Fair Morse Functions for Extracting the Topological Structure of a Surface Mesh

Xinlai Ni Michael Garland John C. Hart

Univ. of Illinois, Urbana-Champaign

Abstract

Morse theory reveals the topological structure of a shape based on the critical points of a real function over the shape. A poor choice of this real function can lead to a complex configuration of an unnecessarily high number of critical points. This paper solves a relaxed form of Laplace's equation to find a "fair" Morse function with a user-controlled number and configuration of critical points. When the number is minimal, the resulting Morse complex cuts the shape into a disk. Specifying additional critical points at surface features yields a base domain that better represents the geometry and shares the same topology as the original mesh, and can also cluster a mesh into approximately developable patches. We make Morse theory on meshes more robust with teflon saddles and flat edge collapses, and devise a new "intermediate value propagation" multigrid solver for finding fair Morse functions that runs in provably linear time.

Keywords: atlas generation, computational topology, Morse theory, surface parameterization, texture mapping

1 Introduction

Morse theory connects the differential geometry of a surface with its algebraic topology. Given a real function over some shape, it describes the connectedness of the shape from the configuration of the points where the function's gradient vanishes, its so-called critical points (e.g. minima, maxima, saddles).

Morse theory has been used in graphics and visualization to analyze different real functions. Terrain data, e.g., is defined by an altitude function on the plane, and Morse theory can identify topographical features, control their simplification [Bajaj and Schikore 1998], and organize them into a multiresolution hierarchy [Bremer et al. 2003]. The zeroset of a real function over 3-space defines an isosurface, and Morse theory can determine its topology for more accurate polygonization [Stander and Hart 1997].

When given only a shape, the critical points of almost all real functions on its surface can be used to interrogate its topology, but some functions are much better choices than others. The Euler characteristic χ reveals the genus of a closed connected manifold mesh by the formula

$$\chi = \text{vertices} - \text{edges} + \text{faces} = 2 - 2g. \qquad (1)$$

The Euler characteristic can also be calculated from the critical points

$$\chi = \text{minima} - \text{saddles} + \text{maxima}. \qquad (2)$$

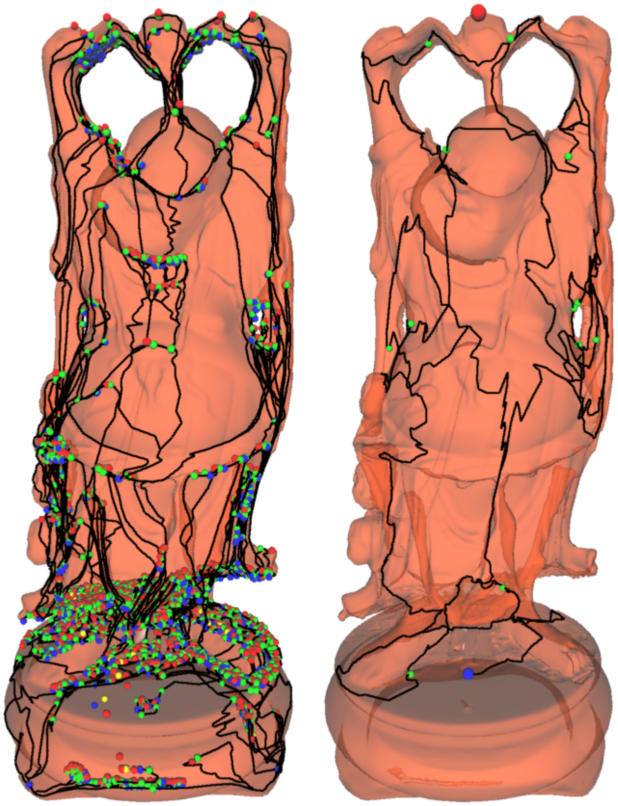


Figure 1: An altitude function (a) yields a complicated arrangement of 3,605 critical points on the genus-6 Buddha. Our method yields a fair Morse function (b) with the least number of critical points, in this case one blue minimum, one red maximum and twelve green saddles. Cutting along the indicated path separates the mesh into a shape topologically a disk suitable for planar parameterization.

By combining these two equations we see that the smallest number of critical points possible on a genus g closed oriented manifold is one minimum, one maximum and $2g$ saddles. However, an arbitrarily chosen Morse function like altitude can yield many more critical points, satisfying the Euler characteristic with any number of additional extrema (minima or maxima) matched by the same number of additional saddles. For example, the altitude function on the genus-6 Buddha model, Fig. 1(a), yields 3,605 critical points. A better choice of function (b) yields the minimum number of 14 Morse critical points for this shape. This paper describes how to find such a function and how to use it to solve various problems in meshed geometry processing.

These extra critical points are caused by the altitude function's sensitivity to surface undulation. A vertical wrinkle in the surface, no matter how small, creates a pair of critical points. These undulations could be removed by smoothing the surface, but it is easier and less destructive to smooth the function. We leave the surface unchanged, and instead apply Laplacian smoothing to the function to remove its "wrinkles" until it converges to a smooth, in fact harmonic, result that yields the least number of critical points. Because we are removing unwanted function variation over the surface, we call this process *Morse fairing* and the resulting function a *fair Morse* function.

Given a fair Morse function, one can trace gradient-descent flowlines down the edges of the mesh from the saddle points to the minimum. These flowlines form a seam that allows the mesh to be cut into a shape topologically equivalent to a disk. Most other methods used in graphics to cut a surface into a disk are based on a region-growing Dijkstra's algorithm. Sec. 2 reveals that Dijkstra's algorithm is in fact an arbitrarily chosen Morse function which leads to extraneous topological events that must be identified and removed.

Sec. 3 reviews Morse theory and the state of the art in its application to meshes. This section offers new solutions that make Morse theory work more robustly on meshes, including managing high-multiplicity saddles, the application of Conley index theory to resolve degenerate "flat" regions, and "teflon" saddles that avoid a degenerate Morse structure.

Sec. 4 describes the Morse fairing process, based on solving a constrained Laplacian over the mesh. A theorem in this section leads to a new "intermediate value propagation" multigrid solver that performs Morse fairing in provably linear time, allowing it to be applied in-core to any size mesh, and outruns an irregular-mesh multigrid Laplacian solver.

Morse fairing can also produce a real function with a user-specified number of critical points, and can place its extrema in user specified positions. Sec. 3 reviews how the gradient-descent paths of this function embed a graph, called the Morse complex, in the meshed surface that contains a vertex for each minimum, a face containing each maximum and the exact arrangement of edges needed to ensure it matches the topology of the mesh. Section 5 describes applications of the faired Morse complex for cutting a surface into a disk, constructing a feature-sensitive topology-correct base domain, clustering faces toward developable charts and visualizing the topology of a complex surface. Sec. 6 concludes with a discussion of the limitations of Morse fairing and directions for further research.

2 Previous Work

Cutting a Surface into a Disk. The problem of cutting a closed genus-g mesh of n vertices into a single flattenable component has been investigated as the polygonal schema problem in computational topology [Vegter 1997; Dey et al. 1999a]. Finding optimal cuts is NP-hard, but cuts within $O(\log^2 g)$ of optimal can be found in $O(g^2 n \log n)$ time [Erickson and Har-Peled 2002], and paths through a common base point can be optimized in polynomial time [de Verdière and Lazarus 2002]. Arbitrary cuts can be found in $O(gn)$ time [Lazarus et al. 2001]. Morse fairing finds the least number of non-optimal cuts through a base point in time linear in the number of vertices, but with an approach that is significantly easier to implement.

Mesh topology methods in computer graphics, including topological noise removal [Guskov and Wood 2001], geometry images [Gu et al. 2002], and feature detection [Zhang et al. 2003], find mesh cuts primarily with region growing [Dey and Schipper 1995]. In a closed manifold, a front expanding from a base point will self intersect, and these self intersections flag the presence of a handle. Front propagation is robust and works well even on meshes with boundaries, and Dijkstra's algorithm can be used when the length of cuts is important.

Front propagation generates a real function over the mesh, when the mesh is sufficiently subdivided, the collisions of these fronts form Morse critical points [Axen and Edelsbrunner 1998]. The choice of distance-to-base-point as the Morse function is arbitrary with respect to the surface topology which makes it unecessarily expensive and prone to generating more critical points than necessary. Morse fairing provides a less expensive real function that generates the same topological information as front propagation, but avoids the maintenance of an prioriry-queued equidistant front and the expense of collision detection.

Morse Theory on Meshes. Morse theory was originally devised for smooth functions on manifolds [Milnor 1963], though it extends elegantly to piecewise linear functions on triangle meshes [Banchoff 1970].

Edelsbrunner *et al.* [2002] further refined the application of Morse theory to meshes. They first define *persistence* as the difference in value between a pair of critical points that would cancel each other after the appropriate perturbation. Persistence prioritizes the cancellation of critical points, which allows one to control the simplification of features in terrain data and general 2-manifolds [Edelsbrunner et al. 2003b; Bremer et al. 2003]. These methods can remove all unnecessary critical points, but do so in order of increasing persistence. Morse fairing leapfrogs this persistence organization and removes unwanted critical points in a single step.

The Reeb graph has also been used in graphics to represent shape topology, e.g. [Shinagawa et al. 1991; Hilaga et al. 2001]. The Reeb graph uses graph topology to represent solid topology (e.g. a Reeb graph cycle represents a torus hole). Similar to us, Steiner & Fischer [2001] also used a mesh Laplacian to simplify topology, but instead generated a simplified Reeb graph that lacked extraneous details from non-topological features. The Morse complex represents topology in a surface embeddable structure, and so makes it more amendable to the application of surface processing.

Laplacian Smoothing. Bajaj *et al.* [1998] observed that smoothing a Morse function with a Gaussian filter cancelled many pairs of unnecessary critical points, which simplified the critical point structure to aid in the visualization of scientific data. Bremer *et al.* [2003] perform iterative Laplacian smoothing steps to cancel critical points in its multiresolution topology hierarchy (and to smooth the jagged 1-cells of the Morse-Smale complex).

Ray & Levy [2003] devise a multigrid Laplacian solver similar to ours to find a least-squares optimal conformal parameterization of a surface triangle mesh. The solution needed for Morse fairing need not be an exact Laplacian, and Sec. 4.3 capitalizes on this property to simplify the multigrid implementation.

3 Morse Theory on Meshes

Morse theory relates the homotopy type of a manifold M with its differential structure specified by the critical points of a Morse function $f : M \to \mathbb{R}$. This section reviews Morse theory for smooth functions, then adapts it to PL functions, defining, classifing and using critical points in the absence of derivatives.

3.1 Critical Points

Milnor [1963] provides a concise, deep but approachable description of Morse theory. Let $\mathbf{p}(\mathbf{u}) \in M \subset \mathbb{R}^3$ be a point on a closed embedded 2-manifold M, in a neighborhood continuously parameterized by $\mathbf{u} = (u_1, u_2)$. Let $f : M \to \mathbb{R}$ be any real function on the manifold. A point is *critical* if its gradient $[\partial f/\partial u_i]$ vanishes, otherwise it is *regular*. A critical point is *Morse* if its Hessian matrix $[\partial^2 f/\partial u_i \partial u_j]$ is non-singular otherwise it is *degenerate*. If and only if all its critical points are Morse, then the function f is a

Morse function. Degenerate critical points are unstable; any non-Morse function can be perturbed into a Morse function. The *index* of a Morse critical point is the number of negative eigenvalues of its Hessian V, indicating the number of “downhill” principal directions (eigenvectors). A Morse critical point on a 2-manifold is either an index-0 minimum, an index-1 saddle or an index-2 maximum.

Banchoff [1970] showed not only that Morse theory extends to triangle meshes, but moreover that its development there is even more elementary than the smooth case. Here f is a piecewise-linear real function. Its values are defined on the vertices of an oriented 2-manifold triangle mesh M, and extend by linear interpolation across the edges and faces of the mesh. For the moment, we assume for each edge $\langle v_1, v_2\rangle \in M$ that $f(v_1) \neq f(v_2)$. Hence the gradient is constant, non-zero and well defined across the interiors of faces and edges; critical points occur at the vertices.

Let $\mathrm{Lk}(v)$ denote the *link* of a vertex v, defined as the graph of m vertices $v_1, v_2, \ldots, v_m$ that share an edge with v, along with the edges $\langle v_1, v_2\rangle, \langle v_2, v_3\rangle, \ldots, \langle v_m, v_1\rangle$. We can decompose the link

$$\mathrm{Lk}(v) = \mathrm{Lk}^{+}(v) \sqcup \mathrm{Lk}^{-}(v) \sqcup \mathrm{Lk}^{\pm}(v) \tag{3}$$

where $\mathrm{Lk}^{+}(v)$ is the *upper link* consisting of the vertices $\{v_i \in \mathrm{Lk}(v) : f(v_i) > f(v)\}$ and the edges $\{\langle v_i, v_j\rangle \in \mathrm{Lk}(v) : f(v_i) > f(v), f(v_j) > f(v)\}$) the *lower link* $\mathrm{Lk}^{-}(v)$ (replacing $>$ with $<$), and the *mixed edges* $\mathrm{Lk}^{\pm}(v) = \{\langle v^{+}, v^{-}\rangle : f(v^{+}) > f(v) > f(v^{-})\}$. The number of mixed edges, $\#\mathrm{Lk}^{\pm}(v)$, is always even.

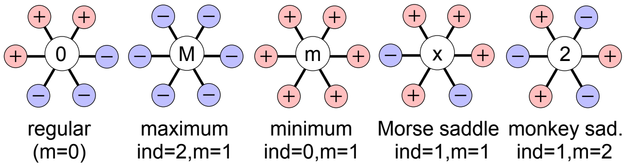


Figure 2: Examples of regular and critical vertices.

We hence classify vertices as

$$\begin{array}{lll}
\mathrm{Lk}^{-}(v) = \emptyset & \Rightarrow & v \text{ is a minimum with index 0,} \\
\mathrm{Lk}^{+}(v) = \emptyset & \Rightarrow & v \text{ is a maximum with index 2,} \\
\#\mathrm{Lk}^{\pm}(v) = 2 & \Rightarrow & v \text{ is regular,} \\
\#\mathrm{Lk}^{\pm}(v) = 2 + 2m & \Rightarrow & v \text{ is a saddle, with index 1} \\
 & & \text{and multiplicity } m \geq 1.
\end{array}$$

Assigning minima and maxima each a multiplicity $m = 1$ allows us to compute the Euler characteristic as

$$\chi(M) = \sum_{v \in \mathrm{Crit}\, M} (-m)^{\mathrm{ind}\, v} \tag{4}$$

as proven by Banchoff [1970]. A saddle of multiplicity $m = 1$ is a Morse saddle, but in the piecewise linear setting we have no need for a neighborhood to be locally quadratic, so saddles of any multiplicity (e.g. $m = 2$ monkey and $m = 3$ dog saddles) can be managed. Edelsbrunner *et al.* [2002] demonstrate that the vertex-split of a monkey saddle perturbs it into two Morse saddles, but our application can process non-Morse saddles directly.

3.2 The Morse Complex

A theorem of smooth Morse theory states that M is homotopic to a cell complex that contains a λ-cell corresponding to each critical point of index λ [Milnor 1963]. If f is Morse-Smale [Edelsbrunner et al. 2002] then this 2-complex can be instantiated geometrically as a graph embedded in M. This graph contains a vertex (0-cell) at each minimum, an edge (1-cell) passing through each saddle (with a minimum at each end), and each face (2-cell) contains exactly one maximum. The 1-cells are constructed as integral curves of $-\nabla f$ extending from the two “downhill” sides of the saddle points to the minima. We call this structure the *Morse complex*[1].

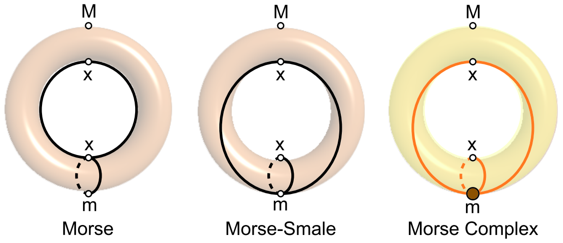


Figure 3: Altitude on a vertical torus (left) is Morse, but integral lines flow from the upper saddle to the lower. Leaning the top of the torus forward slightly (center) yields a Morse-Smale function where integral lines flow from both saddles to the minimum. The Morse complex (right) embeds a brown 0-cell at the minimum, two orange 1-cells along the saddle-point integral lines and a single tan 2-cell in the rest of the torus surface, which contains the maximum.

As before, the embedding of the Morse complex in a smooth manifold also extends to a meshed manifold. For a graph G let $\mathrm{argmin}(G)$ return the vertex v^{-} such that $f(v^{-}) = min_{v \in G} f(v)$ In the event G has multiple vertices that share the same least value, assume $\mathrm{argmin}(G)$ returns only one of them, and always the same one. The function $\mathrm{argmin}(\mathrm{Lk}(v))$ will act as a discrete version of $\nabla(-f(v))$ since it returns the direction of steepest descent.

The analog of an integral curve on a mesh is the *flow path.* For a regular vertex v_0, the *flow path,* $\mathrm{flow}(v_0)$, is the chain of vertices $(v_0, v_1, \ldots, v_n)$ and edges $\langle v_i, v_{i+1}\rangle$ such that $v_i = \mathrm{argmin}(\mathrm{Lk}(v_{i-1}))$ and v_n is a minimum. Flow paths do not cross but they can merge, and once merged paths never separate.

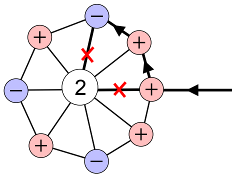


Figure 4: Teflon Saddles: Flow paths are not allowed to reach a saddle, and must travel around the saddle’s link instead in its quest for a minimum.

We do not allow a flow path to penetrate the link of a saddle (other than its origin). If a flow path reaches a vertex v in the link of a saddle v_s, the edge $\langle v, v_s\rangle$ is not considered when computing $\mathrm{argmin}\, v$. Thus flow paths can approach a saddle but do not reach it, and more importantly cannot cross other paths at the saddle. This procedure consistently and robustly reroutes paths leading into the saddle to paths leading away from the saddle, without requiring the careful ordering of paths prescribed in [Edelsbrunner et al. 2002].

These flow paths enable us to embed a 2-D Morse complex X into a meshed manifold M. Let X_0 be the set of 0-cells of X, constructed by placing a 0-cell at each minimum in M. Then for

[1] The Morse complex described here and by Milnor [1963] is different than the Morse-Smale complex [Edelsbrunner et al. 2002; Edelsbrunner et al. 2003b; Bremer et al. 2003] which more closely resembles slope districts [Nackman 1984]. The Morse-Smale complex is the Morse complex of f *refined* by the Morse complex of $-f$.

each Morse saddle v, decompose $\mathrm{Lk}^-(v)$ into its two disjoint connected components A and B, and let $v_A = \mathrm{argmin}(A)$ and likewise for v_B. The 1-cell corresponding to saddle v is then the union of $\mathrm{flow}(v_A), \langle v_A, v\rangle, \langle v, v_B\rangle$ and $\mathrm{flow}(v_B)$ and its ends (minima) are attached to the corresponding 0-cells. The remaining 2-cells, the connected components of $M - X^1$, will each contain a single maximum.

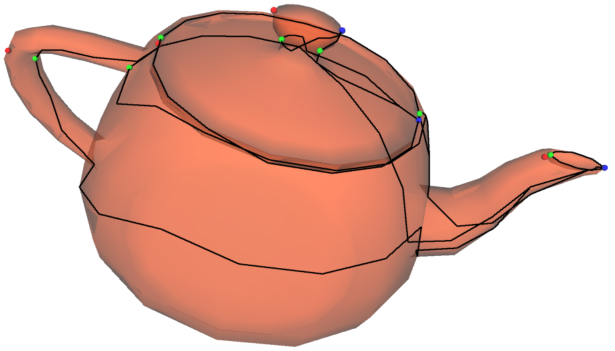


Figure 5: The Morse complex of the x-coordinate of a closed-manifold version of the Utah teapot (the handle-to-spout axis). All critical points lie in the xy-plane: blue = minimum, green = saddle and red = maximum.

We can extend the Morse complex to handle saddles of multiplicity $m > 1$ though it requires the rather inelegant addition of 0-cells at these saddles[2]. First place a 0-cell at each saddle v of multiplicity m. Then decompose $\mathrm{Lk}^-(v)$ into its $m+1$ connected components $\{A_i\}$ and let $v_i = \mathrm{argmin}(A_i)$. For each of these, embed a 1-cell in M corresponding to the flow path $\mathrm{flow}(v_i)$, attaching one end to the 0-cell at v and the other to the 0-cell at the flow path's terminating minimum.

The two flow paths visiting vertices $(v_1, v_2, \ldots, v_n)$ and $(v'_1, v'_2, \ldots, v'_n)$ can merge ($v_i = v'_j$ for some minimal $i < n$ and $j < n'$). These merged paths invalidate the embedding, as it becomes two-to-one on the pair of corresponding 1-cells, and many-to-one on the 2-cell between. We can repair this degeneracy of the resulting complex by inserting a 0-cell at the merge vertex v_i and representing the two merged flow paths by a single 1-cell attached to v_i and the terminating minimum. Since we have added a 0-cell and a 1-cell, the Euler characteristic remains unchanged, though the resulting cell complex contains additional 0-cells that do not correspond to any critical point.

3.3 Flat Regions

The *flat edge* limitation: $\langle v_1, v_2\rangle \in M \rightarrow f(v_1) \neq f(v_2)$, can be overcome by perturbation, but perturbation can introduce numerous low-persistence additional critical points. A better solution for flat edges can be drawn from Conley index theory [Mischaikow 1995; Mischaikow and Mrozek 2002]. The Conley index simplifies the classification of a complicated critical region in a vector field (more general than the gradient field studied in this paper) by analyzing the vector field along the boundary of a neighborhood of the critical region, which effectively contracts the critical region to a single point.

Let F be a simply connected maximal collection of equal-valued vertices, as demonstrated in Fig. 6. Then the boundary of this neighborhood is $\mathrm{Lk}F$, where the link is the boundary of the star and in this case is the set of vertices not in F one edge away from a vertex in F, and the cycle of edges connecting them. (Alternatively a flat edge can be processed as a single vertex by an edge collape, and simply-connected flat regions can be converted to a single vertex by a sequence of flat edge collapses.) Thus the lower link can be evaluated and the flat region classified. Since a flat region will have more edges incident on it than would a single vertex, they are more likely to be saddles of high multiplicity, which we can process directly as previously described.

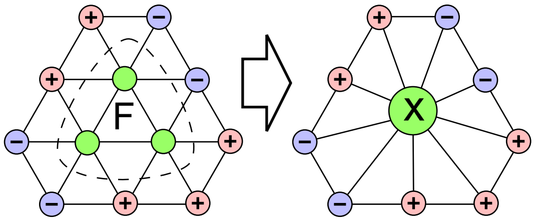


Figure 6: A flat region of vertices can be treated as a single vertex. Each of the vertices in the flat region appears to be regular when ignoring flat edges, but contraction of the flat region to a single vertex reveals it to be a saddle.

The altitude of the Buddha model in Fig. 1(a) contains several flat regions that were removed by flat edge collapse. Perturbation of these edges would have resulted in 18 additional critical points.

One may encounter a multiply-connected flat region, especially in models designed or reconstructed by sampling a rectilinear grid. For example the altitude of the Utah teapot from a triangle mesh derived from its control points contains numerous flat edge cycles. Even the x-coordinate function used in Fig. 5 contains a single flat edge cycle at its girth. Such cases require more sophisticated methods from Conley index theory that deserve a separate and more in-depth treatment than possible here. Such cases can nevertheless be overcome by perturbation techniques at the risk of additional low-persistence critical points, and the teapot in Fig. 5 has been rotated slightly. Such multiply-connected flat regions are far less common in Morse-faired functions.

3.4 Boundaries

We can also extend Morse theory to meshed manifolds with boundary. Such manifolds contain one or more boundaries each represented by a simple closed loop of edges. In preparation for Morse function processing, we sew up each boundary loop by inserting a temporary vertex and creating a face between it and each edge of the boundary loop. The location of the temporary vertex is irrelevant for topological processing, but for geometric concerns can be placed at the centroid of the boundary loop vertices. Likewise the new faces may not generate a valid embedding, but they are only used for combinatorial processing and will be removed once a Morse function has been found. This is similar to the scaffolding triangles of [Zhang et al. 2003] which were used on the much different application of individual patch layout and parameterization.

We will assign an extremal function value to each temporary vertex. If the temporary vertex is not constrained to a global minimum or maximum, then one runs the risk of it becoming a saddle. When the temporary vertex and its face neighborhood are removed, the mesh will be missing a key piece of its critical point structure.

For cutting the surface into a disk, we make the vertex a minimum, which will connect at least one gradient descent flow to the boundary. When the temporary vertex and its star are removed, any flows to it will terminate at, and include, the boundary loop. Alternatively, assigning a maximum value to the temporary vertices

[2] These additional 0-cells do not affect the Euler characteristic because they separate a single 1-cell passing through the saddle into 2 1-cells extending from the saddle.

will encourage (but not guarantee) the flows to avoid the boundary loops, thus isolating and preserving them in the resulting domain.

The temporary vertex increments the Euler characteristic by one (the additional edges and faces cancel), as does the additional extremal point. The agreement of Euler characteristics means that Morse fairing will not introduce any new saddles in response to the new extremum.

If edges of a non-manifold mesh are shared by more than two faces, then these edges can be made into a boundary of their incident faces. This however affects the Euler characteristic, leading to an unexpected critical point structure. In the polygon-soup limit where every face is a disjoint component, all topology information is lost and the Euler characteristic reveals no more than the number of faces.

4 Finding Fair Morse Functions

Morse fairing is based on the observation that low-pass filtering cancels critical points, which leads to solutions of Laplace's equation that preclude the existence of extrema except at its boundary. This section reviews the Laplacian and its solutions, concluding with a new scalable and very simple multigrid solver that rapidly constructs fair Morse functions given the positions of extrema.

4.1 Harmonic Functions and Laplacians

Let f_i denote the value of the real function at the vertex $v_i \in V$. We assume that we are provided with a set $V_C \subset V$ of $k > 0$ constrained points, where f must take on a specified value.

Our goal is to construct a suitable function f that has no local extrema other than the constrained points V_C. The lack of local extrema, except at boundary points, is one of the primary properties of *harmonic functions*, which are solutions of the Laplace equation. Thus, our solution to finding a fair Morse function is to find a solution to the Laplace equation $\Delta f = 0$ subject to the Dirichlet boundary conditions imposed by the constrained vertices V_C. The resulting function will be harmonic, and will be guaranteed to be free of extraneous local extrema.

The standard definition of the Laplacian operator on a piecewise-linear mesh M is the umbrella operator

$$\Delta f_i = \sum_{\langle i,j\rangle \in M} w_{ij}(f_j - f_i) \tag{5}$$

where w_{ij} is a scalar weight assigned to the *directed* edge $\langle i,j\rangle$. It is clear from (5) that any vertex for which $\Delta f_i = 0$ has a value f_i which is a weighted combination of the function values at its neighboring vertices. Thus, we can guarantee that f_i is not a local extremum provided that (1) $\sum_j w_{ij} = 1$ and (2) $w_{ij} > 0$ for all edges $\langle i,j\rangle$. These mirror the validity conditions for linear parameterization methods [Floater 1997].

We assemble the vertices' function values f_i into an n-vector $\mathbf{f}$ and write the Laplace equation in matrix form

$$L\mathbf{f} = 0 \tag{6}$$

where the elements of the $n \times n$ matrix L are given by

$$L_{ij} = \begin{cases} \sum_{\langle i,k\rangle \in M} w_{ik} & \text{if } i = j, \\ -w_{ij} & \text{if edge } \langle i,j\rangle \in M, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

This matrix L is a *weighted* Laplacian matrix [Chung 1997]. Note that (6) implies $\Delta f_i = 0$ everywhere, though we have subtly flipped a sign convention to ease the remaining derivation.

In order to enforce the specified boundary conditions, we construct a constrained (aka "pegged") Laplacian matrix $\hat{L}$ where the row for each constrained vertex is replaced with the corresponding row of the identity matrix:

$$\hat{L}_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } v_i \in V_C, \\ 0 & \text{if } v_i \in V_C, i \neq j, \\ L_{ij} & \text{otherwise.} \end{cases} \tag{8}$$

We then solve

$$\hat{L}\mathbf{f} = \mathbf{b} \tag{9}$$

where

$$b_i = \begin{cases} f_i & \text{if } v_i \in V_C, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

which is unique for $|V_C| \geq 2$. Note that this system of equations is entirely equivalent in form to the systems used in linear parameterization methods [Floater 1997; Desbrun et al. 2002], with the exception that we are solving for a single scalar field.

If we constrain exactly two vertices v_i, v_j to field values f_i, f_j such that $f_i < f_j$, then the function f will have a single minimum at v_i, a single maximum at v_j, and by the Euler characteristic, two saddles for each handle in M. Constraining additional vertex values beyond the first two does not always guarantee them to be minima or maxima. For example, constraining three vertices to three different values could yield a solution where the middle-valued vertex is a saddle. To control the number of minima and maxima, it is best to constrain all maxima vertices to the same global maximum value, and all minima vertices to the same global minimum value (so long as no two maxima share an edge, and likewise for the minima).

Selecting Weights. In the linear system (9), we have the freedom to choose different schemes for assigning the edge weights w_{ij}. One natural choice are the combinatorial weights

$$\text{Combinatorial weights:} \quad w_{ij} = 1/\deg(v_i) \tag{11}$$

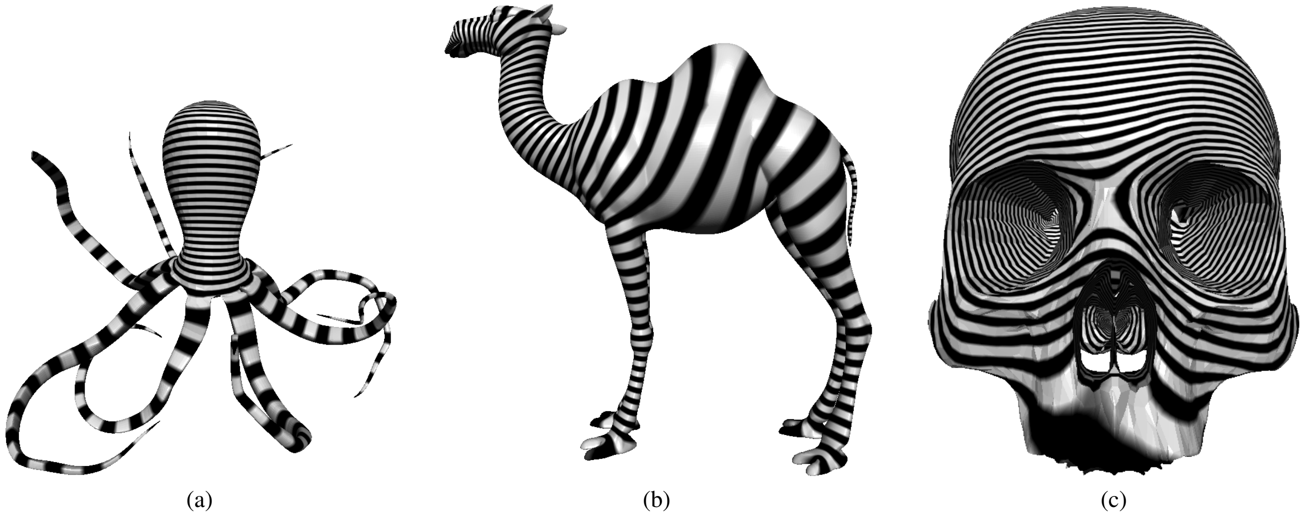
which result in a standard graph Laplacian matrix. This system is purely combinatorial in nature, and these combinatorial weights ignore the geometry of the surface. The graph Laplacian uncovers the topological structure of the surface purely from the connectivity of its mesh.

Some applications, such as base domain construction, are sensitive to the geometric properties of the field, and depend on a smooth and well-shaped function over the manifold. These applications are better served by geometry sensitive weight schemes, such as those used in recent parameterization research.

The mean value weights

$$\text{Mean Value weights:} \quad w_{ij} = \frac{\tan(\theta/2) + \tan(\phi/2)}{\|v_j - v_i\|}, \tag{12}$$

were developed by Floater [2003] as a way to approximate harmonic maps while maintaining the convex validity requirement. The values θ and ϕ are the angles the edge $\langle v_i, v_j\rangle$ makes with its two immediate neighboring edges at v_i. These weights are non-negative which prevents the introduction of unexpected local extrema in the solution. They also produce exceptionally smooth and well-graded scalar fields. Alternative weight schemes that minimize the Dirichlet energy [Pinkall and Polthier 1993; Desbrun et al. 2002] also produce very smooth fields, but can take on negative values in the presence of oblique triangles. This can introduce additional unexpected local extrema in the solution, and makes them unsuitable for our application.



(a) (b) (c)

Figure 7: Morse functions, both smooth and fair, computed with an irregular multigrid Laplacian solver on a "zebrapus" (max: top, min: tentacle ends), a "cazebramel" (max: nose, min: tail, feet) and a psychedelic skull (max: eyes, min: top & bottom).

4.2 Iterative and Sparse Solutions

Depending on the application and mesh size, a variety of techniques exist to produce a fair Morse function by solving (9) or a related system. The simplest of these is the Jacobi iteration

$$\mathbf{f}^{i+1} = D^{-1}\mathbf{b} - D^{-1}W\mathbf{f}^{i} \tag{13}$$

where $\hat{L} = D - W$ decomposes the pegged Laplacian $\hat{L}$ into a diagonal matrix $D_{ii} = \sum_k w_{ik}$ (which inverts elementwise) and the weighted adjacency matrix $W_{ij} = w_{ij}$. For the combinatorial and mean-value weights, $D = D^{-1} = I$, and (13) simply replaces each non-pegged vertex with the weighted average of its neighbors while retaining the value of each pegged vertex. In this case one can simply iterate

$$\mathbf{f}^{i+1} = \bar{L}\mathbf{f}^{i} \tag{14}$$

where

$$\bar{L}_{ij} = \begin{cases} 0 & \text{if } i = j \text{ and } v_i \notin V_C, \\ \hat{L}_{ij} & \text{otherwise.} \end{cases} \tag{15}$$

is a matrix whose diagonal is zeroed at unpegged rows. Iterating in-place performs a faster Gauss-Seidel solve. However iterated Laplacians converge extremely slowly [Hackbusch 1985; Kobbelt et al. 1998]. For example, mesh noise is often filtered by an iterated unpegged Laplacian on vertex position, e.g. [Taubin 1995; Desbrun et al. 1999], but the limit of this iteration would map all vertices to the same position. Morse fairing is instead interested in the solution of a pegged Laplacian.

Incremental Morse Fairing. When solving a pegged Laplacian, the initial guess f^0 is irrelevant. However, the iteration (14) can be used to smooth an initial function. This smoothing reduces and eventually eliminates variation while retaining the characteristics of the original function. This iteration has the effect of cancelling critical point pairs in persistence order [Edelsbrunner et al. 2002] though one does not know *a priori* how many iterations are required for cancellation to occur, and cancellations may occur between unexpected critical point pairs. We use iterative Morse fairing later in Sec. 5 to cancel critical points for a curvature-based function.

Sparse Solution. The Laplacian system is sparse, with only $\deg(v_i) + 1$ out of n elements in each row i. This allows sparse solvers, such as SuperLU or netlib's "sparse" package, to be used which can save both space and time for large meshes. Moreover, Morse fairing could be implemented on the GPU by solving (9) using conjugant gradients [Bolz et al. 2003].

4.3 Multigrid Solutions

For large meshes, even very efficient sparse solution methods can be undesirably slow. Fortunately, the structure of the fields we desire to compute lend themselves very naturally to efficient hierarchical solution techniques.

In order to build a suitable mesh hierarchy, we repeatedly coarsen the mesh via edge contraction. In a single pass, we greedily select a maximal independent set of edges to contract. We do this by ranking all edges according to their cost of contraction as determined by the quadric error metric [Garland and Heckbert 1997]. We then process edges in order of increasing costs, selecting all contractions $v_j \rightarrow v_i$ that meet all of the following validity requirements: (1) the vertex v_j is not a constrained point, (2) Neither v_j nor any of its neighbors are already marked for deletion, and (3) The edge passes the link condition [Dey et al. 1999b] (contracting it will not alter the topology of the surface). The result of this coarsening phase is a sequence of meshes $M^0, M^1, \ldots, M^k$, where M^0 is the original mesh, and M^i is the mesh after the i^{th} coarsening step. The simplest mesh, M^k, we refer to as the base domain mesh.

Having produced a suitable base domain, we can solve a simplified constrained Laplacian system (9) on it. Note that by construction, all constrained vertices must still be present in the base domain, and it must have the same genus as the input. However, unless an unusually large number of vertices are constrained, the base domain can have a very simple structure. For example, the minimal base domain for a *genus*-0 mesh with 2 constrained points is a tetrahedron. Even for the turbine blade model (Fig. 13) with 295 separate connected components and 165 handles, the minimal base domain has only 2,744 triangles and 1,632 vertices. Any reasonably efficient sparse matrix solver can solve systems of this size in a small fraction of a second on modern hardware.

4.3.1 Irregular Multigrid

Having computed an approximate solution field on the base domain M^k, we want to extend this solution back to the original mesh M^0. We do this by a standard irregular multigrid approach quite similar in form to those recently developed independently by Aksoylu *et al.* [2003] and Ray and Levy [2003].

We repeatedly refine the mesh, undoing the contractions performed during coarsening. During each refinement phase, we perform vertex splits corresponding to all the contractions performed during the corresponding coarsening phase. Our validity rules guarantee that, when a vertex v_i is reintroduced into the mesh during refinement, the approximate solution field already exists at all of its neighbors. Our initial estimate for the field value f_i at the new vertex v_i is simply formed by the linear combination of its neighbors $f_i = \sum_j w_{ij} f_j$. This produces an approximate solution, which we must iteratively relax (§4.2) until convergence.

This multigrid algorithm produces the same solution field as a full matrix solver (subject to the convergence tolerance used) as demonstrated in Fig. 7. However, its running time is dependent on the quality of the intermediate mesh approximations, and is not guaranteed to be $O(n)$ if the approximations are poor[3].

4.3.2 Intermediate Value Propagation

Recall that no unpegged vertices of the base domain M^k are extremal. The multigrid Laplacian solver performs a relaxation after each vertex split is to ensure that the function value assigned to each re-introduced vertex creates no new extrema. In fact our interest in the Laplacian is for its elimination of extrema. The following theorem shows that we can avoid extrema during refinement without constructing an expensive Laplacian solution, leading to the *intermediate value propagation* algorithm.

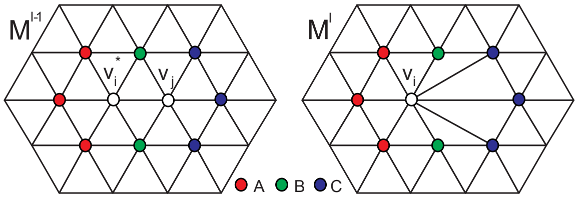


Figure 8: Neighborhood of a vertex split.

Theorem 1. *Let non-extremal vertex v_i split into edge $\langle v_i^*, v_j\rangle$ and assign $f(v_i^*) = f(v_i)$. Define the open intervals*

$$U = \left(f(v_i), \min_{v \in \mathrm{Lk}^+(v_i)} f(v)\right), \tag{16}$$

$$L = \left(\max_{v \in \mathrm{Lk}^-(v_i)} f(v), f(v_i)\right), \tag{17}$$

and let

$$A = \mathrm{Lk}(v_i) \setminus \mathrm{Lk}(v_j) \tag{18}$$

$$B = \mathrm{Lk}\langle v_i^*, v_j\rangle \tag{19}$$

$$C = \mathrm{Lk}(v_i) \setminus \mathrm{Lk}(v_i^*) \tag{20}$$

categorize the vertices sharing an edge with v_i. Then setting the value of the new vertex v_j such that

$$f(v_j) \in \begin{cases} L & \text{if } \mathrm{Lk}^-(v_i) \subset C, \quad \text{(Case 1)} \\ U & \text{if } \mathrm{Lk}^+(v_i) \subset C, \quad \text{(Case 2)} \\ L & \text{if } \mathrm{Lk}^+(v_i) \subset A, \quad \text{(Case 3)} \\ U & \text{if } \mathrm{Lk}^-(v_i) \subset A, \quad \text{(Case 4)} \\ U \cup L & \text{otherwise,} \quad \text{(Case 5)} \end{cases} \tag{21}$$

creates no new extrema.

Proof. Vertex v_i is not extremal: Case 1: $\exists v \in \mathrm{Lk}^+(v_i) \cap (A \cup B)$ such that $f(v_j) < f(v_i) < f(v)$. Case 2: $\exists v \in \mathrm{Lk}^-(v_i) \cap (A \cup B)$ such that $f(v) < f(v_i) < f(v_j)$. Cases 3–5: $\exists v_1 \in \mathrm{Lk}^-(v_i) \cap (A \cup B), v_2 \in \mathrm{Lk}^+(v_i) \cap (A \cup B)$ such that $f(v_1) < f(v_i) < f(v_2)$.

Vertex v_j is not extremal: Cases 1, 2 and 5: $\exists v_1 \in \mathrm{Lk}^-(v_i) \cap (B \cup C), v_2 \in \mathrm{Lk}^+(v_i) \cap (B \cup C)$ such that $f(v_1) < f(v_j) < f(v_2)$. Case 3: $\exists v \in \mathrm{Lk}^-(v_i) \cap (B \cup C)$ such that $f(v) < f(v_j) < f(v_i)$. Case 4: $\exists v \in \mathrm{Lk}^+(v_i) \cap (B \cup C)$ such that $f(v_i) < f(v_j) < f(v)$.

Vertices in A do not become extremal since their links do not change. Vertices in B do not become extremal since v_j is simply added to their links. Vertices in C do not become extremal, for if v_i was a lesser or a greater neighbor of any vertex in C, then for all five cases setting $f(v_j) \in L \cup U$ will continue that role. □

For meshes with bounded valence, single vertex simplification and refinement are constant-time operations. Therefore, a multigrid fairing algorithm using this refinement operation runs in time linear in the number of vertices, making Morse fairing a scalable procedure up to the size constraints of main memory. This refinement results in a Morse function with the least number of critical points.

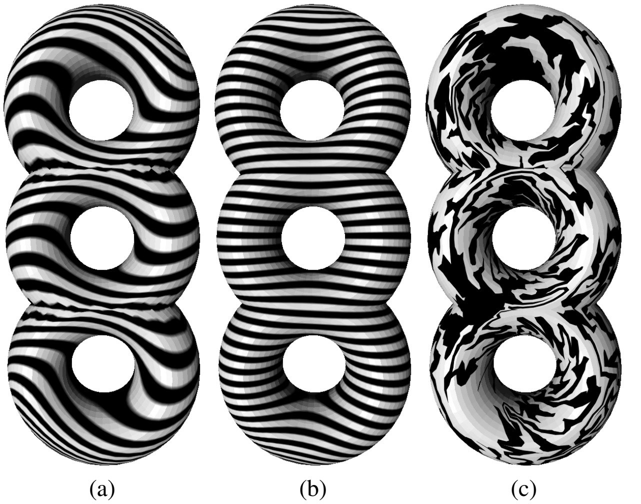


Figure 9: Combinatorial weights (a) ignore the geometry of the mesh, and can produce undesirable field variation (though no new critical vertices). Mean value weights (b) produce a very smooth field. Our new intermediate value propagation solution produces a very random field (c), but surprisingly (and provably) no new critical points.

4.4 Discussion

Properly accounting for the geometry of the surface has a substantial effect on our resulting fields, as demonstrated in Figure 9. Using the purely combinatorial weights $w_{ij} = 1/\deg(v_i)$ produces a field with a great deal of unwanted variation. In contrast, the mean value weights produce a very smooth, pleasingly symmetric field. Intermediate value propagation picks an arbitrary value within bounds during refinement which leads to wild fields that nevertheless avoid extrema and by the Euler characteristic result in the least necessary number of saddles.

Table 1 compares the running times of the two multigrid approaches. Intermediate value propagation avoids the relaxation step that a multigrid Laplacian solution requires, resulting in at least a threefold speedup. Fig. 10 shows the scalability of both multigrid solvers.

[3] Poor approximations are in general unavoidable as no provably-good surface simplification algorithms are yet known.

		Mean-Value	Intermediate Value
Model	Vertices	Laplacian Multigrid (s)	Propagation Multigrid (s)
Teapot	553		0.050
V2	1,923	1.36	0.220
3-torus	4,236	1.99	0.471
Camel (S)	11,225	4.8	1.48
Cranium	12,365		1.76
Octopus	16,554	11.9	2.43
Bunny	34,834		6.17
Camel (M)	44,897	25.9	6.85
Horse	48,485		7.51
Turbine	50,260		8.49
Camel (L)	179,585	110.	34.8

Table 1: Multigrid solver execution times (256MB 1.2GHz P3).

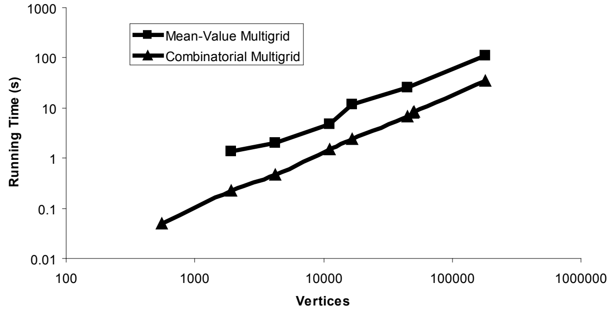


Figure 10: Both the mean-value and propagation multigrid solvers scale linearly (slope of both log-log graphs is one) making them appropriate for in-core processing of any size mesh.

5 Applications

Having derived a more robust Morse theory for meshes and a fast algorithm for fair Morse function construction, we now turn to its application to problems important in computer graphics.

5.1 Cutting a Surface into a Disk

Often a closed meshed surface needs to be cut into a single flattenable piece, which can aid surface texturing, deformation, integration, navigation, sampling and storage, and is a key ingredient of, for example, the recently popular geometry images technique [Gu et al. 2002]. Once cut, the mesh can be flattened into a compact subset of the plane using any number of existing techniques, e.g. [Floater 1997; Piponi and Borshukov 2000; Sheffer and De Sturler 2002].

We cut a surface into a disk by assigning a single minimum that will serve as the base point, and a single maximum representing the single face in the Morse complex. These choices can be made arbitrarily, e.g. as the vertices with the lowest and highest altitude. We then solve the constrained Laplacian, which yields a Morse function with two saddle points for each handle. Each saddle will have a cycle (pair of gradient descent paths) extending from it to the minimum. Each handle will generate two non-separating cycles, one around the hole, and one around the handle. Both of these paths will extend from the base point. Thus cutting and straightening these paths yeilds a polygon with $2g$ sides [Firby and Gardiner 1991]. (In the event a multiple saddle occurs, the saddle is counted with multiplicity and a gradient descent path is extended from each connected component of its lower link.)

5.2 Constructing a Base Domain

The base domain of a polygonal mesh is a highly simplified representation. The base domain is often constructed as the result of a long sequence of mesh simplification steps, and sits at the root of a surface's multiresolution hierarchy. The faces of a base domain correspond to clusters in the mesh, and mapping each face into the plane leads to a multiple chart texture atlas.

A *combinatorial* base domain is an abstract 2-D cell complex X and is often also a simplicial complex. The MAPS method for surface parameterization [Lee et al. 1998] forms a combinatorial simplicial base domain as the end result of repeated vertex removals [Dobkin and Kirkpatrick 1985]. Though the base domain constructed by MAPS simplification matches the topology of the refined mesh, the organization of the domain is rather arbitrary.

The Morse complex embedded in the mesh can be used to construct a combinatorial base domain. The faces of the complex may not be triangular, but the Morse-Smale complex [Edelsbrunner et al. 2002] provides a method for their tessellation. We can trace gradient ascent paths extending from the uphill sides of each saddle, and these uphill paths will lead to a maxima. For a saddle, we construct an ordered list of alternating minima and maxima that are the endpoints of paths extending from the alternating uphill and downhill paths in counterclockwise order about the saddle. We tessellate the 2-cells of the complex with these saddle-minimum-maximum triangles[4].

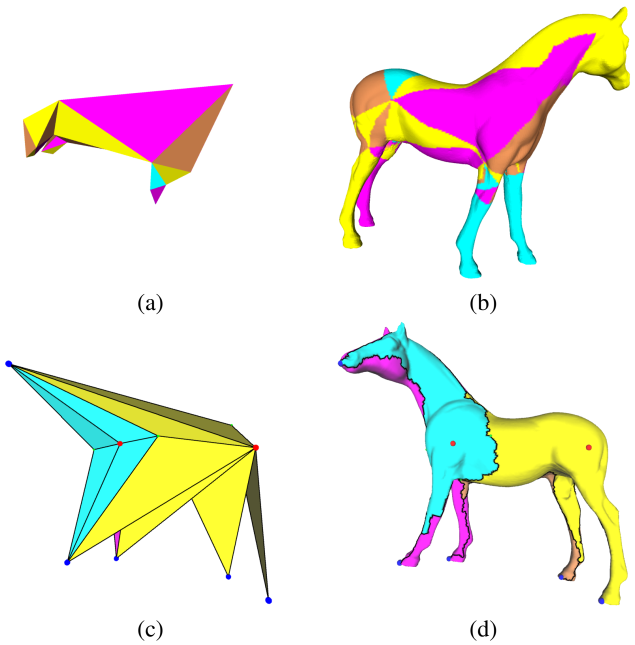

(a)
(b)
(c)
(d)

Figure 11: The greedy base domain (a) corresponding to face clusters (b) constructed by MAPS. Morse fairing allows the user to specify base domain vertices (minima) at feature tips (e.g. nose, feet) and faces (maxima) at feature areas (e.g. shoulder, hip) to create a more geometrically representative base domain (c) corresponding to face clusters (d).

The two steepest descent flows extending from the downhill sides of a saddle can lead to the same minimum, and likewise the two steepest ascent flows can lead to the same maximum, though both conditions cannot occur simultaneously for an individual Morse saddle. The geometry of the triangulated base domain

[4] A Morse saddle is surrounded by four such triangles, creating the "quad" structure used for multires topology processing [Bremer et al. 2003].

may self intersect and may contain degenerate or inverted triangles, but nonetheless serves as a viable combinatorial topological base domain for the purposes of supporting a parameterization. Moreover, as demonstrated in Fig. 11, Morse fairing allows the user to pick extrema which can preserve features in the base domain lost by otherwise local and greedy simplification steps.

The quality of some multiresolution algorithms depend directly on the quality of the base domain. For example, multiresolution mesh morphing [Lee et al. 1999] relies on the ability to construct meaningful correspondences between the base domains of a source and target object. Using Morse fairing to preserve features in a base domain makes it easier to find good source-target correspondences.

5.3 Clustering

When constructing an atlas, it is often desirable to form charts that approximate developable patches. Multichart Geometry Images [Sander et al. 2003] are generated from a rather expensive curvature clustering algorithm that formed clusters using Dijkstra's algorithm, then repeatedly recenters these clusters using Lloyd's algorithm.

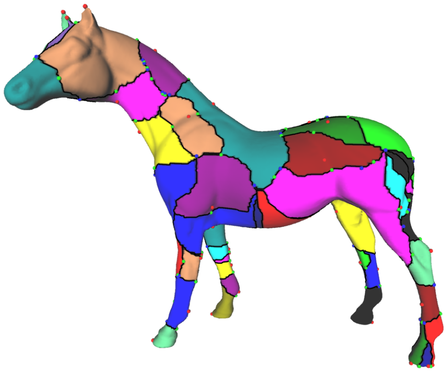


Figure 12: Morse complex of Laplacian-smoothed squared Gaussian curvature yields a rapid clustering toward developable charts.

The Morse complex of Laplacian-smoothed negated squared Gaussian curvature provides a more rapid clustering toward developable patches. Maximal regions will occur at developable regions whereas minima mark vertices of maximal curvature. Sheffer & Hart [2002] showed that forcing chart boundaries through high curvature vertices helps minimize atlas distortion. The paths of the Morse complex will likewise pass through the minima found at high curvature regions. An iterated pegged Laplacian simplifies the Morse complex to the more persistent critical vertices corresponding to curvature features in the mesh, as demonstrated in Fig 12.

5.4 Visualization

We conclude the applications with the fair Morse complex of the cooling tunnels on a jet engine turbine shown in Fig. 13. This polygonal mesh was reconstructed from volume data and contained 294 extraneous small simply-connected components most of which were tetrahedra that we were able to identify and remove. What remained was a single genus-165 component. The 330 paths of the fair Morse complex work their way around or through each of the 165 tunnels in the dataset and could be used for example as fly-through paths for visual inspection.

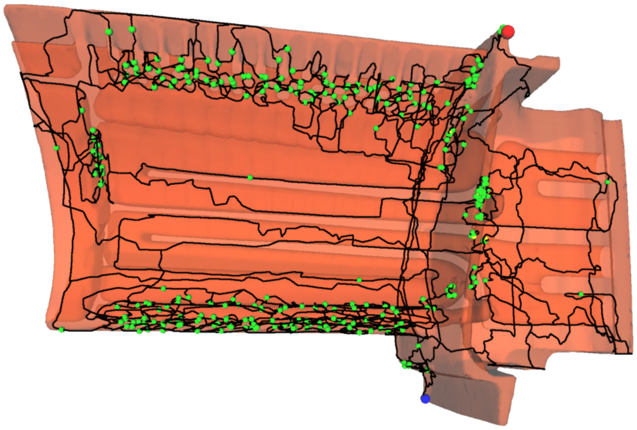


Figure 13: The Morse complex of a genus-165 turbine.

6 Conclusion

Morse theory is an exciting and powerful mathematical tool for reasoning about the global topological structure of a shape based only on local differential information. The unbounded number of critical points has thus far limited the application of Morse theory to a wider variety of problems in computer graphics, mesh processing and scientific computing. We have overcome this limitation with the concept of a fair Morse function that minimizes its variation across a manifold to produce the least possible number of critical points.

Though we have made Morse theory on meshes more robust by now handling simply-connected flat regions, preventing gradient descent paths from reaching saddles and temporarily patching boundaries, we still find it necessary to repair meshes for the method to work properly. Though the critical points and paths can be made to handle non-manifold cases, these special cases make it more difficult to reason about the mesh geometry based on its Euler characteristic, and often our implementation contains hard-coded dependencies based on this reasoning. Stratified Morse theory provides insight into the application of Morse theory to cell complexes and perhaps the key to extending these techniques beyond manifolds.

6.1 Future Work

The 1-cells of the Morse complex are simple gradient descent paths and are by no means optimal. We believe that the gradient descent paths of a harmonic Morse function are probably geodesic, but leave the precise formulation, analysis and proof of such a statement for a future manuscript. Shortening the 1-cell loops like rubberbands would lead to shorter cuts, but these cuts may still not be the most basic cuts, as they can loop around any number of torus holes any number of times. Computational homology provides a way to optimize these cuts to avoid multiple loops and multiple holes [de Verdière and Lazarus 2002], but we leave its integration into Morse fairing for future study.

It is natural to consider the extension of these techniques to piecewise linear 3-manifolds and tetrahedral meshes. For example, Kartasheva [1999] investigated cutting a tetrahedral-mesh solid into a topological 3-ball through the application of homology, and Edelsbrunner *et al.* [2003a] applied their persistence-based topology simplification to 3-manifolds. Unfortunately, 3-manifolds contain two kinds of saddles and its Euler characteristic is the difference between the minima plus one kind of saddle, and the maxima plus the other kind of saddle. Thus one can have a restricted number of extrema and an unbounded number of saddles and still satisfy the Euler charateristic. (If this were not the case, then Morse fairing would have led to a very easy algorithm for the classification of 3-manifolds.)

Acknowledgments. Thanks to Zoe Wood for the genus-6 Buddha model, Patrick Lacz for the manifold teapot, Jesse Hall for video production, and Rob Ghrist for introducing us to the Conley index. This work was supported in part by the NSF and DARPA under the CARGO grant #DMS-0310446.

References

Aksoylu, B., Khodakovsky, A., and Schroeder, P. 2003. Multilevel solvers for unstructured surface meshes. *Siam J. Sci. Comput. (in review)*.

Axen, U., and Edelsbrunner, H. 1998. Auditory morse analysis of triangulated manifolds. In *Mathematical Visualization*, H.-C. Hege and K. Polthier, Eds. Springer-Verlag, Heidelberg, 223–236.

Bajaj, C. L., and Schikore, D. R. 1998. Topology preserving data simplification with error bounds. *Computers and Graphics 22*, 1, 3–12.

Bajaj, C. L., Pascucci, V., and Schikore, D. 1998. Visualization of scalar topology for structural enhancement. *IEEE Visualization '98*, 51–58.

Banchoff, T. F. 1970. Critical points and curvature for embedded polyhedral surfaces. *American Mathematical Monthly 77*, 475–485.

Bolz, J., Farmer, I., Grinspun, E., and Schröder, P. 2003. Sparse matrix solvers on the GPU: Conjugate gradients and multigrid. *ACM Trans. on Graphics 22*, 3 (July), 912–924. (Proc. SIGGRAPH 2003).

Bremer, P.-T., Edelsbrunner, H., Hamann, B., and Pascucci, V. 2003. A multi-resolution data structure for two-dimensional Morse-Smale functions. *Proc. Visualization 03*, 139–146.

Chung, F. R. K. 1997. *Spectral Graph Theory*. American Mathematical Society.

de Verdière, E. C., and Lazarus, F. 2002. Optimal system of loops on an orientable surface. *Proc. Foundations of CS*, 627–636.

Desbrun, M., Meyer, M., Schroeder, P., and Barr, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. *Proc. SIGGRAPH 99*, 317–324.

Desbrun, M., Meyer, M., and Alliez, P. 2002. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum 21* (Sep.), 209–218. (Proc. Eurographics 2002).

Dey, T. K., and Schipper, H. 1995. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete and Computational Geometry 14*, 93–110.

Dey, T. K., Edelsbrunner, H., and Guha, S. 1999. Computational topology. In *Advances in Discrete and Computational Geometry*, B. Chazelle, J. Goodman, and R. Pollack, Eds. Providence.

Dey, T. K., Edelsbrunner, H., Guha, S., and Nekhayev, D. 1999. Topology preserving edge contraction. *Publ. Inst. Math. (Beograd) (N.S.) 66*, 23–45. Also Tech Report RGI-Tech-98-018, Raindrop Geomagic Inc., 1998.

Dobkin, D., and Kirkpatrick, D. 1985. A linear algorithm for determining the separation of convex polyghedra. *J. of Algorithms 6*, 381–392.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. 2002. Topological persistence and simplification. *Discrete and Computational Geometry 28*, 4, 511–533.

Edelsbrunner, H., Harer, J., Natarajan, V., and Pascucci, V. 2003. Morse-Smale complexes for piecewise linear 3-manifolds. *Proc. Symp. on Computational Geometry*, 361–370.

Edelsbrunner, H., Harer, J., and Zomorodian, A. 2003. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry 30*, 1, 87–107.

Erickson, J., and Har-Peled, S. 2002. Optimally cutting a surface into a disk. *Proc. ACM Symp. on Comp. Geom.*, 244–253.

Firby, P., and Gardiner, C. 1991. *Surface Topology*, 2nd ed. Ellis Horwood.

Floater, M. 1997. Parameterization and smooth approximation of surface triangulations. *Computer-Aided Geometric Design 14*, 4, 231–250.

Floater, M. S. 2003. Mean value coordinates. *Computer-Aided Geometric Design 20*, 1 (Mar.), 19–27.

Garland, M., and Heckbert, P. S. 1997. Surface simplification using quadric error metrics. *Proc. SIGGRAPH 97*, 209–216.

Gu, X., Gortler, S. J., and Hoppe, H. 2002. Geometry images. *ACM Trans. on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July), 355–361.

Guskov, I., and Wood, Z. 2001. Topological noise removal. *Proc. Graphics Interface*, 19–26.

Hackbusch, W. 1985. *Multi-Grid Methods and Applications*. Springer-Verlag.

Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. *Proc. SIGGRAPH 2001*, 203–212.

Kartasheva, E. 1999. The algorithm for automatic cutting of three-dimensional polyhedrons of h-genus. *Proc. Shape Modeling Intl. 99*, 26–33.

Kobbelt, L., Campagna, S., Vorsatz, J., and Seidel, H.-P. 1998. Interactive multi-resolution modeling on arbitrary meshes. *Proc. SIGGRAPH 98*, 105–114.

Lazarus, F., Pocchiola, M., Vegter, G., and Verroust, A. 2001. Computing a canonical polygonal schema of an orientable triangulated surface. *ACM Symp. on Comp. Geom.*, 80–89.

Lee, A. W. F., Sweldens, W., Schroeder, P., Cowsar, L., and Dobkin, D. 1998. MAPS: multiresolution adaptive parameterization of surfaces. *Proc. SIGGRAPH 98*, 95–104.

Lee, A. W. F., Dobkin, D., Sweldens, W., and Schröder, P. 1999. Multiresolution mesh morphing. *Proc. SIGGRAPH 99*, 343–350.

Milnor, J. 1963. *Morse Theory*. Princeton Univ. Press.

Mischaikow, K., and Mrozek, M. 2002. Conley index. In *Handbook of dynamical systems, Vol. 2*. North-Holland, 393–460.

Mischaikow, K. 1995. Conley index theory. In *Dynamical systems (Montecatini Terme, 1994)*, vol. 1609 of *Lecture Notes in Math.* Springer, 119–207.

Nackman, L. 1984. Two-dimensional critical point configuration graphs. *IEEE Trans. PAMI 6*, 442–450.

Pinkall, U., and Polthier, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 1, 15–36.

Piponi, G., and Borshukov, D. 2000. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. *Proc. SIGGRAPH 2000*, 471–478.

Ray, N., and Levy, B. 2003. Hierarchical least squares conformal map. *Proc. Pacific Graphics*, 263–270.

Sander, P. V., Wood, Z. J., Gortler, S. J., Snyder, J., and Hoppe, H. 2003. Multi-chart geometry images. *Proc. Symp. Geom. Proc.*, 146–155.

Sheffer, A., and De Sturler, E. 2002. Surface parameterization for meshing by triangulation flattening. *ACM Trans. on Graphics 21*, 4, 874–890.

Sheffer, A., and Hart, J. C. 2002. Seamster: inconspicuous low-distortion texture seam layout. *Proc. Visualization '02*, 291–298.

Shinagawa, Y., Kunii, T. L., and Kergosien, Y. L. 1991. Surface coding based on morse theory. *IEEE Computer Graphics & Applications 11*, 5, 66–78.

Stander, B. T., and Hart, J. C. 1997. Guaranteeing the topology of an implicit surface polygonization for interactive modeling. *Proc. SIGGRAPH 97*, 279–286.

Steiner, D., and Fischer, A. 2001. Topology recognition of 3d closed freeform objects based on topological graphs. *Proc. Pacific Graphics* (Oct.), 82–88.

Taubin, G. 1995. A signal processing approach to fair surface design. *Proc. SIGGRAPH 95*, 351–358.

Vegter, G. 1997. Computational topology. In *Handbook of Discrete and Computational Geometry*. CRC Press, 517–536.

Zhang, E., Mischaikow, K., and Turk, G. 2003. Feature-based surface parameterization and texture mapping. Tech. Rep. GVU 03-29, Georgia Tech.