



TOPOLOGY PRESERVING DATA SIMPLIFICATION WITH ERROR BOUNDS

CHANDRAJIT L. BAJAJ¹ and DANIEL R. SCHIKORE²

¹Texas Institute for Computational and Applied Mathematics, Department of Computer Science,
University of Texas, Austin, TX 78712, U.S.A.

²Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94550, U.S.A.

Abstract—Many approaches to simplification of triangulated terrains and surfaces have been proposed which permit bounds on the error introduced. A few algorithms additionally bound errors in auxiliary functions defined over the triangulation. We present an approach to simplification of scalar fields over unstructured grids which preserves the topology of functions defined over the triangulation, in addition to bounding of the errors. The topology of a 2D scalar field is defined by critical points (local maxima, local minima, saddle points), in addition to integral curves between them, which together segment the field into regions which vary monotonically. By preserving this shape description, we guarantee that isocontours of the scalar function maintain the correct topology in the simplified model. Methods for topology preserving simplification by both point-insertion (refinement) and point-deletion (coarsening) are presented and compared. © 1998 Elsevier Science Ltd. All rights reserved

1. INTRODUCTION

Scientific data is often sampled or computed over a dense mesh in order to capture high-frequency components or achieve a desired error bound. Interactive display and navigation of such large meshes is impeded by the sheer number of triangles required to sufficiently model highly complex data. A number of simplification techniques have been developed which reduce the number of triangles to a particular desired triangle count or until a particular error threshold is met. Given an initial triangulation M of a domain \mathcal{D} and a function $\mathcal{F}(x)$ defined over the triangulation, the simplified mesh can be called M' and the resulting function $\mathcal{F}'(x)$. The measure of error in a simplified mesh M' is usually represented as

$$\epsilon(M') = \max_{x \in \mathcal{D}} (|\mathcal{F}(x) - \mathcal{F}'(x)|). \quad (1)$$

The ability to bound the error $\epsilon(M')$ is very important, but the error definition (1) is inherently a local measure, neglecting to consider global features of the data. We introduce new criteria for the simplification of sampled functions which preserves scalar field features in addition to bounding local errors. Two-dimensional scalar field topology is described by the critical points and arcs between them. Preserving the scalar field criticalities maintains an invariant of the connectivity and combinatorial structure (topological genus) of successively simplified isocontours.

In Section 2 we discuss related work in mesh simplification and feature detection. Section 3 intro-

duces the definition of 2D scalar topology as it will be used in our simplification strategy. In Section 4 we introduce two algorithms for simplification with topology preserving characteristics. The first is an extension to existing *coarsening* techniques which iteratively delete vertices or contract edges in the mesh. The second algorithm adopts an inverse approach, iteratively introducing detail (*refinement*) to an initially sparse mesh, preserving the scalar topology of the fine mesh.

2. RELATED WORK

2.1. Simplification

A wide variety of algorithms have been developed for the simplification of meshes. We present a brief overview of the classes of algorithms which have been proposed for geometry and data simplification.

2.1.1. *Vertex insertion/deletion.* A large class of data and geometry simplification algorithms are based on successive application of one or more topological mesh operators, such as *edge contraction*, which contracts an edge of the mesh to a point, or *vertex deletion*, in which a vertex and adjacent triangles are removed and replaced with a covering of the resulting hole. Point insertion and deletion approaches have been explored by many researchers for application in geographical information systems (GIS). A common technique is to extract key points of data from the originally dense set of points and compute a Delaunay triangulation

[7, 11, 12, 33, 39, 41]. Silva *et al.* [37] use a greedy approach for inserting points into an initially sparse mesh. Schroeder *et al.* [36] compute reduced representations for dense triangular surface meshes such as those computed by marching cubes [25] or similar isosurfacing algorithms. Vertices in the dense mesh are examined and classified based on geometric features in the triangulation surrounding the vertex. If error criteria are satisfied, the vertex is deleted and the resulting hole is retriangulated. Retriangulation is guided by local edges detected in the classification stage and aspect ratios of new triangles. Several passes over the object successively remove vertices until no vertex satisfies the criteria for removal. There is no cumulative error measure, and therefore no guarantee on the amount of accumulated error in the final representation. Hamann [15] applies a similar technique in which triangles are considered for deletion based on curvature estimates at the vertices. Reduction may be driven by mesh resolution or, in the case of functional surfaces, root-mean-square (RMS) error. Ronfard and Rossignac [34] apply successive edge contraction operations to compute a wide range of levels-of-detail for triangulated polyhedra. Edges are extracted from a priority queue based on a computed *edge cost* such that edges of lesser significance are removed first. Guéziec [14] introduces a *tolerance volume* for bounding the error resulting from successive edge contraction operations. The resulting merged vertex is positioned such that the volume remains constant. Cohen *et al.* [6] introduce *Simplification Envelopes* to guide mesh simplification with global error bounds. Envelopes are an extension of *offset surfaces* which serve as an extreme boundary for the desired simplified surface.

2.1.2. *Region merging.* Hinker and Hanson [19] perform “geometric optimization” on triangular surface meshes by grouping faces into contiguous sets which are nearly co-planar. Points interior to a region and points along nearly linear boundaries of regions are deleted and the resulting hole is retriangulated. Kalvin and Taylor [24] cluster mesh faces into *superfaces*, triangulating the resulting polygons for a simplified representation.

2.1.3. *Filtering.* Filtering techniques are capable of producing a large range of simplified models through application of grouping and merging rules. An attractive feature of filtering techniques is the ability to simplify objects to a minimal representation through successive applications. Subsampling is a simple type of filtering which is easily applied to subdivision meshes for which there exists a natural remeshing when nested sets of vertices are successively deleted. The major drawback to subsampling is that there is no bound on the error which is introduced through its application. Rossignac and Borrel [35] use clustering and merging of features of an object based on a regular spatial subdivision. Clustering approaches have the advantage

that small features which are geometrically close but not topologically connected can be grouped and merged for higher rates of simplification. In this scheme long, thin objects may collapse to an edge and small objects may contract to a point. He *et al.* [16] provide more control over subsampling of regular grids by filtering the simplified mesh at each step. The regular grid corresponds to a sampling of the signed-distance function of a 3D surface. A multi-resolution triangle mesh is extracted from the resulting multi-resolution volume buffer using traditional isosurfacing techniques.

2.1.4. *Optimization.* Optimization methods define measures of energies for point sets or triangulations based on an original mesh, and use interactive optimization to minimize these energies in forming a simplified mesh. Turk [40] computes simplified polygonal surfaces at a desired number of vertices. Contrast this with the point insertion and deletion methods which are usually driven by error computations rather than desired resolution. Given the desired number of vertices, point repulsion on the polygonal surface spreads the points out. A mutual tessellation of the original triangulation and the introduced points followed by deletion of the original vertices guarantees that the topology of the polygonal surface is maintained. Point repulsion is adjusted based on estimated curvature of the surface, providing an adaptive triangulation which maintains geometric features. Hoppe *et al.* [21] perform time-intensive mesh optimization based on the definition of an energy function which balances the need for accurate geometry with the desire for compactness in representation. The level of mesh simplification is controlled by a parameter in the energy function which penalizes meshes with large numbers of vertices, as well as a spring constant which helps guide the energy minimization to a desirable result. In [20], Hoppe applies the optimization framework to the simplification of scalar fields.

2.1.5. *Multi-resolution analysis.* Multi-resolution analysis is a structured mathematical decomposition of functions into multiple levels of representation. Through the use of wavelet transforms [10, 27], a hierarchical representation of functions can be obtained by repeatedly breaking the function into a coarser representation and a corresponding set of perturbation coefficients which allow the full recovery of the original representation from the coarse representation. Generally, the wavelet basis is chosen such that the perturbation coefficients have desirable attributes such as direct correlation with some measure of error which is introduced at a given level of representation. During reconstruction from the wavelet representation, sufficiently small wavelet coefficients can be left out, resulting in a coarser approximation to the original data, with a known bound on the amount of error [8, 26, 38]. Further extensions have provided a similar basis for the decomposition of surfaces [9]. Muraki [31]

applies wavelets in 3D to compute multi-resolution models of 3D volume data. Isosurfaces and planar cross sections of the resulting data show little change in image quality with large reductions in the amount of data representing the volume.

2.2. Feature detection

The problem of detecting ridges and valleys in digital terrain has been considered in several papers [12]. McCormack *et al.* [29] consider the problem of detecting drainage patterns in geographic terrain. Interrante *et al.* [22] used ridge and valley detection on 3D surfaces to enhance the shape of transparently rendered surfaces. Extrema graphs were used by Itoh and Koyamada to speed isocontour extraction [23]. A graph containing extreme points and boundary points of a scalar field can be guaranteed to intersect every isocontour at least once, allowing seed points to be generated by searching only the cells contained in the extrema graph. Helman and Hesselink detect vector field topology by classifying the zeros of a vector field and performing particle tracing from saddle points [18]. The resulting partitioning consists of regions which are topologically equivalent to uniform flow. Globus *et al.* [13] describe a software system for 3D vector topology and briefly note that the technique may also be applied to the gradient of a scalar field in order to identify maxima and minima. Bader [1] examines the gradient field of the charge density in a molecular system. The topology of this scalar field represents the bonds linking together the atoms of the molecule. Bader goes on to show how higher level structures in the topology represent chains, rings, and cages in the molecule.

2.3. Our approach

Simplification techniques have advanced to the point at which it is useful to now consider preserving global mesh and data features. In the following section we present a definition and method of computation of *scalar field topology*, which describes scalar field structure in terms of criticalities and relationships between them and captures the topological connectivity of isocontours. Based on these definitions, we develop two approaches for simplification which are motivated by the preservation of the scalar topology, and hence the preservation of the topology of approximate isocontours. The first approach uses a *bottom-up* simplification approach, consisting of a small set of consistency checks which guarantee scalar topology consistency for the case of iterative simplification by point deletion or edge contraction. The second approach inverts the order of operation, beginning with the scalar topology computation, which serves as a sparse base mesh which is refined to meet error-bound criteria.

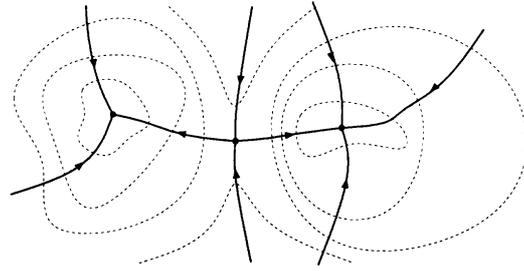


Fig. 1. Isocontours (dashed) of a scalar field along with the critical points and integral curves.

3. SCALAR FIELD TOPOLOGY

Scalar field structure can be characterized by the critical points of the scalar field and higher-order relationships between them [4]. A point x is a critical point on the scalar function $\mathcal{F}(x)$ if all first-order partial derivatives of \mathcal{F} evaluated at x are zero [30]. In degenerate cases, a critical point may be part of a larger critical *curve* or critical *region*. We will restrict our attention to critical points, because our motivating work is not greatly affected by the treatment of these special cases.

For our purpose of feature detection and application to simplification, the topology of a scalar field \mathcal{F} defined over a domain \mathcal{D} consists of the following:

- (i) The local *maxima* of \mathcal{F} .
- (ii) The local *minima* of \mathcal{F} .
- (iii) The *saddle points* of \mathcal{F} .
- (iv) Selected *integral curves* joining each of the above.

Integral curves are defined as curves which are everywhere tangent to the gradient field of \mathcal{F} . In vector field topology, the curves adverted in the flow-field segment the field into regions which are topologically equivalent to uniform flow [17]. In the case of scalar topology, the analogous property holds that integral curves partition the field into regions in which the gradient flow is uniform, or in other words, the scalar function is monotonic. Another way of understanding this property is that within each region, the isocontours of the scalar field will consist of a single connected component.

Figure 1 displays several isocontours of a scalar field and the corresponding scalar field topology.

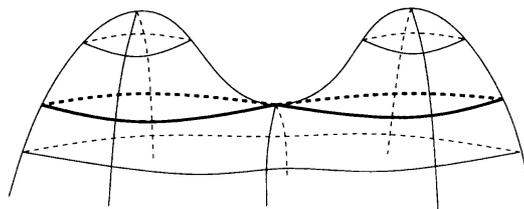


Fig. 2. A surface representation of the function in Fig. 1. A change of contour topology occurs at the saddle point.

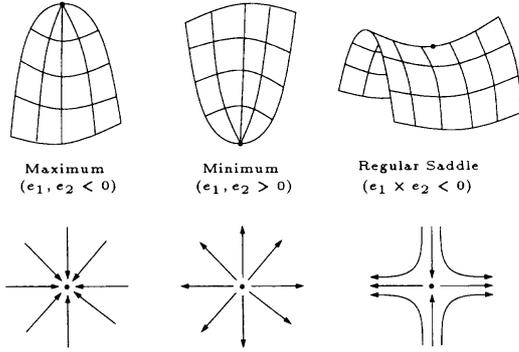


Fig. 3. Scalar critical point classifications.

Figure 2 displays a surface representation of the same field.

The procedure for computation of scalar topology can be broken down into three stages:

- (i) Detect critical points of \mathcal{F} .
- (ii) Classify critical points.
- (iii) Compute selected integral curves in gradient field.

In the following subsections we describe the above components under the assumption that the scalar field \mathcal{F} has continuous first derivatives. In Section 3.4 we will remove this restriction and discuss scalar topology computation for the case of a piecewise C^0 function defined over a triangulation.

3.1. Computing critical points

Critical points of a scalar function are defined as points at which the gradient vanishes [28]. Given a d -dimensional scalar field \mathcal{F} , the critical points correspond to solutions of the system of equations:

$$\begin{pmatrix} \frac{\partial \mathcal{F}}{\partial x_1} \\ \vdots \\ \frac{\partial \mathcal{F}}{\partial x_d} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

3.2. Classification of critical points

Qualitative information about the behavior of the gradient field near a critical point is obtained by analysis of the Hessian of \mathcal{F} (2D):

$$\begin{bmatrix} \frac{\partial^2 \mathcal{F}}{\partial x^2} & \frac{\partial^2 \mathcal{F}}{\partial x \partial y} \\ \frac{\partial^2 \mathcal{F}}{\partial y \partial x} & \frac{\partial^2 \mathcal{F}}{\partial y^2} \end{bmatrix}.$$

The eigenvalues (e_1, e_2) and eigenvectors (v_1, v_2) of the above matrix relate to the behavior of the gradient field, and hence the scalar field, near the critical point, much the same as for the behavior of a general vector field [5, 18]. One difference to note is that for a gradient field the matrix of derivatives is symmetric ($\partial^2 \mathcal{F} / \partial x \partial y = \partial^2 \mathcal{F} / \partial y \partial x$), and therefore the eigenvalues are real. This is intuitively expected,

as imaginary eigenvalues indicate rotation about the critical point, and a gradient field is an irrotational vector field. This observation allows us to simplify the classification of critical points to the cases depicted in Fig. 2 (additional degenerate criticalities exist when $e_1 = 0$ or $e_2 = 0$).

A positive eigenvalue corresponds to gradient flow away from the critical point, while a negative eigenvalue indicates gradient flow toward the critical point. In the case of a saddle point, there is gradient flow toward and away from the critical point, distinguishing them from the field behavior near other critical points. In this case, the eigenvectors corresponding to the positive and negative eigenvalues define the separatrices of the saddle in the directions of flow toward and away from the critical point, respectively. It is this property that is used in the next section to compute integral curves in the gradient field.

3.3. Computing integral curves

Having computed and classified the critical points, the final step for describing the scalar topology is the tracing of selected integral curves between the detected points. Saddle points have the property that the eigenvectors of the Hessian are the separatrices of the saddle. The behavior of the scalar field near the saddle point can be determined based on the behavior of the function in the direction of the eigenvectors. Integral curves are defined as curves which are everywhere tangent to the gradient field of \mathcal{F} . For a 2D scalar field, four integral curves meet at a saddle, dividing the domain into regions. Two curves follow the (forward and backward) direction corresponding to the positive eigenvalue, and two follow the direction corresponding to the negative eigenvalue.

Integral curves can be computed efficiently using a 4th-order adaptive step Runge Kutta integration in the gradient field [32]. The initial position for the iterative stepping is placed a small distance from the saddle point along the appropriate eigenvector. Computation of the integral curve ends when we reach the vicinity of another critical point within a certain ϵ , in which case the curve terminates at that point. Other curves may end at the boundaries of the mesh. The scalar topology for a simple function is shown in Fig. 4.

3.4. Piecewise linear topology

Critical point analysis for functions with continuous derivatives provides a fundamental theory for describing the structure of scalar fields. Extending these definitions for functions which are not smooth requires relaxation of some of our definitions. We address the case of a function which is sampled at the vertices of a 2D triangle mesh and linearly interpolated within each triangle.

For the case of a piecewise linear function, we must revise our definition of ‘‘critical point’’ due to

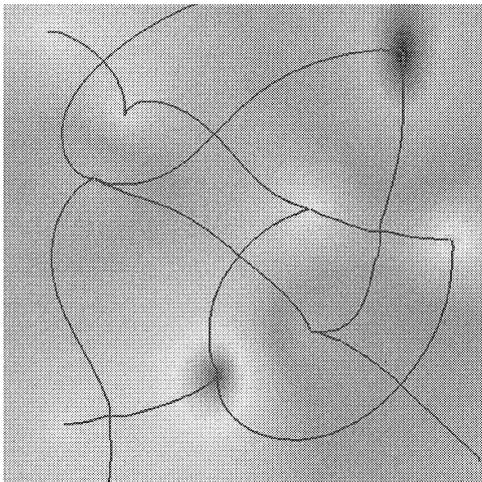


Fig. 4. Scalar topology of a simple scalar field. In this example only interior criticalities are computed. Additional criticalities occur at the boundaries.

the fact that the gradient is not everywhere defined. The feature that we want to capture is the change in the topology of the isocontours, as was illustrated in Fig. 2. It is clear that the contour topology does not change within a triangular cell, and so we confine our attention to the classification of vertices with respect to the local neighborhood. In the general case at a vertex, the gradient is undefined, and may be considered to take on a range of values based on the normals of the adjacent triangles, as illustrated in Fig. 5. As a result, critical points are defined as vertices at which the normal space of the adjacent triangles include the vector $(0, 0, 1)$. Figure 4 illustrates several of the cases which may occur. Fig. 6 shows the cases of pseudo-critical points.

Computation of integral curves requires a similar modification, as the gradient is undefined along the edges of the triangulation as well. Given a saddle point in a piecewise linear triangulation, the local topology is determined by examining the adjacent vertices in a clockwise or counter-clockwise order. Each local maximum or local minimum in this 1D ordering corresponds to a “ridge” or “channel” of the scalar field. Integral curves can be computed by traversing the edges of the triangulation corresponding to ridges and channels. Figure 3 illustrates a the topology of a piecewise-linear function similar to that presented in Figs 1 and 7.

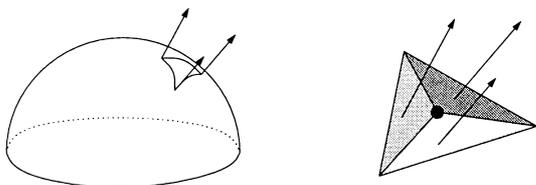


Fig. 5. Normal space (left) for a cluster of three triangles incident on a vertex (right).

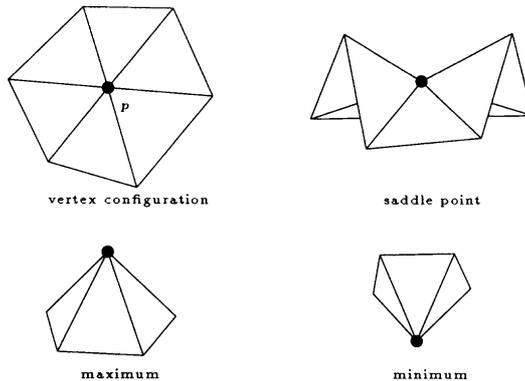


Fig. 6. A subset of the pseudo-critical points for a piecewise linear scalar field.

4. TOPOLOGY PRESERVING SIMPLIFICATION

We describe two approaches to the simplification of functional meshes while maintaining the structure, as defined by the scalar topology. We assume that the domain is discretized by a triangular mesh M of n_t triangles T_i and n_v vertices V_j . The function $\mathcal{F}(x)$ is defined for $x = V_j$ and is linearly interpolated over each triangle. $\mathcal{F}(T_i)$ is used to represent the local interpolant over a triangle.

4.1. Bottom-up simplification

In this section we derive necessary and sufficient conditions which avoid the change of scalar topology in the framework of existing *point-deletion* or *edge contraction* simplification techniques.

The basic framework of our approach is an error-bounded removal of vertices based on an ordering criteria [2, 3]. We restrict our simplification to use the original vertices and to local operations such as vertex deletion and vertex collapse, as illustrated in Fig. 8. The new mesh which results from the k -th step of simplification is called M^k , and its triangles are denoted T_i^k . We further define $\mathcal{F}^k(x)$ to represent the piecework defined height function associated with the mesh M^k . Note that for all triangles not involved in the operation at step k , $T_i^k = T_i^{k-1}$ and thus $\mathcal{F}^k(T_i^k) = \mathcal{F}^{k-1}(T_i^{k-1})$. Initially, we have $M^0 = M$ and $T_i^0 = T_i$.

Error bounds are computed using a two-phase process: (i) measurement of error introduced by a point deletion and (ii) approximation of the accu-

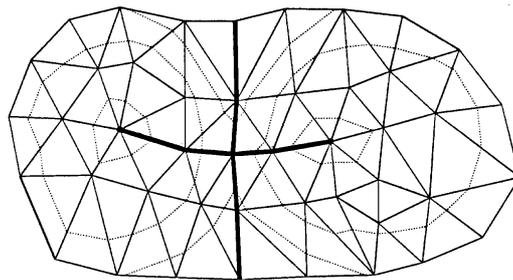


Fig. 7. Topology of a piecewise linear scalar field.

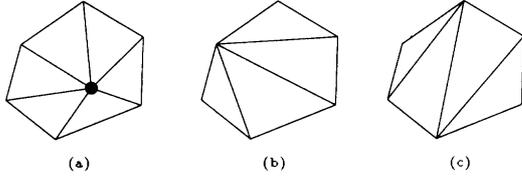


Fig. 8. General simplification operations applied to a local region of the mesh in (a). In (b), the resulting triangulation is formed by collapsing the deleted point to an adjacent point. In (c), the retriangulation is general and may be chosen to minimize the error introduced.

mulated error due to all point deletions in an overlapping region. In the following sections we describe the approach for bounding errors and derive conditions which preserve the topology of our given function.

4.1.1. *Local measurement of error.* The first step in maintaining an error-bound on the simplification is to measure the error *introduced* by deleting a vertex or collapsing an edge. Computing the introduced error is particularly simple due to the limited local influence of the operation.

As illustrated in Fig. 9, a local retriangulation due to removal of a vertex replaces the triangles T_i^{k-1} with triangles T_i^k . The error incurred at any point x in the local domain is quantified by the difference between the interpolated function value in the two triangulations, defined as $e^k(x) = \mathcal{F}^{k-1}(x) - \mathcal{F}^k(x)$. As the difference of two linear functions is again linear, the introduced error is defined as a piecewise linear function over the decomposition implied by the intersections of the triangulations, as illustrated above. The regions of the decomposition are called R_i^k and the (continuous) range of introduced error across the region is $[\epsilon_i^-(R_i^k), \epsilon_i^+(R_i^k)]$. Note that the $\epsilon_i^{\pm}(R_i^k)$ represent signed error values, and that the minimum and maximum errors occur at the extreme points of the convex regions, and thus can be determined by computing the errors at only a few points in the domain.

4.1.2. *Global errors from local errors.* Equally important as the measurement of error introduced through one simplification operation is the measure-

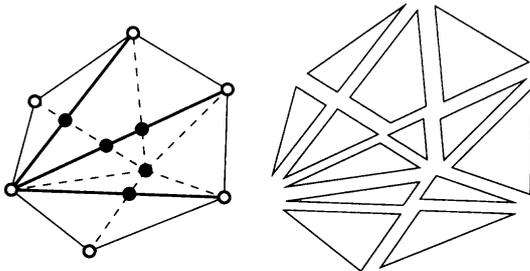


Fig. 9. Measurement of introduced errors at black vertices and the fragmentation into linear error regions. The introduced error at the white vertices and along the boundary is zero.

ment or approximation of errors which accumulate through successive operations. Features which are desirable in, or even required for, an error approximation strategy include:

- The estimation of error must be strict (must not underestimate the actual error).

- The estimation should be tight, such that the estimate provides a useful measure of the actual error.

At the same time, our desire for scalability and efficiency for extremely large scientific datasets requires that the strategy for bounding errors be simple and not impose a significant cost in addition to computing the introduced errors. We have designed a simple error representation and accumulation scheme which fulfills these criteria.

4.1.3. *Error representation.* We associate with each triangle two floating-point errors $\epsilon^-(T_i^k)$ and $\epsilon^+(T_i^k)$ (illustrated in Fig. 10, such that $x \in T_i^k$

$$\mathcal{F}^k(x) - \epsilon^-(T_i^k) \leq \mathcal{F}(x) \leq \mathcal{F}^k(x) + \epsilon^+(T_i^k). \quad (2)$$

Initially, $\epsilon^-(T_i^0) = \epsilon^+(T_i^0) = 0$, indicating that the original representation is exact. Note that one can easily incorporate input data with known source error bounds into this scheme by initializing the error bounds appropriately.

With such a representation in hand, we examine the problem of computing the errors $\epsilon^{\pm}(T_i^k)$ given $\epsilon^{\pm}(T_i^{k-1})$ such that inequality (2) is maintained as an invariant. We noted previously that the local domain is segmented into regions in which the introduced error varies linearly. Each introduced triangle T_i^k maps to a set of regions $\{R_{i,0}^k \dots R_{i,r}^k\}$. For each region $R_{i,j}^k$ we determine the minimal sufficient error range which aggregates the existing error with the accumulated error. Each region $R_{i,j}^k$ corresponds to a portion of a triangle T_j^k with associated error interval $[\epsilon^-(T_j^k), \epsilon^+(T_j^k)]$ and by interval arithmetic we arrive at the following inequality which bounds the error over the region $R_{i,j}^k$:

$$\begin{aligned} \mathcal{F}^k(R_{i,j}^k) - \epsilon^-(T_i^k) - \epsilon_i^-(R_{i,j}^k) &\leq \mathcal{F}(R_{i,j}^k) \\ &\leq \mathcal{F}^k(R_{i,j}^k) + \epsilon^+(T_i^k) + \epsilon_i^+(R_{i,j}^k) \end{aligned}$$

which leads to the accumulated error interval

$$\epsilon^-(R_{i,j}^k) = \epsilon^-(T_i^k) + \epsilon_i^-(R_{i,j}^k)$$

$$\epsilon^+(R_{i,j}^k) = \epsilon^+(T_i^k) + \epsilon_i^+(R_{i,j}^k).$$

For a minimal bound on the triangle T_i^{k+1} it is sufficient to take the maximum of the $\epsilon^{\pm}(R_{i,j}^k)$ for all regions $R_{i,j}^k$, which contribute to the triangle T_i^k . This process is illustrated geometrically for a simple 1D example in Figs 11 and 12.

4.1.4. *Preserving local topology.* Topology preservation is easily integrated into a vertex-delete and edge-contract simplification framework by considering the topology of the contours in the neighborhood of each candidate edge being considered for

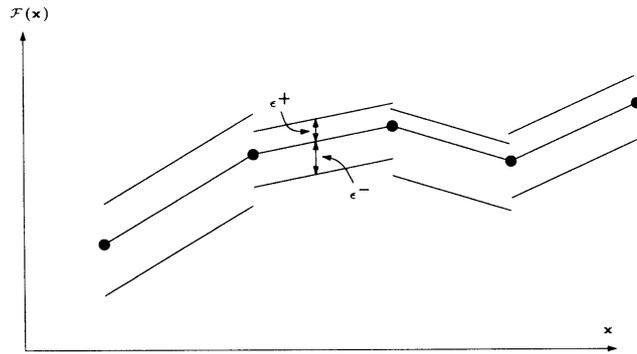


Fig. 10. 1D illustration of the accumulated error bound representation.

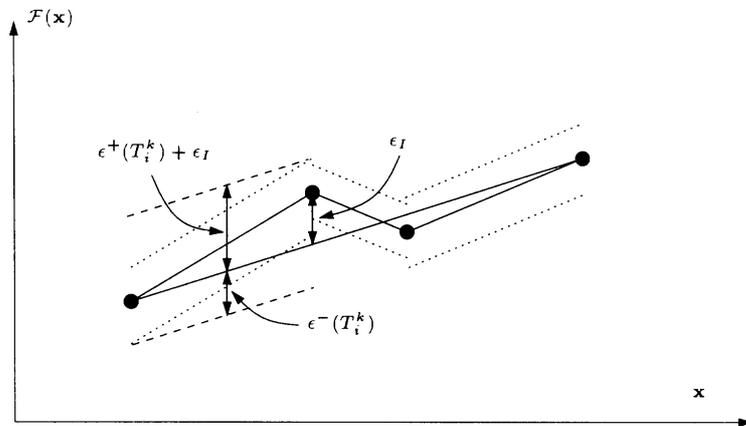


Fig. 11. Geometric interpretation of the accumulation of error bounds. The introduced error ϵ_I contributes to the error interval as illustrated.

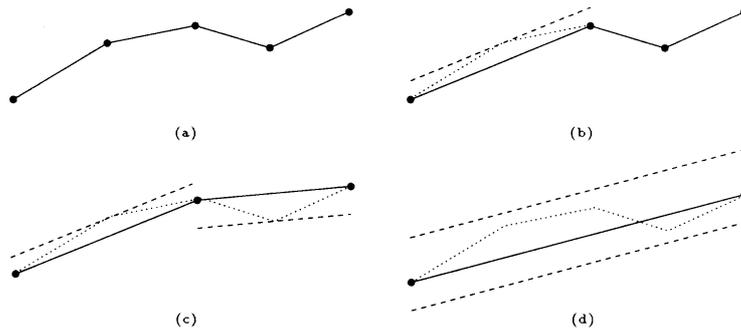


Fig. 12. Illustration of propagation of error bounds. As each successive point is deleted, the error bounds on each segment are updated to reflect the minimum bound between the new segment and the original polyline.

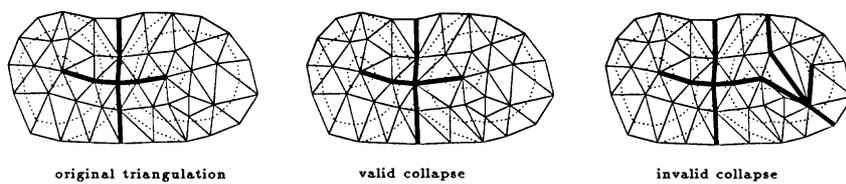


Fig. 13. Illustration of topology preserving and topology modifying edge contraction operations.

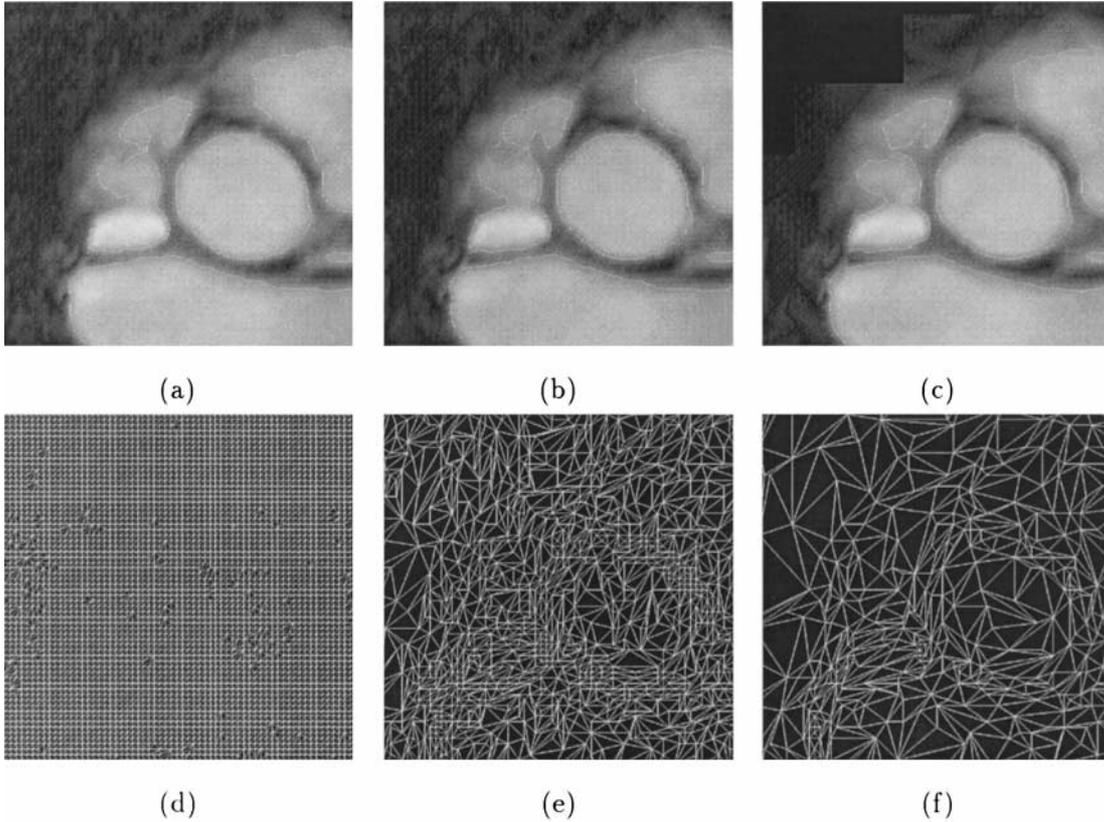


Fig. 14. (a) Close-up of input triangulation of MRI data of a heart, with contour in white; (b) a topology preserving simplified mesh; (c) simplification which violates the original topology; (d–f) the associated triangulations.

contraction. Figure 13 illustrates edge contract operations which preserve the original topology of the field as well as edge contraction operations which modify the original topology.

Topology preservation can be enforced by comparing the classification of the vertices in the local neighborhood before and after the edge contraction. If the collection of classifications change, criticalities have either been introduced or destroyed. Note that criticalities may move from one vertex to a neighboring vertex without modifying the field topology. If this is undesirable, one can enforce that only regular points be contracted to a neighbor. Note that topology preserving simplification of the scalar field in general also simplifies the scalar topology diagram (the piecewise linear arcs between the critical points).

4.1.5. Ordering of operations. Critical to the success of an error approximation approach is the order in which vertices are selected for removal. Vertex selection by priority queue approaches have permitted simplification which spans wide ranges of resolution [34].

We apply a priority queue driven approach which initially measures the error introduced by removal of each vertex. At each simplification step, the vertex which introduces the minimum error is extracted

and the simplification operation is applied. After each operation, neighboring vertices which are affected by the result are reprioritized based on the accumulated error which results from their removal.

4.1.6. Examples. Figure 14 illustrates the features of a topology preserving mesh simplification scheme. Using the same input triangulations, two simplification strategies are applied. The mesh in Fig. 14(b) is the result when topology preserving criteria are enforced. The mesh in Fig. 14(c) is the result when only local error bounds are used to guide the simplification.

4.2. Top-down simplification

Coarsening techniques are desirable for their simplicity. Another class of simplification works in the opposite order, adding detail to an initially coarse triangulation. In this section we consider the application of scalar field topology in guiding the insertion and triangulation. The scalar field topology diagram consists of a subset of the edges of a triangular mesh. Top-down simplification proceeds beginning with a base mesh consisting of the critical points in addition to edges connecting those critical points which are connected by an arc in the scalar topology diagram. Figure 15 illustrates the base

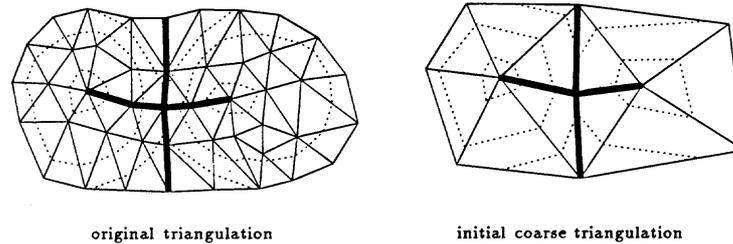


Fig. 15. Generating a base mesh which incorporates the scalar field topology.

mesh construction from an initial dense mesh based on the scalar topology diagram.

Having constructed a base mesh, iterative refinement operations can be applied, in order to reconstruct both the scalar field and the scalar topology diagram to a desired level of detail.

5. CONCLUSIONS

Error-bounded data simplification techniques have progressed rapidly in recent years. We have described a definition of scalar field structure which makes it possible to preserve global features in addition to bounding local errors. The techniques described can be applied to functions defined over any 2D triangulation. The coarsification strategy can also accommodate functions defined over a 3D surface triangulation, preserving the topology of isocontours lying on the surface.

In future work we will extend the techniques described to preserve isosurface topology in 3D volume triangulations. Refinement techniques for 3D surface triangulations will require a method for combining the initial coarse mesh based on the scalar field topology with metrics for constructing a coarse mesh based on geometry alone.

Acknowledgements—This research was supported in part by the Canadian Spinal Research Organization CSRO-96-3409670 and the Indiana Center for Advanced Research ICFAR-97-6712868.

REFERENCES

1. Bader, R., *Atoms in Molecules*. Clarendon Press, Oxford, 1990.
2. Bajaj, C. L. and Schikore, D. R., Decimation of 2D scalar data with error control. *Technical Report CSD-TR-95-005*. Department of Computer Sciences, Purdue University, West Lafayette, Indiana, 1995.
3. Bajaj, C. L. and Schikore, D. R., Error-bounded reduction of triangle meshes with multivariate data. in *Proceedings of SPIE Symposium on Visual Data Exploration and Analysis III*. SPIE, 1996.
4. Bajaj, C. L. and Schikore, D. R., Visualization of scalar topology for structural enhancement. *Technical Report CSD-TR-96-006*. Department of Computer Sciences, Purdue University, West Lafayette, IN, 1996.
5. Boyce, W. E. and DiPrima, R. C., *Elementary Differential Equations and Boundary Value Problems*, 5th edn. John Wiley and Sons, Inc., New York, 1992.
6. Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P., Brooks, F. P., Jr and Wright, W., Simplification envelopes. In *SIGGRAPH 96 Conference Proceedings*, ACM SIGGRAPH, Addison Wesley, Held in New Orleans, LA, August 4-9, 1996 edn, ed. Holly Rushmeier. Annual Conference Series, 1996, pp. 119-128.
7. De Floriani, L., Falcidieno, B. and Pienovi, C., Delaunay-based representation of surfaces defined over arbitrarily shaped domains. *Computer Vision, Graphics and Image Processing*, 1985, **32**, 127-140.
8. DeVore, R., Jawerth, B. and Lucier, B. J., Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*, 1992, **38**, 719-746.
9. DeVore, R. A., Jawerth, B. and Lucier, B. J., Surface compression. *Computer Aided Geometric Design*, 1992, **9**, 219-239.
10. DeVore, R. A. and Lucier, B. J., Wavelets., In *Acta Numerica 92*, ed. A. Iserles. Cambridge University Press, 1992, pp. 1-56.
11. De Floriani, L., Falcidieno, B., Nagy, G. and Pienovi, C., A hierarchical structure for surface approximation. *Computers and Graphics*, 1984, **8**(2), 183-193.
12. Fowler, R. J. and Little, J. J., Automatic extraction of irregular network digital terrain models. In *Computer Graphics (SIGGRAPH '79 Proceedings)*, 1979, **13**, 199-207.
13. Globus, A., Levit, C. and Lasinski, T., A tool for visualizing the topology of three-dimensional vector fields. In *Visualization '91 Proceedings*, ed. G. M. Nielson and L. Rosenblum, 1991, pp. 33-40.
14. Guézic, A., Surface simplification inside a tolerance volume. *Research Report RC-20440*. IBM Research Division, Yorktown Heights, New York, 1996.
15. Hamann, B., A data reduction scheme for triangulated surfaces. In *Computer Aided Geometric Design*, 1994, **13**, 197-214.
16. He, T., Hong, L., Kaufman, A., Varshney, A. and Wang, S., Voxel based object simplification. In *Proceedings of IEEE Visualization '95*, ed. G. M. Nielson and D. Silver, 1995, pp. 296-303.
17. Helman, J. and Hesselink, L., Automated analysis of fluid flow topology. *3D Visualization and Display Technologies (Proc. SPIE)*, 1989, **1083**, 825-835.
18. Helman, J. and Hesselink, L., Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 1991, **11**(3), 36-46.
19. Hinker, P. and Hansen, C., Geometric optimization. In *Proceedings of IEEE Visualization '93 (San Jose, California, October 25-29, 1993)*, ISBN 0-8186-3940-7, ed. Gregory M. Nielson and Daniel Bergeron. IEEE Computer Society, IEEE Computer Society Press, 1993, pp. 189-195.
20. Hoppe, H., Progressive meshes. In *SIGGRAPH 96 Conference Proceedings*, ACM SIGGRAPH, Addison Wesley, Held in New Orleans, LA, 4-9 August 1996 edn, ed. Holly Rushmeier. Annual Conference Series, 1996, pp. 99-108.

21. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., Mesh optimization. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, Vol. 27, ed. James T. Kajiya, 1994, pp. 19–26.
22. Interrante, V., Fuchs, H. and Pizer, S., Enhancing transparent skin surfaces with ridge and valley lines. In *Visualization '95 Proceedings*, ed. G. M. Nielson and D. Silver, 1995, pp. 52–59.
23. Itoh, T. and Koyamada, K., Automatic isosurface propagation using an extrema graph and sorted boundary cell lists. *IEEE Transactions on Visualization and Computer Graphics*, 1995, **1**(4), 319–327.
24. Kalvin, A. D. and Taylor, R. H., Surfaces: Polygonal mesh simplification with bounded error. *IEEE Computer Graphics and Applications*, 1996, **16**(3), .
25. Lorensen, W. E. and Cline, H. E., Marching cubes: A high resolution 3D surface construction algorithm. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, Vol. 21, ed. Maureen C. Stone, 1987, pp. 163–169.
26. Lucier, B. J., Wavelets and image compression. In *Mathematical Methods in Computer Aided Geometric Design II*, ed. T. Lyche and L. Schumaker. Academic Press, 1992, pp. 391–400.
27. Mallat, S. G., A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989, **11**(7), 674–693.
28. Marsden, J. and Tromba, A., *Vector Calculus*, 3rd edn. W. H. Freeman and Company, New York, 1988.
29. McCormack, J. E., Gahegan, M. N., Roberts, S. A., Hogg, J. and Hoyle, B. S., Feature-based derivation of drainage networks. *Int. Journal of Geographical Information Systems*, 1993, **7**(3), 263–279.
30. Morse, M. and Cairns, S. S., *Critical Point Theory in Global Analysis and Differential Topology*, Academic Press, New York, NY, 1969.
31. Muraki, S., Approximation and rendering of volume data using wavelet transforms. In *Proceedings of IEEE Visualization '92 (Boston, Massachusetts, October 19–23, 1992)*, ed. Arie A. Kaufman and Gregory M. Nielson. IEEE Computer Society, IEEE Computer Society Press, 1992, pp. 21–28.
32. Press, W., Teukolsky, S., Vetterling, W. and Flannery, B., *Numerical Recipes in C*, 2nd edn. Cambridge University Press, 1992.
33. Puppo, E., Davis, L., DeMenthon, D. and Teng, Y. A., Parallel terrain triangulation. *Int. Journal of Geographical Information Systems*, 1994, **8**(2), 105–128.
34. Ronfard, R. and Rossignac, J., Full-range approximation of triangulated polyhedra. *Computer Graphics Forum*. 1996, **15**(3) (Proceedings of Eurographics '96).
35. Rossignac, J. and Borrel, P., Multiresolution 3D approximations for rendering complex scenes. In *Modeling in Computer Graphics*, ed. B. Falcidieno and T. L. Kunii, 1993, pp. 455–465.
36. Schroeder, W. J., Zarge, J. A. and Lorensen, W. E. Decimation of triangle meshes. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 1992, **26**, 65–70.
37. Silva, C., Mitchell, J. S. B. and Kaufman, A. E., Automatic generation of triangular irregular networks using greedy cuts. In *Proceedings of IEEE Visualization '95*, ed. G. M. Nielson and D. Silver, 1995, pp. 201–208.
38. Stollnitz, E. J., DeRose, T. D. and Salesin, D. H., *Wavelets for Computer Graphics: A Primer*, University of Washington, 1994.
39. Tsai, V. J. D., Delaunay triangulations in tin creation: an overview and a linear-time algorithm. *International Journal of Geographical Information Systems*, 1993, **7**(6), 501–524.
40. Turk, G., Re-tiling polygonal surfaces. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 1992, **26**, 55–64.
41. Ware, J. M. and Jones, C. B., A multiresolution topographic surface database. *International Journal of Geographical Information Systems*, 1992, **6**(6), 479–496.