

TADD: A Computational Framework for Data Analysis using Discrete Morse Theory

Jan Reininghaus, David Günther,
Ingrid Hotz, Steffen Prohaska, Hans-Christian Hege

Zuse Institute Berlin (ZIB), Takusstr. 7, 14195 Berlin, Germany
{reininghaus,david.guenther,hotz,prohaska,hege}@zib.de
<http://www.zib.de>

Abstract. This paper presents a computational framework that allows for a robust extraction of the extremal structure of scalar and vector fields on 2D manifolds embedded in 3D. This structure consists of critical points, separatrices, and periodic orbits. The framework is based on Forman’s discrete Morse theory, which guarantees the topological consistency of the computed extremal structure. Using a graph theoretical formulation of this theory, we present an algorithmic pipeline that computes a hierarchy of extremal structures. This hierarchy is defined by an importance measure and enables the user to select an appropriate level of detail.

Keywords: Discrete Morse theory, data analysis, scalar fields, vector fields

1 Motivation

We propose a computational framework to extract the extremal structure of scalar and vector fields on 2D manifolds embedded in \mathbb{R}^3 . The extremal structure of a scalar field consists of critical points and separatrices – the streamlines of the gradient field that connect the critical points. The extremal structure of a vector field additionally includes periodic orbits – the streamlines that are closed.

These structures are of great interest in many applications and have a long history [2, 12]. Typically, the critical points are computed by finding all zeros of the gradient or vector field. The critical points of a scalar field are classified into minima, saddles, and maxima by the eigenvalues of its Hessian, while the critical points of a vector field are classified into sinks, saddles, and sources by the eigenvalues of its Jacobian. The respective eigenvectors can be used to compute the separatrices as the solution of an autonomous ODE. For the numerical treatment of these problems and the extraction of the periodic orbits, we refer to [18, 20, 5].

One of the biggest challenges that such numerical algorithms face is the discrete nature of the extremal structure which necessitates a lot of binary decisions. For example, the type of a critical point depends on the sign of the eigenvalues.

Depending on the input data, the resulting extremal structure may therefore strongly depend on the algorithmic parameters and numerical procedures.

From a topological point of view this can be quite problematic. Morse theory relates the extremal structure of a generic function to the topology of the manifold, e.g. by the Poincaré-Hopf Theorem, or by the strong Morse inequalities [13]. The topology of the manifold therefore restricts the set of the admissible extremal structures.

Forman has developed a discrete version of Morse theory [7, 8] for cell complexes. A gradient or vector field is therein directly encoded in the combinatorial structure of the cell complex, and their extremal structure is defined in a combinatorial fashion. A finite cell complex of a 2D manifold can therefore only carry a finite number of combinatorial (gradient) vector fields, and their respective extremal structure is consistent with the topology of the manifold.

The basic idea of our computational framework is to compute a combinatorial (gradient) vector field that represents our input data. Its extremal structure can then be easily extracted and is always consistent with the topology of the manifold. This topological consistency greatly improves the robustness of our algorithm. In some sense it serves as an error correcting code: a single misclassification of a critical point cannot occur, as this would result in an inadmissible extremal structure.

Note that the first implementation of Forman's theory was presented by Lewiner [11]. His combinatorial (gradient) vector fields were thereby based on the construction of hypergraphs and hyperforests.

For the purpose of data analysis, the computed extremal structure is in general too complex. This is especially true if one deals with noisy data. One is therefore interested in a meaningful and consistent simplification of the extremal structure. Our framework allows for this by computing a sequence of combinatorial (gradient) vector fields that represents the input field. The user is then able to select an appropriate level of detail to efficiently analyze the data.

2 Computational Discrete Morse Theory

This section begins with a short introduction to discrete Morse theory in a graph theoretical formulation. We then formulate an optimization problem that results in a hierarchy of combinatorial (gradient) vector fields representing a given (gradient) vector field [14]. For simplicity, we restrict ourselves to 2D manifolds while the mathematical theory for combinatorial (gradient) vector fields is defined in a far more general setting [7, 8].

Let C denote a finite regular cell complex [9] of a 2D manifold. In this paper, we call a cell complex regular if the boundary of each d -cell is contained in a union of $(d - 1)$ -cells.

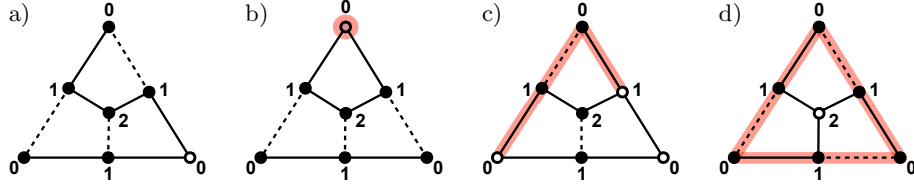


Fig. 1. Basic definitions. a) a combinatorial vector field (dashed) on the cell graph of a single triangle. The numbers correspond to the dimension of the represented cells, and matched nodes are drawn solid. b) a critical point of index 0. c) a 0-separatrix. d) an attracting periodic orbit.

Examples of such cell complexes that arise in practice are triangulations or quadrangular meshes. We first define its cell graph $G = (N, E)$, which encodes the combinatorial information contained in C in a graph theoretic setting.

The nodes N of the graph consist of the cells of the complex C and each node u^p is labeled with the dimension p of the cell it represents. The edges E of the graph encode the neighborhood relation of the cells in C . If the cell u^p is in the boundary of the cell w^{p+1} , then $e^p = \{u^p, w^{p+1}\} \in E$. Note that we label each edge with the dimension of its lower dimensional node.

A subset of pairwise non-adjacent edges is called a matching. Using these definitions, a *combinatorial vector field* V on a regular cell complex C can be defined as a matching of the cell graph G (see Figure 1a). The set of combinatorial vector fields on C is thereby given by the set of matchings \mathcal{M} of the cell graph G .

We now define the extremal structure of a combinatorial vector field. The unmatched nodes are called critical points. If u^p is a critical point, we say that the critical point has index p . A critical point of index p is called sink ($p = 0$), saddle ($p = 1$), or source ($p = 2$). A combinatorial p -streamline is a path in the graph whose edges are of dimension p and alternate between V and its complement. A p -streamline connecting two critical points is called a p -separatrix. If a p -streamline is closed, we call it either an attracting periodic orbit ($p = 0$) or a repelling periodic orbit ($p = 1$). An illustration of the combinatorial extremal structure is shown in Figure 1.

As shown in [4], a *combinatorial gradient vector field* V^ϕ can be defined as a combinatorial vector field that contains no periodic orbits. A matching of G that gives rise to such a combinatorial vector field is called a Morse matching. The set of combinatorial gradient vector fields on C is therefore given by the set of Morse matchings \mathcal{M}^ϕ of the cell graph G . In the context of gradient vector fields, we refer to a critical point u^p as a minimum ($p = 0$), saddle ($p = 1$), or maximum ($p = 2$).

We now define the optimization problem that results in a meaningful combinatorial representative of our input data f . Assume that f leads to edge weights $\omega : E \rightarrow \mathbb{R}$ – we postpone their computation to Section 3.2. Assume further that f is represented well if the weight of the matching is high. We can then compute a combinatorial vector field to represent f by finding the maximum weight matching in G

$$V = \arg \max_{M \in \mathcal{M}} \omega(M). \quad (1)$$

If we want to compute a combinatorial gradient vector field V^ϕ we simply replace \mathcal{M} by \mathcal{M}^ϕ . Note that this restriction of the admissible matchings makes (1) an NP-hard problem in general [10].

Due to the matching property, the number of critical points is given by $|N| - 2|V|$. We can therefore compute a combinatorial vector field with a prescribed number of critical points by computing

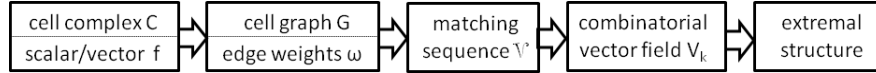
$$V_k = \arg \max_{M \in \mathcal{M}, |M|=k} \omega(M). \quad (2)$$

Let $k_0 = |V|$ denote the size of the maximum weight matching, and let $k_n = \max_{k \in \mathbb{N}} |V_k|$ denote the size of the heaviest maximum cardinality matching. From a data analysis point of view, V_{k_0} is a fine grained while V_{k_n} is the coarsest possible representation of the input data f . A hierarchy of combinatorial (gradient) vector fields \mathcal{V} can now be defined as the sequence of matchings

$$\mathcal{V} = (V_k)_{k=k_0, \dots, k_n}. \quad (3)$$

The main task of our computational framework is to compute the sequence (3).

3 Algorithmic Pipeline



Our computational framework consists of five main parts, which the following subsections describe in detail.

3.1 Input Data

Our computational framework requires as input a finite regular cell complex of a 2D manifold embedded in \mathbb{R}^3 and a scalar or vector valued function f . We assume that f is defined on the 0-cells of the complex. Because we will later need data values on all cells, we extend f to the higher dimensional cells by taking the average value of the incident lower dimensional cells.

3.2 Edge Weighted Cell Graph

Using the regular cell complex, we first construct its cell graph $G = (N, E)$ as defined in Section 2. A spatial embedding $c : N \rightarrow \mathbb{R}^3$ of G can be defined using the embedding of the cell complex in \mathbb{R}^3 . The embedding of the nodes that represent higher dimensional cells is thereby computed by taking the average of the coordinates from the incident lower dimensional cells.

We now define the edge weights $\omega : E \rightarrow \mathbb{R}$ of this graph. In Section 2 we assumed that f is represented well by a matching M , if the weight of M is large (1). Let $e^p = \{u^p, w^{p+1}\}$ denote an edge of the graph. If e^p is a matching edge it can be thought of as an arrow pointing from u^p to w^{p+1} . We therefore assign a large weight to e^p if such an arrow reflects the flow behavior f well. In this paper, we propose to measure the tangential flow of f along e^p to achieve this. Using Stokes Theorem, the edge weight ω for scalar input data f is thereby defined by

$$\omega(e^p) = f(w^{p+1}) - f(u^p), \quad (4)$$

whereas in case of vector field data f (assuming linear interpolation), its edge weight is defined by

$$\omega(e^p) = (f(w^{p+1}) + f(u^p)) \cdot (c(w^{p+1}) - c(u^p)) / 2. \quad (5)$$

3.3 Matching Sequence

Given the edge weighted cell graph G , we now compute the sequence of maximum weight matchings (3). We begin with an algorithm for the vector field case \mathcal{M} . We then use the introduced notation to describe an algorithm that approximates (3) for the scalar field case \mathcal{M}^ϕ .

Due to the assumed regularity of the cell complex C , the cell graph G is bipartite – a simple bipartition can be derived from the dimension of the represented cells. We can therefore employ the Hungarian method. This method is usually employed to compute V_{k_0} , but can be directly applied to compute (3).

The following presentation of the Hungarian method closely follows [16]. The basic idea is to start with V_0 and then to iteratively compute the sequence (2). In each iteration, the augmenting path of maximum weight is computed. An augmenting path of a matching V_j is a path in the graph whose start and end nodes are not covered by the matching and whose links alternate between V_j and its complement. The weight of an augmenting path is defined as the alternating sum of the weights of its links. Given an augmenting path p of maximum weight we can augment the matching V_j to get V_{j+1} by taking the symmetric difference \triangle of V_j and p . For an efficient computation of the augmenting path of maximum weight we refer to [16].

To store the sequence of matchings (3) efficiently, we only store V_{k_n} and the sequence of augmenting paths that lead from V_{k_0} to V_{k_n} [14]. Because the augmenting paths of edge weighted cell graphs are usually rather short, this is a lot

more efficient in our context than storing all matchings of (3) individually.

The computation of (3) for the scalar field case \mathcal{M}^ϕ is a lot more involved – in general it is NP-hard [10]. We therefore propose a simple approximation algorithm for this problem. The basic idea is to make use of Forman’s cancellation theorem [8]. Using the graph theoretic formulation introduced in Section 2 this theorem can be stated as follows:

If two unmatched nodes are connected by a unique p -separatrix s in a Morse matching $M \in \mathcal{M}^\phi$, then $M \triangle s$ is a Morse matching.

The pseudo code for our approximation algorithm is shown in Algorithm 1. The input consists of the cell Graph G and its edge weights ω . The output consists of $V_{k_n}^\phi$ and a list of augmenting paths. Together, these can be used to reconstruct an arbitrary element of the sequence (3). The subfunction *getMaxUniqueSeparatrix(...)* returns the unique p -separatrix of maximum weight of the saddle u^1 . The 2D manifold structure of the cell graph G implies that at most four p -streamlines emanate from u^1 and that these cannot split. The subfunction *getMaxUniqueSeparatrix(...)* therefore simply iterates all (up to four) p -separatrices that start in u^1 . It then checks for uniqueness by comparing their end nodes and returns the unique p -separatrix with the largest weight. If there is no unique p -separatrix at all, then an empty path is returned with a weight of $-\infty$. Note that there are always two 0-streamlines emanating from u^1 and that these are always 0-separatrices. The 1-streamlines that emanate from u^1 however may end in the boundary of the manifold.

Algorithm 1 MorseMatchingSequence(G, ω)

Output: $AugPaths, V_{k_n}^\phi$

```

1:  $M \leftarrow \emptyset, AugPaths \leftarrow \emptyset, heap \leftarrow \emptyset$ 
2: for all  $u^1 \in N$  do
3:    $(path, weight) \leftarrow getMaxUniqueSeparatrix(G, \omega, M, u^1)$ 
4:    $heap.push(u^1, weight)$ 
5: while  $heap \neq \emptyset$  do
6:    $(u^1, weight) \leftarrow heap.pop()$ 
7:    $(path, weight) \leftarrow getMaxUniqueSeparatrix(G, \omega, M, u^1)$ 
8:    $(nextNode, nextWeight) \leftarrow heap.top()$ 
9:   if  $weight \geq nextWeight$  then
10:     $M \leftarrow M \triangle path$ 
11:    if  $weight < 0$  then
12:       $AugPaths.push(path)$ 
13:    else if  $-\infty < weight$  then
14:       $heap.push(u^1, weight)$ 
15:  $V_{k_n}^\phi \leftarrow M$ 

```

Line 1 initializes M as the empty matching, the list of augmenting paths $AugPaths$, and a priority queue $heap$. All nodes representing 1-cells are then inserted into this queue, ordered by the weight of their heaviest unique p -separatrix, in Lines 2-4. We then iterate over the queue (Line 5), remove the top element of the $heap$ (Line 6) and compute its heaviest unique p -separatrix (Line 7). This is necessary, as previous iterations may have affected this node. We now check whether this p -separatrix is the heaviest of all available unique p -separatrices. Assuming that augmenting the matching only decreases the weight returned by $getMaxUniqueSeparatrix(\dots)$, it suffices to check whether the weight of u^1 is larger than the weight of the next element of the $heap$ (Lines 8-9). If this is the case, we augment the matching M by taking the symmetric difference of M and $path$ (Line 10), and store the augmenting path if its weight is negative (Line 11-12). Otherwise, we reinsert u^1 into the $heap$ with its new weight if it is larger than $-\infty$ (Line 13-14). When the $heap$ is empty the algorithm terminates and returns an approximation of the heaviest maximum cardinality Morse matching $V_{k_n}^\phi$.

3.4 Combinatorial (Gradient) Vector Field

The heaviest maximum cardinality (Morse) matching V_{k_n} and the list of augmenting paths computed in the Section 3.3 allows for the reconstruction of an arbitrary element of the sequence (3). Each matching can be restored by iteratively taking the symmetric difference of V_{k_n} with the augmenting paths. This enables the user to select a combinatorial (gradient) vector field with prescribed number of critical points.

Alternatively, we can make use of the associated weight of each matching as an importance measure. The user can set a fraction $\theta \in [0, 1]$ to select a combinatorial (gradient) vector field with a weight as close as possible to $\omega(V_{k_0}) + \theta(\omega(V_{k_n}) - \omega(V_{k_0}))$. This approach can be useful in dealing with noisy data. Noise induces a very complex extremal structure. The augmenting paths corresponding to the spurious extremal structure, however, have a very large weight. Setting θ to a small value therefore removes all spurious extremal structures while the dominant structure remains unchanged.

Note that taking the symmetric difference of a matching M with an augmenting path corresponds to the cancellation of a pair of critical points. In the scalar case, the weight of such an augmenting path equals the difference of the scalar values of these two critical points. The above importance measure is thereby closely related to the persistence measure in [6].

3.5 Extremal Structure

Given a combinatorial (gradient) vector field, we can now extract its extremal structure. As the critical points are the unmatched nodes u^p they can be easily

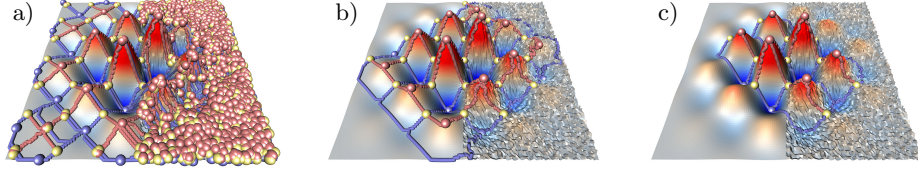


Fig. 2. Synthetic noisy scalar field. Extremal structure of a) $V_{k_0}^\phi$, b) $V_{k_n-23}^\phi$ and c) $V_{k_n-11}^\phi$. Minima, saddles and maxima are depicted as blue, yellow and red spheres, while 0-separatrices and 1-separatrices are shown as blue and red lines.

extracted. The classification into sources, saddles, sinks (minima, saddles, maxima) is given by p . To compute all separatrices, we iterate over all saddles u^1 and compute the incident p -separatrices.

In the vector case, we also need to extract all periodic orbits. Due to the $2D$ manifold structure, p -streamlines can not split when the first node is a 1-node. Therefore, the extraction of periodic orbits is quite simple. First, we iterate over all 1-nodes. Given a node u^1 , we start the computation of the p -streamlines that emanates at u^1 . Each streamline is continued as long as the following node w^1 is not yet labeled, in which case it is labeled with u^1 . If the label of w^1 equals u^1 we add w^1 to a set of seed points. We then iterate over all seed points to compute all periodic orbits.

4 Examples

In this section, we present three applications of the computational framework presented in Section 3. The framework was implemented as a module in the visualization and data analysis software Amira [17]. It can be made available to researchers for evaluation purposes. The integrated visualization capability of Amira was used to assess the relevance of the computed extremal structures and the practical quality of the approximation Algorithm 1.

The visualization of the abstract representation of the input data as a matching in an edge weighted graph proved to be very useful in the development of correct and efficient algorithms.

We first illustrate the ability to extract the extremal structure of a scalar field where noise is present. We then apply our framework to a vector field on a $2D$ manifold to show the physical relevance of the hierarchy of extremal structures (3). The paper is concluded with an application to extremal lines of curvature fields of a discrete $2D$ manifold. All examples were computed on a workstation containing an Intel Core i7 860 CPU.

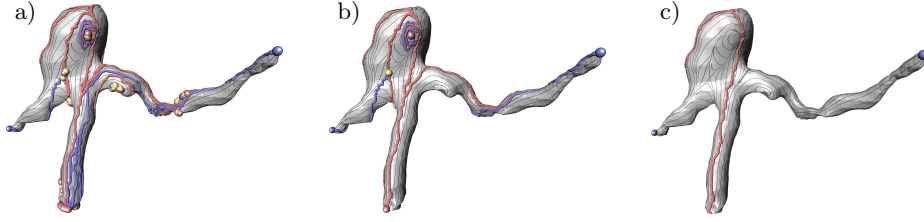


Fig. 3. Vector field from biofluid mechanics. The vector field is visualized using the streamline seeding technique described in [15]. The extremal structures of a) V_{k_0} , b) V_{k_n-4} and c) V_{k_n} are shown. Sinks, saddles and sources are depicted as blue, yellow and red spheres. 0-separatrices and attracting periodic orbits are depicted as blue lines, while 1-separatrices and repelling periodic orbits are shown as red lines.

4.1 A Synthetic Noisy Scalar Field

To illustrate the robustness of our data analysis framework, we applied it to a synthetic data set depicted as a height field in Figure 2. The data set was produced by sampling the analytic function $f : [-1, 1]^2 \rightarrow \mathbb{R}$

$$f(x, y) = \sin(10x) \sin(10y) e^{-3(x^2+y^2)} \quad (6)$$

on a uniform triangulation with $16k$ vertices. We then added uniform noise in the range of $[-0.05, 0.05]$ to the sub domain $[0, 1] \times [-1, 1]$. We applied our algorithmic pipeline presented in Section 3 to this input data. The runtime for the computation of (3) using Algorithm 1 was less than a second on a standard workstation. Figure 2 shows the extremal structure of the initial combinatorial gradient field $V_{k_0}^\phi$, and two elements, $V_{k_n-23}^\phi$ and $V_{k_n-11}^\phi$, of the matching sequence (3). As can be seen in Figure 2a, $V_{k_0}^\phi$ includes the extremal structure induced by the noise. The simplified combinatorial gradient fields, however, only contain the dominant extremal structure present in f .

4.2 A Vector Field from Biofluid Mechanics

Figure 3 depicts a surface velocity field of a simulation of blood flow through a cerebral aneurysm done by the Biofluid Mechanics Lab of the Charité - Universitätsmedizin Berlin [3]. The cell graph of the triangulation consists of $60k$ nodes. The runtime for the computation of (3) using a simple implementation of the Hungarian method was about 30 minutes.

The critical points in this vector field are stagnation points and thus of interest for the flow analysis. Our algorithm delivers a hierarchy of extremal structures which captures the dominant nature of the flow (see Figure 3 bottom-left). The blood enters the aneurysm at the bottom, and leaves it horizontally. This behavior is found by our algorithm and the global separation on the surface is extracted. This reduced flow structure may serve as a basis when comparing different cerebral aneurysms.

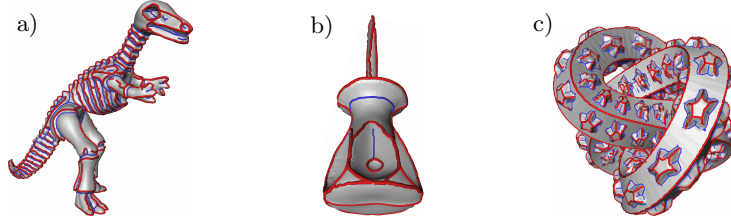


Fig. 4. Extremal lines in curvature fields. For all surface models, the first and second principal curvatures κ_1 and κ_2 are computed. a), b) and c) show the most dominant parts of 0-separatrices (blue) in κ_1 and 1-separatrices (red) in κ_2 .

4.3 Extremal Lines in Curvature Fields

Figure 4 illustrates the extraction of extremal lines in curvature fields of different surfaces. As described in [19], each point of a p -separatrix can be assigned an importance value, called separatrix persistence. The main idea of separatrix persistence is to measure the strength of monotony breaks with respect to the sequence of combinatorial gradient vector fields (3). For details how to incorporate this measure into our computational pipeline, we refer to [19].

Separatrix persistence allows to discriminate spurious from dominant extremal lines. These lines are shown in Figure 4. Note that a reduction to the most dominant extremal parts destroys the connectivity of the extremal structure. The total running time for the computation of (3) using Algorithm 1 and the computation of separatrix persistence is shown in Table 1. The worst case complexity of Algorithm 1 is $O(n^3)$, where n denotes the number of edges in the triangulation. However, the empirical running time for practical applications is almost linear. The models are provided by Aim@Shape [1].

Surface Model	triangles	nodes in G	edges in G	time (sec)
screwdriver	54300	162902	325800	1
dinosaur	112384	337154	674304	2
knot	957408	2872224	5744448	24

Table 1. Running time for Algorithm 1 with separatrix persistence [19] computation.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments that significantly improved the quality of this paper. The first author and third author are funded by the DFG Emmy-Noether research program, while the second author is funded by the Max-Planck Institute of Biochemistry, Martinsried. Finally, we would like to thank Jens Kasten, Michael Koppitz, Britta

Weber, Daniel Baum, and Tino Weinkauff for many fruitful discussions on this topic.

References

1. Aim@Shape: <http://shapes.aim-at-shape.net/>
2. Cayley, A.: On contour and slope lines. *The London, Edinburg and Dublin Philosophical Magazine and Journal of Science* 18, 264–268 (1859)
3. Cebral, J., Castro, M., Appanaboyina, S., Putman, C., Millan, D., Frangi, A.: Efficient pipeline for image-based patient-specific analysis of cerebral aneurysm hemodynamics: technique and sensitivity. *IEEE transactions on medical imaging* 24(4), 457–467 (2005)
4. Chari, M.K.: On discrete Morse functions and combinatorial decompositions. *Discrete Math.* 217(1-3), 101–113 (2000)
5. Chen, G., Mischaikow, K., Laramée, R.S., Pilarczyk, P., Zhang, E.: Vector field editing and periodic orbit extraction using Morse decomposition. *IEEE Transactions on Visualization and Computer Graphics* 13(4), 769–785 (2007)
6. Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical Morse complexes for piecewise linear 2-manifolds. *Discrete Computational Geometry* 30, 87–107 (2003)
7. Forman, R.: Combinatorial vector fields and dynamical systems. *Mathematische Zeitschrift* 228(4), 629–681 (August 1998)
8. Forman, R.: Morse theory for cell complexes. *Advances in Mathematics* 134, 90–145 (1998)
9. Hatcher, A.: *Algebraic Topology*. Cambridge University Press, Cambridge, U.K. (2002)
10. Joswig, M., Pfetsch, M.E.: Computing optimal Morse matchings. *SIAM J. Discret. Math.* 20(1), 11–25 (2006)
11. Lewiner, T.: Geometric discrete Morse complexes. Ph.D. thesis, Department of Mathematics, PUC-Rio (2005), advised by Hélio Lopes and Geovan Tavares
12. Maxwell, J.C.: On hills and dales. *The London, Edinburg and Dublin Philosophical Magazine and Journal of Science* 40, 421–425 (1870)
13. Milnor, J.: *Topology from the differentiable viewpoint*. Univ. Press Virginia (1965)
14. Reininghaus, J., Hotz, I.: Combinatorial 2d vector field topology extraction and simplification. In: Pascucci, V., Tricoche, X., Hagen, H. (eds.) *Topology in Visualization 2009* (2010), (to appear)
15. Rosanwo, O., Petz, C., Prohaska, S., Hotz, I., Hege, H.C.: Dual streamline seeding. In: Eades, P., Ertl, T., Shen, H.W. (eds.) *Proceedings of the IEEE Pacific Visualization Symposium*. pp. 9 – 16 (2009)
16. Schrijver, A.: *Combinatorial Optimization*. Springer (2003)
17. Stalling, D., Westerhoff, M., Hege, H.C.: Amira: A highly interactive system for visual data analysis. *The Visualization Handbook* pp. 749–767 (2005)
18. Weinkauff, T.: Extraction of Topological Structures in 2D and 3D Vector Fields. Ph.D. thesis, University Magdeburg and Zuse Institute Berlin (2008)
19. Weinkauff, T., Günther, D.: Separatrix persistence: Extraction of salient edges on surfaces using topological methods. *Computer Graphics Forum (Proc. SGP '09)* 28(5), 1519–1528 (July 2009)
20. Wischgoll, T., Scheuermann, G.: Detection and visualization of closed streamlines in planar flows. *IEEE Transactions on Visualization and Computer Graphics* 7(2), 165–172 (2001)