Building Morphological Representations for 2D and 3D Scalar Fields

Lidija Ćomić¹, Leila De Floriani² and Federico Iuricich²

¹Faculty of Engineering, University of Novi Sad (Serbia)
²Department of Computer Science, University of Genova (Italy)

Abstract

Ascending and descending Morse complexes, defined by the critical points and integral lines of a scalar field \( f \) defined on a manifold domain \( D \), induce a subdivision of \( D \) into regions of uniform gradient flow, and thus provide a compact description of the morphology of \( f \) on \( D \). We propose a dimension-independent representation for the ascending and descending Morse complexes, and we describe a data structure which assumes a discrete representation of the field as a simplicial mesh, that we call the incidence-based data structure. We present algorithms for building such data structure for 2D and 3D scalar fields, which make use of a watershed approach to compute the cells of the Morse decompositions.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Computational Geometry and Object Modeling—Object Representations

1. Introduction

Representing morphological information extracted from discrete scalar fields is a relevant issue in several application domains, such as terrain modeling, volume data analysis and visualization, and time-varying 3D scalar fields. Morse theory offers a natural and intuitive way of analyzing the structure of a scalar field as well as of compactly representing the scalar field through a decomposition of its domain \( D \) into meaningful regions associated with the critical points of the field. The ascending and the descending Morse complexes are defined by considering the integral lines emanating from, or converging to the critical points of \( f \), while the Morse-Smale complex describes the subdivision of \( D \) into parts characterized by a uniform flow of the gradient between two critical points of \( f \). Computation of an approximation of the Morse and Morse-Smale complexes has been extensively studied in the literature in the 2D case, and recently algorithms have been proposed in 3D. The discrete watershed transform is one of the most popular methods used in image segmentation for 2D and 3D images and has been applied to regular Digital Elevation Models (DEMs). Here, we extend the watershed approach by simulated immersion [VS91] to compute the ascending and descending Morse complexes for simplicial meshes, focusing on triangle and tetrahedral meshes, discretizing the domain of a scalar field. The approach, however, can be easily extended to higher dimensions and our implementation is already dimension independent.

We represent the ascending and descending Morse complexes in arbitrary dimensions as an incidence graph, in which the nodes represent the cells of the Morse complexes in a dual fashion and the arcs their mutual incidence relations. We show how, in the discrete case, the incidence graph can be effectively combined with a representation of the simplicial decomposition of the underlying domain \( D \). This representation, that we call an incidence-based representation of the Morse complexes, is based on encoding the incidence relations of the cells of the two complexes, and exploits the duality between the ascending and descending complexes. The incidence graph is also a combinatorial description of the Morse-Smale complex in case the scalar field is a Morse-Smale function.

Computing the incidence graph from the result of a segmentation produced by a watershed algorithm poses interesting challenges, since we need to extract all incidence relations among the cells by starting from the collections of
maximal cells (2-cells in 2D and 3-cells in 3D) in both the ascending and descending Morse complex produced by the segmentation algorithm. Moreover, we need to ensure the compatibility of the two complexes so as to represent them in one data structure, the incidence-based representation. Computing the incidence-based representation is the first step for implementing simplification operations on the two Morse complexes in a completely transparent and dimension independent fashion.

The remainder of the paper is organized as follows. In Section 2, we review some basic notions on Morse theory and Morse complexes. In Section 3, we discuss some related work. In Section 4, we discuss the watershed approach for simulated immersion and our generalization to triangle and tetrahedral meshes decomposing the domain of 2D and 3D scalar fields. In Section 5, we describe the incidence-based representation of the Morse complexes, and in Section 6 we propose an algorithm to compute it in 2D and 3D. Finally, in Section 7, we draw some concluding remarks and discuss current and future work.

2. Morse Theory and Morse Complexes

Morse theory studies the relationship between the topology of a manifold \( M \) and the critical points of a scalar (real-valued) function defined on the manifold (for more details on Morse theory, see [Mat02, Mil63]). Recall that a closed \( n \)-manifold is a topological space in which every point has a neighborhood homeomorphic to the space \( \mathbb{R}^n \). Let \( f \) be a \( C^2 \) real-valued function defined over a closed compact \( n \)-manifold \( M \). A point \( p \) is a critical point of \( f \) if and only if the gradient \( \nabla f = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}) \) (in some local coordinate system around \( p \)) of \( f \) vanishes at \( p \). Function \( f \) is a Morse function if all its critical points are non-degenerate (i.e., the Hessian matrix \( \text{Hess}_p f \) of the second derivatives of \( f \) at \( p \) is non-singular). The number \( i \) of negative eigenvalues of \( \text{Hess}_p f \) is called the index of critical point \( p \), and \( p \) is called an \( i \)-saddle. A 0-saddle, or an \( n \)-saddle, is also called a minimum, or a maximum, respectively. An integral line of \( f \) is a maximal path which is everywhere tangent to the gradient of \( f \). Each integral line originates at a critical point of \( f \), called its origin, and converges to a critical point of \( f \), called its destination.

Integral lines that converge to (originate at) a critical point \( p \) of index \( i \) form an \( i \)-cell \( (n - i) \)-cell \( p \) called a descending (ascending) cell, or manifold, of \( p \). The descending and ascending cells decompose \( M \) into descending (stable) and ascending (unstable) Morse complexes, denoted as \( \Gamma^d \) and \( \Gamma^a \), respectively. Figure 1 shows a 2D example. In Figure 1 (a), \( p_1 \) are the descending 2-cells corresponding to maxima, \( r_t \) and \( c_t \), and \( q \) are the descending 1-cells corresponding to 1-saddles and \( z_2 \) are the descending 0-cells corresponding to minima. In Figure 1 (b), \( p \) and \( p' \) are the ascending 0-cells corresponding to maxima, \( t_1 \) and \( c_1 \) and \( q \) are the ascending 1-cells corresponding to 1-saddles and \( z_1 \) are the ascending 2-cells corresponding to minima. We will denote as \( p \) the descending \( i \)-cell of an \( i \)-saddle \( p \). A Morse function \( f \) is called a Morse-Smale function if the descending and the ascending manifolds intersect transversally. If \( f \) is a Morse-Smale function, then complexes \( \Gamma^d \) and \( \Gamma^a \) are dual to each other.

3. Related Work

In this Section, we review related work on morphological representations of scalar fields based on Morse or Morse-Smale complexes. Specifically, we concentrate on the computation and on the simplification of Morse and Morse-Smale complexes.

Several algorithms have been proposed in the literature for decomposing the domain of a 2D scalar field \( f \) into an approximation of a Morse, or a Morse-Smale, complex. Recently, some algorithms in higher dimensions have been proposed. For a review of the work in this area, see [BFF08].

Algorithms for decomposing the domain \( D \) of \( f \) into an approximation of a Morse, or of a Morse-Smale complex in 2D can be classified as boundary-based [BS98, BEHP04, EHZ01, Pas04, TIKU95], or region-based [CCL03, DDM03, MDD07]. In [EHNP03], an algorithm for extracting the Morse-Smale complex from a tetrahedral mesh is proposed. The algorithm, while interesting from a theoretical point of view, exhibits a large computation overhead, as discussed in [GNPH07].

Discrete methods rooted in the discrete Morse theory proposed by Forman [For98] are computationally more efficient. In [DDM03], a dimension-independent approach based on region growing has been proposed which implements the discrete gradient approach and computes the descending and ascending Morse complexes. In [GNPH07], a region growing method, inspired by the watershed approach, has been proposed to compute the Morse-Smale complex in 3D. It classifies the vertices of a simplicial complex \( \Sigma \) as interior or boundary vertices of ascending manifolds, and then assigns the tetrahedra, triangles and edges of \( \Sigma \) spanned by such vertices to the ascending 3-, 2-, and 1-manifolds, respectively. Descending manifolds are constructed inside ascending ones, to preserve the structure of the Morse-Smale complex. Only the minima are classified using the lower

© The Eurographics Association 2010.
link, while other critical points are created in correspondence to the constructed ascending manifolds. In [GBHP08], a formal gradient vector field $V$ is defined, and an approximation of the Morse-Smale complex is computed by tracing the integral lines defined by $V$.

Simplification algorithms have been developed in order to eliminate less significant features from a Morse-Smale complex. Simplification is achieved by applying an operator called cancellation, defined in Morse theory [Mat02]. It cancels pairs of critical points in the order usually determined by the notion of persistence (absolute difference in function values between the paired critical points) [EHZ01]. In 2D Morse-Smale complexes, cancellation operator has been investigated in [BEHP04, EHZ01, TIKU95, Wol04]. Cancellation operator on Morse-Smale and Morse complexes of a 3D scalar field has been investigated in [GNPH07] and [ČomićD08], respectively. Unfortunately, the application of such operators to 1-saddles and 2-saddles increases the number of cells in the Morse-Smale complex, and the number of incidences in the two Morse complexes.

4. Computing Morse Complexes through a Watershed Approach

In this Section, we recall the definition of the watershed transform which, in the continuous case [Mey94], produces a segmentation of the domain of a scalar field $f$ into ascending cells of minima. In the discrete case, the watershed transform has been introduced for segmentation of gray-scale images into regions of influence of minima, which approximate those ascending cells. The watershed transform can be modified in an obvious way to obtain the descending cells related to at least two connected components of $B$ that the geodesic distance between two points $p$ and $q$ in $A$ is the length of a minimal path which connects $p$ to $q$ and stays within $A$. The influence zone of a component $B_i$ of $B$ is the set of points in $A$ which are closer to $B_i$ than to any other connected component $B_j$ of $B$. Note that the skeleton by influence zones $C$ of $B$ within $A$ is the complement of the union of influence zones of $B_i$ within $A$.

The method in [VS91] recursively extracts catchment basins and watershed lines, starting from the minimal value of the elevation function $f$ and going up. At each level of recursion, new minima are found, or already created catchment basins are expanded. The expansion process continues until, at a given level $h$, a potential catchment basin $CB_h$ (related to level $h$) contains at least two catchment basins (for example $CB_{h-1}, CB_{h-1}$) already present at level $h - 1$. This is the case in which the definition of skeleton by influence zones comes up: $CB_h$ is partitioned into three elements, the two influence zones of $CB_{h-1}$ and $CB_{h-1}$ and the set of points in $CB_h$ equally distant from $CB_{h-1}$ and $CB_{h-1}$ (skeleton by influence zones). The influence zones of $CB_{h-1}$ and $CB_{h-1}$ will be part of the final set of catchment basins in the output of the algorithm. The process stops when the maximal level is reached. The watershed is defined as the complement of the set of catchment basins.

We have extended the watershed-by-simulated-immersion algorithm to simplicial meshes in arbitrary dimensions. The vertices of the simplicial mesh $\Sigma$ are sorted in increasing order with respect to the values of the scalar field $f$. In the second phase, the vertices of $\Sigma$ are processed level by level in increasing order of elevation values. For each minimum $m$, a catchment basin $CB_m$ is constructed iteratively through a breadth-first traversal of the graph which forms the 1-skeleton of the simplicial mesh $\Sigma$. We first label each vertex at level $h$ with a neutral label. Then, for each vertex $p$, we examine its adjacent vertices in the mesh and, if they

![Figure 2: Segmentations provided by watershed algorithm in the 2D case. (a) Ascending and (b) descending Morse complexes built from a synthetic function and (c) ascending and (d) descending Morse complexes built from real data.](image-url)
all belong to the same catchment basin $\beta_m$, or some of them are watershed points, then we mark $p$ as belonging to $CB_m$.

If they belong to two or more catchment basins, then $p$ is marked as a watershed point. Vertices that are not connected to any previously processed vertex are new minima and get a new label corresponding to a new catchment basin.

Finally, each maximal simplex (an $n$-simplex if we consider an $n$-dimensional simplicial mesh) is assigned to a basin based on the labels of its vertices. If all vertices of $\sigma$, that are not watershed points, have the same label corresponding to the basin $CB_m$, then we assign $\sigma$ to $\beta_m$. Otherwise, if the vertices belong to different basins corresponding to minima $m$, then $\sigma$ is assigned to the basin corresponding to the lowest such minimum.

Figure 2 illustrates segmentations obtained, in the 2D case, from a synthetic terrain, built by sampling a function which is a combination of two planes and 64 Gaussian surfaces, and from real data, part of a real terrain model, representing Mount Marcy (courtesy of USGS), formed by 16384 vertices and 32258 triangles. It has 128 minima and 113 maxima. Figure 3 illustrates the results, in the 3D case, from a synthetic function and from real data sets representing a component of a nuclear reactor Super Phoenix constructed in 1968, opened in 1981 on Reno, and closed after numerous accidents in 1997 (courtesy of Lawrence Livermore National Laboratory). It is composed of 2896 vertices and 12936 tetrahedra, and has 14 maxima and 9 minima.

We have also compared this approach in 3D with the region-based algorithm in [DDM03,MM09] using different metrics (extended from the ones used from TINs in [Vit10]) and we have obtained more promising results with the watershed approach described here [Iur10].

5. A Dual Incidence-Based Representation for Morse Complexes

In this Section, we discuss a dual representation for the ascending and descending Morse complexes $\Gamma_d$ and $\Gamma_u$, that we call the incidence-based representation. The underlying idea is that we can represent both the ascending and the descending complex as a graph by considering the boundary and co-boundary relations between cells in the two complexes. In the discrete case, we consider a representation for the simplicial mesh which generalizes an indexed data structure commonly used for triangle and tetrahedral meshes, and we relate the two representations into the incidence-based data structure.

We encode manifold simplicial meshes by storing the 0-simplices (vertices) and $n$-simplices explicitly plus some topological relations. For every $n$-simplex $\sigma$, we encode the $n+1$ vertices of $\sigma$ and the $n+1$ $n$-simplices which share an $(n-1)$-simplex with $\sigma$. For every 0-simplex, we also encode one $n$-simplex incident in it.

Recall that there is a one-to-one correspondence between $i$-saddles $p$ and $i$-cells $q$ in the descending complex $\Gamma_d$, and dual $(n-i)$-cells in the ascending complexes $\Gamma_u$, $0 \leq i \leq n$. We exploit this duality to define a representation which encodes both the ascending and the descending complexes at the same time, as an incidence graph [Ede87]. The incidence graph encodes the cells of a complex as nodes, and a subset of the boundary and co-boundary relations between cells as arcs. The incidence graph associated with an $n$-dimensional descending and ascending Morse complexes $\Gamma_d$ and $\Gamma_u$ is a graph $G = (N,A)$, in which

1. the set of nodes $N$ is partitioned into $n+1$ subsets $N_0, N_1, \ldots, N_n$, such that there is a one-to-one correspondence between nodes in $N_i$ (which we will call $i$-nodes) and the $i$-cells of $\Gamma_d$ (and thus the $(n-i)$-cells of $\Gamma_u$),
2. there is an arc joining an $i$-node $p$ with an $(i+1)$-node $q$ if and only if $i$-cell $p$ is on the boundary of $(i+1)$-cell $q$ in $\Gamma_d$ ($q$ is on the boundary of $p$ in $\Gamma_u$),
3. each arc connecting an $i$-node $p$ to an $(i+1)$-node $q$ is labeled by the number of times $i$-cell $p$ (corresponding to $i$-node $p$ in $\Gamma_d$) is incident to $(i+1)$-cell $q$ (corresponding to $(i+1)$-node $q$ in $\Gamma_d$)

Attributes are attached to the nodes in $N_0$ and $N_n$, containing information about geometry, and function values, while arcs have no associated (geometric) attributes. The incidence graph provides also a combinatorial representation of the 1-skeleton of a Morse-Smale complex. Figure 1 (c) shows a portion of the incidence graph encoding the connectivity of

![Figure 3: Segments obtained by watershed algorithm extended to the 3D case. (a) Ascending and (b) descending Morse complexes obtained from the synthetic function $w = \sin(x) + \sin(y) + \sin(z)$ and (c) ascending and (d) descending Morse complexes obtained from real data.](image-url)
the descending Morse complex in Figure 1 (a), and of the ascending Morse complex in Figure 1 (b).

We have designed and implemented a data structure based on combining the incidence graph and the underlying representation of the complex discussed above. We associate with each node representing a minimum the list of the \( n \)-simplexes forming its ascending cell, and with each node representing a maximum the list of the \( n \)-simplexes forming its descending cell. We call this data structure the incidence-based representation.

In the incidence-based representation, the incidence graph \( G = (N, A) \) is encoded as three arrays of nodes (one for minima, one for maxima and one for saddles) plus an array of arcs. Each element of the array of the nodes corresponding to minima encodes a minimum \( p \) and contains: the coordinates of \( p \), the list of the \( n \)-simplexes forming the corresponding ascending \( n \)-cell plus a list of pointers to the arcs incident in \( p \). The array of the nodes corresponding to the maxima is dual. Each element of the array of the saddles contains the lists of all saddles with the same index \( i \), and for each of them, the list of the arcs incident in it. More precisely, for a saddle \( s \) of index \( i \), there are two lists of arcs, those joining \( s \) to nodes of index \( i + 1 \) and those joining \( s \) to nodes of index \( i - 1 \).

Arcs are also stored in an array of lists. The \( j \)-th element of the array contains a list of arcs connecting nodes corresponding to saddles of index \( j \) to nodes corresponding to saddles of index \( j + 1 \). Each element of any of such lists corresponds to an arc \( a \) and contains the indexes of the two nodes in which \( a \) is incident plus an integer indicating how many times the nodes are incident to each other.

The resulting data structure is completely dimension independent. We have compared the 3D instance of the incidence-based representation with the data structure proposed in [GNP’06] for encoding 3D Morse-Smale complexes. This latter encodes the critical points (together with their geometric location) and, for each critical point \( p \), the list of the \( n \)-simplexes forming its corresponding ascending \( n \)-cell, and with each node representing a maximum the list of the \( n \)-simplexes forming its descending cell. We call this data structure the incidence-based representation.

In the preprocessing step, for each descending region in \( \Gamma_d \), a maximum node is created and inserted in the array of maximum nodes, and the same is done for each ascending region in \( \Gamma_u \), in which case a minimum node is created. The index of the corresponding vertex in \( \Sigma \) is also stored for each extremum. Then, for each \( n \)-simplex \( \sigma \) of \( \Sigma \), we add the index of \( \sigma \) to the maximum node which represents the region in \( \Gamma_d \) containing \( \sigma \) and to the minimum node which represents the region in \( \Gamma_u \) containing \( \sigma \).

The preprocessing step is common to both the 2D and 3D algorithms, while the other steps are dimension-specific and are described in the following two subsections.

6.1. Construction of the Incidence Graph in 2D

In the 2D case, after the preprocessing step, we perform two steps: (i) creation of the nodes corresponding to saddles, and (ii) creation of the arcs of the incidence graph.

To create the saddle nodes, we need to generate the 1-cells either of the ascending or of the descending complex. Each 1-cell is a chain of edges of the triangle mesh. We work on the ascending complex \( \Gamma_u \). We initialize a queue \( Q \) of triangles with an arbitrary triangle, and we label all triangles as non-visited. We repeat the following process while \( Q \) is not empty:

- extract the first triangle \( t \) from \( Q \);
- for each triangle \( t_i \) adjacent to \( t \):
  - insert \( t_i \) in \( Q \)
  - if \( t \) and \( t_i \) have not been visited, let \( m_1 \) and \( m_2 \) be the nodes representing ascending regions containing \( t \) and \( t_i \), respectively
  - if \( m_1 \) is different from \( m_2 \), check if there is a node \( s \) representing the saddle separating the ascending regions. If there is such a node, add edge \( e \) common to triangles \( t \) and \( t_i \) to it. Otherwise, create a new node \( s \) and add it to edge \( e \), as well as a reference to adjacent nodes \( m_1 \) and \( m_2 \).
- if one of the triangles adjacent to \( t \) is missing
  - consider node \( m_1 \) modeling the ascending region containing \( t \)
  - create a node which will model the adjacency between \( m_1 \) and the node modeling the boundary

6. Building an Incidence-Based Representation

We have developed algorithms for constructing the incidence graph in 2D and 3D starting from the decomposition of the simplicial complex \( \Sigma \) into regions associated with minima and maxima, and the information on the geometry of maxima and minima.

The input of the algorithms consists of the simplicial mesh \( \Sigma \) over which a scalar function \( f \) is defined, encoded in the data structure described in the previous section. The \( n \)-cells of the descending Morse complex \( \Gamma_d \) and of the ascending Morse complex \( \Gamma_u \) are expressed as collection of \( n \)-simplexes of \( \Sigma \), and they are labeled by the indexes of vertices of \( \Sigma \) which are minima and maxima of \( f \).

The preprocessing step is common to both the 2D and 3D algorithms, while the other steps are dimension-specific and are described in the following two subsections.
Middle node triangles we find three different cases. Triangle $t$.

- if the end-points of a 1-cell do not correspond to maxima,
- if a 1-cell has two different end-points corresponding to maxima, and the boundary.
- if the number of triangles incident in it belong to the same descending 2-cell.

If the 1-cell is on the boundary, and there is a minimum on 1-cell, that minimum is cancelled in order to maintain the duality in the incidence graph.

Figure 4 illustrates how the algorithm creates saddle nodes. Let us consider triangle $t_0$. By looking at its adjacent triangles we find three different cases. Triangle $t_2$ belongs to the same 2-cell as $t_0$, so no saddle node has been created. Triangle $t_1$ belongs to a different 2-cell than $t_0$ and thus saddle node $A$ is created with two references to minimum nodes $m_1$ and $m_3$. The last adjacent triangle is missing, and thus we create saddle node $B$ to model the adjacency between $m_1$ and the boundary.

Next, we create the arcs between saddle nodes and nodes corresponding to maxima.

- if a 1-cell has two different end-points corresponding to maxima, the corresponding nodes are connected in the incidence graph.
- if one of the end-points of a 1-cell belongs to the boundary, then an arc is created between a virtual maximum and the node corresponding to the other end-point of the 1-cell.
- if the end-points of a 1-cell do not correspond to maxima, we check for each of the end-points if all triangles occurring in it belong to the same descending 2-cell. If this is the case, we create a maximum at the end-point. Otherwise, we delete the saddle node, since we regard it as an error of segmentation algorithm.

If a 1-cell has no end-points, it circumscribes one of the 2-cells in $\Gamma_d$. In this case, we add a dummy maximum on the 1-cell, thus creating a loop, to maintain topological consistency.

If there is some maximum $p$ not connected to any saddle, then that maximum must be inside some 2-cell in $\Gamma_d$.

In this case, a 1-saddle is created by looking at the 2-cells corresponding to $p$ and at its adjacent 2-cells in $\Gamma_d$.

6.2. Construction of the Incidence Graph in 3D

The construction of the incidence graph requires, after the preprocessing, other three steps, namely, (i) generation of the nodes corresponding to 1-saddles and 2-saddles, (ii) generation of the arcs between 1-saddles and minima and 2-saddles and maxima and (iii) generation of the arcs joining 1- and 2-saddles.

The first two steps directly generalize the 2D algorithm. Nodes corresponding to 1-saddles and 2-saddles are constructed in a similar way as we construct saddle nodes in the 2D case. 1-saddles are generated by considering the triangulated surfaces separating 3-cells in the ascending Morse complex (recall that 3-cells correspond to minima), while 2-saddles are generated by considering the triangulated surfaces separating 3-cells in the descending Morse complex (which correspond to maxima). This is simply a generalization of the algorithm we have seen before in the 2D case: the difference is that here we consider tetrahedra instead of triangles, and that we look for triangles separating the 3-cells of the Morse complexes instead of edges separating the 2-cells.

Again, the algorithm for connecting 1-saddle nodes to minimum nodes and 2-saddle nodes to maximum nodes is a simple extension of the algorithm for computing the arcs between saddle and extrema in the 2D case.

The third step consists of generating the arcs connecting the nodes corresponding to 1-saddles to those corresponding to 2-saddles. We work first on the ascending complex $\Gamma_a$. For each 2-cell $s_1$ in $\Gamma_a$ (which corresponds to a 1-saddle), we consider the set $M_s$ of maxima connected to $s_1$, which correspond to the vertices of 2-cell $s_1$. Then, if there is more than one maximum in $M_s$ and the 2-cell $s_1$ is not on the boundary of the domain, we check, for each pair of maxima $m_1$ and $m_2$ in $M_s$, if there exists in the descending complex $\Gamma_d$ a 2-cell $s_2$ (i.e., a 2-saddle) between the 3-cells corresponding to $m_1$ and $m_2$. If $s_2$ exists, then we connect in the incidence graph the two nodes corresponding to 1-saddle $s_1$ and 2-saddle $s_2$.

Otherwise, if the 2-cell $s_1$ is on the boundary, we consider the minimum $p$ associated with the only 3-cell bounded by $s_1$ and the set of edges in the tetrahedral mesh $\Sigma$ incident into vertex $p$. For each of such edges $e$, we consider the set of tetrahedra $T$ incident in $e$ and, for each tetrahedron $t$ in $T$, we find the 3-cell in the descending complex $\Gamma_d$ containing $t$. We select only the edges $e'$ incident in $p$ such that $T'$ contains tetrahedra belonging to different 3-cells. We denote the set of such 3-cells $C_e$. We replace node $s_1$ in the graph with nodes $q$ corresponding to the selected edges. Nodes $q$ are connected to the same minimum node to which $s_1$ was connected. For each 1-saddle node $q$, we consider all pairs of maxima corresponding to the 3-cells in $C_e$ and, for each of

© The Eurographics Association 2010.
We have defined dimension-independent simplification operators for Morse complexes in arbitrary dimensions. We have proposed an algorithm for computing the incidence-based data structure, for both the ascending and descending Morse complexes of a scalar field \( f \) based on exploiting the duality of the two complexes. We have developed a watershed algorithm that we have used for constructing the maximal cells of the descending and ascending Morse complexes.

The next step in our research is to develop a simplification algorithm for Morse complexes in arbitrary dimensions. We have defined dimension-independent simplification operators called removal and contraction \([\text{CD}09]\). The removal and contraction operators have a dual effect on the descending and the ascending Morse complexes. The effect of a contraction on \( \Gamma_d (\Gamma_u) \) is the same as the effect of a removal on \( \Gamma_u (\Gamma_d) \). The effect of a removal \( \text{rem}(p,q,p') \) on the descending Morse complex \( \Gamma_d \) deletes \( i \)-cell \( q \) and merges \( (i+1) \)-cell \( p \) into \( (i+1) \)-cell \( p' \). A contraction \( \text{contr}(p,q,p') \) on the descending Morse complex \( \Gamma_d \) deletes \( i \)-cell \( q \) and merges \( (i-1) \)-cell \( p \) into \( (i-1) \)-cell \( p' \) in \( \Gamma_d' \); \( i \)-cell \( q \) is contracted, and each \( i \)-cell in the co-boundary of \( p \) is extended to include a copy of \( i \)-cell \( q \).

These operators form a minimal basis of operators for simplifying Morse complexes in arbitrary dimensions. Unlike the operators for the 3D case defined in \([\text{GNPH}07]\), these operators never increase the number of cells in the complexes. Also the operators in \([\text{GNPH}07]\) can be expressed as macro-operators in terms of our operators.

We have implemented the simplification operators on the incidence-based data structure in a completely dimension-independent way, since the incidence graph is dimension independent. We have experimented with Morse complexes for 2D and 3D scalar fields. An example of the application of simplification operators to 2D and 3D Morse complexes is shown in Figure 5.

7. Concluding Remarks

We have described a compact and dimension-independent representation, the incidence-based data structure, for both the ascending and descending Morse complexes of a scalar field \( f \) based on exploiting the duality of the two complexes. We have developed a watershed algorithm that we have used for constructing the maximal cells of the descending and ascending complexes.

The next step in our research is to develop a simplification algorithm for Morse complexes in arbitrary dimensions. We have defined dimension-independent simplification operators called removal and contraction \([\text{CD}09]\). The removal and contraction operators have a dual effect on the descending and the ascending Morse complexes. The effect of a contraction on \( \Gamma_d (\Gamma_u) \) is the same as the effect of a removal on \( \Gamma_u (\Gamma_d) \). The effect of a removal \( \text{rem}(p,q,p') \) on the descending Morse complex \( \Gamma_d \) deletes \( i \)-cell \( q \) and merges \( (i+1) \)-cell \( p \) into \( (i+1) \)-cell \( p' \). A contraction \( \text{contr}(p,q,p') \) on the descending Morse complex \( \Gamma_d \) deletes \( i \)-cell \( q \) and merges \( (i-1) \)-cell \( p \) into \( (i-1) \)-cell \( p' \) in \( \Gamma_d' \); \( i \)-cell \( q \) is contracted, and each \( i \)-cell in the co-boundary of \( p \) is extended to include a copy of \( i \)-cell \( q \).

These operators form a minimal basis of operators for simplifying Morse complexes in arbitrary dimensions. Unlike the operators for the 3D case defined in \([\text{GNPH}07]\), these operators never increase the number of cells in the complexes. Also the operators in \([\text{GNPH}07]\) can be expressed as macro-operators in terms of our operators.

We have implemented the simplification operators on the incidence-based data structure in a completely dimension-independent way, since the incidence graph is dimension independent. We have experimented with Morse complexes for 2D and 3D scalar fields. An example of the application of simplification operators to 2D and 3D Morse complexes is shown in Figure 5.

8. Acknowledgments

This work has been partially supported by the MIUR-FIRB project SHALOM under contract number RBIN04HWR8 and by the National Science Foundation under grant CCF-0541032.

References


[CD09] ˇComi´c L., De Floriani L.: Modeling and Simplifying Morse Complexes in Arbitrary Dimensions. In Workshop on Topological Methods in Data Analysis and Visualization (TopoInVis 09) (Snowbird, Utah, February 23 – 24 2009). 7


