

# Computing Morse-Smale Complexes with Accurate Geometry

Attila Gyulassy, Peer-Timo Bremer, *Member, IEEE*, and Valerio Pascucci, *Member, IEEE*

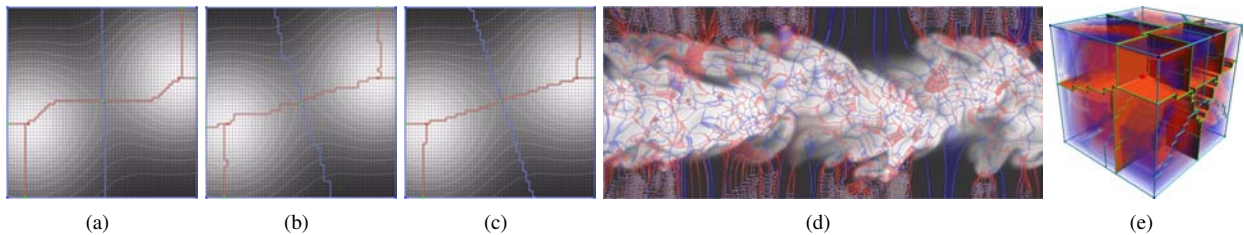


Fig. 1. Popular approaches for computing Morse-Smale complexes assign discrete gradient arrows in the direction of steepest descent, aligned with cells of the input mesh. Given a highly resolved sampling of two Gaussians, we show the results of the corresponding complex (a) which demonstrates severe artifacts in the positioning and direction of one-manifolds as can be seen from the level sets, which should be orthogonal. This paper introduces a randomized algorithm that better represents the gradient flow (b); and a deterministic variant that integrates probabilities to achieve near-optimal geometric reconstruction of the MS complex (c). We show the high-quality geometry, as well as the probability fields we compute for a two-dimensional jet (d); and a three-dimensional tetrahedrane molecule (e).

**Abstract**—Topological techniques have proven highly successful in analyzing and visualizing scientific data. As a result, significant efforts have been made to compute structures like the Morse-Smale complex as robustly and efficiently as possible. However, the resulting algorithms, while topologically consistent, often produce incorrect connectivity as well as poor geometry. These problems may compromise or even invalidate any subsequent analysis. Moreover, such techniques may fail to improve even when the resolution of the domain mesh is increased, thus producing potentially incorrect results even for highly resolved functions. To address these problems we introduce two new algorithms: (i) a randomized algorithm to compute the discrete gradient of a scalar field that converges under refinement; and (ii) a deterministic variant which directly computes accurate geometry and thus correct connectivity of the MS complex. The first algorithm converges in the sense that on average it produces the correct result and its standard deviation approaches zero with increasing mesh resolution. The second algorithm uses two ordered traversals of the function to integrate the probabilities of the first to extract correct (near optimal) geometry and connectivity. We present an extensive empirical study using both synthetic and real-world data and demonstrates the advantages of our algorithms in comparison with several popular approaches.

**Index Terms**—Topology, topological methods, Morse-Smale complex.

## 1 INTRODUCTION

Since the development of the first combinatorial algorithms to compute the Morse-Smale (MS) complex from sampled data [1, 3, 9, 17] it has become widely used in a large variety of applications [5, 7, 15, 19, 22, 25]. Since then most of the algorithmic development has been focused on robust computation, topological consistency, and computational efficiency. As a result there now exist comparatively simple, yet highly efficient, streaming and/or parallel algorithms to compute MS complexes from large data sets [13, 28–30]. These algorithms perform very well as long as one is interested in structural topological attributes of the data such as how many critical points exist [22] or average connectivity of 1-manifold networks [15]. More recently researchers have also begun to analyze the geometric information of topological features. For example Bennett et al. [2] compute length scales of local contours and Kasten et al. [19] find and track two dimensional vortex structures as unstable manifolds in the acceleration magnitude. However, geometric fidelity so far has been mainly ignored when designing algorithms and, as shown in Fig. 1, all existing techniques tend to produce poor geometry even for well-resolved functions. Furthermore, as discussed in more detail in Section 6, incorrect geometry often leads to incorrect connectivity thus casting doubts on subsequent analysis.

Even more problematic is the fact that traditional approaches will not converge with increasing mesh resolution to the correct solution.

In this context it is important to remember that there exists no MS complex for a sampled function since such functions are not “Morse-Smale.” Instead, all algorithms, discrete or otherwise, compute a quasi-MS complex as defined by Edelsbrunner et al. [9]. Intuitively, a quasi-MS complex is one that is consistent with the underlying theory, *i.e.*, there exists a Morse function with this complex, yet not necessarily matching the flow behavior of the function sampled. In particular, Edelsbrunner et al. introduce a rarely used procedure call a Handle-Slide to correct the connectivity of a quasi-MS complex assuming the ability to perfectly compute the geometry of integral lines. By ignoring geometric fidelity most existing techniques may produce severely incorrect MS complexes making geometric quality a fundamental issue of correctness rather than only of visual improvement.

The geometry of the MS complex is defined by integrating gradient lines of the underlying flow which current discrete techniques construct through a local, greedy optimization. However, the locally optimal choice may accumulate an arbitrarily large global error in the integration process and also make the algorithms highly dependent on local mesh orientation. Instead, we propose two new algorithms that both extract significantly better geometry than existing approaches and are guaranteed to converge to the correct geometry and connectivity under subdivision. The first algorithm uses the same algorithmic kernel as the best existing techniques but replaces the local optimization with a carefully designed random selection. It can easily be integrated into existing approaches, on average produces the correct complex, and its standard deviation from the mean converges to zero with increasing mesh size.

The second algorithm exploits the fact that typically only small portions of the gradient field matter to the geometry, namely those areas containing cell boundaries. By integrating the local probabilities of the first algorithm we construct a deterministic variant of the algo-

- Attila Gyulassy is with the SCI Institute, University of Utah, E-mail: jediati@sci.utah.edu.
- Peer-Timo Bremer is with Lawrence Livermore National Laboratory, E-mail: bremer5@llnl.gov.
- Valerio Pascucci is with the SCI Institute, University of Utah, E-mail: pascucci@sci.utah.edu.

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

rithm that extracts near optimal cell boundaries through a simple region growing approach. Both algorithms are based on the common discrete framework and thus are combinatorial in nature, guaranteed to be topologically consistent, and applicable in any dimension. Our four main contributions are listed briefly below.

1. We introduce a randomized algorithm to compute MS complexes that significantly improves the geometric quality of the result when compared to existing techniques and converges with increasing mesh resolution. The algorithm is simple to implement, efficient, applicable in any dimension, and easily parallelized.
2. We introduce a second deterministic algorithm that directly extracts near optimal geometry without the need for refinement.
3. We provide an extensive empirical study comparing our approach with state of the art techniques using both synthetic and real-world data.
4. We provide a broad range of experiments demonstrating the superior quality and stability with respect to sampling density and mesh orientation.

## 2 RELATED WORK

Despite the fact that the initial ideas were already discussed more than a century and a half ago [6,26] the first practical algorithm to compute MS complexes of two dimensional, piecewise linear functions was introduced fairly recently [9]. In their seminal paper, Edelsbrunner et al. not only describe the first robust algorithm but also introduce the notion of a quasi-MS complex indicating a complex consistent with the theory yet not necessarily correct. Nevertheless, the algorithm relies on fairly complex data structures to maintain strict separation between one manifolds.

Bremer et al. [3] propose a simpler version of the same idea by demonstrating that by ordering the computations appropriately much of the algorithmic complexity can be avoided. They also introduce the first multi-resolution encoding of the complex. The algorithm proposed in [3] is interesting in so far it has been the only one capable of producing correct geometry for piecewise linear functions by locally refining the input mesh. However, the 3D version of the same approach [8] has proven too complex to be practical and instead the focus has shifted to approaches based on discrete Morse theory, as discussed below.

Since the initial research, related concepts have seen rapid adoption in a wide variety of scientific applications. Laney et al. [22] use MS complexes to analyze Rayleigh-Taylor instabilities, Gyulassy et al. [15] to study the structure of porous media, Bremer et al. [4] to analyze turbulent combustion simulations, and Kasten et al. [19] to study vortical structures, to name just a few. In a related field, Koch et al. [21] used a probabilistic method for tractography in diffusion tensor images. This work introduced a randomized approach, taking discrete, axis-aligned steps in tracing out probable diffusion paths, with probabilities weighted by alignment with the local tensor.

The most successful algorithms for computing Morse-Smale complexes for large, especially volumetric data rely on a discrete interpretation of Morse theory introduced by Forman [11]. In this framework, the key challenge is to compute a discrete gradient vector field from which the MS complex can then be extracted in a fairly straightforward manner. Existing techniques share many characteristics yet each computes a discrete gradient vector field in a unique manner. At the core of every algorithm is the choice of discrete gradient vector that best represents the flow behavior of the underlying function and in this aspect all approaches are remarkably similar. Every approach assigns weights to pairings based on the difference in function values of the cells to be paired. Here, we survey each technique highlighting the differences in order of computation and philosophy while pointing out the similarities of the results.

Lewiner [23] presented the first technique for constructing a discrete gradient field that agrees with the flow behavior of a scalar function. In this approach, recently proved to be robust [24], the discrete gradient field is represented by a hyperforest. At each iteration of the GREEDY algorithm, the potential pair with maximum weight is selected and paired if it does not create a cycle. In this approach, the weight of a pairing is computed as the difference in function values at the barycenters of the two cells.

King et al. [20] introduced a technique to generate discrete Morse functions on simplicial complexes by modifying the Hasse diagram. The resulting diagram encodes both the discrete gradient arrows and the face relations needed to guarantee the construction of a discrete Morse function. At each step, the EXTRACTRAW algorithm first creates a directed edge in the diagram between a vertex and the edge in the direction of steepest descent, and then assigns pairs in the rest of the lower link. In effect, the “weight” of the initial vertex-edge pairing is given by the difference in function values of the two vertices incident on the edge. An alternative, that scales this weight by one over the length of the edge, is also mentioned.

Gyulassy et al. [13] introduced an algorithm that assigns cells in order of increasing function value and increasing dimension, using the ordering to avoid acyclicity checks. In an extension to the algorithm [12], simply homotopy expansions are performed to avoid spurious critical points. In this approach, the maximal weight pair is chosen for a cell, with weight being defined as the difference in value between the cell and the lowest vertex of its co-facet pair. This approach is used in [18] to design a blocked, parallel construction of MS complexes.

Reininghaus et al. [27,28] presented an approach for generating discrete vector fields at multiple scales by computing matchings of a cell graph using the Hungarian method. The mesh is represented using a Hasse diagram, where the weight of each edge is given by its alignment to the gradient direction. Each iteration of the MORSEMATCHINGSEQUENCE algorithm adds to the current matching the alternating path of heaviest weight. The unpaired cells of the maximum weight matching for any scalar function occur at critical points of the function. Although this technique finds a maximal global weight, the global sum is composed of aggregating local weights, therefore, in effect, gradient pairs are assigned in the steepest direction.

Robins et al. [29] present a technique that computes the discrete gradient on the lower star of a vertex. The PROCESSLOWERSTARS algorithm pairs each vertex with the edge in its lower star that touches the lowest vertex in the lower link. Subsequently, remaining cells in the lower star are paired using simple homotopy expansions, when possible, or assigning critical cells. Effectively, this approach maximizes the weight of a vertex-edge pairing, the weight being the difference in function values of the two vertices incident on the edge. The independent assignment of gradient arrows in this approach allows an embarrassingly parallel implementation.

Shivashankar et al. [30] also present an embarrassingly parallel technique for computing the gradient. First, a discrete Morse function is computed in which every cell is critical. However, the function is defined such that the ASSIGNGRADIENT algorithm simply looks for the co-facet with lowest value to pair a cell. The recursively defined function ensures that a vertex will always be paired with the edge whose other endpoint has lowest value, *i.e.*, in the direction of steepest descent.

In summary, while all approaches use slightly different techniques they all assign arrows based on steepest-descent and thus produce very similar geometry. For example, two functions for which *all techniques [13,20,23,27–30] produce identical discrete gradient fields* are (1) distance from a point,  $f(x) = c||x - p||$ , and (2) any constant slope,  $f(x) = cx + b$ . The differences for more complex functions are minimal, and each technique follows the same pattern of greedy assignment. In the examples in this paper, we use the algorithm of Gyulassy et al. [13] as a representative of this class of approaches.

## 3 BACKGROUND

Scalar valued volumetric data is most often available as discrete samples at the vertices of an underlying mesh. Morse theory has been well-studied in the context of smooth scalar functions, and has been adapted to such discrete domains. We first present some basic definitions from smooth Morse theory, and then present the discrete analogue.

### 3.1 Morse Functions and the MS Complex

Let  $f$  be a real-valued smooth map  $f : \mathbb{M} \rightarrow \mathbb{R}$  defined over a compact  $d$ -manifold  $\mathbb{M}$ . A point  $p \in \mathbb{M}$  is critical when  $|\nabla f(p)| = 0$ , *i.e.* the gradient is zero, and is non-degenerate when its Hessian (matrix of second partial derivatives) is non-singular. The function  $f$  is a *Morse function* if all its critical points are non-degenerate and no two critical points have the same function value. In this case the *Morse Lemma*

states that there exists local coordinates around  $p$  such that  $f$  has the following *standard form*:  $f_p = \pm x_1^2 \pm x_2^2 \cdots \pm x_d^2$ . The number of minus signs in this equation gives the *index* of critical point  $p$ . In three-dimensional functions, minima are index-0, 1-saddles are index-1, 2-saddles are index-2, and maxima are index-3.

An integral line in  $f$  is a path in  $\mathbb{M}$  whose tangent vector agrees with the gradient of  $f$  at each point along the path. The integral line passing through a point  $p$  is the solution to  $\frac{\partial}{\partial t} L(t) = \nabla f(L(t)), \forall t \in \mathbb{R}$ , with initial value  $L(0) = p$ . Each integral line has an origin and destination at critical points of  $f$ . *Ascending* and *descending* manifolds are obtained as clusters of integral lines having common origin and destination respectively. The descending manifolds of  $f$  form a cell complex that partitions  $\mathbb{M}$ ; this partition is called the *Morse complex*. Similarly, the ascending manifolds also partition  $\mathbb{M}$  in a cell complex. A Morse function  $f$  is a *Morse-Smale function* if ascending and descending manifolds of its critical points only intersect transversally. An index- $i$  critical point has an  $i$ -dimensional descending manifold and a  $(d-i)$ -dimensional ascending manifold.

## 3.2 Discrete Morse Theory

Discrete Morse theory is at the heart of current techniques for efficiently computing Morse-Smale complexes. We provide a brief overview with basic definitions from Forman [11], and we refer the reader to this introductory work for an intuitive description. A  $d$ -cell is a topological space that is homeomorphic to a Euclidean  $d$ -ball  $B^d = \{x \in \mathbb{E}^d : |x| \leq 1\}$ . For cells  $\alpha$  and  $\beta$ ,  $\alpha < \beta$  means that  $\alpha$  is a *face* of  $\beta$  and  $\beta$  is a *co-face* of  $\alpha$ , i.e., the vertices of  $\alpha$  are a proper subset of the vertices of  $\beta$ . If  $\dim(\alpha) = \dim(\beta) - 1$ , we say  $\alpha$  is a *facet* of  $\beta$ , and  $\beta$  is a *co-facet* of  $\alpha$ , and denote this  $\alpha < \beta$ . When necessary to clarify the discussion, we may denote the dimension of a  $d$ -cell  $\alpha$  with  $\alpha^{(d)}$ . The *star* of a cell  $\alpha$ , denoted  $St(\alpha)$ , is the set of co-faces of  $\alpha$ . The *lower star* of  $\alpha$ , denoted  $St^-(\alpha)$  is subset of  $St(\alpha)$  where each element has lower function value. The *link* of  $\alpha$  is the closure of the star, minus the star itself,  $Lk(\alpha) = St(\alpha) - St(\alpha)$ . Similarly, the *lower link* of  $\alpha$  is  $Lk^-(\alpha) = St^-(\alpha) - St^-(\alpha)$ .

Let  $K$  be a regular complex that is a mesh representation of  $\mathbb{M}$ . The *barycenter*  $B : K \rightarrow \mathbb{M}$  of a  $d$ -cell  $\alpha \in K$ ,  $B(\alpha)$ , is the average of its vertices. A function  $F : K \rightarrow \mathbb{R}$  that assigns scalar values to every cell of  $K$  is a *discrete Morse function* if for every  $\alpha^{(d)} \in K$ , its number of co-facets  $|\{\beta^{(d+1)} > \alpha | F(\beta) \leq F(\alpha)\}| \leq 1$ , and its number of facets  $|\{\gamma^{(d-1)} < \alpha | F(\gamma) \geq F(\alpha)\}| \leq 1$ . A cell  $\alpha^{(d)}$  is critical if its number of co-facets  $|\{\beta^{(d+1)} > \alpha | F(\beta) \leq F(\alpha)\}| = 0$  and its number of facets  $|\{\gamma^{(d-1)} < \alpha | F(\gamma) \geq F(\alpha)\}| = 0$ , and has index equal  $d$ .

A *vector* in the discrete sense is a pairing of cells  $\langle \alpha^{(d)}, \beta^{(d+1)} \rangle$ , where  $\alpha < \beta$ . We say that an arrow points from  $\alpha^{(d)}$  to  $\beta^{(d+1)}$ . The *direction* of the arrow relates the combinatorial notion of the pairing to the geometric interpretation of the flow, and is given by  $B(\beta^{(d+1)}) - B(\alpha^{(d)})$ . Intuitively, this vector simulates a direction of flow. A *discrete vector field*  $V$  on  $K$  is a collection of pairs  $\langle \alpha_i^{(d)}, \beta_i^{(d+1)} \rangle$  of cells of  $K$  such that each cell is in at most one pair of  $V$ . A critical cell is unpaired. Given a discrete vector field  $V$  on  $K$ , a *V-path* is a sequence of cells

$$\alpha_0^{(d)}, \beta_0^{(d+1)}, \alpha_1^{(d)}, \beta_1^{(d+1)}, \alpha_2^{(d)}, \dots, \beta_r^{(d+1)}, \alpha_{r+1}^{(d)}$$

such that for each  $i = 0, \dots, r$ , the pair  $\langle \alpha_i^{(d)}, \beta_i^{(d+1)} \rangle \in V$ , and  $\alpha_i^{(d)}$  and  $\alpha_{i+1}^{(d)}$  are both facets of  $\beta_i^{(d+1)}$ . A V-path is the discrete equivalent of a streamline in a smooth vector field. A discrete vector field in which all V-paths are monotonic in  $F$  and do not contain any loops is a *discrete gradient field*, denoted  $G$ , of a discrete Morse function. When constructing a discrete gradient field, we say that  $G$  is *valid* if these two conditions are met. The discrete equivalent of *flow* in a continuous gradient is taking a *step* in a V-path, i.e., we say that  $\alpha_{i+1}^{(d)}$  is one step from  $\alpha_i^{(d)}$ . The critical cell  $\alpha_{r+1}^{(d)}$  at the end of a V-path is the *destination* of the V-path. We also say  $\alpha_{r+1}^{(d)}$  *terminates* the V-path.

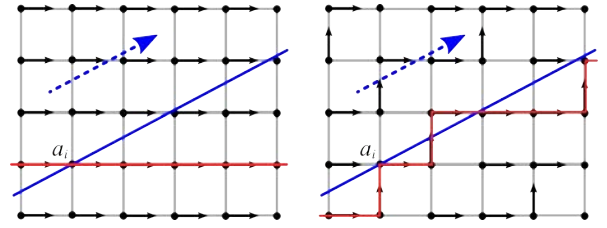


Fig. 2. For a function with a constant negative gradient (blue arrow), approaches using the locally steepest descent produce identical pairings at all vertices (left). The error of the V-path (red) from the integral line (blue) passing through a vertex  $\alpha_i$  accumulates with each step. In our randomized approach, a vertex can be paired with any vertical edge with some probability, producing a V-path with lower overall error (right).

## 3.3 Generic kernel for computing discrete gradient fields

The algorithms surveyed in Section 2, while quite varied in their specific approaches, roughly implement the same underlying functionality. The algorithmic kernel proceeds in the following four steps:

1. Pick a yet unassigned cell
2. Find its potential pairings and assign weights
3. Check for the validity of the pairing
4. Assign gradient arrow of highest weight or declare a critical cell

Each algorithm may perform these steps in unique ways. For example, Lewiner [23] picks the cell whose potential pairing has the highest weight, and uses a search structure on a graph representation of the gradient to check that the pairing not create a cycle. In contrast, Shivshankar et al. [30] first define a discrete Morse function for which the order in which cells are picked is irrelevant. Each cell has one unique possible pair derived directly from the function, and the validity of the pairing is guaranteed by the definition of the function. Gyulassy et al. [13] picks cells in order of increasing function value and dimension to guarantee that any pairing is valid. Robins et al. [29] selects any unassigned vertex, pairs it with the edge in the steepest downwards direction, and assigns the rest of the cells in the lower star of the vertex. Acyclicity is guaranteed by simulation of simplicity used in generating the lower link.

## 4 RANDOMIZED GRADIENT FIELDS

One of the most significant drawbacks of existing techniques is their poor geometric approximation of gradients which leads to incorrect connectivity. Due to their local, greedy assignment of gradient arrows they may produce arbitrary large errors even in regions of constant gradients. Consider the example shown in Fig. 2: The (inverse) gradient of a function is indicated by the blue arrow and assumed to be constant for all cells. Current techniques will, at each vertex, determine the locally best pairing, causing all gradient arrows to be chosen horizontally. Thus, the discrete V-path passing through a vertex  $\alpha_i$  (shown in red) will diverge drastically from the integral line  $L$  in  $f$ . The fundamental problem is that while each arrow is chosen to minimize the local error, all arrows deviate in the same direction from  $-\nabla f$  and collectively cause major artifacts.

We propose a randomized approach that on average will produce significantly better results. Instead of choosing the locally optimal gradient arrow at  $\alpha$ , we pick among all potential valid pairs with a certain probability designed such that the expected V-path will approximate the integral line.

### 4.1 Algorithm

To construct a randomized gradient field we use the standard discrete kernel discussed in Section 3.3 with a modified pairing. Similar to existing techniques, given an unassigned cell  $\alpha$  we compute for each of its co-facets  $\beta_i$  a weight for the potential pair  $\langle \alpha, \beta_i \rangle$ . However, instead of picking the pair with maximal weight, we instead choose *at random* from the potential pairs, with probabilities assigned proportional to the weights. We pair each vertex with an edge probabilistically using these weights, and use the simple homotopy expansion described by Robins et al. [29] to complete the pairings for all other cells in the lower star of the vertex.

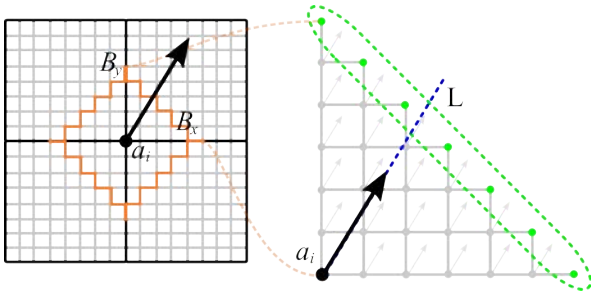


Fig. 3. On the left, the original grid (black) is uniformly subdivided (gray lines) until the a neighborhood (orange outline) around  $\alpha_i$  can be modeled using a piecewise constant gradient. The right shows the lower right quadrant of  $N$ , the cells that have a non-zero probability of being in a V-path passing through  $\alpha_i$ . The green dots denote the potential exit point of  $P^\alpha$  from  $N$ , while the blue dot denotes the exit point of  $J^\alpha$  from  $N$ .

More specifically, let  $\alpha$  be a 0-cell, and  $-\nabla f(B(\alpha))$  be the negative gradient of  $f$  at  $B(\alpha)$  the barycenter  $\alpha$ . The weight of a potential pair  $\langle \alpha, \beta_i \rangle$  is defined as the dot-product between the geometric realization of the pair  $B(\beta_i) - B(\alpha)$  and the negative gradient  $-\nabla f(B(\alpha))$ . Furthermore, all negative weights - indicating invalid pairs - are clamped to zero:

$$w(\langle \alpha, \beta_i \rangle) = \max\{(B(\alpha) - B(\beta_i)) \cdot (-\nabla f(B(\alpha))), 0\} \quad (1)$$

Using these weights, the probability of picking  $\langle \alpha, \beta_i \rangle$  is defined as

$$Pr(\langle \alpha, \beta_i \rangle) = w(\langle \alpha, \beta_i \rangle) / \sum_{\beta_j \in Sr^-(\alpha)} w(\langle \alpha, \beta_j \rangle). \quad (2)$$

Note, that the probabilities are invariant under uniform scaling of both the mesh as well as the gradient magnitudes.

#### 4.2 Geometric Convergence

One of the advantages of our approach is that under subdivision, V-paths in the resulting gradient field will converge to integral lines in  $f$ . We say a V-path containing  $\alpha$  converges to the integral line passing through  $B(\alpha)$  when the Hausdorff distance between the V-path and integral line goes to zero. It is well known that with increasing mesh resolution a piecewise constant approximation of a gradient field will converge to the continuous gradient. In this section we will show that V-paths of the randomized field will converge to integral lines in a given piecewise constant gradient field and thus ultimately to any gradient field. In contrast, existing approaches do not faithfully reproduce even constant gradients, and thus will produce incorrect results even for well resolved functions.

For simplicity let  $\gamma = [0, 1]^2$  be the unit square with a constant gradient  $\nabla f = (-f_x, -f_y)$  and wlg. assume  $f_y > f_x > 0$ . Then the integral line passing through the origin is the line  $L: y = f_y/f_x x$ . Here we will show that if  $\gamma$  is subdivided into a regular  $n \times n$  grid with gradient arrows assigned as discussed above then after  $n$  steps the expected deviation from  $L$  is 0 with a standard deviation that behaves like  $1/\sqrt{n}$ .

Given the assumptions above, when pairing a vertex, only potential pairs in positive  $x$  or positive  $y$  direction in  $\gamma$  have non-zero weights. All potential pairs in the positive  $x$  direction in  $\gamma$  will have a weight of  $w(\langle \alpha, \beta_x \rangle) = (1, 0) \cdot (f_x, f_y) = f_x$  and all potential pairs in the positive  $y$  direction have weight  $w(\langle \alpha, \beta_y \rangle) = f_y$ . Therefore, the probability of pairing a vertex with its horizontal edge  $\beta_x$  is:

$$Pr(\langle \alpha, \beta_x \rangle) = \frac{f_x}{f_x + f_y} = \frac{f_x + f_y - f_y}{f_x + f_y} = 1 - Pr(\langle \alpha, \beta_y \rangle).$$

Consider assigning gradient arrows starting at the origin one pair at a time. Each assignment extends the V-path containing the origin by one pair. Each time we step to the right with probability  $Pr(\langle \alpha, \beta_x \rangle)$  and step upwards with probability  $Pr(\langle \alpha, \beta_y \rangle)$ . The number of horizontal steps in  $n$  trials follows a binomial distribution with parameters  $n$  and  $Pr(\langle \alpha, \beta_x \rangle)$ . Therefore, the expected number of horizontal steps after  $n$  trials is  $nPr(\langle \alpha, \beta_x \rangle)$  and the expected number of ver-

tical steps is  $n - nPr(\langle \alpha, \beta_x \rangle)$ . Normalizing by the grid size, it follows that the expected endpoint of the V-path starting at the origin is  $(Pr(\langle \alpha, \beta_x \rangle), Pr(\langle \alpha, \beta_y \rangle))$ , right on  $L$ . Furthermore, the standard deviation of the binomial distribution normalized by the grid size is  $Pr(\langle \alpha, \beta_x \rangle)Pr(\langle \alpha, \beta_y \rangle)/\sqrt{n}$ .

The same argument holds for unit cubes of higher dimensions using multinomial distributions. Let the probabilities of pairing an edge in the direction  $x_i$  be  $Pr(\langle \alpha, \beta_{x_i} \rangle)$  then the expected endpoint of the V-path after  $n$  steps will be  $(Pr(\langle \alpha, \beta_{x_0} \rangle), \dots, Pr(\langle \alpha, \beta_{x_d} \rangle))$  with the standard deviation in dimension  $i$  given by  $(Pr(\langle \alpha, \beta_{x_i} \rangle)(1 - Pr(\langle \alpha, \beta_{x_i} \rangle)))/\sqrt{n}$ . This proves that with increasing grid resolution, V-paths of a randomized gradient field will converge to integral lines of a given piecewise constant gradient with standard deviation approaching zero. Given that any gradient field can be approximated up to an arbitrarily small error by a piecewise constant field, V-paths in a randomized gradient field will converge to integral lines of the continuous gradient under subdivision.

#### 4.3 Randomized Gradient Fields of Sampled Functions

In practice, we are primarily dealing with sampled functions in which case we must estimate the gradient vector at vertices. There exist two options: First, one can use any of the standard gradient estimation techniques to compute a gradient per vertex and apply the algorithm as described above. Second, one may compute a modified version of the weight that admits gradient discontinuities on the boundary of cells. When using a per-vertex gradient computed with a gradient estimation technique, such as central differences, it is possible that the gradient arrow point in a direction that is locally "uphill", e.g. in a direction of increasing function value. To avoid potential cycles the weight of each potential pair in the lower star will be set to zero, and the vertex will be marked critical.

For simplicity and/or in a parallel environment it may be convenient to compute a gradient per highest dimensional cell (i.e. quads in 2D, voxels in 3D). In this case all lower dimensional cells have multiple gradient vectors assigned to them. In this case Equation 1 must be changed to use the appropriate gradient for each  $\beta_i$ . Note that for regular grids the dot product of, for example, an edge in a two dimensional mesh with its two gradients on either side will naturally produce consistent values. In fact, since the probabilities are scale independent one can assume a regular grid with edges of length one in which case weights can be computed as the difference in function values at their endpoints.

### 5 A DETERMINISTIC APPROACH FOR ACCURATE GEOMETRY

The algorithm discussed above will construct a gradient field that on average is correct and will converge with increasing sampling rates to the gradient of  $f$ . However, in practice subdivision is typically infeasible for larger data sets. Therefore, the true value of the randomized gradient field is that it is expected to produce a good approximation of  $f$ 's gradient even though the quality of any individual V-path cannot be guaranteed. This lack of guarantees is inconvenient especially for large data sets where the computation cannot be easily repeated but geometric quality is important.

In this section we present an extension of randomize gradient fields that leads to a deterministic algorithm guaranteed to extract high quality geometry. The algorithm is based on the insight that the geometry of the MS complex depends only on a small fraction of V-paths, those on the boundaries between ascending and descending manifolds. In particular, one may rephrase the problem of constructing a high quality MS complex as constructing a discrete gradient in which arrows do not cross a-/descending manifold boundaries. To this end we integrate the probabilities defined in Section 4 in order of increasing function value to compute for each cell the probability of its V-path ending at a particular critical cell. We then assign gradient arrows to minimize the number of arrows crossing manifold boundaries. Assigning all gradient arrows of, for example, a two dimensional grid according to their minima distribution leaves exactly those cells unassigned that form the boundary of the ascending manifolds. These cells are filled in by assigning cells according to their saddle distributions. In a second pass we then construct the distributions according to maxima to compute the boundaries between descending manifolds but restrict the computation to be consistent with the first pass.

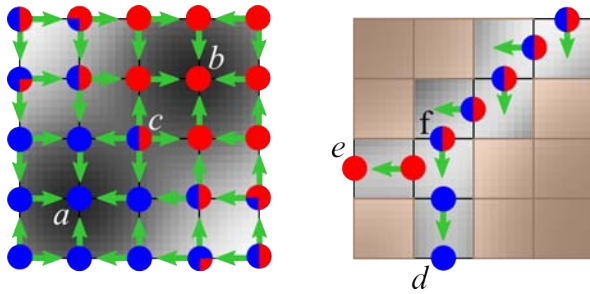


Fig. 4. This figure represents member distributions of cells as a pie-charts, *i.e.*, the area of a particular color indicates the probability that the cell will flow to the critical point represented by the color. On the left, there are two minima,  $a$  and  $b$ , colored blue, and red, respectively. The green arrows indicate the discrete gradient arrows that have non-zero probability of being picked in a randomized approach. At vertex  $c$ , all four directions have equal probability of being paired and therefore, the member distributions of the vertices at the other end of each arrow are averaged to get  $\mu_c$ . The image on the right illustrates a scenario where a V-path splits. In this case,  $\mu_f$  represents this split by assigning equal probability to  $e$  and  $d$ . The shaded area indicates cells that have zero probability of belonging to any V-path terminating at a critical 1-cell.

### 5.1 Computing Membership Distributions

We represent the probabilities of a (randomized) V-path ending at certain critical cells as a *membership distribution*  $\mu_\alpha : K \rightarrow \mathbb{R}$  which for each  $d$ -cell  $\alpha$  and all cells  $\kappa \in K$  defines the probability of the V-path containing  $\alpha$  to end at  $\kappa$ . Membership distributions are defined recursively starting at critical cells. In the following, let  $\beta$  be a co-facet of  $\alpha$ , and  $Lk_\beta^-(\alpha)$  denote the set of  $d$ -cells in the lower link of  $\alpha$  that are also facets of  $\beta$ . Let  $Pr$  denote the selection probability function from Equation 2 in Section 4.1. Then:

$$\mu_\alpha(\kappa) = \begin{cases} 0 & \text{if } \dim(\alpha) \neq \dim(\kappa) \\ 1 & \text{if } \alpha \text{ is critical and } \kappa = \alpha \\ 0 & \text{if } \alpha \text{ is critical and } \kappa \neq \alpha \\ \sum_{\beta \in St^-(\alpha)} Pr(\langle \alpha, \beta \rangle) \sum_{\rho \in Lk_\beta^-(\alpha)} \frac{\mu_\rho(\kappa)}{|Lk_\beta^-(\alpha)|} & \text{otherwise.} \end{cases}$$

The first case reflects the fact that  $\mu_\alpha$  will only be used to assign arrows between  $d$ -cells and  $(d+1)$ -cells and thus we are only concerned with the index- $d$  critical cells that can be reached from  $\alpha$ . In the second case,  $\alpha$  itself is critical, therefore it has probability one since all possible V-paths containing  $\alpha$  must terminate there. Consequently, in the third case, if  $\alpha$  is critical the probability of reaching any other cell is zero. The final case defines the membership distribution of a cell recursively as the weighted combination of the membership distributions of cells in its lower link. Intuitively, the probabilities indicate the likelihood that a V-path containing  $\langle \alpha, \beta \rangle$  exists which when multiplied with the probability that a V-path containing  $\rho$  ( $\beta$ 's other facet) ends at  $\kappa$  defines the probability that a path containing  $\alpha$  reaches  $\kappa$  via  $\langle \alpha, \beta \rangle$ . Summing over all  $\beta$ 's, thus, computes the probability that any V-path containing  $\alpha$  terminates at  $\kappa$ .

Since the membership distribution of any cell depends only on membership distributions of cells with lower function value, these distributions can be computed efficiently by processing cells in order of increasing value.

### 5.2 Gradient Computation

The general framework we use for assigning discrete gradient arrows borrows heavily from published techniques [12–14, 23, 29]. For example, we extend a function sampled at vertices to values at every cell to assigning a cell the value of its highest vertex. We use the region-growing approach introduced by Gyulassy et al. [13] to assign discrete gradient arrows on the interior of an ascending manifold. We then perform simple homotopy expansions [12, 29]. Indeed, as with every previous technique, we pair a discrete gradient arrow in the direction that maximizes the weight of the pairing. However, the new weight function derived below assumes a particular order of assignment, which

requires subtle variations of existing approaches.

In our algorithm, we first compute the  $d$ -cell to  $(d+1)$  cell arrows on the interior of an ascending  $(D-d)$ -manifold, where  $D$  denotes the maximum dimension of a cell in the mesh  $K$ . We then assign all possible discrete gradient arrows that preserve the simple homotopy type of the ascending manifolds. For volumetric data, we first assign all cells in the interior of ascending three-manifolds, then ascending two-manifolds, etc. until all cells are assigned. Fig. 5 illustrates these steps for a simple two-dimensional example. Below we report pseudocode for the corresponding algorithm.

In the following, let  $K$  be the underlying mesh,  $F$  a function mapping a cell to a scalar value, and  $G$  the discrete gradient field. The gradient field is defined as a set of pairs of cells, where each pair represents an arrow pointing from the lower to the higher dimensional cell. We pair a cell with itself to denote criticality. We call cells *assigned* if they can be found in  $G$ .

```

1: ComputeGradient(K, F) :
2: G = {}
3: for d ∈ [0, D] do
4:   G = AssignArrows(d, K, F, G)
5:   G = HomotopyExpand(K, F, G)
6: end for
7: return G

```

The functions `AssignArrows()` and `HomotopyExpand()` simply add new pairs to the discrete gradient field  $G$ .

**Weight of a Pairing.** In the process of computing the discrete gradient field, our technique assigns a weight to a potential pair  $\langle \alpha, \beta_i \rangle$  based on (1) the known destinations (critical cells) of V-paths assigned so far in  $G$ , and (2) the membership distribution  $\mu_\alpha$  of the lower dimensional cell. Let  $\alpha$  be an unassigned  $d$ -cell, and  $C$  be the set of  $(d+1)$ -cells it can be paired with. Formally,  $C$  is the set of unassigned cells in the lower star of  $\alpha$  that are its co-facets, *i.e.*,  $C = \{St^-(\alpha_i) \mid \beta_i \succ \alpha, \beta_i \notin G\}$ . We call  $C$  *candidates*, since pairing  $\alpha$  with any cell in  $C$  produces a valid gradient field. For each  $\beta_i \in C$ , we assign a weight to the potential pair  $\langle \alpha, \beta_i \rangle$  to minimize the number of arrows crossing manifold boundaries. When pairing  $\alpha$ , each cell in the lower link  $\gamma_i \in Lk^-(\alpha)$  has already been paired. In fact,  $\gamma_i$  is part of V-paths that do not change below  $\gamma_i$  with any subsequent gradient arrow assignments, and we can find their terminating critical cells. It is well-known that in dimensions higher than one, V-paths can split, for example, as illustrated in Fig. 4 (right). Therefore, let  $D_\gamma$  be the set of critical cells that terminate V-paths flowing through  $\gamma_i$ . The weight of  $\langle \alpha, \beta_i \rangle$  is defined as:

$$w(\langle \alpha, \beta_i \rangle) = \sum_{\gamma \in Lk^-(\alpha), \gamma \prec \beta_i} \left( \sum_{\kappa \in D_\gamma} \mu_\alpha(\kappa) \right) \quad (3)$$

Intuitively, this weight represents the likelihood that  $\alpha$  belongs to the same ascending manifolds as the facets of  $\beta$ . Therefore, higher weights indicate potential pairs that are less likely to cross boundaries of ascending manifolds. Just as the definition of the distributions, the weight depends on the fact that every  $d$ -cell in the sub-level complex of  $F(\alpha)$  has been assigned. Therefore every cell in the lower link of  $\alpha$  belongs to assigned V-paths terminating at critical cells in  $G$ . Processing cells in order of increasing function value as described below guarantees this property.

**Assigning Gradient Arrows.** The following algorithm computes the  $d$ -cell to  $(d+1)$ -cell gradient arrows on the interior of an ascending  $(D-d)$ -manifold. The unassigned  $d$ -cells in  $K$  are processed in order of increasing function value, with ties broken by simulation of simplicity [10]. By processing cells in sorted order, in effect, we are growing the spanning trees of  $d$ -cells of the sub-level complex of  $K$ . As each  $d$ -cell is visited by the algorithm, its potential pairs, the candidates  $C$  are identified. If there are no candidates for pairing, the  $d$ -cell is assigned critical. Otherwise, the one maximizing the weight function given in Equation 3 is chosen. The algorithm terminates when all  $d$ -cells are assigned.

```

1: AssignArrows(d, K, F, G) :
2:  $K_d = \{\alpha^d \in K \mid \text{dimension of } \alpha \text{ is } d\}$ 
3: sort( $K_d$ )

```

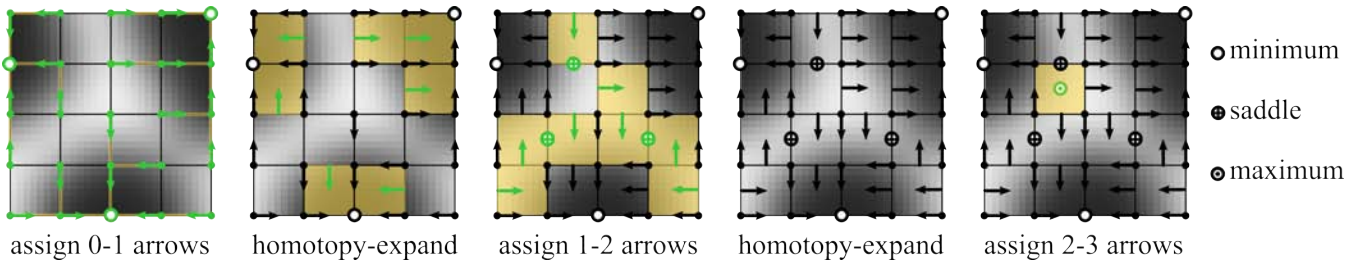


Fig. 5. A discrete gradient field of a sampled function indicated in gray is computed using the algorithms of Section 5. From left to right: First, we pair vertices and edges according to the maximal weight, then pair edges and faces in a simple homotopy expansion. Second, we pair the remaining edges and cells according to their weight and faces and voxels (not shown) in the corresponding homotopy expansion. Finally, we pair faces and voxels by weight and mark all remaining unpaired voxels as critical.

```

4: for  $\alpha_i \in K_d$  do
5:    $C = \{\beta \in St^-(\alpha_i) \mid \beta \succ \alpha\}$ 
6:   if  $C = \emptyset$  then
7:      $G = G \cup \langle \alpha_i, \alpha_i \rangle$ 
8:   else
9:      $\beta_j = \operatorname{argmax}_{\beta_i \in C}(w(\langle \alpha, \beta_i \rangle))$ 
10:     $G = G \cup \langle \alpha_i, \beta_j \rangle$ 
11:   end if
12: end for
13: return G

```

The function  $\operatorname{argmax}()$  simply returns the argument that maximizes its value, *i.e.*, the  $(d+1)$ -cell where the pair  $\{\alpha, \beta_i\}$  has highest weight.

**Simple Homotopy Expansion** Let  $K^n \subseteq K$  be the subcomplex of assigned cells of  $K$  after selecting  $n$  pairs. Assigning a gradient arrow adds exactly two cells to this subcomplex. An assignment of a gradient arrow is a *simple homotopy expansion* if  $K^n$  is homotopic to  $K^{n+1}$ . In practice, a  $d$ -cell to  $(d+1)$ -cell arrow can be inserted without changing the homotopy type of the subcomplex when (1) all faces of the  $d$  cell are assigned, and (2) the  $d$ -cell is the only unassigned face of the  $(d+1)$ -cell. The only time we prohibit this expansion is when the arrow would point “uphill”, *i.e.*, the value of the  $d+1$  cell is strictly larger than the value of the  $d$  cell in  $F$ .

```

1: HomotopyExpand( $d, K, F, G$ ) :
2:  $H = \{\alpha \in K \mid \# \text{ unassigned facets in } G \text{ is } 1\}$ 
3: while  $H \neq \emptyset$  do
4:    $\alpha = \operatorname{PopFirst}(H)$ 
5:    $\beta = \text{unassigned facet of } \alpha$ 
6:   if  $F(\alpha) \leq F(\beta)$  then
7:      $G = G \cup \langle \alpha, \beta \rangle$ 
8:     update( $H$ )
9:   end if
10: end while
11: return G

```

When a gradient arrow  $\langle \alpha, \beta \rangle$  is assigned, the number of unassigned facets of all co-facets of  $\alpha$  and  $\beta$  decreases, and  $\operatorname{update}()$  inserts the co-facets having exactly one unassigned facet into  $H$ .

### 5.3 A Two-Pass Approach

The algorithm described above only takes the geometric accuracy of ascending manifolds into account. To get both accurate ascending and descending manifolds, we use the result of the first pass to restrict a second pass. In the second pass, we repeat the algorithm on the complement of  $K$  using the negative of  $F$  with one change: We only consider candidates for pairing cells belonging to the interior of the same dimensional ascending manifold. Figure 6 illustrates this approach.

## 6 RESULTS

In this section we present experiments comparing the geometric quality of the MS complex computed using previous steepest-descent techniques, the new randomized approach, and the two-pass, deterministic variant. We compute and visualize the MS complex from a discrete gradient field using the techniques described by Gyulassy et al. [16]. For clarity we begin with simple two-dimensional examples. Fig. 7

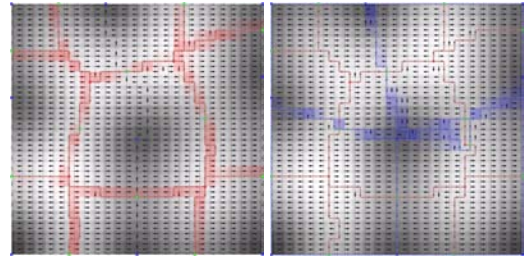


Fig. 6. A first pass computes accurate geometry for the ascending manifolds (left). In the second pass (right), arrows in ascending 1-manifolds (pink lines) may only be paired with one another. In each image, the hazy cells indicate the maximum probability of a cell belonging to a particular manifold - the darker the color, the lower the probability. The boundaries between cells in our MS complex naturally occur where probability of membership is lowest.

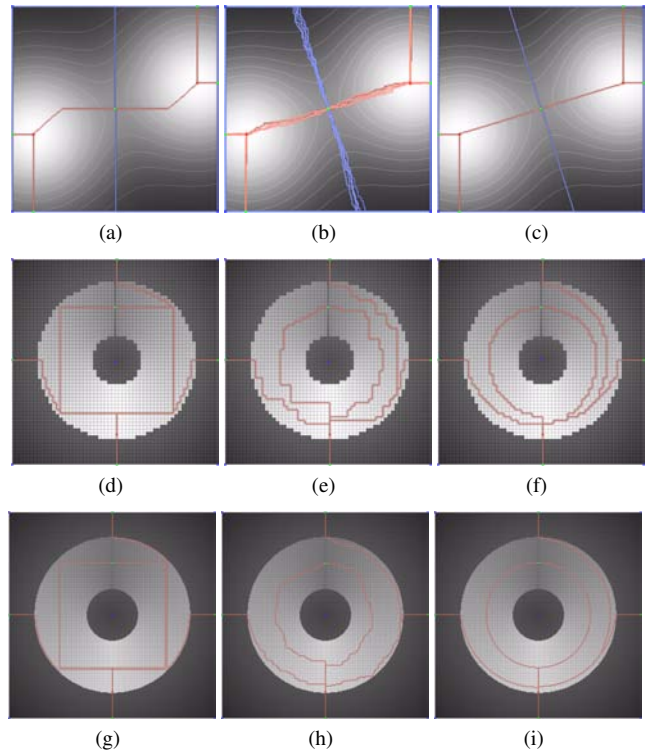


Fig. 7. The top row, (a-c) show the function from Fig. 1, up-sampled four times in each direction. Note that the steepest-descent construction in (a) does not improve with increased mesh resolution. We show five realizations of our randomized approach (b), and our near-optimal approach (c). In (d-f) we show the same techniques applied to  $f(x,y) = \arctan(-y/|x|)$  for  $r_1 < \|(x,y)\| < r_2$ . In these examples, the smooth integral line starting at the saddle forms a circle. The same function is used sampled four times finer in each direction in (g-i).

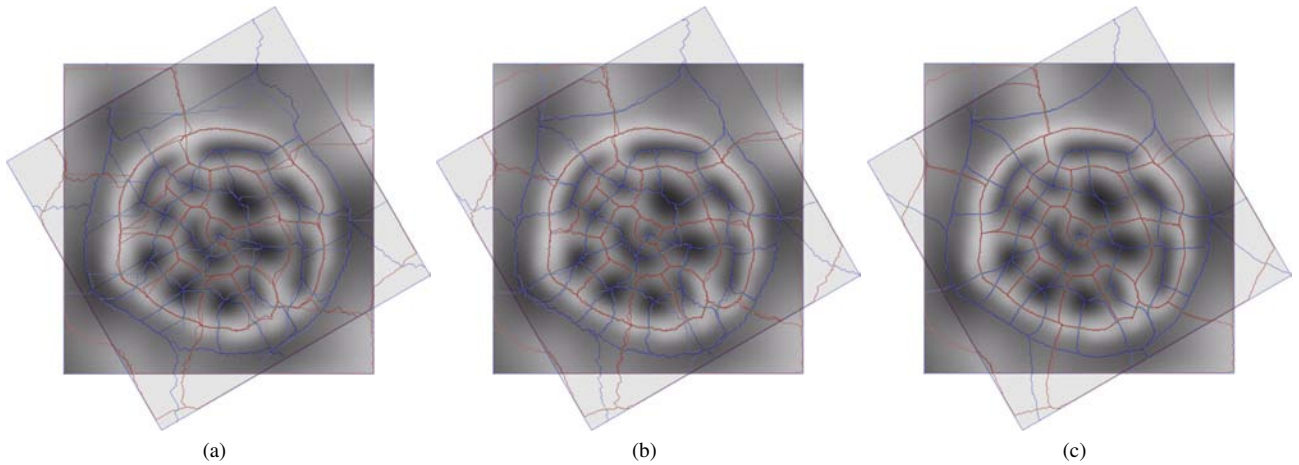


Fig. 8. These images illustrate the dependence of the steepest-descent construction (a) on the mesh orientation, and the resilience of the two proposed techniques (b, c). The same underlying function is re-sampled onto one grid oriented vertically, and another grid rotated 30 degrees from vertical. We overlay the complexes computed for each, aligning them with the orientation of the underlying function. Excluding boundary artifacts, the complexes of the locally maximizing approach (a) vary significantly, the randomized approach (b) displays better behavior, and the complexes of our two-pass approach (c) vary only by the width of a cell.

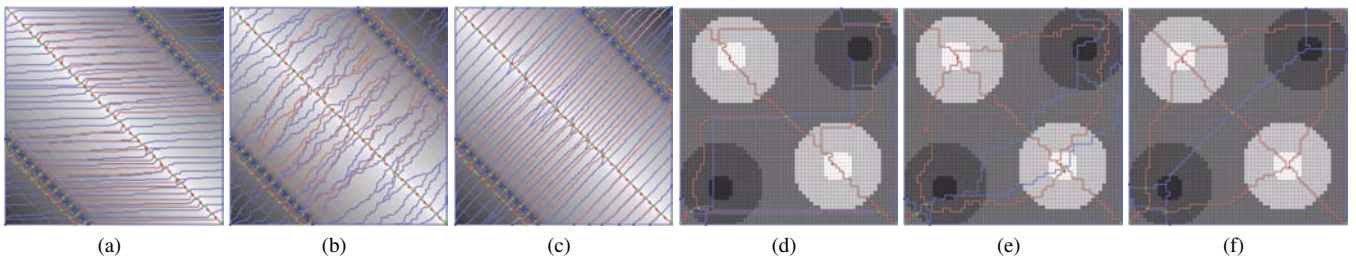


Fig. 9. In (a-c), minima along two “valleys” and maxima along a “ridge” are separated by saddles. While the complex generated by the steepest descent approach (a) is *consistent*, it connects the saddles and extrema poorly. Our randomized approach performs better (b), and the two-pass approach (c) produces a very accurate solution. In (d-f) we show the resilience of our techniques to flat regions in the data. (d) The region growing variant of the steepest-descent algorithm described by Gyulassy et al. [12] provides a reasonable approximation to ascending 1-manifolds (pink lines). However, the descending lines simply follow the steepest-descent direction given by simulation of simplicity [10]. We apply the same order of assignment, but with our randomized kernel (b), resulting in a more intuitive traversal of flat regions. In the two-pass approach, distance (in the number of steps in a V-path) naturally factors into the weight function, steering the geometry of the complex perpendicularly away from the boundary of a flat region.

and 1 show how each algorithm responds when the mesh is subdivided. We sample two smooth functions on a regular grid: The first is the sum of two Gaussians; The second is a function with semi-circular integral lines. In both cases existing approaches create severe artifacts in both direction and shape of the one-manifolds. Moreover, the artifacts are unaffected by the increase in resolution. The randomized algorithm, while producing somewhat wavy patterns, already significantly improves the geometry and converges to the correct solution with increasing mesh resolution. The two pass approach extracts the correct geometry up to the resolution of the mesh, and a higher resolution mesh unsurprisingly allows more accurate geometry.

Fig. 8 shows the resilience of each algorithm to changes in the orientation of the underlying mesh. Here, an underlying function is sampled by two grids, rotated 30 degrees from one another. The computed complexes are overlaid, for each technique, for visual comparison. Again the steepest-descent assignment shows large variations and artifacts, while the randomized approach shows a good correspondence, and the two-pass algorithm extracts identical geometry to the extent possible.

Fig. 9(a-c) compares the topological correctness of each technique. In this case, we connect saddles along a perturbed valley to maxima on a perturbed ridge, and saddles from the ridge to the minima. In this case we consider the topological information encoded by an arc of the MS complex to be *correct* when the critical points at the ends are connected by an integral line in the underlying function. An infinitesimal perturbation of the function can redirect integral lines, therefore any MS complex that is *consistent* is a valid output (a quasi-MS complex in the sense of [9]). However, the output ideally should be both

consistent and correct for some smooth interpretation of the sampled function. In this sense, our randomized and two-pass techniques produce a far more correct complex.

Imaged data is often captured with limited precision or quantized to reduce its size. This often leads to degenerate regions with zero gradients everywhere. Therefore, the behavior of algorithms in such “flat” regions is of significant interest. In Fig. 9(d-f) we compare the results of our new algorithms against the best know previous technique of Gyulassy et al. [14], where gradient arrows are assigned in a breadth-first order to route V-paths efficiently across flat regions.

Finally, Fig. 10 compares our algorithms against steepest-descent techniques for two volumetric data sets. As before, the steepest-descent algorithm produces significant artifacts as exemplified by the shape of the one-manifolds for the tetrahedrane. Note that in the jet example, the mangled geometry causes entire Morse crystals to disappear leading to significant changes in the structure of the complex.

Results were generated on a commodity laptop, with 4Gb main memory and 2.2GHz Intel processor. Our algorithms were implemented in C++. The running time and total memory used for each example is summarized in table 1. The table shows results for both a serial implementation of the randomized approach, as well as the deterministic algorithm for computing accurate geometry. Storing the membership distributions was responsible for the large memory footprint of the latter approach.

## 7 DISCUSSION

Our results highlight the difficult cases for a local steepest-descent algorithm. In many cases, such an algorithm produces acceptable re-

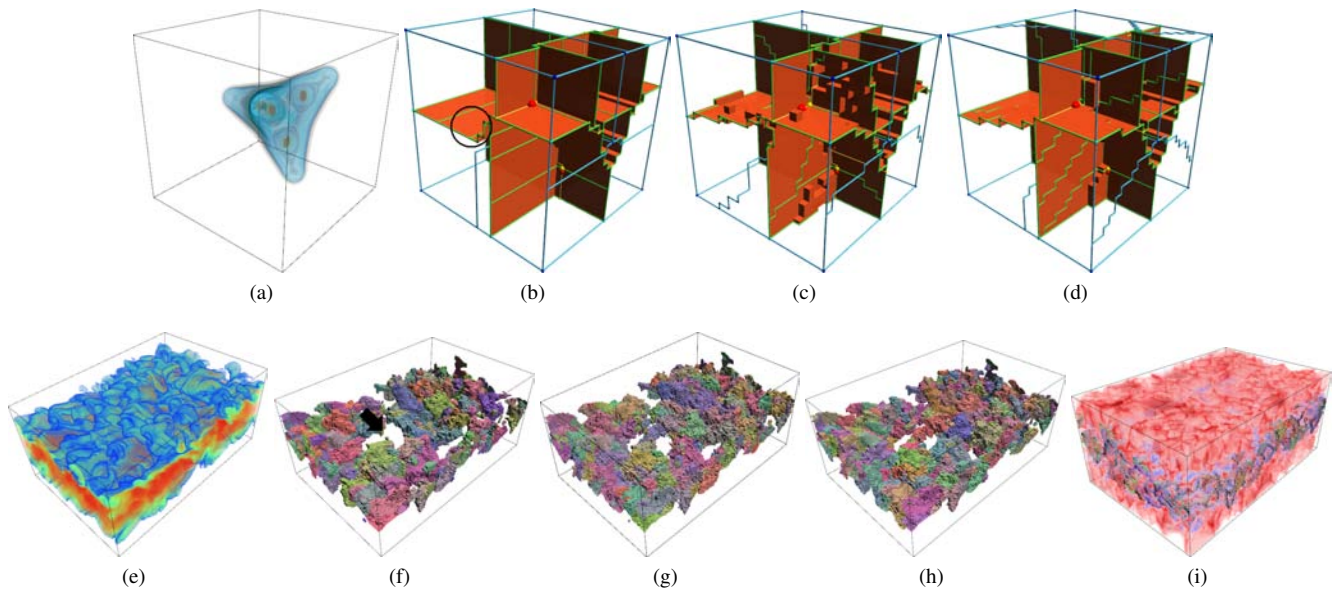


Fig. 10. We show results for two three dimensional datasets. The top row shows the ascending 2-manifolds, arcs, and nodes of the electron probability density function of a tetrahedrane( $C_4H_4$ ) molecule (a). (b) Uses the steepest-descent, (c) the randomized, and (d) the two-pass approach. Note the shape of the arcs, and the gradual decline of the surface of (d) with respect to (b)(circled). Figure 1(e) shows the integrated probability functions for this dataset. The bottom row (e-i) displays the *dissipation elements* in a cross-flow jet flame. Each element is defined by the boundary of an ascending 3-manifold on the interior of the flame, and the same filters are used to select from each complex computed using steepest-descent (f), randomized (g), and two-pass (h) gradient fields. Several of the ascending 3-manifolds in the complex computed using steepest-descent (f) are missing (arrow), possibly due to handle slides. (i) shows the integrated probabilities for our two-pass approach, with blue indicating low ascending manifold membership probability, and red indicating low descending manifold membership probability.

Table 1. Run time and maximum memory usage to construct the discrete gradient field using the randomized approach(section 4) and the deterministic one(section 5). The times reported for the randomized approach are for a single thread.

	XxY(xZ)	# Vert	Time(s)	Randomized		Deterministic	
				Memory(Mb)	Time(s)	Memory(Mb)	
gauss60	60x60	3600	0.047	<1	0.124	2	
gauss240	240x240	57,600	0.546	4	2.044	12	
JetSlice	768x512	393,216	4.509	14	16.54	49	
C4H4	24x24x24	13,824	0.374	5	5.7	18	
Fuel	64x64x64	262,144	4.509	12	14.64	40	
JetChi	192x84x128	2,064,384	57.02	100	256.3	411	

sults. In particular, sharp features, such as sharp ridge and valley lines are typically well-represented by a steepest-descent algorithm. In these cases, our approaches perform just as well, since there tends to be little variation in the probable membership of a cell.

In practice, we represent membership distributions as a sparse map, and compute them during the region-growing construction. In this case, only the membership distributions on the growing front of our region need to be available. When every  $d$ -dimensional neighbor of a  $d$ -cell has been assigned, its membership distribution can safely be discarded. In our implementation, the memory requirements of membership distributions is given by a constant times the number of cells crossed by the largest isosurface. The constant is bounded by the maximum number of critical  $d$ -cells  $\alpha$  could flow to. In equation 3, the set  $D_\gamma$  of critical  $d$ -cells that terminate the assigned V-paths containing a cell is also computed incrementally. When a gradient arrow is assigned into a pair  $\langle \alpha, \beta \rangle$ ,  $D_\alpha$  it is simply the union of  $D_\gamma$  for  $\gamma < \beta$ . In practice, the running time of our algorithm is  $C^2 * n \log n$ .

The one case our two-pass algorithm does not handle correctly is the interior of a topological *strangulation*. The geometric accuracy of our approach depends on finding the boundary between distinct regions - in a strangulation, an a-/descending manifold borders itself. In this case, when each potential pairing has equal weight, we resort to steepest-descent.

## 8 CONCLUSION AND FUTURE DIRECTIONS

We have presented two new approaches for computing discrete gradient fields that better approximate the gradient flow a scalar function.

Our first technique is simple and can be plugged in as-is to parallel computation of the MS complex. Our second technique provides accurate geometry, but at the cost of serial computation. As the MS complex is becoming more widely used in analysis, geometric guarantees become necessary. We will investigate techniques for providing numerical error estimates for analysis performed using the MS complex. The most significant current limitation is the heavier memory footprint of the two-pass approach. Although the two-pass algorithm computation is serial, there are several potential avenues for its parallelization. We are investigating a GPU implementation of the membership distribution computation, maintaining efficiency by only keeping the most probable elements of  $\mu$ . We plan on addressing the memory limitation initially through a divide-and-conquer approach on a distributed memory system. Alternatively, the complex could be computed in parallel using any previous technique, with a multi-threaded approach applying the two-pass algorithm to “fix” the gradient on independent sub-regions delimited by the boundaries of existing ascending and descending manifolds.

## ACKNOWLEDGMENTS

This work is supported in part by NSF OCI-0906379, NSF OCI-0904631, DOE/NEUP 120341, DOE/MAPD DESC000192, DOE/LLNL B597476, DOE/Codesign P01180734, and DOE/SciDAC DESC0007446.

## REFERENCES

- [1] E. Babson and P. Hersh. Discrete Morse functions from lexicographic orders. *Transactions of the American Mathematical Society*, 3(457):509–534, 2005.
- [2] J. Bennett, V. Krishnamurthy, S. Liu, V. Pascucci, R. Grout, J. Chen, and P.-T. Bremer. Feature-based statistical analysis of combustion simulation data. *IEEE Transactions Visualization and Computer Graphics*, 17(12):1822–1831, 2011.
- [3] P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):385–396, 2004.
- [4] P.-T. Bremer, G. Weber, V. Pascucci, M. Day, and J. Bell. Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):248–260, 2010.
- [5] P.-T. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. B. Bell. Interactive exploration and analysis of large scale simulations using topology-based data segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 17(99), 2010.
- [6] A. Cayley. On contour and slope lines. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, XVIII:264–268, 1859.
- [7] F. Cazals, F. Chazal, and T. Lewiner. Molecular shape analysis based upon the morse-smale complex and the connolly function. In *Proceedings of the 19th Symposium on Computational Geometry*, SCG '03, pages 351–360, New York, NY, USA, 2003. ACM.
- [8] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *Proceedings of the 19th Symposium on Computational Geometry*, pages 361–370, 2003.
- [9] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete Computational Geometry*, 30:87–107, 2003.
- [10] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics (TOG)*, 9:66–104, 1990.
- [11] R. Forman. A user's guide to discrete Morse theory. In *Séminaire Lotharinen de Combinatoire*, volume 48, 2002.
- [12] A. Gyulassy. *Combinatorial Construction of Morse-Smale Complexes for Data Analysis and Visualization*. PhD thesis, University of California, Davis, 2008.
- [13] A. Gyulassy, P.-T. Bremer, V. Pascucci, and B. Hamann. A practical approach to Morse-Smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008.
- [14] A. Gyulassy, P.-T. Bremer, V. Pascucci, and B. Hamann. Practical considerations in Morse-Smale complex computation. In *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, Mathematics and Visualization, pages 67–78. Springer, 2011.
- [15] A. Gyulassy, M. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann. Topologically clean distance fields. *IEEE Transactions on Computer Graphics and Visualization*, 13(6):1432–1439, 2007.
- [16] A. Gyulassy, N. Kotava, M. Kim, C. D. Hansen, H. Hagen, and V. Pascucci. Direct feature visualization using Morse-Smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints), 2011.
- [17] A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, and B. Hamann. Topology-based simplification for feature extraction from 3D scalar fields. *IEEE Transactions on Computer Graphics and Visualization*, 12(4):474–484, 2006.
- [18] A. Gyulassy, T. Peterka, V. Pascucci, and R. Ross. Characterizing the parallel computation of Morse-Smale complexes. In *Proceedings of IPDPS '12*, Shanghai, China, 2012.
- [19] J. Kasten, J. Reininghaus, I. Hotz, and H.-C. Hege. Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2080–2087, 2011.
- [20] H. King, K. Knudson, and M. Neza. Generating discrete Morse functions from point data. *Experimental Mathematics*, 14(4):435–444, 2005.
- [21] M. A. Koch, D. G. Norris, and M. Hund-Georgiadis. An investigation of functional and anatomical connectivity using magnetic resonance imaging. *NeuroImage*, 16(1):241–250, 2002.
- [22] D. Laney, P.-T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Transactions Visualization and Computer Graphics*, 12(5):1052–1060, 2006.
- [23] T. Lewiner. Constructing discrete Morse functions. Master's thesis, Department of Mathematics, PUC-Rio, 2002.
- [24] T. Lewiner. Critical sets in discrete morse theories: relating forman and piecewise-linear approaches. *Computer Aided Geometric Design*, 2012.
- [25] T. Lewiner, H. Lopes, and G. Tavares. Applications of forman's discrete morse theory to topology visualization and mesh compression. *IEEE Transactions on Visualization and Computer Graphics*, 10:499–508, 2004.
- [26] J. C. Maxwell. On hills and dales. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, XL:421–427, 1870.
- [27] J. Reininghaus and I. Hotz. Combinatorial 2d vector field topology extraction and simplification. In *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, Mathematics and Visualization, pages 103–114. Springer, 2011.
- [28] J. Reininghaus, C. Lowen, and I. Hotz. Fast combinatorial vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 17:1433–1443, 2011.
- [29] V. Robins, P. Wood, and A. Sheppard. Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1646–1658, 2011.
- [30] N. Shivashankar, M. Senthilnathan, and V. Natarajan. Parallel computation of 2d Morse-Smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, to appear, 2012.