Ridge Extraction from Isosurfaces of Volumetric Data using Implicit B-Splines

Suraj Musuvathy, Tobias Martin, Elaine Cohen
School of Computing, University of Utah

Abstract—Ridges are extremal curves of principal curvatures on a surface that indicate salient intrinsic features of its shape. This paper presents a novel approach for extracting ridges of improved quality from isosurfaces of volumetric scalar-valued grids by converting them to implicit trivariate B-spline representations. A robust tracing approach demonstrated to extract ridges accurately from parametric B-spline surfaces is extended to extract ridges directly from the implicit representations resulting in accurate and hence smoother, connected ridge curves as compared to approaches that extract ridges directly from discrete representations. This approach can also be used to extract ridges directly from smooth representations such as isosurfaces of volumetric B-Spline CAD models and algebraic functions, and extended to extract ridges from polygonal meshes, as demonstrated in the paper. Most of the existing approaches for ridge extraction address only crests, a certain subset of the ridges on a surface. The approach presented in this paper enables extraction of all types of generic ridges on a surface thereby presenting a complete solution.

Keywords—ridge; isosurface; volumetric grid; polygonal mesh; implicit; B-Spline

1. INTRODUCTION

Ridges on a surface are shape intrinsic feature curves that describe higher order differential properties of surface geometry. Ridges are defined as loci of points where one of the principal curvatures ($\kappa_1 \geq \kappa_2$) attains a local extremum along its principal direction ($t_1$ or $t_2$)[20, 43].

$$\phi_i = \langle \nabla \kappa_i, t_i \rangle = 0, \quad i = 1 \text{ or } 2$$

(1)

This paper refers to $\phi_i$ as the ridge function and to Equation (1) as the ridge equation. While other definitions of ridge-like structures in the existing literature include that of height ridges [15] and watershed ridges [36], we address the extraction of principal curvature ridges in this work. Crests, valleys, ravines are other frequently used terms in the existing literature to describe extremal curves of curvature. Ravines and valleys are ridges of $\kappa_2$. Crests are perceptually salient ridges where $\kappa_1$ attains a local maximum along its principal direction and $|\kappa_1| > |\kappa_2|$ or $\kappa_2$ attains a local minimum along its principal direction and $|\kappa_2| > |\kappa_1|$ [8], [20].

$$t_1^T H_{\kappa_i} t_1 < 0, \quad |\kappa_1| > |\kappa_2|; \quad \kappa_1\text{-crest}$$

$$t_2^T H_{\kappa_2} t_2 > 0, \quad |\kappa_2| > |\kappa_1|; \quad \kappa_2\text{-crest}$$

$$H_{\kappa_i}(u, v) = \begin{bmatrix} \kappa_{iuv} & \kappa_{iuv} \\ \kappa_{iuv} & \kappa_{iuv} \end{bmatrix}, \quad i = 1 \text{ or } 2$$

(2)

Generically, ridges form continuous closed curves and do not end abruptly. Ridges of a principal curvature do not cross each other except at umbilics where three such ridges can meet. At an umbilic point, either one or three ridges of $\kappa_1$ meet a corresponding number of $\kappa_2$ ridges. Crests do not contain umbilic points.

1.1 Contributions

The main contribution of this paper is a novel approach for extracting all types (crest and non-crest) of ridges with improved quality from isosurfaces of 3D volumetric grid data. The proposed approach uses trivariate implicit B-Spline representations of isosurfaces, created by filtering the grid data using 3D B-Spline filters of appropriate degree, in conjunction with a generalization of a previously presented approach for robustly tracing ridges on parametric surfaces [39], to extract ridges directly from the smooth representation.

\[t_1\] and $t_2$ are 2D vectors representing the coefficients of the basis vectors of the tangent plane of the surface.

\[\text{See Appendix A for explanation of parameterization invariance of Equation 2.}\]
3D data grids are abundantly available in the form of medical images (MRI, CT scans), simulation data where the resulting grids approximate solutions of discretized partial differential equations and in the field of graphics and visualization in the form of level set models [6]. The proposed approach can also be used to extract ridges directly from smooth implicit surface representations such as those occurring from isogeometric analyses [23] and implicit modeling systems [44]. Direct processing of volumetric grid data avoids potentially difficult problems associated with extracting isosurfaces in the form of discrete representations such as polygonal meshes. Trivariate B-Splines have been previously used for rendering [45] and filtering [34] of volumetric data and implicit modeling [44]. Our proposed method complements this approach with a tool for extracting important shape features directly from volumetric grids and smooth implicit surface representations. In this paper, the proposed approach is demonstrated to extract ridges directly from medical image data, discrete simulation data, isosurfaces of volumetric B-Splines occurring from isogeometric analysis and algebraic functions. Further, the method is extended to compute ridges from discrete representations such as polygonal meshes by converging them to trivariate B-Spline representations.

The consequence of tracing on a smooth representation is that the extracted ridge curves conform to generic behavior and are therefore continuous connected curves. B-Splines act as low pass filters on the data grid and tend to smooth out high frequency characteristics such as noise and thus the ridges do not have unexpected undulations. In addition, a smooth representation allows robust detection of isolated umbilics, and thus, ridges around umbilics, to present a complete solution. Figure 1 shows ridges and crests extracted using our approach from an isosurface of a 3D grid resulting from a simulation. This isosurface has cylindrical regions with near circular cross-sections that are non-generic. So ridges can terminate abruptly as they get closer to such regions. Other types of ridges including elliptic and hyperbolic ridges are distinguished based on [39] as demonstrated in Figure 6 (c).

1.2 Applications of Crest and Non-crest Type Ridges

Ridges have been used as landmarks for shape matching and registration [19], [26], [42], [49], as indicators of the quality of product designs [25], [22], as visual cues for effective visualization [12], [24], [31] and several other shape analysis tasks [29], [50]. Most of the previous approaches use crests mainly because there are few methods for robust extraction of all types of ridges. Non-crest ridges are more sensitive to subtle variations in geometry than crests and along with their topology, indicate higher order local geometric variation. These local curvature variations may not be desirable for smooth product designs. Therefore non-crest ridges are useful for evaluating product designs where undesirable curvature variations of a freeform surface are detected. Since these are higher order surface properties, they may not be immediately perceptible even on high quality renderings of the objects. Non-crest ridges are useful for statistical shape analysis tasks over a group of similar objects such as anatomical organs. Since crests are more stable, they may occur at very similar locations and may seem to have similar structure across the group of objects. In this case, the sensitivity of non-crest ridges to local geometric variation will reveal additional geometric differences. In addition, computing the full set of ridges helps in understanding the relationship between crests and the topological structure of ridges. Non-crest ridges may connect two seemingly separate crest segments. This information is useful for shape analysis tasks. Umbilics represent important surface features and have been used for shape fingerprinting [28]. Since non-crest ridges exhibit topological changes at umbilics, it is also essential to compute ridges around umbilics accurately.

1.3 Issues with Extracting Ridges Directly from Discrete Representations

Earlier approaches for extracting ridges from discrete data representations, including isosurfaces of volumetric data and polygonal meshes, tend to result in sets of disconnected ridge segments or tend to have undesirable undulations. There are several factors that potentially contribute to this problem. First, most of the earlier techniques address only crests. However, crest curves on a smooth surface can turn into non-crest ridges that may in turn change back to crests. Extracting only crests therefore results in a disconnected subset of ridges. Second, curvatures and derivatives are typically not available with discrete data and are hence estimated. Being functions of the second and third order surface derivatives respectively, this process is very sensitive to noise. Consequently, the quality of ridges extracted depends on the quality and consistency with which these quantities are estimated. In addition, fragmentation of the ridges can occur due to inconsistent choices of principal direction vector orientations [21] and when ridges are near parallel to mesh edges [53]. The aim of the method presented in this paper is to overcome the aforementioned problems with the discrete techniques and to extract all types of ridges on discrete surfaces for which generic conditions hold.

2. PREVIOUS WORK

We first review previous work on extracting ridges from discrete representations and then present existing techniques that address smooth representations. For discrete representations, estimating curvatures and their derivatives is a significant challenge. Smooth functions are typically used to estimate these quantities on the vertices of the tessellation. Approximate ridge points are identified on the edges and faces of the tessellation by linear interpolation of the ridge function estimates at the
corresponding vertices. The main improvement of the proposed approach over existing methods is that ridges are extracted directly from the smooth representation resulting in a robust, accurate and complete solution.

**Volumetric data.** The Marching Lines algorithm [52] presented a discrete technique to compute ridge curves on level sets of volumetric scalar fields such as medical images (MRI, CT scans). The technique computes intersection curves of an isosurface and the ridge function \( \phi_i \) on the voxels of the data set. The Gaussian extremality, which is the product of the ridge functions \( \phi_1 \) and \( \phi_2 \), was introduced in [51] and used to extract ridges from 3D images. The Gaussian extremality overcomes the problem of finding consistent principal direction orientations for evaluating the ridge functions. However, \( \kappa_1 \) and \( \kappa_2 \) ridges cannot be distinguished when computed as zeros of the Gaussian extremality. This causes additional errors in determining the topology of ridges around regions where a \( \kappa_1 \) ridge intersects a \( \kappa_2 \) ridge as noted in [10]. An image filtering approach is presented in [37] to first identify points on an isosurface and classify them as ridge points if they satisfied the ridge equation. Curvatures and their derivatives are estimated at required points using image filters in both techniques. In the work of [19], parametric B-splines are fit to isosurfaces and a discrete sampling technique is used to determine ridge curves on the isosurfaces.

**Implicits.** A discrete method for computing intersections of an implicit function defining the surface and the ridge function is presented in [3]. An analytic solution for computing solutions of a system of equations describing ridges of a polynomial implicit function using a singularity theory approach is presented in [4]. The authors suggest using the implicit representation for discrete data but no results are presented. In our work, we adopt this idea but use a different approach to extract ridges using a different representation, piecewise polynomial implicit B-Splines, that enables a global representation of large complicated discrete data sets.

**Polygonal Meshes.** Curvatures and their derivatives are estimated at mesh vertices by fitting smooth surfaces locally or over the entire mesh such as compactly supported radial basis functions [40], polynomials [53], [10], MLS based implicit functions [27] or using discrete methods [21], [53]. Ridges are traced by detecting zero crossings of the ridge function on the vertices and edges of the meshes. Umbilics and ridges around them are detected in the method presented in [10]. All other approaches address only crests. Smoothing of the ridge function (as opposed to the surface) as well as smoothing crest space curves themselves was proposed in [21] to obtain crests with fewer undulations. Local angle measures between end points of crest segments have been used for connecting disjoint segments[53].

**Parametric Surfaces.** For smooth surface representations, curvatures and their derivatives can be computed exactly at any point. Ridges form continuous curves on smooth surfaces and extracting them accurately is more difficult. Three approaches have been presented to extract ridges curves from smooth parametric surfaces. A lattice method for single patch polynomial (Bézier) surfaces is presented [9], [7] where solutions of the ridge equations are computed on a dense grid of isoparametric lines. Ridge curves are obtained by connecting ridge points on the grid. Sampling based methods are presented in [22], [25], [19], [26], [38] where the ridge function is evaluated on various tessellations of the parametric domain and ridges are identified on the tessellation. A robust tracing method computationally suitable for complicated B-spline surfaces is presented in [39]. In this paper, the tracing approach of [39] is extended for extracting ridges from level sets of implicit trivariate B-splines.

### 3. Implicit B-Spline Representation of Isosurfaces of Volumetric Data

Given a parallelepiped region \( \Omega \subset \mathbb{R}^3 \), where \( \Omega = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3] \), let \( f: \Omega \rightarrow \mathbb{R} \) be a \( C^4 \) trivariate function that maps a point \((x_1, x_2, x_3)\) in \( \Omega \) to a scalar value. Given a specific isovalue \( \hat{a} \in \mathbb{R} \),

\[
I = \{(x_1, x_2, x_3) : f(x_1, x_2, x_3) = \hat{a} \}
\]

forms an implicit surface also called an isosurface or level set at isovalue \( \hat{a} \). In this paper, it is assumed that \( \nabla f \neq [0 \ 0 \ 0]^{T} \) \( \forall (x_1, x_2, x_3) \in I \) in which case the isosurface is guaranteed to be a 2-manifold [5].

The implicit function theorem states that around every point on the isosurface there exists a neighborhood in which the isosurface can be represented as a Monge surface using at least one of \((x_1, x_2), (x_2, x_3), (x_3, x_1)\) as parameter variables. For example, when \( \frac{\partial f}{\partial x_3} \neq 0 \), there exists a scalar field \( g(x_1, x_2) \) such that the isosurface can be represented as \( S(x_1, x_2) = (x_1, x_2, g(x_1, x_2)) \) where \( f(x_1, x_2, g(x_1, x_2)) = \hat{a} \). The first, second and third order partials of \( S(x_1, x_2) \) are computed using this framework which are in turn required to evaluate principal curvatures, principal directions and curvature gradients at any point \((x_1, x_2, x_3) \in I \) as given in Appendix B. The reader is referred to [47], [52] for derivation of the formulæ for computing principal curvatures, principal directions and curvature gradients defined on the implicit surface.

In this paper, \( f(x_1, x_2, x_3) \) is a trivariate B-spline defined as

\[
f(x_1, x_2, x_3) := \sum_{i=1}^{n} c_i B_{1,d,r}(x_1, x_2, x_3),
\]

where \( c_i \in \mathbb{R} \) are the coefficients of a \( n_1 \times n_2 \times n_3 \) control grid and \( i = (i_1, i_2, i_3) \) and \( n = (n_1, n_2, n_3) \) are multi-indices. Every coefficient has an associated piecewise polynomial basis function

\[
B_{1,d,r}(x_1, x_2, x_3) := \prod_{j=1}^{3} B_{i_j, d_j, r_j}(x_j),
\]
where \( B_{i_1,i_2,\tau_j}(x_j), j = 1, 2, 3 \) are linearly independent B-Spline basis functions. \( B_{i_1,i_2,\tau_j}(x_j) \) as defined in [11] is a piecewise polynomial of degree \( d_j \) with knot vector \( \tau_j = \{ t_k \}_{k=1}^{n_j+1} \) that has local support and is \( C^{d_j-1} \). In order for the ridge functions \( \phi_1 \) and \( \phi_2 \) to be continuous, third order derivative smoothness is required \( (d_j = 4) \). To distinguish crests from other types of ridges, fourth order derivative smoothness \( (d_j = 5) \) is required to compute second derivatives of curvatures. \( \tau_j \) is a uniform and open knot vector, i.e. the first five and last five knots of \( \tau_j \) are \( a_j \) and \( b_j \) respectively.

For instance, the result of a CT scan is a uniform grid of densities. If these densities are used as coefficients \( c_i \) in Equation (5), the corresponding B-spline basis can be viewed as a smoothing low pass reconstruction filter [34] of the samples \( c_i \) that does not introduce additional geometric features on the isosurface.

Implicit representations of polygonal meshes are created by computing Euclidean distance fields of surrounding regions. Given an unstructured polygonal mesh boundary \( T \), \( c_1 = \pm |p_i - p_0|/2 \), where \( p_0 \in \Omega \) corresponds to \( c_1 \) [11] and \( p_i \) is a point on \( T \) that is closest to \( p_0 \), where the sign of \( c_1 \) depends whether \( p_0 \) lies within \( T \) or not. The implicit surface at distance 0, i.e. \( f(x_1, x_2, x_3) = 0 \) approximates \( T \). Note that if the input is a point cloud, a corresponding distance field can be computed.

In addition, given any algebraic function \( a(x, y, z) \), there exists a set of coefficients \( c_i \) such that

\[
a(x, y, z) \equiv f(x, y, z) = \sum_{i=1}^{n} c_i B_{i,4,\tau}(x, y, z), \tag{6}
\]

defined over the parallelepiped \( \Omega \in \mathbb{R}^3 \), where the B-spline basis matches the highest degree of \( a(x, y, z) \). The set of coefficients \( c_i \) can be derived by a multivariate version of Marsden’s identity [35].

4. TRACING RIDGES ON ISOSURFACES OF IMPLICIT B-SPLINES

This section first reviews the approach of [39] for tracing ridges on parametric B-spline surfaces and then presents extensions to the algorithm for tracing on level set isosurfaces of implicit trivariate B-splines.

4.1 Parametric Surfaces

**Seed points.** Let \( S(u, v) : R^2 \rightarrow R^3 \) be a \( C^3 \) smooth parametric surface. Seed points for tracing, including principal curve extrema and umbilics, are first computed using efficient subdivision based B-spline constraint solving techniques [17], [18]. Extrema of principal curvature trivially satisfy the ridge equation since the curvature gradient vanishes at such points. As noted in Section 1, ridges of a particular curvature can meet only at umbilics. The complexity of resolving the topology of ridges is significantly reduced when umbilics are used as seed points. Extremal points of \( \kappa_i \) are computed using Equation (7).

\[
\frac{\partial \kappa_i(u, v)}{\partial u} = 0; \quad \frac{\partial \kappa_i(u, v)}{\partial v} = 0 \tag{7}
\]

Umbilics are computed by obtaining solutions of Equation (8).

\[
\frac{\partial Q(u, v)}{\partial u} = 0; \quad \frac{\partial Q(u, v)}{\partial v} = 0 \tag{8}
\]

where

\[
Q(u, v) = B^2 - 4AC \\
A(u, v) = EG - F^2 \\
B(u, v) = 2FM - GL - EN \\
C(u, v) = LN - M^2
\]

and \( E, F, G \) are coefficients of the first fundamental form of \( S(u, v) \), and \( L, M, N \) are coefficients of the second fundamental form of \( S(u, v) \) [14]. Note that the left hand sides (LHS) of Equation (7) are not rational functions. They are converted to rational functions by squaring and rearranging terms. The coefficients of the LHS of the equations are computed using symbolic B-spline techniques. See [32], [41] for further details.

![Fig. 2. Tracing ridges by advancing and sliding along principal directions](image)

**Tracing.** The ridges of \( \kappa_1 \) and \( \kappa_2 \) are traced independently. Since principal directions \( T_i = t_i^1 S_u + t_i^2 S_v \in R^3, i = 1, 2 \) at a non-umbilic point of the surface are orthogonal, they are used as local coordinate systems for tracing. Each trace step consists of two phases, an advance phase followed by a slide phase, as illustrated in Fig. 2. From a point on a ridge the advance phase for a \( \kappa_1 \) ridge steps along the \( T_2 \) direction and the slide phase steps along the \( T_1 \) direction until a new ridge point is reached. Principal directions and curvature gradients are recomputed at each step. Since a \( \kappa_1 \) ridge intersects the integral curvature lines of \( T_1 \) transversely except at turning points, the trace is guaranteed to progress along the ridge. For a ridge of \( \kappa_2 \), the advance is performed along the \( T_1 \) direction and the slide is performed along the \( T_2 \) direction. At umbilics, limit principal directions for tracing are computed using the approach presented in [33], [41]. Details of selecting robust step sizes, selecting consistent principal direction orientations and dealing with other issues at umbilics and turning points
are presented in [39]. Since tracing is performed in $\mathbb{R}^3$, points at each step are projected back onto the surface using a two dimensional Newton’s algorithm.

Crests are identified by evaluating the second order derivatives of the curvatures and testing the crest conditions given in Equation (2) at each point of all extracted ridges.

4.2 Extension to Implicit Trivariates

The tracing algorithm for implicit trivariates follows the same framework of advancing and sliding from seed points. This section presents techniques for addressing new challenges that arise with computing seed points and tracing with the implicit trivariate representation. Curvatures, principal directions and curvature gradients required for evaluating the ridge function are computed using a local parameterization of the isosurface given by the implicit function theorem as presented in Section 3.

**Seed points.** Extremal points of $\kappa_1$ and umbilics are computed as simultaneous roots of three equations in three unknowns as given in Equations (10) and (11) respectively,

$$f(x_1, x_2, x_3) \equiv \hat{a} \quad \frac{\partial f}{\partial x_1} = 0 \quad \frac{\partial f}{\partial x_2} = 0$$

$$f(x_1, x_2, x_3) \equiv \hat{a} \quad \frac{\partial f}{\partial x_1} = 0 \quad \frac{\partial f}{\partial x_2} = 0$$

wherein the isosurface is locally parameterized using $(x_1, x_2)$ and coefficients of the LHS are computed symbolically. When using the $(x_1, x_2)$ parameterization, it is assumed that $\frac{\partial f}{\partial x_3} \neq 0$ so that the implicit function theorem is valid. However it is possible that $\frac{\partial f}{\partial x_3} = 0$ within $\Omega$. Therefore, similar equations are derived for $(x_2, x_3)$ and $(x_3, x_1)$ parameterizations and seed points are computed using these parameterizations as well. In practice, obtaining roots of these equations is computationally very demanding even for reasonably sized trivariate B-splines. We present techniques to reduce computation time in Section 5.

**Tracing.** The following additional issues are addressed for tracing:

1) At each step of the trace, it is imperative that the orientation of the normal of the isosurface is globally consistent since the convention of which curvature, $\kappa_1$ or $\kappa_2$, is the larger one is dependent on it. Since $\nabla f$ is the normal of the global implicit representation at any point and is oriented consistently, the normal computed using the local parametrization of the isosurface $(S_{x_1} \times S_{x_2})$ is compared with it. If the directions of the two vectors are opposite, then $\kappa_1$, $t_1$, $\nabla \kappa_1$ and $\kappa_2$, $t_2$, $\nabla \kappa_2$ are swapped and the signs of $\kappa_i$, $\nabla \kappa_i$, $i = 1, 2$ are changed.

2) It is assumed that $\nabla f \neq [0 \ 0 \ 0]$. However, it is possible that up to two of the quantities $f_{x_1}$, $f_{x_2}$, $f_{x_3}$ are zero at a point. The algorithm selects an appropriate parameterization for evaluating the ridge function at every advance and slide step of the trace. This enables tracing of ridges passing through such points and even lying exactly on such points. All the ridges of the ellipsoid shown in Figure 6 lie along curves where either $f_{x_1}$, $f_{x_2}$ or $f_{x_3}$ are zero since $f_{x_i} = 0$ at $x_i = 0$, $i = 1, 2, 3$.

3) At every step, the trace moves off the isosurface (along one of the principal directions) and is projected back onto the isosurface using a standard technique of iteratively marching along $\nabla f$ until the isosurface is reached.

5. OPTIMIZATION OF SEED POINT COMPUTATION

Robust and efficient subdivision based multivariate B-Spline constraint solving techniques have been presented in [17], [18]. These techniques bound the range of values a function can take, typically using axis aligned bounding boxes (AABBs) of appropriate dimension, recursively subdividing if necessary until a user specified resolution is reached. Then a numerical technique, such as a multivariate Newton’s method, is used to converge to more accurate solutions. The subdivision phase requires symbolically computing the coefficients of the terms in the equations from the coefficients of $f$. Computing principal curvature extrema and umbilics using Equations (10) and (11) is computationally demanding since the equations have a high degree and therefore symbolic computation of the coefficients of the terms of the equations is very expensive in terms of both time and memory. We have developed the following optimizations to reduce compute time.

First, subregions of the trivariate that potentially contain the isosurface are extracted. The domain of the trivariate in Section 3 is $[t_{d_1+1}^1, t_{n_1-1}^1] \times [t_{d_2+1}^2, t_{n_2-1}^2] \times [t_{d_3+1}^3, t_{n_3-1}^3]$. Every knot span subdomain $[k_{b_1}^1, k_{b_1+1}^1] \times [k_{b_2}^2, k_{b_2+1}^2] \times [k_{b_3}^3, k_{b_3+1}^3]$ is extracted as a Bézier trivariate and retained if the range of $f(x_1, x_2, x_3)$ within the subdomain contains the isovalue of interest. The convex hull property of the trivariate Bézier representation enables an efficient test of checking the 1D AABB of the coefficients of $f(x_1, x_2, x_3)$ of the region representing the subdomain for this purpose. Since the subdomains are typically very small for a reasonably high resolution data set, a large number of subdomains are rejected in this step. Table 1 compares the percentage of subdomains retained for seed point computation for the various models used in this paper.

Second, computing coefficients of the LHS of the equations symbolically is still computationally expensive even though they are in Bézér form. An ex-
pression tree approach was presented in [17] to reduce computational demands of multivariate B-Spline constraint solvers especially when the different terms in an equation are functions of different independent variables. In our work, the high degree terms in the equations for computing seed points are functions of the same independent variables $x_1$, $x_2$ and $x_3$. We have developed a variant of the expression tree approach to address this situation. The equations are represented as expression trees as in [17] and coefficients are computed only for $f(x_1, x_2, x_3)$ and its partial derivatives up to third order, which are low degree terms. It should be noted that a term involving a partial derivative of $f$ appears multiple times in the LHS of the equations. In the expression tree approach of [17], a copy of this term is stored in every repeated leaf node. This can lead to redundant subdivisions of each copy. In our method only one global copy of each term is stored and only the global copies of the terms are subdivided during the subdivision step. This approach is similar in spirit to the idea presented for efficient data structure management using reference counted pointers but were proposed for cases when the leaf nodes of an expression tree are functions of different independent variables. The AABBs for a subdomain are computed using interval arithmetic as presented in [17]. In our experiments, we have found that this reduces computation time by over an order of magnitude. Constraint solving by computing the coefficients for a Bézier trivariate subdomain took about four minutes on average per parameterization. The optimized approach presented here required a few seconds.

In addition, as presented in Section 4, in some cases, seed points may have to be computed for all three possible parameterizations of the isosurface. However, in our experiments we noticed that most of the time a single parameterization is sufficient for a given subdomain. In order to determine an appropriate parameterization, we first select one such parameterization $x_i, x_{i[1]}$. If the range of $\frac{\partial f}{\partial x_{i[2]}}$ potentially does not have a zero within the subdomain, then this parameterization is the only one used for computing seed points. If $\frac{\partial f}{\partial x_{i[2]}}$ does potentially have a zero within the subdomain, then one of the other two candidate parameterizations are similarly tested. It is possible that $\frac{\partial f}{\partial x_{i[1]}}, \frac{\partial f}{\partial x_{i[2]}}$ and $\frac{\partial f}{\partial x_{i[3]}}$ all potentially have a zero (but not at the same point) in which case all three parameterizations are used.

Further, the constraint solver for different subdomains are executed in parallel since they are independent. These optimizations significantly reduce time and thus enable seed point computation for large data sets.

### Table 1

<table>
<thead>
<tr>
<th>Model</th>
<th>Source</th>
<th>Size of trivariate control grid</th>
<th>Subdomains with isosurface</th>
<th>% of subdomains with isosurface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skull</td>
<td>CT scan</td>
<td>128 x 128 x 128</td>
<td>73447</td>
<td>3.5</td>
</tr>
<tr>
<td>Silicium</td>
<td>Simulation data</td>
<td>34 x 34 x 96</td>
<td>15231</td>
<td>13.4</td>
</tr>
<tr>
<td>Pensatore</td>
<td>Distance field of mesh</td>
<td>57 x 64 x 62</td>
<td>99995</td>
<td>4.4</td>
</tr>
<tr>
<td>Buddha</td>
<td>Distance field of mesh</td>
<td>128 x 52 x 52</td>
<td>15802</td>
<td>4.5</td>
</tr>
</tbody>
</table>

6. Results and Discussion

We demonstrate the method presented in this paper on a CT scan, a 3D data grid arising from simulation results, on implicit B-Spline representations of isosurfaces resulting from isogeometric analyses on a volumetric B-Spline model, on algebraic surfaces, and polygonal meshes. Ridges of $\kappa_1$ are shown in blue and ridges of $\kappa_2$ are shown in red. Crests are shown as thicker curves. $|\phi|$ is used as a measure of the accuracy of the ridges extracted. A user specified accuracy (typically $10^{-2}$ or $10^{-3}$) is used as an input parameter for the tracing algorithm and all ridges extracted in generic regions using our method satisfy the accuracy requirement.

Our algorithm has been implemented in the Irit modeling environment [16]. The results presented here have been generated on an Intel Xeon X7350 processor with 32 cores and computation times are shown in Table 2. Since seed point computation for different subregions are independent processes, a speedup roughly equal to the number of threads is achieved using a parallel implementation. Similarly, ridge tracing from different start points are also independent processes and a significant speedup could be achieved using a parallel implementation. However, this has not been currently implemented.

For comparison, high resolution isosurfaces are extracted from the trivariate B-Spline representations using the Marching Cubes algorithm [30], and the method presented in [10] is used to compute results since this method extracts ridges as well as crests. We use the implementation of this algorithm available in CGAL [2].

Figure 3 shows ridges and crests extracted from a CT scan where the skull isosurface is identified at intensity value 69.5. This result can be compared to the results in Figure 15 of [52] and Figures 12, 13, 19 and 20 of [37] that show crests extracted from volumetric images of skulls. Figure 3 shows that our method captures a very high level of detail with smooth crest curves whereas previous grid based methods result in a sparse collection of fragmented crest segments. The crests on the top and side of the skull correspond to scanning artifacts of the data set and are accurately captured by our method.

We attempted creating a polygonal mesh representation of the isosurface using marching cubes but due to the geometric and topological complexity, the mesh failed to be suitable for use with the ridge extraction method in...
CGAL even after considerable manual effort to correct the errors. Afront [46] generated a suitable mesh for the algorithm in CGAL after several hours of computation but many of the features presented in the original data set were missing. The method presented in this paper avoids issues related to mesh generation for complex data sets and extracts ridges directly from B-Spline filtered smooth representations of the 3D grids.

Figure 4 compares ridges extracted on a 3D grid resulting from the Silicium simulation\(^3\) using the method presented in this paper (Figure 4.(a)) with the method of [10] on a high resolution isosurface mesh (335,000 triangles) extracted using marching cubes (Figure 4.(b)) and by isocontouring zeros of the Gaussian extremality on the isosurface mesh using ParaView [1] based on the method presented in [51](Figure 4.(c)). However, the isocontouring method has the drawback that the ridge type is unknown (See Section 2). As shown in the close up views of part of the data set, ridges extracted using the proposed method are smoother than the ridges extracted using the method of [10], which are fragmented, have undesirable undulations and do not capture many ridges. The ridges extracted using the isocontouring method also have large undulations. In addition, the topology of the ridges extracted using the isocontouring approach is incorrect in many areas where \(\kappa_1\) and \(\kappa_2\) ridges cross each other, as noted in [10].

There has been recent impetus in the area of isogeometric engineering analysis [23], where partial differential equations are solved directly on CAD representations of objects by avoiding any conversion into conventional finite element representations such as hexahedral meshes that only approximate the CAD models. Simulation results are obtained directly on the CAD representations and are therefore more reliable. In this paper we solve linear elasticity equations on a cube represented as a trivariate B-Spline to examine the vertical displacements resulting from loads applied at the top of the cube. The isosurface at a particular displacement value (that is now an implicit B-Spline) identifies all locations within the volume that have the same vertical displacement. Ridges and crests are extracted directly from the higher order trivariate implicit B-Spline representation of the isosurface to reveal additional structural information about the distribution

\(^3\)available at http://www.volvis.org

---

### TABLE 2

<table>
<thead>
<tr>
<th>Model</th>
<th>Seed points (32 threads) (minutes)</th>
<th>Tracing (1 thread) (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silicium</td>
<td>31</td>
<td>52</td>
</tr>
<tr>
<td>Skull</td>
<td>122</td>
<td>167</td>
</tr>
<tr>
<td>Cube (isogeometric)</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>Pensatore</td>
<td>15</td>
<td>34</td>
</tr>
<tr>
<td>Buddha</td>
<td>22</td>
<td>14</td>
</tr>
<tr>
<td>Camel</td>
<td>24</td>
<td>26</td>
</tr>
<tr>
<td>Moai</td>
<td>18</td>
<td>22</td>
</tr>
<tr>
<td>Screwdriver</td>
<td>10</td>
<td>17</td>
</tr>
</tbody>
</table>

---

Fig. 3. Ridges and crests extracted from an isosurface corresponding to a skull from a CT scan. \(\kappa_1\) ridges are in blue and \(\kappa_2\) ridges are in red. Crests are indicated by thicker curves.

---

\(\kappa_1\) and \(\kappa_2\) ridges cross each other, as noted in [10].
of this vertical displacement within the cube. Crests indicate areas where there is a sharp change in the stress-strain relationship within the cube and may provide better insight for engineering analysis. Figure 5 (a) and (b) show the results for the cube under slightly different vertical loads and the variation in geometry of the corresponding isosurfaces. While crests indicate major variations in the geometry of the two isosurfaces, the non-crest ridges also indicate higher order structural differences in the stress-strain relationships.

Figure 6 shows all ridges extracted on an ellipsoid, a tangle surface and a smooth dodecahedron represented as algebraic functions. Exact trivariate Bézier representations of the algebraic functions are determined using the multivariate version of Marsden’s identity [35]. The exact structure of ridges and umbilics on ellipsoids are well documented in the existing literature [43] and Figure 6.(a) validates that our approach accurately extracts all ridges and umbilics on the ellipsoid. One of the partial derivatives of the algebraic function $f(x_1, x_2, x_3)$ is zero along each ridge and two of the partial derivatives are zero at the six poles. Similar issues are present on the tangle function and the smooth dodecahedron as well. In addition, the surface normals for both examples computed using local parameterizations given by the implicit function theorem do not always agree with the function gradient direction. The results show that our method is robust to both situations. Figure 6. (c) shows the different types of ridges on a smooth dodecahedron including crests, non-crest elliptic ridges and hyperbolic ridges distinguished based on [39]. Figure 5 of Ohtake et al. [40] shows only the crests extracted from a polygonal
7. CONCLUSIONS, LIMITATIONS AND FUTURE WORK

Ridges and crests are important feature curves on surfaces and have several shape analysis applications. In this paper, a novel method for extracting ridges from isosurfaces of volumetric grid data using smooth trivariate implicit B-Spline representations in conjunction with a robust tracing method extended for trivariate implicit B-Splines is presented. Smooth 3D B-Spline filters are applied to volumetric scalar fields to represent isosurfaces as level sets of implicit B-Splines. Since ridges are extracted directly from smooth representations of grid data, the proposed method avoids potentially difficult problems associated with extraction of polygonal mesh representations of isosurfaces with complicated geometry and topology. The method is extended to polygonal meshes and point clouds by representing smooth approximations of their geometry as zero level sets of Euclidean distance fields.

By using smooth representations that also act as low pass filters on discrete data, our approach tends to generate smoother and more accurate ridge curves that conform to generic behavior. Smooth representations also enable robust and accurate computation of umbilics, and ridges around umbilics, thereby presenting a complete solution. Ridges computed directly from discrete data types and using discrete algorithms on smooth representations tend to result in fragmented segments with undesirable undulations. Post processing is typically performed to obtain smooth connected segments. Most of the earlier techniques for extracting ridges from discrete data representations do not address ridge computation at umbilics due to the complexity in identifying and discerning the topology of ridges around umbilics on discrete representations.

One of the major limitations of the approach presented in this paper is the computational demands of seed point computation. We have developed several optimizations to reduce computational cost. However, there is still scope for reducing computational demands and is a topic for further research. Optimized parallel implementations of the tracing algorithm is currently in progress. The
tracing algorithm is designed to extract ridges that conform to generic behavior. Hence, another area for future work is to address non-generic surface regions such as those with singularities, symmetries or non-isolated umbilic regions including flat areas, spherical areas or cylindrical regions with circular cross-section.

ACKNOWLEDGMENTS

This work was supported in part by NSF (CCF0541402). All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

REFERENCES


Fig. 9. Crests extracted from the level set of the distance field of the (a) Camel mesh; (b) Moai mesh; (c) Screwdriver mesh.


APPENDIX A

INVALENCE OF CURVATURE CONDITIONS

In the neighborhood of any point $p$ on a surface $S(u, v)$, define $\sigma_i(u, v) = (u, v, \kappa_i(u, v))$ $(1 \leq i \leq 2)$, where $\kappa_i$ is a principal curvatures of $S(u, v)$. Let $t_i = [t_{i1} \ t_{i2}]^T$ such that $T_i = t_{i1}\mathbf{S}_u + t_{i2}\mathbf{S}_v$ is the principal direction corresponding to $\kappa_i$. Then the second fundamental form of the surface $\sigma_i(u, v)$ in the $t_i$ direction is

$$
\hat{I}_p(t_i) = t_i^T \begin{pmatrix} \sigma_{iuu} & \sigma_{iuv} \\ \sigma_{iuu} & \sigma_{iuv} \end{pmatrix} t_i = t_i^T \begin{pmatrix} \kappa_{iuu} & \kappa_{iuu} \\ \kappa_{iuu} & \kappa_{iuu} \end{pmatrix} t_i
$$

(12)

Since the second fundamental form of a surface is invariant under coordinate transformations [14], the invariance of the crest conditions follows.

APPENDIX B

DIFFERENTIAL PROPERTIES OF ISOSURFACES

This section summarizes formulae for computing principal curvatures, principal directions and curvature gradients of an isosurface (See [47], [52] for details). Let $f(x_1, x_2, x_3) : R^3 \mapsto R$, $f \in C^4$. $\mathcal{I} = \{(x_1, x_2, x_3) : f(x_1, x_2, x_3) = \delta\}$ is an isosurface. Assuming $f_{x_3} \neq 0$, $\mathcal{I}$ can be locally represented as $S(x_1, x_2) = \{(x_1, x_2, g(x_1, x_2))\}$.

$$
n = S_{x_1} \times S_{x_2} = \left(\frac{f_{x_1}}{f_{x_3}}, \frac{f_{x_2}}{f_{x_3}}, 1\right)
$$

(13)

We assume that the surface normal $n(x_1, x_2)$ of $S(x_1, x_2)$ is oriented toward the interior of the region bounded by $S(x_1, x_2)$.

The coefficients of the first fundamental form are given by

$$
E = 1 + \frac{f_{x_1}^2}{f_{x_3}^2}; F = \frac{f_{x_1} f_{x_2}}{f_{x_3}}; G = 1 + \frac{f_{x_2}^2}{f_{x_3}^2}
$$

(14)

The coefficients of the second fundamental form are given by

$$
L = \frac{(f_{x_2} f_{x_3} f_{x_3, x_3} - f_{x_1} f_{x_1, x_3}^2 - f_{x_1} f_{x_3}^2 f_{x_1, x_3})}{f_{x_3}^2 \sqrt{f_{x_1}^2 + f_{x_2}^2 + f_{x_3}^2}}
$$

(15)

$$
M = \frac{(f_{x_1} f_{x_2} f_{x_3, x_3} + f_{x_1} f_{x_3} f_{x_1, x_3} - f_{x_1} f_{x_2} f_{x_1, x_3} - f_{x_2} f_{x_1} f_{x_1, x_3})}{(f_{x_3}^2 \sqrt{f_{x_1}^2 + f_{x_2}^2 + f_{x_3}^2})}
$$

(16)

$$
N = \frac{(f_{x_2} f_{x_3} f_{x_3, x_3} - f_{x_1} f_{x_3}^2 f_{x_1, x_3} - f_{x_1} f_{x_2}^2 f_{x_1, x_3})}{f_{x_3}^2 \sqrt{f_{x_1}^2 + f_{x_2}^2 + f_{x_3}^2}}
$$

(17)

The principal curvatures, $\kappa_i$, $i = 1, 2$ are the roots of the following quadratic equation as given in [11]:

$$
a \kappa^2 + b \kappa + c = 0
$$

(18)

$$
a = EG - F^2
$$

$$
b = 2FM - GL - EN
$$

$$
c = LN - M^2
$$

(19)

$$
\kappa_1 = -\frac{b + \sqrt{b^2 - 4ac}}{2a}; \kappa_2 = \frac{b^2 - 4ac}{2a}
$$

(20)

$$
b = (f_{x_1} f_{x_2} + f_{x_1}^2 + f_{x_2}^2) + f_{x_3} (f_{x_1}^2 + f_{x_2}^2) + 2f_{x_1} f_{x_2} f_{x_3, x_3} - 2f_{x_3} f_{x_1} f_{x_2} f_{x_1, x_3}
$$

(21)

$$
c = (f_{x_1}^2 f_{x_2, x_3} - f_{x_2} f_{x_3, x_3}) + f_{x_2} (f_{x_3, x_3} f_{x_1, x_3} - f_{x_1, x_3} f_{x_2}) + 2f_{x_1} f_{x_2} f_{x_3, x_3} - 2f_{x_2} f_{x_1} f_{x_3, x_3}
$$

(22)

$$
t_i = \begin{pmatrix} - (M - \kappa_i F) \\ L - \kappa_i E \end{pmatrix} or \begin{pmatrix} - (N - \kappa_i G) \\ M - \kappa_i F \end{pmatrix} \quad i = 1, 2
$$

(23)

$$
\nabla \kappa_i = [\kappa_{ix_1}, \kappa_{ix_2}]^T
$$

(24)

$$
\kappa_{ix_1}(x_1, x_2, g(x_1, x_2)) = \kappa_{ix_1}(x_1, x_2, x_3) - \kappa_{ix_2}(x_1, x_2, x_3) f_{x_3}
$$

(25)
\[ \kappa_{i2}(x_1, x_2, g(x_1, x_2)) = \kappa_{i2}(x_1, x_2, x_3) - \kappa_{i3}(x_1, x_2, x_3) \frac{f_{x_2}}{f_{x_3}} \] (26)