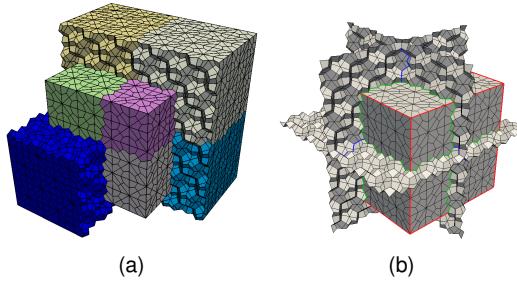


Figure 4: Geometrical representation of a Morse-Smale complex computed on a synthetic dataset. (a) Filtered view of the (macro) MS complex 3-cells (unique colors) composed of a set of micro-hexahedra. (b) The MS 2-cells bounded by the MS 1-skeleton. Each 2-cell is composed of a set of micro-quads and is bounded by 1-cells (saddle connectors).

encoded as a pair (τ, v) . A micro-quad γ in Σ_S separates two hexahedral cells, which can either share a tetrahedron τ or a vertex v , depending on whether they are part of the same primal tetrahedron or the same dual 3-cell (corresponding to v). This leads to an encoding for γ either as a triple (τ, v_1, v_2) , where v_1 and v_2 are the vertices of τ defining the two hexahedral cells (which are pairs (τ, v_1) and (τ, v_2)), or as a triple (τ_1, τ_2, v) , where τ_1 and τ_2 are the two tetrahedra defining the two hexahedral cells (which are pairs (τ_1, v) and (τ_2, v)).

The (macro) cells of the MS complex consist of elements from the dually subdivided mesh Σ_S . A 3-cell in the MS complex is a collection of micro-hexes. A 2-cell α of the MS complex is a collection of micro-quads on the boundary of the two macro 3-cells of the MS complex sharing α . Figure 4 illustrates the geometrical representation of an MS complex on a synthetic dataset.

The 1-cells of the MS complex are of the following three types and are all sequences of *micro-edges* in Σ_S .

Minimum-1-saddle connector: Each micro-edge in the sequence connects a primal vertex and edge. Thus, the connector is formed by primal edges, and can be encoded as a sequence of primal vertices, where the last two vertices define the critical edge. In Figure 3 the black edges connect primal vertices (black dots) to primal edges (blue dots).

Maximum-2-saddle connector: Each micro-edge in the sequence connects a primal tetrahedron and face (i.e. a dual vertex and edge). Thus, the connector is formed by a sequence of dual vertices and is encoded as a sequence of primal tetrahedra. In Figure 3, the green edges connect primal tetrahedra (red dot) to primal faces (green dots).

Saddle-connector: Each micro-edge in the sequence connects a primal triangle and edge (or, dually, a dual edge and 2-cell). In Figure 3, the blue edges connect primal edges (blue dots) to primal faces (green dots). We encode it as a sequence of primal triangles, whose first triangle is critical and whose last two intersect in the critical edge.

Note that a saddle connector is not a subset of 1-cells of the ascending or descending Morse complexes.

Figure 1g concludes our 2D example by illustrating the Morse-Smale complex associated with the discrete Morse gradient field of Figure 1b. Note that each micro-quad is defined by the intersection of a triangle (a primal 2-cell) and a dual 2-cell associated with one of its boundary vertices, and that each (macro) 2-cell of the MS complex is defined by a maximum (red critical point), a minimum (blue critical point) and two saddles (green critical points).

6. Extracting morphological features

In this section, we discuss how to retrieve the cells of the Morse complexes (i.e. the descending and ascending manifolds), the cells of the MS complex, and the combinatorial structure of the 1-skeleton of the MS complex from a tetrahedral mesh Σ endowed with a local discrete gradient field, encoded as in Section 4. These structures are the basis for computing morphological features of 3D scalar fields [GKK^{*}12]. We discuss how to extract the descending and ascending manifolds based on the simplices of the primal mesh Σ and the topological relations involved. Generally speaking, a descending or ascending i -manifold is extracted by traversing the primal/dual mesh following the pairings of the gradient field, and starting from the i -simplex corresponding to the critical point associated with the i -manifold. The resulting manifolds are encoded as discussed in Section 5.

A *descending 3-manifold* for a maximum p_{max} is computed by starting from the critical tetrahedron τ corresponding to p_{max} and following all paired simplices of V from the boundary triangles of τ into its adjacent tetrahedra. It is encoded as a label attached to the tetrahedra forming it. Dually, an *ascending 3-manifold* for a minimum p_{min} is computed by starting from the critical primal vertex v (corresponding to a dual 3-cell) and following all paired simplices of V from the co-boundary edges of v to its adjacent vertices in Σ (i.e. it traverses into adjacent dual 3-cells paired in V). It is encoded as a label attached to the primal vertices forming it.

A *descending 2-manifold* for a 2-saddle p_{2s} is computed by starting from the critical triangle σ corresponding to p_{2s} and following all paired simplices in V from boundary edges of σ to its adjacent triangles. Dually, an *ascending 2-manifold* for a 1-saddle p_{1s} is computed by starting from the critical primal edge ε corresponding to p_{1s} (corresponding to a dual 2-cell), and following all gradient arrows from the co-boundary triangles of ε to their boundary edges (i.e. it traverses along the adjacencies of dual 2-cells).

A *descending 1-manifold* for a 1-saddle p_{1s} is computed using the same relations as for ascending 3-manifolds, and similarly for *ascending 1-manifolds* and descending 3-manifolds. In both cases, duality reverses the roles of boundary and co-boundary.

Note that for all traversals, we can view the discrete gradient field as a directed acyclic graph (DAG), whose root is the critical i -cell, whose nodes are i -cells in Σ or Σ_d , and whose arcs are the subset of the adjacent i -cells from the mesh paired in V along boundary facets. Since triangles in the primal mesh are in the co-boundary of at most two tetrahedra, and edges have at most two boundary vertices, their DAGs are degenerate (i.e. they are rooted trees) and no additional bookkeeping is required. However, for the 2-manifold extractions, visited triangles and edges need to be marked as we traverse the arcs of the DAG.

A 3-cell of the MS complex corresponds to a pair (p_{\max}, p_{\min}) of critical points and is encoded as a collection of micro-hexes in the dually subdivided mesh Σ_S obtained by intersecting the descending 3-cell (for p_{\max}) and the ascending 3-cell (for p_{\min}), which are collections of tetrahedra and primal vertices, respectively. It is computed by collecting all pairs (τ, v) , where τ is labeled with p_{\max} and v with p_{\min} .

A 2-cell in the MS complex corresponds to a saddle pair (p_{2s}, p_{1s}) and it is a collection of micro-quads. For each pair of face-adjacent micro-hexes, both containing p_{2s} and p_{1s} on their boundary, the common micro-quad is part of the MS 2-cell if the labels of the two hexahedra are different.

The MS 1-cells corresponding to a maximum-2-saddle, or a minimum 1-saddle, are the ascending and descending 1-manifolds, respectively. A saddle connector, which connects a 1-saddle p_{1s} and a 2-saddle p_{2s} , is computed by extracting the descending and ascending 2-manifolds associated with p_{1s} and p_{2s} . The descending 2-manifold extraction is performed first, and all the traversed triangles are marked as visited. We then start from the critical primal edge ε corresponding to p_{1s} and its adjacent edges. Among the triangles shared by ε and its adjacent edges, only the triangles previously marked as visited are considered. We then apply the same process as for extracting ascending 2-manifolds.

The combinatorial structure of the MS complex is a graph in which the nodes correspond to the critical points (simplices in Σ) and the arcs to the adjacencies of these points on the 1-skeleton of the MS complex. We observe that this is also the incidence graph representation of either Morse complex. Thus, the extraction is performed by computing all the manifolds in one of the two complexes, for instance, the descending complex, saving one node for each critical simplex and connecting two nodes in the graph with an arc if there is a path in the discrete gradient connecting the two corresponding critical simplices.

7. PR-star octree implementation

In contrast to topological data structures, which explicitly encode the connectivity among mesh elements, or to spatial indexes, which index the elements for efficient location-based queries, a PR-star octree [WFDV11] uses the spatial index induced by an octree to efficiently generate local application-dependent topological data structures at runtime. Here, we

briefly describe the PR-star octree and discuss the extraction of the morphological features discussed in Section 6 on the PR-star octree. The computation of the discrete Morse gradient on the PR-star octree is performed by extending to simplicial meshes encoded in a PR-star octree the algorithm in [RWS11], which requires the computation of the tetrahedra incident in each vertex, a particularly efficient operation on the PR-star octree.

The PR-star octree is based on the *Point Region octree* (*PR octree*) [Sam06], a spatial index on a set of irregularly distributed points P . The domain decomposition is controlled by a single parameter, that we denote as k_v , which determines the maximum number of points indexed by a leaf node. The insertion of a new point into a *full* leaf in the tree causes the leaf to split and its indexed points to be redistributed among its children. Thus, the domain decomposition induced by a PR octree is independent of the insertion order of its points.

The *PR-star octree* for a tetrahedral mesh Σ encodes the vertices and the tetrahedra of Σ and consists of (a) an array P of vertices, encoding the geometry of Σ ; (b) an array T of indexed tetrahedra, where each element is encoded in terms of the indices of its four vertices within P ; and (c) an augmented PR octree N , whose leaf nodes index a subset of vertices from P , as well as all tetrahedra in T incident in these vertices.

We use a more compact representation for the leaves of the PR-star octree compared with the one in [WFDV11], by exploiting the spatial locality provided by the octree through a reindexing of arrays P and T . Besides the hierarchical information associated with the octree (e.g. pointers to the parent node and to the set of children nodes), each leaf node N_l encodes: the range of indices v_{start} and v_{end} in P of the vertices contained in N_l ; the range of indices t_{start} and t_{end} in T of the tetrahedra that are completely contained in N_l ; and a pointer to the list of the remaining tetrahedra from T incident in these vertices. i.e. each such tetrahedron has at least one vertex inside and outside the domain of N_l .

The basic paradigm for performing operations on a mesh encoded as a PR-star octree is to locally process the mesh in a streaming manner by iterating through the leaf nodes of the octree. For each leaf node, a local application-dependent data structure, which we refer to as an *expanded leaf node* is generated and used to process the local geometry. After we finish processing a leaf node, we discard this local data structure and move on to the next leaf node. For efficiency, we use an auxiliary cache LRU-CACHE based on a least-recent-used replacement policy that maintains a subset of expanded leaf nodes. Since connectivity relations are reconstructed within leaf nodes, the PR-star is ideally suited for situations in which the geometry will be processed in batches. In such cases, the connectivity reconstruction costs can be amortized over multiple mesh processing operations and a more verbose application-dependent local data structure can be utilized.

For morphological feature extraction, we locally reconstruct a subset of the topological connectivity relations among the elements of the simplicial mesh Σ . Since the gradient information is associated with tetrahedra, we encode all expanded relations in terms of tetrahedra in the co-boundary:

- Descending 3-manifolds and ascending 1-manifolds utilize the *Tetrahedron-Tetrahedron (TT)* adjacency relation. For the PR-star, we expand the *Face-Tetrahedron (FT)* relations for all triangles inside the leaf node, i.e. the two tetrahedra sharing a given triangle. We use the convention that the first tetrahedron associated with a triangle σ in the FT relation is the one paired with σ in V (or σ is critical).
- Ascending 3-manifolds and descending 1-manifolds utilize the *Vertex-Vertex (VV)* relation. Instead, we expand the *Vertex-Tetrahedron* relation, i.e. the set of tetrahedra incident in each vertex inside the leaf node and use the convention that, for a vertex v , the first such tetrahedron contains the edge paired with v (or v is critical).
- Descending 2-manifolds utilize the *Face-Face (FF)* relation. We encode the *Edge-Tetrahedron (ET)* relation and use the convention that, for an edge e , the first encoded tetrahedron contains a face paired with e (or e is critical).
- Ascending 2-manifolds utilize the *Edge-Edge(EE)* relation. We expand the ET relation for all edges e in the leaf node, with the convention that the first tetrahedron in the list contains the triangle paired with e .

Following the PR-star conventions, we extract all ascending or descending i -manifolds, with a fixed i , during a traversal of the octree. When computing an i -manifold, we traverse the mesh inside a leaf node and expand the current path only within the current leaf node and the cached leaf nodes. When we need to process simplices that are outside the current and cached nodes, we interrupt the current path, which we refer to as a *dangling path*. The information needed to complete the path, e.g., the indices within P or T of the path to continue, are saved in an auxiliary data structure.

8. Experimental results

We have evaluated the performance of our discrete gradient encoding and morphological feature extraction algorithms through implementations of several data structures: the *Indexed data structure with Adjacencies (IA)* [PBCF93], an optimized version of the IA that encodes additional information about the edges (IA_{ET}) and the PR-star octree. We present experiments on five tetrahedral meshes whose sizes vary from 6 to 30 million tetrahedra. The semi-regular meshes (BONSAI, VISMALE, FOOT) were extracted from a regular grid using Regular Simplex Bisection [WD11] and irregularized through a half edge collapse-based simplification process that removed approximately 15% of the vertices. We also simplified about 10% of the vertices of irregular dataset SAN FERNANDO to remove ‘flat’ regions (i.e. regions with very low persistence) from the mesh, yielding a more meaningful feature extraction.

The IA data structure explicitly encodes, for each tetrahedron, the indices of its four vertices and of its four face-adjacent tetrahedra (the *TT relation*), as well as the index of one incident tetrahedron per vertex. Since each vertex v must reference a single (arbitrary) incident tetrahedron, we use the convention that the encoded tetrahedron either contains the edge that is paired with v in V , or v is critical. We find this tetrahedron during our gradient vector field generation. This optimization accelerates the descending 1-manifold and the ascending 3-manifold extraction steps without any impact on storage cost. We also noticed room for optimization in the descending 2-manifold extraction by using a similar trick to encode the Edge-Face gradient relation. That is, for each edge e paired with a face, we encode a single tetrahedron whose gradient contains the face pointed to by e . We refer to the IA data structure with this optimization as IA_{ET} .

We first compare the IA data structure enhanced with the gradient encoding of Section 4 with the Incidence Graph (IG), which is the natural choice for encoding cell complexes endowed with a discrete Morse gradient field. In addition to encoding all simplices of the input tetrahedral mesh, the IG encodes, for each i -simplex σ , the indices of all its facets and cofacets. The discrete gradient field can then be directly encoded with one additional bit for each node and arc of the IG. Encoding the IG with gradient attached to all simplices requires about four times as much space as the IA with gradient attached to the tetrahedra. For our largest datasets, the storage cost for the IG are almost 6 GB.

Next, we evaluate both the storage costs of the underlying data structures (IA data structure and PR-star octree) as well as the storage costs and time requirements for computing the gradient on each of them (see Table 1). We use the same encoding for the gradient field on the IA data structures and on the PR-star octree, as discussed in Section 4. For the sake of brevity, we report only a single value of k_v for each dataset. In general, increasing the value of k_v reduces the overall storage requirements but increases its local storage requirements and connectivity reconstruction times.

8.1. Gradient field generation

Our implementations of the gradient field computation on the data structures share the same encoding but differ in their connectivity reconstruction algorithms. In all cases, we have adapted the algorithm by Robins et al. [RWS11] from regular grids to simplicial meshes. Since the PR-star octree efficiently reconstructs the local connectivity for the entire submesh indexed by a leaf node, rather than for each individual vertex (as in the IA) it is able to compute the gradient field in about half the time (see column GRADIENT in Table 1).

For storage comparisons, we considered the *topological (connectivity) overhead* of the data structures. The IA requires about the same amount of information for connectivity as it does for the base mesh (P and T arrays), the IA_{ET}

Table 1: Absolute and relative timings (in seconds) and storage costs (expressed in MBs) for the implementations based on the PR-star octree and the two version based on the IA (with and without the ET* relation explicitly stored). The first two datasets are irregular tetrahedral meshes, while the final three are irregularized tetrahedral meshes derived from regular grids.

Data set	$ T $	k_v	Storage						Timing					
			Mesh	Connectivity		Max total		Gradient		Descending		Ascending		Total
				tot	%	tot	%	tot	%	tot	%	tot	%	tot
F16	6.35M	IA	115	—	—	231	—	206.85	—	56.64	—	133.88	—	397.37
		IA_{ET}	114	189	164	305	132	212.13	102	23.45	40	133.88	100	369.47
		800	38	33	178	77	100.67	49	313.30	549	591.92	442	1005.89	253
SAN FERNANDO	12.4M	IA	225	—	—	448	—	358.02	—	6.90	—	260.89	—	625.81
		IA_{ET}	223	370	164	593	134	362.97	101	2.48	29	260.89	100	626.34
		800	68	30	329	73	186.27	52	89.85	1286	728.69	279	1004.81	161
BONSAI	24.4M	IA	445	—	—	823	—	732.76	—	75.23	—	147.61	—	955.60
		IA_{ET}	437	723	162	1101	134	748.19	102	15.96	21	147.61	100	911.76
		800	130	29	577	70	370.31	50	309.33	412	353.97	240	1033.61	108
VISMAL	26.5M	IA	484	—	—	959	—	796.68	—	113.75	—	217.87	—	1128.30
		IA_{ET}	475	786	162	1261	131	809.64	102	22.19	19	217.87	100	1049.70
		800	141	29	725	76	400.85	50	288.44	253	355.59	163	1044.88	92
FOOT	29.5M	IA	541	—	—	1068	—	868.29	—	138.78	—	201.60	—	1208.67
		IA_{ET}	527	875	162	1402	131	892.89	103	27.15	19	201.60	100	1121.64
		800	164	30	691	65	454.52	52	699.86	504	395.69	196	1550.07	128

requires about 1.6 times as much space, and the PR-star requires less than a third of the storage. (see column CONNECTIVITY in Table 1). In terms of overall storage requirements, which includes the base mesh, gradient, topological overhead and auxiliary data structures, the PR-star requires about 30% less storage space than the IA, while the IA_{ET} requires about 30% more space.

8.2. Feature extraction

All data structures require a small amount of additional memory to perform feature extraction. The IA requires a global queue to perform the graph traversal of the gradient field, while the PR-star utilizes a cache of octree nodes with expanded connectivity information, as well as a list of *dangling paths* for each visited leaf node. We have experimented with different sizes for the PR-star’s LRU-CACHE in an attempt to balance the size of cache with the overhead of reconstructing the connectivity within each expanded octree node. We found that for the datasets derived from regular grids, a fixed cache size of 100 yields optimal results, while for the irregular datasets, optimal performance was achieved with a cache of around 300 expanded nodes. We believe that this is due to the less regular access patterns induced by the features within the irregular datasets. For both data structures, this additional storage was negligible (0.01%–0.1% the size of the mesh).

As can be seen in Table 1, the IA data structures perform best on descending manifold extraction, where the relevant topological connectivity relations are explicitly encoded. Furthermore, the ET* optimization in IA_{ET} accelerates the 2-manifold extractions, reducing the overall extraction times for descending manifolds to 20%–40% that of IA. In contrast,

since the PR-star needs to explicitly reconstruct these relations, it can take several times as long to extract the descending manifolds (see column DESCENDING). The timings are significantly closer for the ascending manifold extractions (see column ASCENDING), where the connectivity relations need to be extracted for all data structures. Overall, the PR-star is the smallest data structure, but requires additional time to reconstruct the connectivity of the mesh at runtime.

Figure 5 illustrates features extracted from the BUCKY dataset, including the 3-cells of the MS complex, the intersection of ascending and descending cells of the Morse complexes, the 1-skeleton of the Morse-Smale complex and its combinatorial structure. Note that many arcs of the extracted 1-skeleton are shared (Figure 5a), while they are explicit in the combinatorial representation (Figure 5d) (compare the lower left corners).

9. Concluding remarks

We have presented a primal/dual interpretation of the descending and ascending Morse manifolds in terms of simplices of the primal tetrahedral mesh associated with a 3D scalar field and of the cells of its dual mesh. The MS complex was described combinatorially as collections of cells from the dually subdivided mesh. This lead to simple descriptions of morphological features in terms of only the vertices and tetrahedra of the primal mesh.

We have also proposed a compact encoding of discrete vector fields using the *local frame* representation, which associates information with the tetrahedra in the primal mesh, and which we apply to the discrete Morse gradient field. A

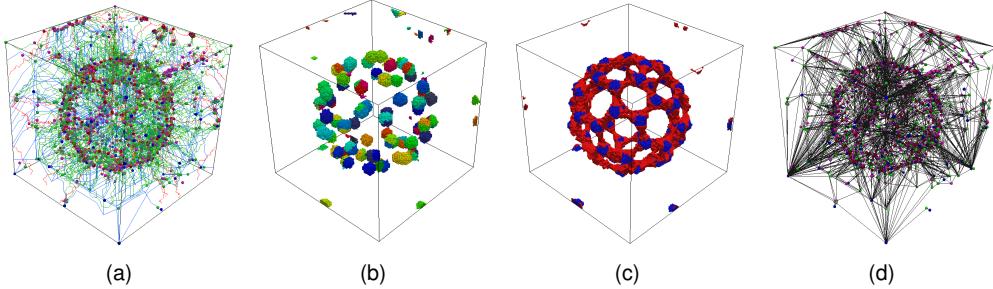


Figure 5: Example of features extracted from the BUCKY dataset. (a) The Morse-Smale 1-manifolds: maxima–2-saddle connectors (red), 2-saddle–1-saddle connectors (green) and 1-saddle–minima connectors (blue). (b) The Morse-Smale 3-cells, thresholded by region sizes to highlight the larger 3-cells decomposing the inner spheres. (c) The intersection of 3-cells from the ascending (blue) and descending (red) Morse complexes, filtered to highlight 3-cells decomposing the inner spheres. (d) The graph representing the combinatorial structure of the MS complex.

further analysis of this representation through the lens of our primal/dual interpretation reveals that the discrete vector field is a labeling of the 1-skeleton of the dually-subdivided mesh Σ_S . In this interpretation, there is a one-to-one correspondence between the nodes of Σ_S 's 1-skeleton and the simplices of Σ , and between the edges of Σ_S 's 1-skeleton and the incidence relations of Σ whose dimensions differ by one (i.e. the relations encoded in the Incidence Graph). The discrete vector field restrictions discussed in Section 2 imply that regular nodes in the 1-skeleton have valence one and critical nodes have valence zero. This interpretation opens the possibility of extending our representation to more general cell complexes, such as irregular hexahedral meshes.

The proposed gradient encoding on the IA data structure is a very compact alternative to the IG, which is the common encoding of cell complexes endowed with Forman gradient. We have shown that an even greater savings has been achieved by using the PR-star octree, at the expense of some additional computation. Alternatively, by explicitly encoding more relations, as in the IA_{ET} , we can achieve faster extraction times.

Both our vector field encoding and our formulation of morphological features are independent of the topological data structure used to encode the mesh (as long as it encodes the vertices and the tetrahedra), of the method in which the gradient field is computed and of the algorithms used for feature extraction. Thus, we anticipate our approach benefiting from theoretical advances in data structures on the one hand and in computational topology on the other.

Due to the high degree of independence of the individual calculations for computing the gradient field and for feature extraction, we are currently looking into parallel and out-of-core extensions to our data structures and algorithms. A preliminary OpenMP extension of our IA implementation on our 3.2 GHz Intel i7 quad-core testing machine achieved a 2–3x speedup compared to our single core results presented

in Table 1. For the PR-star, we are looking into using the implicit ordering on the vertices (as discussed in Section 7), to enable processing the mesh in parallel.

We are currently implementing an algorithm for persistence-based simplification, which is based on the simplification operators in [ČD11], expressed in terms of gradient field simplification. Persistence-based simplification will allow us to extract morphological features, such as the various manifolds, the extrema graphs (which are formed by the ascending and descending 1-manifolds) and the 1-skeleton of the MS complex at different levels of persistence, as in [GKK^{*}12]. It can also remove the need for the preprocessing mesh simplification step (as discussed in Section 8).

In our future work, we plan to use a modified PR-star octree to encode the extracted i -manifolds, or the MS cells, which would allow us to reconstruct the topological connectivity of the various complexes as well as to efficiently perform spatial queries on them. Our approach can also be extended to time-varying datasets defined on simplicial meshes, and to tetrahedral shapes in 4D space (such as isosurfaces of time-varying fields), since the PR-star octree, the gradient encoding and the feature extraction are all dimension independent.

Acknowledgments

We thank the reviewers for their many helpful comments and suggestions. 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. It has also been performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-JRNL-608773).

References

- [Ban70] BANCHOFF T.: Critical points and curvature for embedded polyhedral surfaces. *American Mathematical Monthly* 77, 5 (1970), 475–485. 3
- [BDF^{*}08] BIASOTTI S., DE FLORIANI L., FALCIDIENO B., FROSINI P., GIORGI D., LANDI C., PAPALEO L., SPAGNUOLO M.: Describing shapes by geometrical-topological properties of real functions. *ACM Computing Surveys* 40, 4 (October 2008), 12:1–12:87. doi:10.1145/1391729.1391731. 3
- [CCL03] CAZALS F., CHAZAL F., LEWINER T.: Molecular shape analysis based upon the Morse-Smale complex and the Connolly function. In *Proceedings Symposium on Computational Geometry* (New York, USA, 2003), ACM Press, pp. 351–360. doi:10.1145/777792.777845. 3
- [ČD11] ČOMIĆ L., DE FLORIANI L.: Dimension-independent simplification and refinement of Morse complexes. *Graphical Models* 73, 5 (2011), 261–285. doi:10.1016/j.gmod.2011.05.001. 9
- [CDW11] CANINO D., DE FLORIANI L., WEISS K.: IA*: An adjacency-based representation for non-manifold simplicial shapes in arbitrary dimensions. *Computers & Graphics* 35, 3 (2011), 747–753. doi:10.1016/j.cag.2011.03.009. 2
- [CLB11] CORREA C., LINDSTROM P., BREMER P.-T.: Topological spines: A structure-preserving visual representation of scalar fields. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (December 2011), 1842–1851. doi:10.1109/TVCG.2011.244. 2
- [DFIW12] DE FLORIANI L., FELLEGARA R., IURICICH F., WEISS K.: A spatial approach to morphological feature extraction from irregularly sampled scalar fields. In *Proceedings ACM SIGSPATIAL International Workshop on GeoStreaming* (November 2012), IWGS ’12, pp. 40–47. doi:10.1145/2442968.2442974. 3
- [DFM10] DE FLORIANI L., FELLEGARA R., MAGILLO P.: Spatial indexing on tetrahedral meshes. In *Proceedings ACM SIGSPATIAL GIS* (2010), GIS ’10, ACM, pp. 506–509. doi:10.1145/1869790.1869873. 3
- [Ede87] EDELSBRUNNER H.: *Algorithms in Combinatorial Geometry*. Springer Verlag, Berlin, 1987. 2
- [EHN⁰³] EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Morse-Smale complexes for piecewise linear 3-manifolds. In *Proceedings Symposium on Computational Geometry* (2003), ACM, pp. 361–370. doi:10.1145/777792.777846. 3
- [EHZ01] EDELSBRUNNER H., HARER J., ZOMORODIAN A.: Hierarchical Morse complexes for piecewise linear 2-manifolds. In *Proceedings Symposium on Computational Geometry* (2001), ACM, pp. 70–79. doi:10.1145/378583.378626. 3
- [For98] FORMAN R.: Morse theory for cell complexes. *Advances in Mathematics* 134 (1998), 90–145. 1, 2, 3
- [GBHP08] GYULASSY A., BREMER P. T., HAMANN B., PASCUCCI V.: A practical approach to Morse-Smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1619–1626. doi:10.1109/TVCG.2008.110. 2, 3
- [GBP12] GYULASSY A., BREMER P.-T., PASCUCCI V.: Computing Morse-Smale complexes with accurate geometry. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2014–2022. doi:10.1109/TVCG.2012.209. 3
- [GKK^{*}12] GYULASSY A., KOTAVA N., KIM M., HANSEN C., HAGEN H., PASCUCCI V.: Direct feature visualization using Morse-Smale complexes. *IEEE Transactions on Visualization and Computer Graphics* 18, 9 (2012), 1549–1562. doi:10.1109/TVCG.2011.272. 2, 5, 9
- [GR09] GURUNG T., ROSSIGNAC J.: SOT: Compact representation for tetrahedral meshes. In *Proceedings ACM Symposium on Solid and Physical Modeling* (2009), ACM Press, pp. 79–88. doi:10.1145/1629255.1629266. 2, 3
- [GRWH12] GÜNTHER D., REININGHAUS J., WAGNER H., HOTZ I.: Efficient computation of 3D Morse-Smale complexes and persistent homology using discrete Morse theory. *The Visual Computer* 28, 10 (2012), 959–969. doi:10.1007/s00371-012-0726-8. 2, 3
- [GSW12] GÜNTHER D., SEIDEL H.-P., WEINKAUF T.: Extraction of dominant extremal structures in volumetric data using separatrix persistence. *Computer Graphics Forum* 31, 8 (2012), 2554–2566. doi:10.1111/j.1467-8659.2012.03222.x. 2, 3
- [KKN05] KING H. C., KNUDSON K., NEZA M.: Generating discrete Morse functions from point data. *Experimental Mathematics* 14, 4 (2005), 435–444. doi:10.1080/10586458.2005.10128941. 3
- [Lew12] LEWINER T.: Critical sets in discrete Morse theories: Relating Forman and piecewise-linear approaches. *Computer Aided Geometric Design*, 0 (2012), -. doi:10.1016/j.cagd.2012.03.012. 3
- [LLT04] LEWINER T., LOPEZ H., TAVARES G.: Applications of Forman’s discrete Morse theory to topology visualization and mesh compression. *IEEE Transactions on Visualization and Computer Graphics* 10, 5 (September 2004), 499–508. doi:10.1109/TVCG.2004.18. 3
- [Mil63] MILNOR J.: *Morse Theory*. Princeton University Press, New Jersey, 1963. 1
- [PBCF93] PAOLUZZI A., BERNARDINI F., CATTANI C., FERRUCCI V.: Dimension-independent modeling with simplicial complexes. *ACM Transactions on Graphics* 12, 1 (January 1993), 56–102. doi:10.1145/169728.169719. 2, 3, 7
- [RWS11] ROBINS V., WOOD P., SHEPPARD A.: Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8 (August 2011), 1646–1658. doi:10.1109/TPAMI.2011.95. 2, 3, 6, 7
- [Sam06] SAMET H.: *Foundations of Multidimensional and Metric Data Structures*. The Morgan Kaufmann series in computer graphics and geometric modeling. Morgan Kaufmann, 2006. 3, 6
- [SN12] SHIVASHANKAR N., NATARAJAN V.: Parallel computation of 3D Morse-Smale complexes. *Computer Graphics Forum* 31, 3 (2012), 965–974. doi:10.1111/j.1467-8659.2012.03089.x. 3
- [SSN12] SHIVASHANKAR N., SENTHILNATHAN M., NATARAJAN V.: Parallel computation of 2D Morse-Smale complexes. *IEEE Transactions on Visualization and Computer Graphics* 18, 10 (October 2012), 1757–1770. doi:10.1109/TVCG.2011.284. 3
- [WD11] WEISS K., DE FLORIANI L.: Simplex and diamond hierarchies: Models and applications. *Computer Graphics Forum* 30, 8 (2011), 2127–2155. doi:10.1111/j.1467-8659.2011.01853.x. 7
- [WFDV11] WEISS K., FELLEGARA R., DE FLORIANI L., VELLOSO M.: The PR-star octree: A spatio-topological data structure for tetrahedral meshes. In *Proceedings ACM SIGSPATIAL GIS* (November 2011), GIS ’11, ACM, pp. 92–101. doi:10.1145/2093973.2093987. 2, 3, 6