UC Davis

UC Davis Previously Published Works

Title

TOPOLOGY-GUIDED TESSELLATION OF QUADRATIC EIEMENTS

Permalink

https://escholarship.org/uc/item/3s5064dp

Journal

International Journal of Computational Geometry & Applications, 19(02)

ISSN

0218-1959

Authors

DILLARD, SCOTT E
NATARAJAN, VIJAY
WEBER, GUNTHER H
et al.

Publication Date

2009-04-01

DOI

10.1142/s0218195909002903

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Topology-guided Tessellation of Quadratic Elements

Scott E. Dillard[1,2], Vijay Natarajan[3], Gunther H. Weber[4], Valerio Pascucci[5] and Bernd Hamann[1,2]

[1] Department of Computer Science, University of California, Davis
[2] Institute for Data Analysis and Visualization, University of California, Davis
[3] Department of Computer Science and Automation, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore
[4] Computational Research Division, Lawrence Berkeley National Laboratory
[5] Lawrence Livermore National Laboratory

Abstract. Topology-based methods have been successfully used for the analysis and visualization of piecewise-linear functions defined on triangle meshes. This paper describes a mechanism for extending these methods to piecewise-quadratic functions defined on triangulations of surfaces. Each triangular patch is tessellated into monotone regions, so that existing algorithms for computing topological representations of piecewise-linear functions may be applied directly to the piecewise-quadratic function. In particular, the tessellation is used for computing the Reeb graph, a topological data structure that provides a succinct representation of level sets of the function.

1 Introduction

Scalar functions are used to describe several physical phenomena like temperature, pressure, etc. Scientists interested in understanding the local and global behavior of these functions often study their level sets. A *level set* consists of all points $\mathbf{p}$ in the domain of a function f where $f(\mathbf{p}) = c$, and c is a constant. Various methods have been developed for the purpose of analyzing the topology of level sets of a scalar function. These methods primarily were developed for piecewise-linear functions. We focus on an extension of these methods to bivariate piecewise-quadratic functions defined over a triangulated surface.

A *contour* is a single connected component of a level set. Level sets of a smooth bivariate function are curves. The *Reeb graph* of f is obtained by contracting each contour to a single point [1], see Figure 1. The connectivity of level sets changes at *critical points* of the function. For smooth functions, the critical points occur where the gradient becomes zero. Critical points of f are the *nodes* of the Reeb graph, connected by *arcs* that represent families of topologically equivalent contours.

⋆ A preliminary version of this paper appeared in Proceedings of the 17th International Symposium on Algorithms and Computation, Kolkata, India, Lecture Notes in Computer Science (LNCS) 4288, 2006, 722–731.

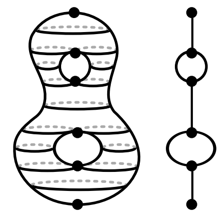


Fig. 1. Reeb graph of a height function defined on a double torus. Critical points of the surface become nodes of the Reeb graph.

1.1 Related Work

Methods to extract contours from bivariate quadratic functions have been explored in the context of geometric modeling applications [2]. *A priori* determination of the topology of contours has been studied also in computer graphics and visualization [3, 4]. Much of this attention has been focused on functions defined by bilinear or trilinear interpolation of discrete data given on a rectilinear grid. Work in this area has led to efficient algorithms for computing contour trees, a special Reeb graph that has no cycles [5, 6]. More general algorithms have been developed for computing Reeb graphs and contour trees for piecewise-linear functions [7–9].

Topological methods have also been used in computer graphics and scientific visualization as a user interface element, to describe high-level topological properties of a dataset [10]. They can be used to selectively explore large scientific datasets by identifying important function values corresponding to topological changes in a function's behavior [11, 12], and for selective visualization [13, 14]. Visualization frameworks for higher order finite elements take topological considerations into account [15]. Reeb graphs have also provided a basis for searching large databases of shapes [16], and for computing surface parametrization of three-dimensional models [17].

1.2 Our Results

Given a triangulated surface and a piecewise-quadratic function defined on it, we tessellate the surface into monotone regions. A graph representing these monotone regions is a valid input for existing algorithms that compute Reeb graphs and contour trees for piecewise-linear data. The essential property we capture in this tessellation is that the Reeb graph of the function restricted to a single tile of the tessellation is a straight line. In other words, every contour contained in that tile intersects the triangle boundary at least once and at most twice. We tessellate each triangular patch by identifying critical points of the function within the triangle and connecting them by arcs to guarantee the required property. We prove monotonicity for a single triangular quadratic patch, but the result applies to any piecewise-defined surface made of such patches.

2 Preliminaries

We consider bivariate piecewise-quadratic functions defined over a triangulated domain, and we represent each triangular element (or patch) by a function $f : \mathbb{R}^2 \to \mathbb{R}$ of the form

$$f(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F.$$

A critical point of f is a point $\mathbf{x}$ where $\nabla f(\mathbf{x}) = 0$. The partial derivatives are given by the linear expressions

$$\frac{\partial f}{\partial x} = 2Ax + By + D \text{ and } \frac{\partial f}{\partial y} = 2Cy + Bx + E.$$

The location of a critical point $(\hat{x}, \hat{y})$ is

$$\hat{x} = \frac{-2CD + BE}{4AC - B^2} \text{ and } \hat{y} = \frac{-2AE + BD}{4AC - B^2}.$$

Bivariate quadratics and their critical points can be classified based on their contours. The contours are conic sections. Let $H = 4AC - B^2$ be the determinant of the Hessian matrix of f. We partition the set of bivariate quadratic functions into three classes:

1. $H > 0$: Contours are ellipses; a critical point is a maximum or a minimum.
2. $H < 0$: Contours are hyperbolas or intersecting lines; a critical point is a saddle.
3. $H = 0$: Contours are parabolas, pairs of parallel lines, or single lines; critical points do not exist.

We refer to members of these classes as *elliptic*, *hyperbolic*, and *parabolic*, respectively. We further classify the critical points of elliptic functions using a second-derivative test. The critical point is a minimum when $A > 0$, and a maximum when $A < 0$. When $A = 0$, the sign of C discriminates between a maximum and a minimum.

2.1 Line Restrictions

Let $\ell(t) = \begin{pmatrix} x_0 + tx_d \\ y_0 + ty_d \end{pmatrix}$ be a parametrically defined line passing through (x_0, y_0) in direction (x_d, y_d). We restrict the domain of the bivariate quadratic $f(x, y)$ to only those points on $\ell(t)$. We obtain a univariate quadratic $r(t) = \alpha t^2 + \beta t + \gamma$ where

$$\begin{aligned}
&\alpha = Ax_d^2 + Bx_d y_d + Cy_d^2, \\
&\beta = 2Ax_0 x_d + B(y_0 x_d + x_0 y_d) + 2Cy_0 y_d + Dx_d + Ey_d \text{ and} \\
&\gamma = Ax_0^2 + Bx_0 y_0 + Cy_0^2 + Dx_0 + Ey_0 + F.
\end{aligned}$$

We call $r(t)$ a *line restriction* of f. If $\alpha \neq 0$, r is a parabola with one critical point at $\hat{t} = -\beta/2\alpha$; if $\alpha = 0$, r is linear. We refer to a critical point of this univariate quadratic function as a *line-critical point*, while we refer to critical points of the bivariate function as *face-critical points.* The line restrictions satisfy several useful properties:

1. There is at most one line-critical point on a line restriction: this is a consequence of the function along the line being either quadratic or linear.
2. If a line intersects any contour twice, it must contain a line-critical point between these intersections: let us consider the case where the function value on the contour is zero. The sign of the line-restriction changes each time it crosses the contour. Applying the mean value theorem to the derivative of the line-restriction, we can show that there must be a critical point between two zero crossings.
3. A line-critical point is located at the point where the line is tangential to a contour: this fact follows from the previous property.
4. Any line restriction passing through a face-critical point has a line-critical point which is coincident with the face-critical point: this fact is a consequence of the gradient at the face-critical point being zero. Thus, all directional derivatives are zero, and, in particular, the derivative of the line restriction is zero.

2.2 Contours and Critical Points

A critical point is a point where the number of contours or the connectivity of existing contours changes. When the gradient is not defined, we may classify critical points based on the behavior of the function in a local neighborhood [18]. Figure 2 shows this classification. Consider a plane of constant height moving upward through the graph surface $(x, y, f(x, y))$ of f. The intersection of the surface and the plane is a set of contours, each homeomorphic to either a closed loop or a line segment. When this plane passes a minimum, a new contour is created. When the surface passes a maximum, an existing contour is destroyed. When the surface passes a saddle, two types of events occur: (a) Two segments may merge into a new segment or a segment may split into two; (b) the endpoints of a segment may connect with each other to form a loop or a loop may split into a segment.

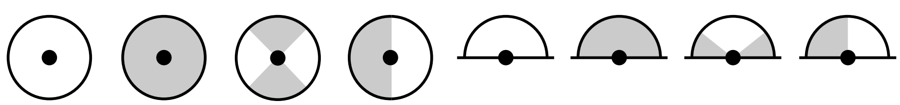


Fig. 2. From left to right: interior minimum, maximum, saddle and regular point, and boundary minimum, maximum, saddle and regular point. The shaded areas are regions with function value less than the value at the indicated point.

We consider critical points of a function restricted to a triangular patch. A face criticality of an elliptic function creates or destroys a loop when the sweep passes it. A face criticality of a hyperbolic function interchanges the connectivity of two segments. A line criticality can create or destroy a segment, merge two segments into a new segment or split a segment in two, transform a segment into a loop or a loop into a segment. However, a line-critical point cannot create or destroy loops, see Figure 3. We determine whether a line criticality is an extremum or a saddle by examining the directional derivative perpendicular to the edge, at the critical point. Vertices, when not located exactly at a hyperbolic saddle point, can only create or destroy segments within a triangular patch. We provide a proof in Sect. 5.

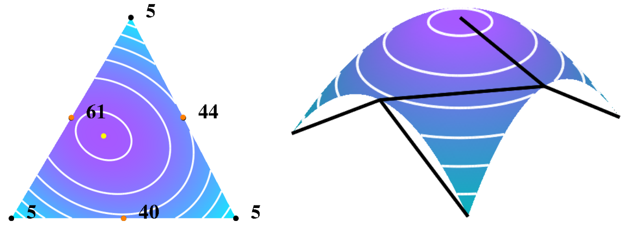


Fig. 3. A patch with one face-critical point and three line-critical points. Interpolated function values are shown. The Reeb graph is overlaid in black, and contours are shown in white. Note that the line-critical point at value 61 converts a closed-loop contour into a segment contour, while those at 44 and 40 split segment contours.

3 Tessellation

We are given a scalar-valued bivariate function defined by quadratic triangular patches. We aim to tessellate the patches into monotone regions, for the purpose of analyzing the topology of level sets of one patch, and, by extension, of a piecewise-defined function composed of many patches. The tessellation will decompose each triangular patch into subpatches, so that each subpatch has a Reeb graph which is a straight line. More specifically, every level set within each subpatch is a single connected component and is homeomorphic to a line segment. We achieve this by ensuring that each subpatch contains exactly one maximum and one minimum, and no saddles. Since we are interested in the topology of the subpatches but not their geometry, we only compute a combinatorial graph structure which captures the connectivity of contours. Some of the arcs of this graph come from the patch, such as boundary edges, and thus their geometry can be inferred. The embeddings of remaining arcs are not computed since they are not required to construct the Reeb graph.

The construction of the tessellation proceeds by using a case analysis. For each patch, we count the number of line-critical points ($L = 0, 1, 2, 3$) and face-critical points ($F = 0, 1$). Each pair $\langle F, L\rangle$ is handled as an individual case to determine an appropriate tessellation. We first describe the composition of the tessellation. The nodes of a tessellation graph include: (1) all three vertices of the triangle; (2) all line-critical points that exist on the triangle boundary and are not coincident with a triangle vertex; and (3) the face-critical point, assuming that it lies within the triangle and is not coincident with the boundary. We are given (1) as input, but (2) and (3) must be computed in a pre-processing step. Numerical problems can arise. Root-finding is not needed to compute (2) and (3), so exact arithmetic may be used. However, if speed and consistency are favored over accuracy then we only need to ensure that the tessellation graphs of every patch agree on their boundary edges. The existence and location of (3) does not affect the tessellation boundary, thus no consistency checks are needed for computing these points. The computation of (2) must agree between two triangles sharing an edge. To ensure this, we do not treat edges as line restrictions of bivariate quadratics, but rather as univariate quadratics defined by data prescribed for the edge.

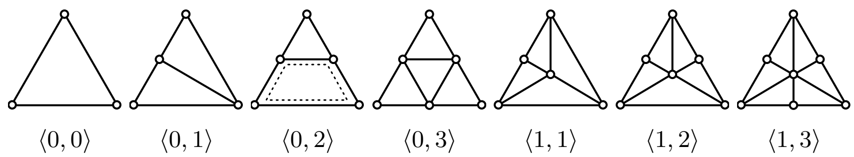


Fig. 4. Tessellation cases $\langle F, L\rangle$, where F is the number of face-critical points and L is the number of line-critical points.

The arcs are included in the tessellation based on the following rules:

- If a line-critical point exists on an edge, we connect that node to both triangle vertices on that edge. Otherwise, if an edge has no line-critical point, then we connect its vertices by an arc.
- If a face criticality does not exist, then we do the following:
  - If there is only one line criticality, we connect it to the opposite vertex, as in Figure 4 $\langle 0,1\rangle$.
  - If there are two line-critical points, we connect each one to the other. The tessellation is not yet a triangulation. There are two possible arcs that can be added to triangulate the quadrilateral shown in Figure 4 $\langle 0,2\rangle$. One, but not both, may be incident on a boundary saddle. We include this arc in the tessellation.
  - If there exist three line criticalities, we connect each one to the other two, as shown in Figure 4 $\langle 0,3\rangle$.
- Otherwise, if a face-critical point exists in the triangle, we connect it by an arc to every other node, as shown in Figure 4, $\langle 1,1\rangle$, $\langle 1,2\rangle$ and $\langle 1,3\rangle$.

4 Application to Reeb Graphs

We show how these tessellations may be used to compute a Reeb graph. Reeb graph algorithms, such as the algorithm of Cole-McLaughlin *et al.* [8], operate by tracking contours of a level set during a sweep of the range of function values.

Since all arcs of our tessellation are monotone, any contour intersects an arc at most once. Contours of function values that are not equal to any node intersect the boundary of every triangular patch twice, or not at all. We can follow a contour by starting at one arc that it intersects, and moving to another arc of the same triangle that intersects the contour. From the perspective of the Reeb graph algorithm, the tessellation is just a piecewise-linear triangulated surface. This allows Reeb graph algorithms to be used, without modification, on the tessellated surface, and Reeb graphs of general surfaces (of any genus) can be constructed from piecewise-quadratic meshes.

When the domain of a function is planar, such as in images or terrains (height fields), the Reeb graph contains no cycles and is called a contour tree. Efficient contour tree algorithms proceed in two distinct steps [5, 7]. First, two separate trees, called the *join tree* and *split tree*, are constructed using one of several known of methods. The join tree tracks topological changes in the subdomain lying above a contour, and the split tree tracks topological changes in the subdomain lying below a contour. In the second step, the join and split trees are merged to yield the contour tree. Our tessellation graph is a valid input for the join and split tree construction algorithms. Applying any of these algorithms to the tessellation graph of a piecewise-quadratic function will yield the correct contour tree for that function. The methods described in Carr *et al.* [7] and Chiang *et al.* [9] utilize the following properties to ensure the correctness of their join-tree and split-tree construction algorithms [6]:

1. All critical points of the piecewise function appear as nodes in the graph.
2. For any value h, a path above (below) h exists in the graph if and only if a path above (below) h exists in the domain.

We explicitly include every potential critical point of f in the tessellation, so that the first property is satisfied. We assert that the monotonicity property of our tessellation ensures that the second property is also satisfied.

The correctness of many topological algorithms relies on the *Morse property* of the function in question: all critical points of the function are isolated, i.e., there are no degenerate critical points or entire flat regions, plateaus for example. This is not the case in most scientific datasets. In a piecewise-linear setting, non-isolated critical points can occur when there is a constant function value across an entire edge or face of the triangle mesh. In addition to these instances, bivariate quadratic functions can also exhibit non-isolated critical points along a line. The graph surface of such a function is that of a swept parabola, and the non-isolated critical point is the line swept by the parabola's vertex.

Existing approaches to topological analysis of scientific data handle non-isolated critical points by applying *symbolic perturbation* to the function values. In practice, this means using a tie-breaker paradigm: If two points in the domain have equal function value, then some other property of the points is used to deterministically choose a greater and lesser value for the respective points. This other property can be the address of the points in computer memory, or a lexicographical ordering of the point locations. We apply this same technique to our tessellated graph, treating it as a piecewise-linear function. In the case of a swept-parabola patch, the line swept out by the parabola's vertex will intersect the patch boundary at two points: a triangle vertex and a line-critical point, or two line-critical points. Our tessellation method establishes arcs between these two nodes, and the non-isolated critical point can be handled just as in the piecewise-linear setting.

5 Case Analysis

The tessellation constructed in Sect. 3 has the property that every arc is monotone. To prove this fact, we first state and prove some useful results about configurations of triangle vertices, line-critical points and face-critical points. We assume that all triangles in the triangulation are non-degenerate, i.e., every interior angle is between zero and π and every triangle has non-zero, finite area. In the proofs given below, we make use of two properties of the Reeb graph of the function defined over a triangular patch. The first property is that the Reeb graph does not contain any cycles because the domain (a single triangular patch) is a topological disk [8]. The second property is that the number of extrema in the Reeb graph is two more than the number of saddles. The is a property of trees with interior nodes of degree no more than three. This latter property does not hold when there is a hyperbolic saddle point in the strict interior of the patch, which has degree four, as shown in Figure 5. We do not rely on this property in this case.

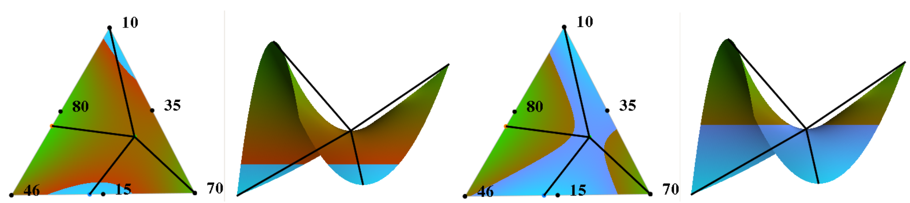


Fig. 5. A hyperbolic patch in Case $\langle 1, 2\rangle$, shown immersed to elevations of 24 and 42. In general, the number of "beaches" at a given elevation is equal to the number of arcs spanning that elevation.

Lemma 1. *If a triangle vertex is a boundary saddle of the function restricted to the triangle, then it lies at the intersection of the two hyperbolic asymptotes.*

Proof. We prove this lemma by contradiction. We assume that there exists a vertex v that is a boundary saddle, but not the hyperbolic saddle point. As we sweep the level sets downward in function value and pass v, two contours merge or a single contour splits into two. We assume, without loss of generality, that a contour splits into two, as shown in Figure 6. Above the value of v, the contour is contained entirely in the triangle. Below the value of v, the contour passes outside the triangle and then back in; the triangle cuts the contour into two segments. (These segments may be joined elsewhere, but locally they are distinct.) We consider the segment of the contour approaching v from the right. If this segment is to remain strictly inside the triangle until the sweep arrives at v, its tangential direction is constrained by the triangle edge to the right of v. Similarly, the tangential direction of the contour segment approaching from the left is constrained by the triangle edge to the left of v. All contours except for hyperbolic asymptotes are smooth, and thus the tangent on the right of v must agree with the tangent on the left. The edges must be parallel, and therefore the triangle must be degenerate, which violates our assumption.

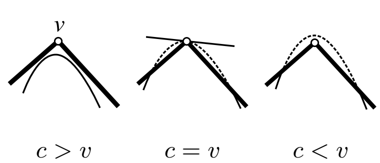


Fig. 6. A smooth contour c cannot both touch the triangle boundary at v and lie completely in the interior.

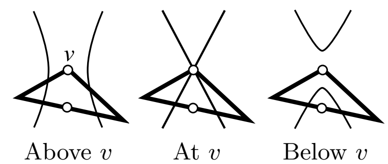


Fig. 7. A triangle containing a boundary saddle at a vertex v must contain a line-critical point on the opposite edge.

Lemma 2. *If a vertex of a triangular patch is a boundary saddle, then the triangle has zero face-critical points and one line-critical point.*

Proof. Considering Lemma 1, if a vertex v is a boundary saddle then v is a face criticality of a hyperbolic function. Therefore, the face criticality is not in the interior. Now, the edges incident at v cannot have line criticalities since v is necessarily the line-critical point of all lines that pass through it. The edge opposite v intersects the asymptotes twice and therefore must have a line criticality, see Figure 7.

Lemma 3. *If a triangular patch contains exactly one line criticality and no face criticality, then that line criticality can be reached by a monotone path from any other point in the triangle.*

Proof. We first assume that none of the vertices is a boundary saddle. The line criticality may be an extremum or a saddle. If it is an extremum, the triangle does not contain any boundary saddle. The Reeb graph of the triangular patch is a straight line, and every contour in the patch is homeomorphic to a line segment. We can reach any point from the extremum by walking along contours that monotonically sweep the patch. If the line criticality is a saddle, then it is the only point at which the connectivity of contours in the patch changes, because no other saddles exist. The Reeb graph has one internal node and three leaves. Starting from the saddle, we can walk to any other point in the patch by choosing the appropriate contour component to follow as it is swept away from the saddle. Now we assume that one vertex is a boundary saddle, as shown in Figure 7. If the line-critical point is also a boundary saddle, the Reeb graph has two saddles, which implies at least four extrema. This is impossible because there are only two vertices left. Therefore, the line criticality must be an extremum. We assume without loss of generality that it is a maximum. The two vertices on that edge must be minima, so the Reeb graph has one maximum, one saddle and two minima. A monotone path exists from the maximum to all points in the triangle.

Lemma 4. *If a triangular patch contains zero face-critical points and two line-critical points, the two line-criticalities are connected by a monotone path.*

Proof. Lemma 2 implies that there are no vertex saddles. Let e_0 and e_1 be the line criticalities. If both e_0 and e_1 are saddles, then they are connected by a monotone path because they are the only two interior nodes in the Reeb graph of the triangular patch. If both are extrema, then one must be a maximum and the other a minimum and the Reeb graph is a straight line. If e_0 is a boundary saddle and e_1 is an extremum, then there is a monotone path from that boundary saddle to every point in the patch.

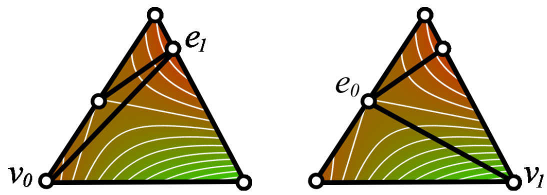
We use the above lemmata to prove that in all cases our tessellations consist of monotone subpatches, each with one maximum and one minimum.

Case $\langle \mathbf{0}, \mathbf{0} \rangle$: No line-critical point exists on any edge, and no face-critical point exists in the triangle. The function is monotone along the edges. There is exactly one maximum and one minimum, which occur at vertices. All contours in the triangle are homeomorphic to a line segment.

Case $\langle \mathbf{0}, \mathbf{1} \rangle$: One line-critical point e exists on an edge, and no face-critical point exists in the triangle. Let v_0, v_1 and v_2 be the triangle vertices, where v_0 is the vertex opposite e. We split the triangle in two, creating patches e, v_1, v_0 and e, v_2, v_0, each containing zero face and line criticalities. Considering Lemma 3, the arc e, v_0 is guaranteed to have a monotone embedding.

Case $\langle \mathbf{0}, \mathbf{2} \rangle$: Two line-critical points exist, e_0 and e_1, and no face-critical point exists. We first subdivide the triangle along the monotone arc between e_0 and e_1. Considering Lemma 4, we know that this arc exists. This arc splits the patch into a triangular subpatch and a quadrilateral subpatch. The triangular subpatch belongs to Case $\langle 0, 0 \rangle$. Both e_0 and e_1 may not be boundary saddles because this implies that the Reeb graph of the triangular patch has two saddles and two extrema, which is an impossible configuration. Let e_0 be a saddle and e_1 an extremum. Figure 8 shows a patch in this configuration. If we triangulate the quadrilateral by adding an arc which does not terminate at e_0, then e_0 still is a saddle of its triangular subpatch, which violates our desired monotonicity property. To prevent this, we always triangulate the quadrilateral by adding an arc which has e_0 as an endpoint. If e_0 and e_1 are both extrema, the Reeb graph is a line because no other saddles exist.

Case $\langle \mathbf{1}, \mathbf{0} \rangle$: This case is impossible. Face-critical points occur in elliptic and hyperbolic functions only. We first consider a triangular patch containing an



(a) Incorrect (b) Correct

Fig. 8. A patch in Case $\langle 0, 2\rangle$. Tessellation (a) is incorrect because the edge v_0e_1 crosses the same contour twice and cannot be monotone. The boundary saddle at e_0 is correctly connected to v_1 in tessellation (b).

elliptic minimum. Tracking the topology of level sets during a sweep in the increasing function direction, we note that a loop, contained entirely inside the triangle, is created at the minimum. This loop cannot be destroyed by a vertex without first being converted into a segment by a line criticality. Therefore, the triangle boundary should contain at least one line criticality. A similar argument holds when the triangular patch contains an elliptic maximum. Now we consider a triangular patch containing a hyperbolic face-critical point. The level set at this critical point consists of a pair of intersecting asymptotes. These asymptotes intersect the triangle boundary at four unique points. Since there are only three edges, at least one edge intersects the asymptotes twice. This triangle edge contains a line-critical point between the two intersection points.

Cases $\langle 1, 1\rangle$, $\langle 1, 2\rangle$, $\langle 1, 3\rangle$: We tessellate the patch by connecting the face criticality to all vertices and line criticalities on the boundary. These new arcs are monotone because any line-restriction containing the face criticality as an end-point cannot contain a line criticality. All possible critical points appear as nodes in the tessellation, all new subpatches are triangular and contain zero face and line criticalities. Figures 3 and 5 show elliptic and hyperbolic patches in this configuration.

Case $\langle 0, 3\rangle$: Monotone arcs exist between all three pairs of line criticalities. We require a few more results in order to prove the above statement. For the remainder of this paper we assume that all bivariate quadratics are given in standard form, i.e., $B, D, E = 0$ and $F = 1$. (Any bivariate quadratic can be transformed into the standard form using homeomorphisms that preserve the topological properties we are interested in.)

Lemma 5. *Let f be a parabolic or elliptic bivariate quadratic function. If at least one line restriction of f contains a line-critical point, then all line-critical points are of the same type.*

Proof. The determinant of the Hessian matrix of f is $H = 4AC$. If $H \geq 0$, f is parabolic or elliptic. The type of line criticality on a line-restriction is determined by the sign of second derivative of the univariate quadratic defined over that line, which is 2α, where $\alpha = Ax_d^2 + Cy_d^2$. Since x_d^2 and y_d^2 are both positive, the sign of α is determined by A and C. When $H = 0$, either A or C is zero, so, if the other is non-zero then its sign is the sign of α. If both A and C are zero, then f is linear. When $H > 0$, then A and C have the same sign, so α has this sign as well. For any parabolic or elliptic function f the sign of A and C determine the sign of α for any line-restriction, and so all line-restrictions of f must have the same sign of α, and thus the same type of line-critical point.

Lemma 6. *It is impossible for an elliptic or parabolic patch in Case $\langle 0, 3\rangle$ to have two line-criticalities which are both minima, or both maxima.*

Proof. We assume that two line-criticalities are minima of the triangular patch, i.e., they each create a component. Because they are minima of the patch, they must also be minima of the function restricted to their edges. Lemma 5 implies that the third line criticality must also be a minimum of its edge. If all three line criticalities are minima of their edges, then all three vertices are maxima. The Reeb graph then has two minima and three maxima, implying at least three saddles, which is a larger number of critical points than we have assumed.

Lemma 7. *Consider a triangular patch in Case $\langle 0, 3\rangle$ with six critical points. The only possible Reeb graph for this patch is the one where all three vertices are minima (maxima), two line-criticalities are saddles, and one line-criticality is a maximum (minimum), as shown in Figure 9.*

Proof. All three vertices must be critical points; if they were not, then the tree could not have six nodes. We assume that one vertex is a minimum and another is a maximum. The edge between these vertices cannot have a line-criticality, because the maximum vertex has a greater value than every point on the edge, and the minimum vertex has a lower value than every point on the edge, and the edge cannot have any inflection points. Thus, if this edge does not have a line-criticality, then there are fewer than six critical points, which violates the assumption. Thus, all vertices must be minima, or all must be maxima.

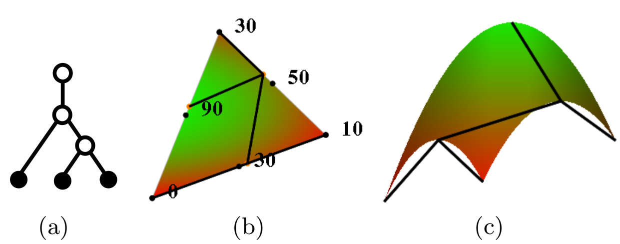


Fig. 9. (a) The only possible Reeb graph in Case $\langle 0, 3\rangle$ with six nodes. The solid nodes are triangle vertices, the open nodes are line-critical points. Two views of such a patch are shown in (b) and (c).

Lemma 8. *Let T be the set of all line-critical points of line-restrictions which are parallel to the direction (x_d, y_d). All elements $(x, y) \in T$ satisfy $Ax_dx + Cy_dy = 0$.*

Proof. We recall that the (now simplified) definition of the univariate quadratic along a line restriction of f is $\alpha t^2 + \beta t + \gamma$, and $\hat{t} = \beta/2\alpha$, where $\alpha = Ax_d^2 + Cy_d^2$ and $\beta = 2Ax_0x_d + 2Cy_0y_d$. Instead of defining a line restriction with an arbitrary point, let (x_0, y_0) be the line critical point by setting $\hat{t} = 0$, leading to: $Ax_dx_0 + Cy_dy_0 = 0$ and $Ax_d^2 + Cy_d^2 \neq 0$. Thus any line-critical point $x = x_0, y = y_0$ satisfies $\beta/2 = Ax_dx + Cy_dy = 0$, with $\alpha \neq 0$ implying that the line-restriction direction (x_d, y_d) is not parallel to an asymptote. So all line-restrictions which share a common direction have all of their line-critical points on this line, $Ax_dx + Cy_dy = 0$, which we refer to the *critical line*.

There are two possible Reeb graphs for Case $\langle 0, 3\rangle$, one with six nodes and one with four nodes. If there are six nodes, then Lemma 7 implies that the tree looks like the example shown in Figure 9, in which case all line-critical points are connected by a monotone path. We now consider the case that there are four nodes. Because we have three line criticalities, one line criticality must be a saddle and the other two are extrema. If the extremal line criticalities are a maximum and a minimum, then all three line criticalities are connected by a monotone path. The only case in which two line criticalities could not be connected by a monotone path is the case when they are both minima or both maxima. If so, Lemma 6 implies that the patch is hyperbolic. We show that in order for a hyperbolic triangular patch to contain two line criticalities that are extrema, it must also contain the origin, and hence a face-criticality, so it is not in Case $\langle 0, 3\rangle$. We assume that the two line criticalities are minima. (By inverting the function, the result will hold for two maxima.)

We partition the set of line restrictions based upon which "spokes" the line intersects. A spoke is one half of a hyperbolic asymptote, a ray originating at the origin, see Figure 10(a). There are four spokes, and a line may intersect one, two, or four of them. If a line hits only one spoke, it is parallel to an asymptote and contains no line criticality. If a line hits four spokes, it passes through the origin and, by convention, any triangle having this line as an edge contains a face-critical point. Thus, if a triangle edge intersects one or four spokes, then the triangle is not in Case $\langle 0, 3\rangle$. We narrow our focus to the case when a line intersects two spokes. There are four possible pairs of spokes a line might intersect: (1,2), (2,3), (3,4) and (4,1). We further divide each of these four cases into two subcases, based on the sign of the slope of the line. (If the line has zero or infinite slope, we can arbitrarily assign it to either the + or − subcase without effect.)

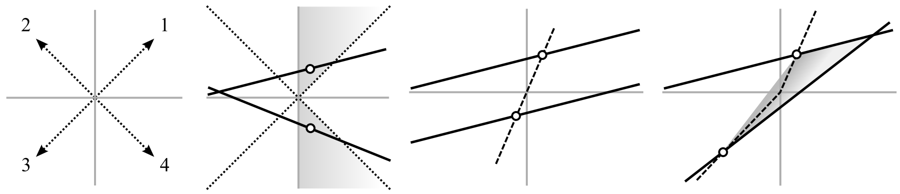
We consider pairs of line restrictions, to be taken as edges of a triangular patch. To avoid considering many redundant cases, we first fix one line of the pair as hitting spokes one and two, and having positive slope. We can assume this without loss of generality because any line can be transformed into this configuration by rotating or mirroring about an axis. Further, we assume that the regions between spoke pairs (4,1) and (2,3) have positive function value, and the regions between spoke pairs (1,2) and (3,4) have negative value. This is equivalent to setting the coefficients $A > 0$ and $C < 0$, which we can ensure by swapping the x and y coordinates. Having fixed one line of the pair, we consider the eight possible configurations of the free line. For each configuration, we show that either the triangle is not in Case $\langle 0, 3 \rangle$, or that the line criticalities cannot both be minima. The subcases are labeled as (m, n, s), meaning that the free line intersects spokes m and n with slope of sign s. (In the following, the symbol $*$ means *don't care*.)

Subcases $(\mathbf{4}, \mathbf{1}, *)$ and $(\mathbf{2}, \mathbf{3}, *)$: The line criticality has positive value, and thus cannot be a minimum.

Subcase $(\mathbf{1}, \mathbf{2}, *)$: The fixed line and the free line intersect the same pair of spokes, which implies that they are tangential to two hyperbolic curves belonging to the same family of curves, one which smoothly evolves into another by changing the contour value. As we track this evolution by sweeping the contour value, the contour first enters the triangle at the lower-valued of the two line criticalities. When we reach the value of the other line criticality, the contour already exists within the triangle. The second line criticality cannot create a second contour segment, so it cannot also be a minimum.

Subcase $(\mathbf{3}, \mathbf{4}, -)$: For this case, we make use of the following property: If a line is perpendicular to a gradient direction (x_d, y_d), then the tangency point (x_p, y_p) has $\text{sign}(x_d) = \text{sign}(x_p)$ and $\text{sign}(y_d) \neq \text{sign}(y_p)$. This fact follows from the definition of the gradient of f and $A > 0$, $C < 0$, see Figure 10(b). The free line has negative slope, meaning it is perpendicular to a direction (x_d, y_d) with $x_d, y_d > 0$, so the line-critical point has $x > 0, y < 0$. The fixed line has positive slope and thus the line-critical point has $x > 0, y < 0$. The intersection of the two lines (a vertex of the triangle) has $x < 0$. If a triangle contains all three of these points, it must also contain the origin, which is a face-criticality, and hence the triangle is not in Case $\langle 0, 3 \rangle$.

Subcase $(\mathbf{3}, \mathbf{4}, +)$: See Figure 10(c). Both lines have positive slopes, but lie on opposite sides of the origin. Considering Lemma 8, we know that a line restriction parallel to the direction (x_d, y_d) has a critical line with equation $Ax_d x + Cy_d y = 0$, which has slope $-Ax_d/Cy_d$. Since $\text{sign}(A) \neq \text{sign}(C)$, making $-A/C$ positive, we therefore have that the slope of the line restriction, y_d/x_d, is inversely proportional to the slope of its critical line. In other words, as we rotate a line restriction, its critical line is rotated in the opposite direction. We consider that the fixed line and the free line are parallel, and thus they share a common critical line. No triangle can have two parallel edges, so we must rotate one line by some angle. We rotate the free line counter-clockwise, which rotates its critical line clockwise, as shown in Figure 10(d). The two critical lines form an angle which is smaller in the direction *opposite* that of the triangle vertex formed by the two lines. The third triangle edge cannot be placed to contain both line-critical points and also exclude the origin, and so the triangle is not in Case $\langle 0, 3 \rangle$. The situation is the same if we rotate the free line clockwise.



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

Fig. 10. (a) The asymptote "spokes." (b) Subcase $(3, 4, -)$. The line-critical points lie in the shaded regions. If a triangle includes these two points, it must also include the origin. (c) Subcase (3,4,+). (d) As the bottom line rotates counter-clockwise, its critical line (dashed) rotates clockwise. The triangle must contain the shaded area, and thus the origin.

This completes the subcase analysis of Case $\langle 0, 3\rangle$. The above case analysis implies that none of the tessellated patches contains a critical point in its interior or on its boundary. Thus we have the following result:

Theorem. *Given a quadratic function defined on a triangle, the tessellation defined in Section 3 partitions the triangular patch into monotonic subpatches.*

6 Conclusions and Future Work

We have described a theoretically sound tessellation scheme for piecewise-quadratic functions defined on triangulated surfaces. Our tessellation scheme allows existing Reeb graph and contour tree construction algorithms to be applied to a new class of inputs, thereby extending the applications of these topological structures. We intend to develop a similar tessellation scheme for trivariate quadratic functions defined over tetrahedral meshes in a piecewise fashion. We believe that this work can contribute to the development of numerically robust and topologically consistent methods for analysis of functions specified as higher-order interpolants or approximation functions over two- and three-dimensional domains.

Acknowledgments

This work was supported in part by the National Science Foundation under contract ACI 9624034 (CAREER Award) and by a large Information Technology Research (ITR) grant. Vijay Natarajan was also supported by a faculty start-up grant from the Indian Institute of Science. We thank the members of the Visualization and Computer Graphics Research Group at the Institute for Data Analysis and Visualization (IDAV) at the University of California, Davis.

References

1. Reeb, G.: Sur les points singuliers d'une forme de Pfaff completement integrable ou d'une fonction numrique. Comptes Rendus Acad. Sciences **222** (1946) 847–849

2. Worsey, A., Farin, G.: Contouring a bivariate quadratic polynomial over a triangle. Comput. Aided Geom. Des. **7**(1-4) (1990) 337–351
3. Nielson, G., Hamann, B.: The asymptotic decider: resolving the ambiguity in marching cubes. In Nielson, G.M., Rosenblum, L.J., eds.: Proc. Visualization '91, IEEE Computer Society Press (1991) 83–91
4. Nielson, G.: On marching cubes. IEEE Trans. Visualization and Comp. Graph. **9**(3) (2003) 283–297
5. Pascucci, V., Cole-McLaughlin, K.: Efficient computation of the topology of level sets. In Gross, M., Joy, K., Moorhead, R., eds.: Proc. Visualization '02, IEEE Computer Society Press (2002) 187–194
6. Carr, H.: Topological Manipulation of Isosurfaces. PhD thesis, University of British Columbia (2004)
7. Carr, H., Snoeyink, J., Axen, U.: Computing contour trees in all dimensions. Comput. Geom. Theory Appl. **24**(2) (2003) 75–94
8. Cole-McLaughlin, K., Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Loops in Reeb Graphs of 2-Manifolds. Discrete and Computational Geometry **32**(2) (2004) 231–244
9. Chiang, Y.J., Lenz, T., Lu, X., Rote, G.: Simple and optimal output-sensitive construction of contour trees using monotone paths. Comput. Geom. Theory Appl. **30**(2) (2005) 165–195
10. Bajaj, C.L., Pascucci, V., Schikore, D.R.: The contour spectrum. In Yagel, R., Hagen, H., eds.: Proc. Visualization '97, IEEE Computer Society Press (1997) 167–174.
11. Takahashi, S., Nielson, G.M., Takeshima, Y., Fujishiro, I.: Topological volume skeletonization using adaptive tetrahedralization. In Hu, S.M., Pottmann, H., eds.: Proc. Geometric Modeling and Processing 2004, IEEE Computer Society Press (2004) 227–236
12. Weber, G.H., Scheuermann, G., Hagen, H., Hamann, B.: Exploring scalar fields using critical isovalues. In Gross, M., Joy, K., Moorhead, R., eds.: Proc. Visualization '02, IEEE Computer Society Press (2002) 171–178
13. Carr, H., Snoeyink, J., van de Panne, M.: Simplifying flexible isosurfaces using local geometric measures. In mushmeier, H., Turk, G., van Wijk, J.J., eds.: Proc. Visualization '04, IEEE Computer Society Press (2004) 497–504
14. Weber, G.H., Dillard, S.E., Carr, H., Pascucci, V., Hamann, B.: Topology-Controlled Volume Rendering. IEEE Transactions on Visualization and Computer Graphics **13**(2) (2007) 330–341
15. Schroeder, W., Bertel, F., Malaterre, M., Thompson, D., Pebay, P.: Methods and Framework for Visualizing Higher-Order Finite Elements. IEEE Transactions on Visualization and Computer Graphics **12**(4) (2006) 446–460
16. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology matching for fully automatic similarity estimation of 3d shapes. In Fiume, E., ed.: Proc. 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01, ACM Press (2001) 203–212
17. Zhang, E., Mischaikow, K., Turk, G.: Feature-based surface parameterization and texture mapping. ACM Trans. Graph. **24**(1) (2005) 1–27
18. Milnor, J.W.: Morse Theory. Princeton University Press, Princeton, New Jersey (1963)