Computer Aided Geometric Design 28 (2011) 215–232

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Ridge extraction of a smooth 2-manifold surface based on vector field [☆]

WuJun Che [a,*], XiaoPeng Zhang [a], Yi-Kuan Zhang [a], Jean-Claud Paul [b,c], Bo Xu [a]

[a] *LIAMA-NLPR-Digital Content Technology Research Center, Institute of Automation, CAS, Beijing 100190, PR China*
[b] *School of Software, Tsinghua University, Beijing 100084, PR China*
[c] *INRIA, France*

ARTICLE INFO

Article history:
Received 20 August 2009
Received in revised form 28 February 2011
Accepted 19 March 2011
Available online 2 April 2011

Keywords:
Implicit surface
Invariant feature
Line of curvature
Parametric surface
Principal curvature
Principal direction
Ridge
Umbilical point
Vector field

ABSTRACT

This paper presents a general scheme to compute ridges on a smooth 2-manifold surface from the standpoint of a vector field. A ridge field is introduced. Starting with an initial ridge, which may or may not be umbilical, a ridge line is then traced by calculating an associated integral curve of this field in conjunction with a new projection procedure to prevent it from diverging. This projection is the first that can optimize a ridge guess to lie on a ridge line uniquely and accurately. In order to follow this scheme, we not only develop practical ridge formulae but also address their corresponding computational procedures for an analytical surface patch, especially for an implicit surface. In contrast to other existing methods, our new approach is mathematically sound and characterized by considering the full geometric structures and topological patterns of ridges on a generic smooth surface. The resulting ridges are accurate in the numerical sense and meet the requirement of high accuracy with complete topology. Although the objective of this paper is to develop a mathematically sound framework for ridges on a smooth surface, we give a comprehensive review of relevant works on both meshes and smooth surfaces for readers.

1. Introduction

Due to quickly developing computer technology and the extensive availability of 3D models, time-consuming computer-aided analysis of high-order differential geometry becomes more and more doable, increasing interest in techniques of 3D shape analysis in recent years. A set of geometric features consists of points describing a non-trivial region of a local surface with prescribed geometric properties, providing a solid foundation for shape analysis, design and recognition. Significant features usually involve high-order surface derivatives such that it is a difficult task to extract them precisely and robustly. Therefore, feature analysis and detection is currently a subject of intensive research in which invariant features play an important role.

1.1. Ridge

Koenderink (1990) recognized that ridges are significant features of a surface shape, conveying most of the shape's virtually essential characteristics. Several definitions for ridges in image and shape analysis can also be found in the literature (Eberly et al., 1994; Porteous, 2001). A generally acceptable definition is via principal curvature extrema along

[☆] This paper has been recommended for acceptance by G.E. Farin.
[*] Corresponding author.
E-mail addresses: chewj@liama.ia.ac.cn (W.J. Che), xpzhang@nlpr.ia.ac.cn (X.P. Zhang), y-k.zhang@163.com (Y.-K. Zhang), paul@tsinghua.edu.cn (J.-C. Paul), xubo@hitic.ia.ac.cn (B. Xu).

0167-8396/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cagd.2011.03.005

associated lines of curvature so as to be invariant under rotations, translations and scaling.[1] This kind of ridges usually constitute piecewise smooth curves of extremal curvature and thereby find use in applications such as surface segmentation and shape registration. Another significant property of ridges is that they are robust compared to geodesics and lines of curvature under slight deformation of the surface. Ridges provide us a pleasing geometrical representation of important physical properties. Therefore, they have been intensively studied in research for geometric modeling (Koenderink, 1990; Hosaka, 1992; Porteous, 2001; Hisada et al., 2002), image analysis (Eberly et al., 1994; Stylianou and Farin, 2004; Thirion, 1996a, 1996b), machine vision (Kent et al., 1996; Hallinan et al., 1999) and other fields (Ohtake et al., 2004; Yoshizawa et al., 2008).

Assume that $\mathbf{c}(s)$ is a line of maximum or minimum curvature parameterized by arc length s. Denote κ_i, $i = 1, 2$, their principal curvatures, respectively. Because ridges relevant to κ_1 and κ_2 can be taken up separately, $\mathbf{c}(s)$ is supposed to be either maximum or minimum when it is not specified explicitly, and its associated principal curvature is denoted as κ_n instead. For the sake of simplicity, we will also use the notation $d\mathbf{c}/ds = \dot{\mathbf{c}}$ and $d^2\mathbf{c}/ds^2 = \ddot{\mathbf{c}}$. Consider now $\dot{\kappa}_n$, evaluated along $\mathbf{c}$. According to the above definition, $\mathbf{p} = \mathbf{c}(s)$ is a ridge if $\dot{\kappa}_n(\mathbf{p}) = 0$. Because an umbilical point is a singularity of $\mathbf{c}$, this definition is unsuitable at an umbilical point; which must be a ridge as pointed out by definition via contact of the surface with osculating spheres (Cazals and Pouget, 2005a; Porteous, 2001). Hence a different analysis is called for.

Some applications, however, count only certain perceptually salient ridges, i.e. crest lines, or even further filtered ridges of prominence (Hildebrandt et al., 2005; Ohtake et al., 2004; Stylianou and Farin, 2004; Thirion, 1996b). These lines consist of a subset of ridges, where κ_n is maximal in absolute value. They are designed for a mesh model, where a configuration of ridges fails to match that of a smooth surface and losing not only some sharp ridges but significant topological information of ridges as well (Cazals and Pouget, 2005b). Consequently, the ridges, as used in this article, are more than just crest lines since full set of ridges can exhibit significant topological structure of features.

Ridges has been examined deeply in mathematics and much work has been performed on their numerical calculation, but an efficient and numerical framework is absent to extract full set of ridges on a smooth surface with safeguard. The goal of this work is to make some significant contributions toward it. The paper is organized as follows. In the next two subsections, we give a broad overview of existing work on ridges followed by our contributions. Given some mathematical facts on ridges, we introduce the concepts of ridge field on a manifold surface and its topological structure – ridge graph – in Section 2. Then we lay out our scheme to extract the ridge graph in the ridge field through a series of computational procedures in Section 3. In Section 4 we depict a new projection for a ridge guess onto the ridge line and briefly discuss orientation problems of $\dot{\kappa}_n$. We show numerical experiments in Section 5 and finally conclude our work in the last section. All practical formulae are given in Appendix A.

1.2. *Previous work*

The desirable method for ridges on a smooth surface is essentially different from that on a mesh surface because the accuracy of the former is required much more highly than that of the latter. The focus of this paper is on smooth implicit and parametric surfaces. But most existing methods are proposed for ridges on meshes. We still attempt to give a comprehensive review of all relevant methods for both meshes and smooth surfaces. We emphasize that our objective is to develop a new framework for ridges which is more amenable to smooth surfaces instead of discrete surfaces.

1.2.1. *Existing methods for ridges*

Ridges are at least third-order derivative phenomena and difficult to detect perfectly. It comes natural to consider ridge lines as the bi-iso-lines between the original surface and the ridge trimming surface of $\dot{\kappa}_n = 0$. Therefore, finding ridges is essentially a problem of solving a system of non-linear equations, and some practical approaches have been proposed from different angles.

Generally speaking, four key issues need to be addressed to extract complete ridges:

Issue 1 *(Optimizing ridges).* It is a core issue to properly project a guess point to a ridge as no existing methods can provide an immediate ridge without an optimization procedure. For a smooth surface, where $\dot{\kappa}_n$ is well defined for non-umbilical points, a binary subdivision algorithm is used to converge to the solution (Morris, 1996; Belyaev et al., 1998). Another more appropriate alternative is to determine a ridge by tracing its corresponding line of curvature, as Hosaka (1992) and Musuvathy et al. (2009) do on a parameterized surface.

For a mesh surface, however, we are usually unconcerned about this issue since we cannot do so faithfully. A commonly used method is a simple linear interpolation based on the zero-crossing of a mesh edge transversally crossing a ridge line (Gueziec, 1993; Morris, 1996; Kent et al., 1996; Cazals and Pouget, 2004, 2005b; Ohtake et al., 2004). The quality of resulting ridges relies closely on the mesh fineness.

[1] Ridges can be defined on an n-dimensional hypersurface (Eberly et al., 1994). For the sake of technical simplicity, only 2-manifold surfaces are considered in this paper.

Issue 2 *(Estimating ridges).* Before accurately optimizing ridges locally as above, a method should be available to find where ridges may appear roughly on the surface. In contrast, it may be classified as global because it is designed to extract the full set of ridges in some area of interest. The main current approaches typically fall into two categories:

- **Tracing method.** The basic idea is to start with a certain intersection point and then trace the intersection curve by solving a set of differential equations formulated for surface–surface intersection as an initial value problem. Hosaka (1992) developed differential equations for ridges, but they are too complicated to exploit in practice; instead, an intuitive method is supposed to search points of extremal curvature by tracing each line of curvature and then making their interpolation curves. In order to simplify the vector computation for the tracing, Musuvathy et al. (2009) adopted a branch of principal direction vectors to approximate the required tangent vectors to the underlying ridge line, which is perpendicular to the other branch of principal direction vectors relevant to the underlying ridges; for instance, the current ridge is relevant to κ_1 and the advance step follows the principal direction relevant to κ_2. This approximation is very rough so correction is required to shift each solution point to the ridge line. In contrast, Clémençon et al. (2008) formally computed the integration vectors as those perpendicular to the gradients of $\dot{\kappa}_n$, i.e. $\nabla\dot{\kappa}_n^{\perp}$. The resultant vectors are accurate, so a single ridge line can be computed with acceptable accuracy compared to that of Musuvathy et al. (2009). All the above work only deals with a parametric polynomial surface; our work in this paper can be regarded as of this class.

  A simplified variation is the widely used zero-following method for a mesh surface (Gueziec, 1993; Morris, 1996; Kent et al., 1996; Ohtake et al., 2004; Cazals and Pouget, 2005b). In this method, each line is tracked as it crosses a side of a primitive in the piecewise linear surface, e.g. a triangle or a quadrangle where the line is expected to cross a second side. We can then repeat the process of an adjacent primitive. The capability of this method for recovering topological information of ridges is weak and the extent rests with polygonization resolution. So no researchers involved opine that the zero-following method sounds reasonable for full set of ridges on a smooth surface, although it can be applied trivially to a smooth surface in a brute-force manner (Morris, 1996; Cazals and Pouget, 2005b).
- **Surface–surface intersection techniques.** The intersection of the ridge trimming surface with the original surface provides the ridge lines. If the original surface is implicit, both the original and the trimming surfaces can be approximated by polygons, and the ridges are obtained by intersecting two resulting polyhedra. Thirion and Gourdon (1996) proposed a Marching Lines algorithm, which characterizes the ridge lines as the intersection of two implicit surfaces, later strengthened by introducing the so-called Gaussian extremality whose sign is independent of the orientation problems (Thirion, 1996b). However, this method is still unsuitable for an umbilical point. Belyaev et al. (1998) improved the method further in an adaptive way with a refinement of the triangulation approximating the implicit surface near ridges. Bogaevski et al. (2003) used a formal algebraic computation to determine a system of ridge trimming equations on an implicit polynomial surface, and the intersection with the original surface is then computed by the Marching Lines algorithm. The Marching Lines algorithm can be applied to a pair of images or functions defined on a regular grid. To recover topological information of ridges, it is essentially no more than the zero-following method for a mesh surface.

  If the original is parameterized, the coordinate variables of the implicit ridge surface can be substituted by those of the parameterized such that the problem reduces to finding roots of an implicit function with only two parameters. A representative method was proposed by Cazals et al. (2005, 2006), in which an implicit equation is exhibited, describing and encoding all ridges on a polynomial surface. But the method results in expressions of very high degree and suffers from numerical instability. It is well-suited for implementation in symbolic mathematical systems but is difficult to implement in floating point arithmetic.

Both zero-following method and Marching Lines algorithm are brute-force methods, based on a high-quality discrete sampling of the underlying surface. They are inappropriate solvers for ridges on a smooth surface.

An interesting work is to exploit the invariant theory to derive the ridge line equations as the zero set of an invariant function (Gravesen, 2005); the invariant form trivially implies that ridges are independent of the given parametrization. This invariant function can be rearranged to match, up to a constant factor, an implicit polynomial function characterizing the ridges if the input surface is polynomial. Hence the set of all ridges and umbilical points is globally described by an algebraic curve, and this problem of approximating ridges can be switched into the field of algebraic geometry to report ridge lines certified topologically (Cazals et al., 2005). But the resultant implicit function is of high degree; this method requires complex algebraic tools and runs too slowly to use in practice, as pointed out above.

Tracing methods for an analytic surface are usually employed to extract ridges and are computationally less demanding than other methods. However, such methods are by themselves *incomplete* in that they require starting points to trace every branch of the solution. A lot of methods have been well developed in the CAGD community. In order to identify all connected components of the intersection curve, a set of important singular points on the intersection curve, e.g. border, turning and singular points of the intersection, can be exploited as the starting point (Patrikalakis and Maekawa, 2002). In the ridge extraction scenario, Musuvathy et al. (2009) adopted points of local extrema of principal curvature, i.e. those of $\nabla\kappa_n = \mathbf{0}$, and umbilical points on a B-spline surface, and later furnish the necessary starting points with border points, i.e. the intersections of ridge lines with a boundary edge of the parameter space and with the knot line where the surface could be with deficient smoothness lower than C^3 (Musuvathy and Cohen, 2009). If the given surface is non-polynomial, however, no robust and efficient solving techniques are available for this purpose; we will discuss this further in Section 3.2.

Issue 3 *(Reporting singular points).* Singular points deeply reflect the topological structure of ridge lines. A common singular point is umbilical. In theory, locating an umbilical point can be done faithfully only on a smooth surface because a discrete representation of a smooth surface usually loses more or less information to recover all those points. For example, differential quantities cannot be estimated absolutely reliably for a discrete model containing noise; consequently, it is a non-trivial task to extract umbilical points consistent with the configuration of the resultant ridge lines. Umbilical points are important in shape analysis and quite a few methods have been developed on surfaces in various surface representations; see Maekawa et al. (1996) and Che et al. (2007) for a brief overview.

To the best of our knowledge, only work has been reported that calculates full sets of ridges on a parametric polynomial surface. Additionally, a self-intersection point is also singular, but rarely discussed in the literature. We will depict them in Section 2.

Issue 4 *(Certifying topology).* This issue stems from the complex patterns made on a manifold surface by ridge lines and their singularities. Both theory and practice have shown that a ridge graph may have complex topological patterns. It is also the essential reason why a conventional method for a mesh surface cannot be applied straightforwardly to a smooth surface. Cazals and Pouget (2005b) considered the pattern of ridges in the neighborhood of an umbilical point based on a mesh counterpart of an implicit or a parametric surface. Thirion (1996a) extracted an extremal point (one type of self-intersection points) in a fashion consistent with the Marching Lines algorithm; Clémençon et al. (2008) and Musuvathy et al. (2009) tried to sweep around an umbilical point for a ridge by putting scout points at a small distance from the umbilical point. All of these methods are too heuristic to be applied further to certify the topology of ridges. The pioneering work describing topological difficulties, as we know, was given by Cazals et al. (2005, 2006), and is based on computer algebra to report a topologically certified approximation of ridges, including umbilical points, extremal points and turning points.[2] Yet this method is only suitable for a parameterized polynomial surface at present, and runs very slowly.

Besides the aforementioned methods focusing on the surface itself, other methods exist by introducing an associated object, such as the focal-surface based method (Watanabe and Belyaev, 2001) or the medial-axis based method (Hisada et al., 2002). But these methods are still quite limited and have received little attention.

1.2.2. Existing work on ridges over a smooth surface

To sum up, a zero-following method is still the most commonly used, based on zero-crossing edges of a piecewise linear model. To such the polygonal mesh, it is acceptable that only crest lines are usually considered since the notion of continuous geometric attributes, especially high-order ones, is not trivial.

For an analytically-represented surface, ridges are defined analytically and, at least in theory, can be recovered using computer algebra or numerical techniques. But a zero-following method is too crude to extract complete ridges from a smooth surface. Those methods aiming at an analytical smooth surface, however, are limited to a polynomial surface of low degree (Bogaevski et al., 2003; Cazals et al., 2005, 2006). The methods dedicated to ridges on a smooth surface, including umbilical points, are those of Cazals et al. (2005, 2006), Clémençon et al. (2008) and Musuvathy et al. (2009). Some of them deal with a parameterized polynomial surface with computer algebra and validate the topology of ridges reliably, but are prohibitively computationally expensive (Cazals et al., 2005, 2006). The other methods use either integral methods or tracing methods to trace ridge lines, but they only handle the patterns of ridges at a singular point in a heuristic and rough way (Clémençon et al., 2008; Musuvathy et al., 2009). All of these works are still limited to a polynomial surface.

As described above, little work benefits from available analytical representations and the accuracy depends on the fineness of surface polygonization. Work on topological patterns of ridges has been even less. The framework by Musuvathy et al. (2009) bears a superficial resemblance to ours because we both use tracing method, but in mathematical respects our techniques and theirs are quite different.

1.2.3. Practical formulae and orientation problems

As a ridge involves differentiating κ_n, related practical formulae are quite desirable to calculate explicitly.

For the sake of compact representation, we use vector notation and a vector is represented as a column. Given a map $f : \mathbb{R}^n \to \mathbb{R}^m$, we write its k-th order Fréchet derivative at $\mathbf{p}$ as $f^{(k)}(\mathbf{p})$; if $f(\mathbf{p})$ is continuously differentiable up to order k, it is a k-times linear form on $\mathbb{R}^n$, simplified as $f^{(k)}$, and the first, second and third derivatives are f', f'' and f''', respectively. $f^{(k)}\mathbf{t}$ means $f^{(k)}$ acts on $\mathbf{t}$ and results in a $(k-1)$-times linear form of $f^{(k)}\mathbf{t}$ on $\mathbb{R}^n$. We assume that the surface of interest is smooth enough so that it possesses a sufficient number of continuous derivatives required for analysis. If the reader is unaccustomed to vector notation, Einstein notation can also be used to rewrite the concerned formulae so as to help understanding; when an index variable appears twice in a single term, once in a superscript and once in a subscript position, it implies that all of its possible values are being summed. For example, $H'''\alpha^3$ in Eq. (1) means $H_{ijk}\alpha^i\alpha^j\alpha^k$, i.e.

[2] The ridge line is tangent to a line of curvature at a turning point.

$\sum_{i,j,k=1}^{3} H_{ijk}\alpha^i\alpha^j\alpha^k$ and $H''\alpha\mathbf{n}$ means $H_{ij}\alpha^i n^j$, i.e. $\sum_{i,j=1}^{3} H_{ij}\alpha^i n^j$, where α^i and n^j are the i-component and j-component of α and $\mathbf{n}$, respectively. Here H_{ij} is the mixed second-order partial derivatives with respect to the ith and jth variables.

- In Hosaka (1992) and Morris (1996), equations governing ridges for an implicit or a parametric surface are provided, but not $\dot{\kappa}_n$ itself. For an implicitly defined surface of $H(\mathbf{p}) = 0$, its expression is given by:

$$\dot{\kappa}_n = -\frac{H'''\alpha^3 + 3\kappa_n H''\alpha\mathbf{n}}{H'\mathbf{n}} \tag{1}$$

  where $\mathbf{n}$ is the unit normal to the oriented surface and α the normalized vector of principal direction (Belyaev et al., 1998; Monga and Benayoun, 1995; Porteous, 2001, Exercise 11.8). Eq. (1) is so compact that most existing work has been based on the representation.
  In this article, $\ddot{\kappa}_n$ is also necessary for optimizing a ridge guess by projecting it efficiently along $\mathbf{c}(s)$ onto the ridge line. Its computable formula defined on a parametric surface is rather more technical according to classical differential geometry than that of an implicit surface (see Appendix A). Except in a special case of $\ddot{\kappa}_n$ holding only at a ridge (Belyaev et al., 1998), we cannot find solution for $\ddot{\kappa}_n$ for an implicit surface elsewhere in the literature. The practical formula of $\ddot{\kappa}_n$ on an implicit surface is difficult to obtain because it involves the differential geometry of lines of curvature, and it is only recently that an explicit analysis has been developed by Che et al. (2007).
- Eq. (1) obviously indicates that $\dot{\kappa}_n$ depends on a given orientation of α and its opposite orientation changes $\dot{\kappa}_n$ in sign. So it is a correct definition only when α is oriented coherently along $\mathbf{c}(s)$. In particular, tracking a zero-crossing of $\dot{\kappa}_n$ from a sign change along a curve segment on the surface requires finding a coherent orientation of the principal frame, i.e. the orientation problem. One heuristic way to find such a local orientation is to choose two principal vectors at endpoints so that they make an acute angle. This Acute Rule has been widely used in computer examination of ridges (Cazals and Pouget, 2004). However, this method usually suffers from frequent failures, e.g. in the vicinity of an umbilical point. An alternative is to design a measure function which is invariant with respect to the orientation of α but still encodes the ridges, e.g. Gaussian extremality (Thirion, 1996b).

1.3. Contributions

The goal of our work in this paper is to propose a new and comprehensive scheme to extract the ridge structure on a smooth manifold surface using a vector field. Remarkably different from existing methods, our method is the first to cover all the four issues in the same framework from a novel aspect of the vector field. It works in fully numerical fashion with safeguard to extract the topological patterns near a singular ridge.

This paper's contributions are:

- A novel projection algorithm for optimizing a ridge guess. Its advantage is that all points on a line of curvature result in the same ridge image locally.
- A new framework to extract ridges. It works in concert with the new projection, based on a vector field defined on the surface. An integral curve through a ridge in the field leads to a ridge line.
- Developing high-order practical formulae relevant to ridges. These formulae are represented not only explicitly but computationally as well.
- Algorithms for locating singular points to precisely extract topological structures of ridges, some of which have not been described in the literature. All other existing methods, which are comparable to our approach, are confined to polynomial surfaces.

Our scheme gives a rich collection of recent investigation on ridges from the computational and mathematical viewpoint and provides sufficient mathematical soundness for the proposed framework to compute ridges on a smooth surface.

2. Ridge graph and ridge field

The complete set of ridges on a generic surface forms a collection of piecewise smooth ridge lines – a geometric structure of almost entirely one-dimensional differential manifold. In this section we list some essential facts of differential geometry on ridges and introduce the concepts of a ridge graph and a ridge field from the viewpoint of a vector field, on which our framework is grounded.

2.1. Global quality of ridges and graph structure

A ridge line is usually a locally smooth curve and thus their collection can be regarded as piecewise smooth curves except at some singular points (see Section 2.2). Both umbilical points and self-intersection (but non-umbilical) points constitute singular points. Those self-intersection points can be separated into two classes: one is a point from which ridge lines fanning out are relevant to both κ_1 and κ_2, given the name of an *extremal point* (Thirion, 1996b); the other is only related to either κ_1 or κ_2, and called a *critical point* here. In this paper we do not consider extremal points because ridges of κ_1 and κ_2 are processed independently and it can be recovered without too much difficulty compared to critical points.

Thus, the ridge structure is usually a geometric graph. In this graph, a vertex is a singular point, either umbilical or critical, and an edge is a single curve, *either* closed, *or* open and ending at a vertex or at infinity. We call this a *ridge graph*.

2.2. Local quality of ridges and vector field

A potential method to examine ridges is to employ the vector field relevant to ridges. This method is not so intuitive because many computational issues should be addressed carefully. In this subsection, we will introduce a *ridge field* on a smooth surface and describe how to compute the field.

2.2.1. Monge form and ridges

We choose coordinates in $\mathbb{R}^3$ so that a point of interest, say $\mathbf{p}_0$, is at the origin and the x- and y-axes coincide with two principal directions if $\mathbf{p}_0$ is non-umbilical and the x-axis is set as of interest, say relevant to κ_1. If $\mathbf{p}_0$ is umbilical, the x-axis is set as we want so long as it is located on the tangent plane. The surface is then expressible in Monge form as

$$z = \frac{1}{2}\left(\kappa_1 x^2 + \kappa_2 y^2\right) + \frac{1}{6}\left(b_0 x^3 + 3b_1 x^2 y + 3b_2 x y^2 + b_3 y^3\right) + \frac{1}{24}\left(c_0 x^4 + 4c_1 x^3 y + 6c_2 x^2 y^2 + 4c_3 x y^3 + c_4 y^4\right) + O(x, y)^5 \qquad (2)$$

where $O(x, y)^5$ is a Peano remainder of a power series of degree five or more.

Given a non-umbilical point $\mathbf{p}_0$, $b_0 = 0$ if and only if κ_1 has an extremum along $\mathbf{c}(s)$, i.e. $\dot{\kappa}_1 = 0$. The formulae of attributes with respect to κ_1 and κ_2 ridges through Eq. (2) can turn into each other by the symmetry of x, y and coefficients of Eq. (2). Furthermore, we construct Eq. (2) separately for κ_1 and κ_2 in implementation because of their independence in topology, so we consider only the κ_1 ridges without loss of generality in the following text.

2.2.2. Non-umbilical points

Denote a ridge line as $\mathbf{e}$. It has a local equation (Hallinan et al., 1999):

$$P \cdot x + Q \cdot y + O(x, y)^2 = 0 \qquad (3)$$

where $P = (\kappa_1 - \kappa_2)(c_0 - 3\kappa_1^3) + 3b_1^2$ and $Q = (\kappa_1 - \kappa_2)c_1 + 3b_1 b_2$.

The tangent vector at $\mathbf{p}_0$ in the Monge frame is

$$\mathbf{t} = (-Q, P)^T \qquad (4)$$

When either of P or Q is non-vanishing, we deduce that a tangent $-Q\alpha_1 + P\alpha_2$ to $\mathbf{e}$ can be defined at $\mathbf{p}_0$ and $\mathbf{e}$ is a smooth 1-manifold locally; here α_1 and α_2 are principal directions relevant to κ_1 and κ_2 respectively. The case of $P = Q = 0$ means a critical point; the topological pattern in its vicinity is then determined by non-vanishing terms of lowest degree in Eq. (3), e.g. quadratic. Suppose the quadratic part of Eq. (3), say $ax^2 + 2bxy + cy^2$, is not vanishing. This means that, locally, the picture of ridges is an isolated point if $b^2 < ac$, or two smooth intersecting curves if $b^2 > ac$. We call the latter a *generic critical point*. In our work, an isolated critical point is not considered.

2.2.3. Umbilical point

If $\mathbf{p}_0$ is a generic umbilical point, $\kappa_1 = \kappa_2$ and Eq. (3) fails. Its analysis can be made via the cubic form of Eq. (2):

$$C_3(x, y) = b_0 x^3 + 3b_1 x^2 y + 3b_2 x y^2 + b_3 y^3 \qquad (5)$$

The classifying form of the patterns made by ridge lines in the neighborhood of $\mathbf{p}_0$ is just Eq. (5). The number of ridge lines traveling outwards is given by the number of real root lines of this cubic, but they are not the limiting tangents at $\mathbf{p}_0$ to $\mathbf{e}$. However, the tangent directions can be computed from Eq. (5). Hallinan et al. (1999) propose such a method, but the Monge frame needs to be readjusted to have the x-axis aligned along a root line to make b_0 equal to zero. Here we develop another procedure to facilitate the computation directly in terms of $C_3(x, y)$.

Consider a parameterized regular surface

$$\mathbf{r} : \mathbb{R}^2 \rightarrowtail \mathbb{R}^3; \qquad \mathbf{u} = (u, v)^T \mapsto \mathbf{r}(\mathbf{u}) \qquad (6)$$

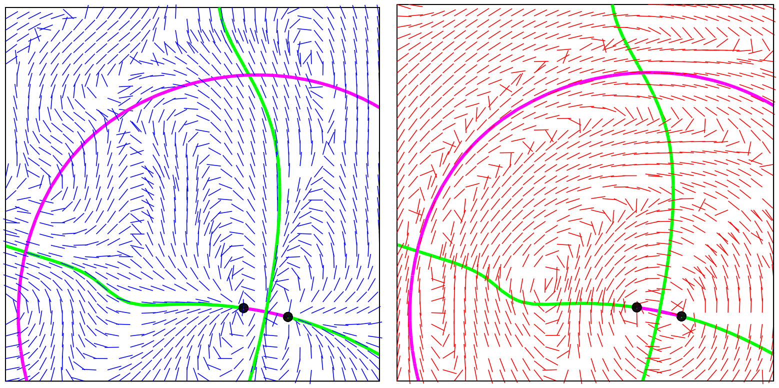
For a certain $\mathbf{p} \in \mathbb{R}^3$, the contact function between the surface and the sphere centered at $\mathbf{p}$ is defined by $V = \mathbf{p} \cdot \mathbf{r} - \frac{1}{2}\mathbf{r} \cdot \mathbf{r}$. A tangent vector $\mathbf{t}$ at $\mathbf{p}_0$ satisfies the following equations, there being $\mathbf{w}$ and $\mathbf{v}$ (see, Porteous, 2001, Theorem 10.11):

$$V'''\mathbf{twv} = 0 \qquad (7)$$

$$V'''\mathbf{w}^3 = 0 \qquad (8)$$

$$I''\mathbf{wv} = 0 \qquad (9)$$

Here I'' is the first fundamental form (see Eq. (11)).



(a) Ridge field relevant to κ_1 (b) Ridge field relevant to κ_2

Fig. 1. Ridge field and ridge graph on a Gaussian blend surface. Ridge lines of κ_1 are green; of κ_2 are magenta; the black points are umbilical. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Eq. (8) implies $\mathbf{w}$ is a vector of a root line of Eq. (5) and Eq. (9) implies $\mathbf{w}$ and $\mathbf{v}$ are mutually orthogonal. Supposing $(x, y)^T$ is a root line, we can then obtain $\mathbf{v} = (-y, x)^T$, and thus according to Eq. (7), the corresponding tangent direction is given by

$$\mathbf{t} = \begin{pmatrix} b_2x^2 - (b_1 - b_3)xy - b_2y^2 \\ -b_1x^2 + (b_0 - b_2)xy + b_1y^2 \end{pmatrix} \tag{10}$$

Eqs. (4) and (10) can now be applied to defining a ridge field on a smooth 2-manifold.

2.2.4. Ridge field

Thus far, a vector flow has been defined on the ridge graph. We further extend Eq. (4) to other points on the surface and a vector field across the surface, called the *ridge field*, is obtained as a result:

- A singular point in this field is either umbilical or critical, at which vector directions are indeterminate. The ridge pattern in the vicinity of a generic critical point is determined by the quadratic terms of Eq. (3), while the ridge pattern of an umbilical point is a bit more complicated. The limiting direction approaching a generic umbilical point along a ridge line is computable by Eq. (10); the interested readers can refer to Hallinan et al. (1999), Porteous (2001) and Cazals and Pouget (2005a) for more details.
- At a non-singular point, the vector is given by Eq. (4).

A ridge graph is actually a geometric graph through singular points in the ridge field, reflecting its topological information. Both umbilical points and some critical points serve as its vertices. In implementation, ridges relevant to κ_1 and κ_2 can be computed separately in topology. A ridge subgraph of κ_1 is topologically connected with that of κ_2 at umbilical points and either one or three ridge lines pass through a generic umbilical point where each one changes from being an extremum of one curvature to being an extremum of the other. But the ridge graph does not yet cover all cases, e.g. a ridge graph can contain no singular points.

The primary goal of this paper is not to study the properties of ridge field, so we do not investigate deeply into this aspect. Instead, we interpret the ridge field and the ridge graph by a vivid example here, an explicit surface as a blend between two translated and scaled Gaussian distributions defined by

$$h(u, v) = (1 - u)^2 \exp\bigl(-u^2 - (v + 0.5)^2\bigr) - (1 - v)^2 \exp\bigl(-u^2 - v^2\bigr)$$

where $(u, v) \in [-1, 1] \times [-1, 1]$. Fig. 1 illustrates its ridge field and associated ridge graph in the parametric u–v plane. It can also be observed that the ridge graph does not pass through some critical points in the ridge field.

3. Framework

In Section 2, we defined a ridge field and showed how to compute it on a manifold surface. We will continue to outline our framework in this section to extract a ridge graph based on a corresponding ridge field, partly influenced by the work of Zhang et al. (2009). For the sake of technical simplicity, we only make use of coefficients in Eq. (2) up to the fifth order, i.e. c_i, $i = 0, \ldots, 4$, provided in Appendix A. On this assumption, the tangent vector at a generic critical point fails to be represented explicitly; hence among singular points, a generic critical point is unable to serve as a starting point in our work to integrate the field for a ridge line like a generic umbilical point.

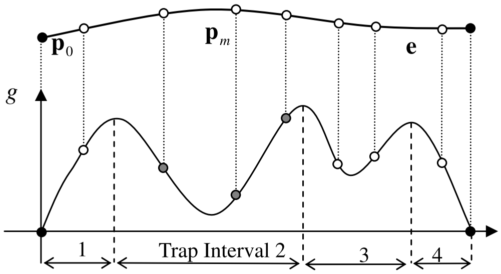


Fig. 2. A tracing method for singular points in integrating $\mathbf{e}$: $\mathbf{p}_0$ is a starting point; any gray one can work as a guess for a singular point in the trap interval 2, e.g. $\mathbf{p}_m$.

3.1. Integration of a ridge line

Given a starting ridge $\mathbf{p}_0$, either a generic umbilical point or a non-singular ridge, and an initial ridge vector $\mathbf{t}_0$, computing a ridge line – **Integration**($\mathbf{p}_0$, $\mathbf{t}_0$) – can now be reduced to an initial value problem: an integral curve $\mathbf{e}$ in the ridge field. Any routine for this problem can be employed, e.g. the Runge–Kutta–Fehlberg method with adaptive stepsize control in this article. **Integration** is a crucial component of our framework. Two important operations are performed simultaneously for a stable numerical integration. One is to locally optimize solution points in integration to prevent the computed ridge line from diverging; the other is related to where the ridge line ends, involving the possible failure of the integration near a singular point. These operations are closely interrelated to Issues 1 and 3; we now depict these two basic operations, respectively.

Local optimization. The computed $\mathbf{e}$ could suffer from drift over progress distance by a simple ODE integrator because of numerical integration errors. Fortunately, in contrast to lines of curvature, a solution point of the integral curve can be corrected robustly and we develop a projection algorithm to do so. Given a ridge guess, whose target ridge is non-umbilical, our approach is to propagate it along its relevant $\mathbf{c}(s)$ to the target ridge by combining the information from Newton's method. Because $\mathbf{c}(s)$ is a space curve, Newton's method cannot be applied directly and we will elaborate on the details in Section 4. If the target is umbilical, the optimization is carried out as singular.

Singular points. Smooth ridge lines are connected at singular points to comprise the ridge graph. Furthermore, a singular point is often a branching vertex of the graph to prevent integration from being performed favorably. Thus we need to monitor these points along $\mathbf{e}$ in **Integration**. We apply the tracing method of lines of curvature proposed by Zhang et al. (2009) for singular points; but $\mathbf{c}$ in the field of principal directions in their framework is replaced with $\mathbf{e}$ in the ridge field in our framework. As illustrated in Fig. 2, with the integration advance, we keep a close watch on a function g at a solution point and measuring its violation of singularity, i.e. indicating how close it is to being singular. If the function begins to decrease, the present solution point steps into the neighborhood of a local minimum like a trap of an extremum. A trap interval consists of consecutive points along $\mathbf{e}$, with g values first decreasing and then increasing. The tracing proceeds in a trap-by-trap fashion until the integration terminates. For a guess of a potential singularity in each trap interval, a standard method for minimization optimization is used to fix the target point, i.e. umbilical points and critical points.

- Umbilical points. The evaluation function g can be defined respectively for a parametric surface and an implicit surface (Zhang et al., 2009). The numerical methods proposed by Maekawa et al. (1996) and Zhang et al. (2009) can be employed.
- Critical points. No algorithms are available to compute a critical point yet. Here, a novel approach is developed to do so, although not so efficient as Newton's method used in optimization of umbilical points. A critical point is naturally to be evaluated by $g = P^2 + Q^2$; yet differentiating g leads to complicated formulae of g' since both P and Q rely on the Monge coordinates. Thus we use a Downhill Simplex method which requires only function evaluations, but no derivatives (Nelder and Mead, 1965). In consideration of the fact that a critical point, characterized by $b_0 = P = Q = 0$, does not occur on a generic surface, we believe that our approach is reasonable in our scheme.

Singular points are solved during the running of **Integration** while a stack is maintained to preserve those newly discovered generic umbilical points for starting ridges (see the next subsection). The integration terminates at a singular point or an intersection point with the domain bounds. One technical point to consider is the additional possibility where $\mathbf{e}$ is closed. In this case, a trivial process is required to verify whether the last solution point is approaching the starting one in a nearly parallel direction.

Algorithm 1: Basic framework to extract ridge graph on a generic smooth surface.

```
Triangulate the input surface as a sparse $\mathbb{M}$; $G = \emptyset$;
Compute umbilical points and push generic ones onto a stack $\mathbb{S}$ based on $\mathbb{M}$;
For (Each edge $\ell$ of $\mathbb{M}$)
  If ($\ell$ is an edge guess and $\ell \cap G = \emptyset$)
    Find a ridge $\mathbf{p}_0$ on $e$ with ridge vector $\mathbf{t}_0$;
    $\mathbf{e} = \mathbf{Integration}(\mathbf{p}_0, \pm\mathbf{t}_0)$; $G = [G, \mathbf{e}]$;
    Mark tangents to $\mathbf{e}$ at ending generic umbilical points;
  EndIf
  If ($\mathbb{S}$ is not empty)
    $\mathbf{p}_0 = \mathrm{Pop}(\mathbb{S})$;
    For (all unmarked tangents $\mathbf{t}_0$ at $\mathbf{p}_0$)
      $\mathbf{e} = \mathbf{Integration}(\mathbf{p}_0, \mathbf{t}_0)$; $G = [G, \mathbf{e}]$;
      Mark tangents to $\mathbf{e}$ at ending generic umbilical points;
    EndFor
  EndIf
EndFor
```

3.2. Extraction pipeline of a ridge graph

In the previous subsection, we described how to calculate a ridge line. We now set out to develop a framework to extract the ridge graph. This procedure belongs to a global computation, roughly addressing Issues 2 and 4.

Estimating ridges. In Section 3.1, we handled tracing **e** as an initial value problem, but did not mention how to get a starting ridge.

Polynomial surfaces are ubiquitous in CAD applications; many global and efficient methods exist to solve a system of polynomial equations. Thus if the input surface is polynomial, a set of important points on the ridge graph, such as border points, umbilical points, and locally extreme points of principal curvatures, can serve as starting points to identify all connected ridge lines (Cazals et al., 2006; Che et al., 2007; Maekawa et al., 1996; Musuvathy et al., 2009; Musuvathy and Cohen, 2009). See Patrikalakis and Maekawa (2002) for a good overview on polynomial solvers.

If the given surface is non-polynomial, however, no robust and efficient solving techniques are available to find those points. An intuitive method is proposed to find them, accompanying the ridge tracing in our framework. Our approach is to first use a polygonizer of a smooth surface to obtain a sparse mesh, e.g. a polynomizer of an implicit surface or a mesh produced by the regular grid in the parametric space for a parametric surface. We use heuristics for an umbilical guess: a vertex with local minimum of g amongst its 1-ring vertices. For a guess of a non-singular ridge, a zero-crossing detection of $\dot{\kappa}_n$ is applied by going over the mesh edges, leading to an *edge guess* instead (see Issue 1 in Section 1.2.1). Finally, a guess is optimized to be a starting ridge. This approach is heuristic but works fairly well in our experiments because of the connectivity of the ridge graph and information redundancy collected from all guesses.

Topological assemblage. Both theory and practice have shown that a ridge graph may have complex topological patterns. Since umbilical points deeply reflect the topology of a ridge graph, our framework centers on these points to trace ridge lines, taking full advantage of known ridge patterns near a generic umbilical point (Hallinan et al., 1999; Porteous, 2001; Cazals and Pouget, 2005a) to guarantee the topology. If no more information is provided from the umbilical points, tracing starts from a non-singular ridge. Due to our robust and accurate approaches to Issues 1, 2, and 3, respectively, the graph extraction is quite intuitive: one ridge line by one ridge line.

We synthesize all of the above procedures to get an extraction routine of the ridge graph as shown by the pseudocode in Algorithm 1. The algorithm centers on generic umbilical points with starting priority compared with the non-singular points resulting from edge guesses. The maintenance of the stack $\mathbb{S}$ is actually performed in **Integration**: a new generic umbilical point is pushed into $\mathbb{S}$ when it is discovered. To avoid tracing **e** repeatedly, only one starting ridge of **e** is picked up amongst ending umbilical points from which **e** radiates out and those ridges from which edge guesses arise. This is done by excluding edge guesses crossing the latest ridge graph G, and by marking the tangent directions at generic umbilical points, from which computed ridge lines radiate out. We learned by experience that the second step of computing umbilical points as the early starting ridges in Algorithm 1 can be ignored because of the stronger impact of edge guesses on starting ridges for a sparse mesh.

If the input surface is polynomial, characteristic points can be computed using a solver of polynomial system as starting ridges (Cazals et al., 2006; Musuvathy et al., 2009). Therefore, the spare mesh $\mathbb{M}$ is unnecessary any more and Algorithm 1 is reduced to Algorithm 2.

It is worthwhile to note that operations are performed in the parametric domain if the input surface is parameterized. Otherwise, if it is implicit, the computations are performed directly on the curve surface itself rather than the flat parametric plane and it is necessary to project a solution point onto the surface. See Zhang et al. (2009, Section 6.2).

Algorithm 2: Basic framework to extract ridge graph on a parameterized polynomial surface.

```
Compute characteristic points by a solver of polynomial system and put them onto $\mathbb{S}$.
While ($\mathbb{S} \neq \emptyset$)
  $\mathbf{p}_0 = \mathrm{Pop}(\mathbb{S})$;
  For (all unmarked tangents $\mathbf{t}_0$ at $\mathbf{p}_0$)
    $\mathbf{e} =$ Integration($\mathbf{p}_0, \pm\mathbf{t}_0$);
    Mark tangents to $\mathbf{e}$ at ending characteristic points;
  EndFor
EndIf
```

Algorithm 3: Ridge projection along $\mathbf{c}(s)$.

```
Initialize $\mathbf{p}_0$ as a starting point with principal direction $\alpha$ by Zero Rule and set $s = -\dot{\kappa}_n/\ddot{\kappa}_n|_{\mathbf{p}_0}$;
While ($s > \varepsilon$)
  Fix minimal $m$ by tracing $\mathbf{c}(s)$ from $\mathbf{p}_0$ with a maximal stepsize $s$ in direction $\alpha$, resulting in $\mathbf{p}_1, \ldots, \mathbf{p}_m$ such that $|\dot{\kappa}_n(\mathbf{p}_m)| \geqslant |\dot{\kappa}_n(\mathbf{p}_0)|$ or $\sum_{i=0}^{m} |\mathbf{p}_i\mathbf{p}_{i-1}| \geqslant s$;
  If ($m > 1$)
    Let $\mathbf{p}_0 = \mathbf{p}_{m-1}$, $\alpha =$ principal direction at $\mathbf{p}_{m-1}$ oriented by Zero Rule; $s = -\dot{\kappa}_n/\ddot{\kappa}_n|_{\mathbf{p}_0}$;
  else $s = s/2$;
  EndIf
EndWhile
```

4. Projection and optimization

In this section, we will give a novel projection algorithm along a line of curvature based on Newton's method to compensate for the error effects in **Integration**.

4.1. Projection

Suppose $\mathbf{p}_0$ to be a ridge guess on $\mathbf{c}(s)$. The next point $\mathbf{p}$ can be improved with $\triangle s = -\dot{\kappa}_n/\ddot{\kappa}_n|_{\mathbf{p}_0}$ along $\mathbf{c}(s)$ by the Newton–Raphson formula. Since it is impossible to get an explicit representation of $\mathbf{c}(s)$, the difficulty becomes how to link s with $\mathbf{c}(s)$. We propose a numerical method to address this issue.

The basic idea is to approximate the surface-arc length $\triangle s = \widehat{\mathbf{p}_0\mathbf{p}}$ by a polyline. Using a numerical integration procedure to trace $\mathbf{c}(s)$ on a parametric surface (Maekawa et al., 1996) or on an implicit surface (Zhang et al., 2009), we can obtain a sequence of solution points $\mathbf{p}_1, \ldots, \mathbf{p}_m$ on $\widehat{\mathbf{p}_0\mathbf{p}}$ with $\sum_{i=0}^{m-1} |\mathbf{p}_i\mathbf{p}_{i-1}| < |\widehat{\mathbf{p}_0\mathbf{p}}|$ but $\sum_{i=0}^{m} |\mathbf{p}_i\mathbf{p}_{i-1}| > |\widehat{\mathbf{p}_0\mathbf{p}}|$. Basically, we think of $\mathbf{p}_{m-1}$ as an approximation of $\mathbf{p}$. A simple fact is, however, that $|\widehat{\mathbf{p}_i\mathbf{p}_{i-1}}| > |\mathbf{p}_i\mathbf{p}_{i-1}|$. As a consequence, a potential danger arises that $\mathbf{p}_{m-1}$ could be a poor guess of $\mathbf{p}$, e.g. $|\widehat{\mathbf{p}_0\mathbf{p}_{m-1}}| \gg |\widehat{\mathbf{p}_0\mathbf{p}}|$ for the resultant $\mathbf{p}_{m-1}$. To overcome the side effect related to such failures, we take the idea of the Downhill method into consideration by checking $\dot{\kappa}_n$ values of solution points in implementation, as listed in Algorithm 3. There, the Zero Rule is for addressing orientation problems in estimating $\dot{\kappa}_n$ and we will explain it in the following subsection.

The performance of the method relies on the extent of $\sum_{i=0}^{m-1} |\mathbf{p}_i\mathbf{p}_{i-1}|$ approximating $\sum_{i=0}^{m-1} |\widehat{\mathbf{p}_i\mathbf{p}_{i-1}}|$. The closer $\mathbf{p}_0$ is to the target ridge, the closer the projection runs in the line of Newton's method. So our method is actually a globally convergent Newton's method incorporated with the Downhill method.

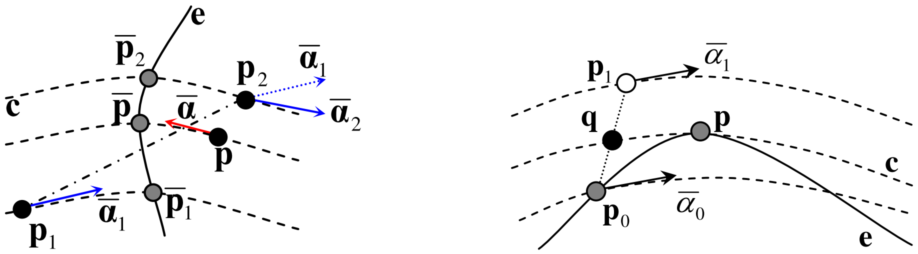
4.2. Orientation problems

A crucial issue of $\dot{\kappa}_n$ is the orientation of principal direction α, since $\dot{\kappa}_n$ will change in sign if the opposite orientation of α is taken. The Acute Rule suffices for an edge guess, but this rule does not suit the new projection since a ridge guess is accessible instead of an edge guess. We observe that the sign of $\dot{\kappa}_n$ is related to the orientation of α but $\ddot{\kappa}_n$ is not, and we are only interested in an orientation in which $\mathbf{c}$ immediately traverses $\mathbf{e}$. Hence our approach is to determine the orientation so that $\mathbf{c}$ crosses $\mathbf{e}$ towards the immediate zero-point of $\dot{\kappa}_n$. This direction satisfies $\dot{\kappa}_n\ddot{\kappa}_n < 0$; if not, the opposite direction is taken. We call this the Zero Rule in contrast to the Acute Rule, as shown in Fig. 3(a).

4.3. Tracing near a turning point

Another potential danger of the new projection arises from solution points approaching a turning point where the projection along $\mathbf{c}$ would not work properly. As illustrated in Fig. 3(b), $\mathbf{p}$ is a turning point of $\mathbf{e}$ and $\mathbf{p}_1$ is a resulting point with improper stepsize in integration at which the projection fails to reach a ridge because $\mathbf{c}$, through $\mathbf{p}_1$, does not traverse $\mathbf{e}$ locally. We have rarely observed such a failure in our numerical experiments. Because it usually covers a tiny distance along $\mathbf{e}$, Runge–Kutta–Fehlberg provides sufficient accuracy to entirely ignore the correction on this occasion, losing nothing in practice.

For the sake of algorithmic completeness, however, we still intend to address this issue by the following tricks. Since the integration would fail at a turning point, we can deal with it as a singular point. If the input surface is polynomial, it can



(a) Orientation of α in estimation of $\dot{\kappa}_n$ (b) Integration near a turning point

Fig. 3. Two technical issues. Because α_1 and α_2 have their specified significance for κ_1 and κ_2 in the context of our work, we use $\bar{\alpha}$ in this figure to stand for the corresponding principal direction of $\mathbf{p}$ (resp. $\mathbf{p}_1$ and $\mathbf{p}_2$) instead. (a) Acute Rule: the angle between $\bar{\alpha}_1$ at $\mathbf{p}_1$ and $\bar{\alpha}_2$ at $\mathbf{p}_2$ is acute; Zero Rule: $\bar{\alpha}$ at $\mathbf{p}$ takes the orientation towards $\bar{\mathbf{p}}$ subject to $\dot{\kappa}_n(\mathbf{p})\ddot{\kappa}_n(\mathbf{p}) < 0$. (b) In the close neighborhood of a turning point $\mathbf{p}$, $\mathbf{p}_1$ is a solution point with respect to $\mathbf{p}_0$. Any point in segment $\mathbf{p}_1\mathbf{q}$ cannot find a ridge of $\mathbf{e}$ along their lines of curvature so the correction projection of $\mathbf{p}_1$ fails. But it is negligible in our framework.

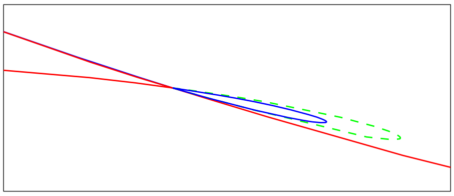


Fig. 4. A ridge loop in blue is with projection; but the dashed in green is without. The desired accuracy is 5×10^{-4} to adjust stepsize in Runge–Kutta–Fehlberg method. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

be exploited like an umbilical point by a solver of a polynomial system (Cazals et al., 2006). If the input surface is generic, it can be handled like a critical point. A turning point is characterized by $\ddot{\kappa}_n = 0$.

4.4. With projection vs without projection

An accurate starting point is crucial for integrating a ridge line. Whether or not a projection in integration is needed to correct a solution point relies on the status of the ridge line in operation. The well-developed technologies of numerical integration, e.g. Runge–Kutta–Fehlberg method employed in this work, usually provide favorable accuracy to get an integral curve in the ridge field even without correction. However, the extraction of a complete ridge line is a procedure of global computation across the surface; thus a simple numerical integration cannot always restrain accumulated round-off error and can cause shift after a long-distance progress. As illustrated in Fig. 4, there is a ridge loop in blue clinging to an umbilical point on a Bézier patch (see also Section 5 and Fig. 5). If the integration is performed without correction projection, a green loop is yielded, significantly different from the real one in blue, or even diverges if a bigger tolerance error bound is taken.

Hence a measure should be taken to correct gradually diverging solution points without desired bounds. Fortunately, we can take precautions against unexpected divergence by the proposed correction projection. On the other hand, since Runge–Kutta–Fehlberg method guarantees high accuracy by adaptively determining an appropriate stepsize, the correction of most solution points can be passed over to make integration more efficient. An alternative is to use a threshold of $\dot{\kappa}_n$ to command its performance and thereby obtaining a quite robust integration with minimum computational effort.

5. Results and analysis

In this section we set out to demonstrate our framework through varied examples of how to accurately extract the ridge graph on a smooth surface. In the following text, ridge lines of κ_1 are drawn in blue, ridge lines of κ_2 are in red, and a black spherical point stands for an umbilical point. Degenerate surfaces with constant principal curvatures, such as spherical and cylindrical surface patches, are not taken into account.

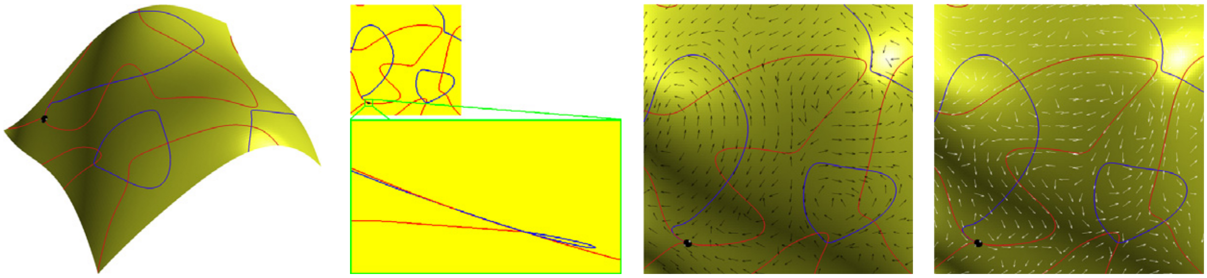
Our framework is implemented in Matlab 7.0, performed on an Intel Core 2 Dual 2.5 GHz processor with 2 GB of RAM. The desired accuracy to control the Runge–Kutta–Fehlberg is 10^{-4}.

5.1. Experiments and analysis

Let us now first consider a Bézier patch defined by the control polyhedron listed in Table 1. In order to calculate its ridge graph, we use a sparse mesh formed by a regular grid of 10×10 as initial in $[0,1] \times [0,1]$. Its ridge graph is illustrated in Fig. 5(a). There is a small ridge loop in the vicinity of an umbilical point. It is easily computed by our method but a mesh of extremely high resolution is required using a zero-following method; see Fig. 5(b) and also refer to Fig. 4. We also display its ridge field together with its ridge graph in Figs. 5(c) and 5(d), respectively, to illuminate their relation.

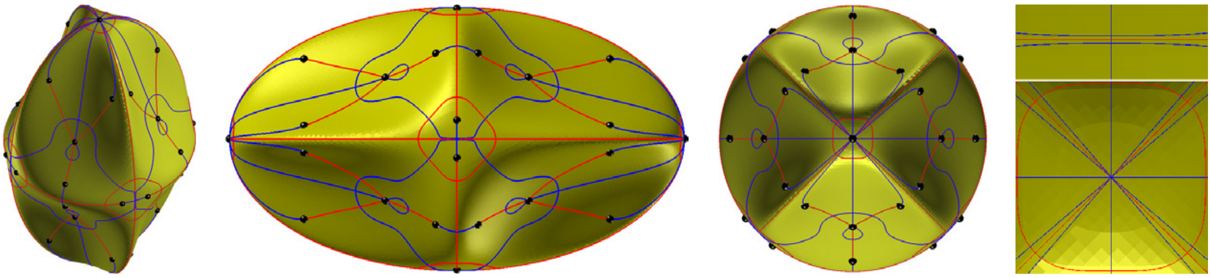
Table 1
Control points of a Bézier surface in the first example.

(0.00, 0.00, 0.25)	(0.25, 0.00, 0.25)	(0.50, 0.00, 0.50)	(0.75, 0.00, 0.50)	(1.00, 0.00, 0.25)
(0.00, 0.25, 0.30)	(0.25, 0.25, 0.70)	(0.50, 0.25, 0.80)	(0.75, 0.25, 1.00)	(1.00, 0.25, 0.50)
(0.00, 0.50, 1.00)	(0.25, 0.50, 0.90)	(0.50, 0.50, 0.10)	(0.75, 0.50, 1.00)	(1.00, 0.50, 0.60)
(0.00, 0.75, 0.70)	(0.25, 0.75, 0.80)	(0.50, 0.75, 1.00)	(0.75, 0.75, 1.00)	(1.00, 0.75, 0.90)
(0.00, 1.00, 0.50)	(0.25, 1.00, 0.30)	(0.50, 1.00, 0.50)	(0.75, 1.00, 0.70)	(1.00, 1.00, 0.50)



(a) 3D view of ridge graph (b) Ridges near an umbilical point (c) κ_1 ridge field (d) κ_2 ridge field.

Fig. 5. Ridge field and ridge graph of a Bézier patch based on an initial mesh of a regular grid 10 × 10.



(a) 3D view (b) Side elevation of ridge graph (c) Planform of ridge graph (d) Critical points

Fig. 6. The ridge graph on a fruit-like surface. (d) indicates the centric regions of (b) and (c) where critical points appear as branching vertices of the ridge graph in the same color.

The second example is an algebraic surface of a fruit-like shape defined by

$$x^2y^2z^2 + 3x^2 + 3y^2 + z^2 - 30 = 0$$

Its ridge graph is exhibited in Figs. 6(a), 6(b) and 6(c) viewed from different angles, respectively. Critical points appear for the first time in this example as illustrated in Fig. 6(d). The centric umbilical point is non-generic, displayed in Fig. 6(c) and at the bottom of Fig. 6(d), which is a *source* or *sink* of twelve-ridge line flows with respect to κ_1 and four with respect to κ_2. The points are difficult to be worked out accurately by other methods.

In Fig. 7(a) is the third example, indicating a stick-handle shape described as

$$x^2 + 2y^2 + \cos(z\pi)\exp(z) - 1 = 0$$

in $[-1.5, 1.5] \times [-1.1, 1.1] \times [-3.5, +\infty)$.

The ridge characteristic of the stick-handle shape is that quite a few critical points appear in the graph.

Fig. 8(a) is a composite Bézier surface of an arm-like shape by 4 × 4 patches, but adjacent ones are joined with at least C^3-continuity. Its traits are that all blue ridge lines are closed and that there are no singular points in its ridge graph, as illuminated with the graph flattened on the parametric plane in Fig. 8(b). Thus, each starting ridge has to come from an edge guess. We note that this graph is actually connected by extremal points. Such an interesting observation suggests that we could inject a singular point unconsidered in this article, i.e. an extremal point, into the ridge graph so as to traverse the ridge graph by one starting ridge alone. An extremal point is governed by $\dot{\kappa}_1^2 + \dot{\kappa}_2^2 = 0$, so we can certainly handle it as a singular point and, as such, more sparse initial mesh is possible. We do not do so because the topologically independent ridges of κ_1 and κ_2 are destroyed into pieces by extremal points as interpreted in this example and additional computing resources will cost in their detection. Readers have the opportunity to take their choice.

Finally, let us consider the piecewise smooth 2-manifold of the Utah teapot model shown in Fig. 9. Its tessellation consists of bi-cubic Bézier patches, with C^1-continuity of the blend between adjacent patches, and not being joined smoothly as

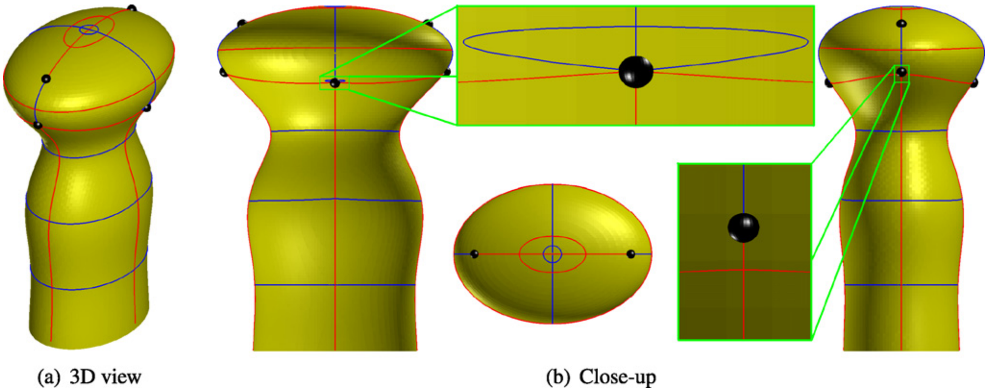


Fig. 7. The ridge graph on a stick-handle-like shape.

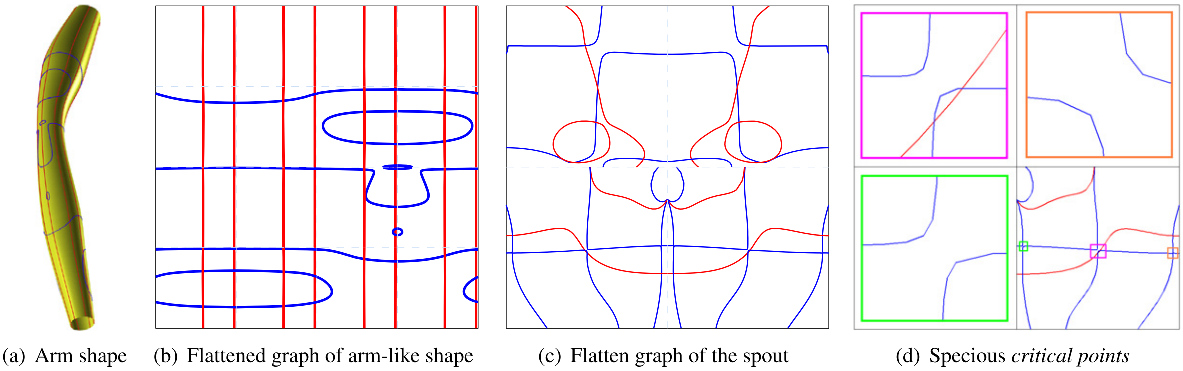

(a) Arm shape
(b) Flattened graph of arm-like shape
(c) Flatten graph of the spout
(d) Specious *critical points*

Fig. 8. The ridge graphs are flattened into the parametric space. In (a) and (b) are the ridge graphs of an arm-like shape on the surface and in the parametric space, respectively. In (c) and (d) are the ridge graphs of the spout of the Utah teapot in the parametric space and close-ups of several specious *critical points*. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

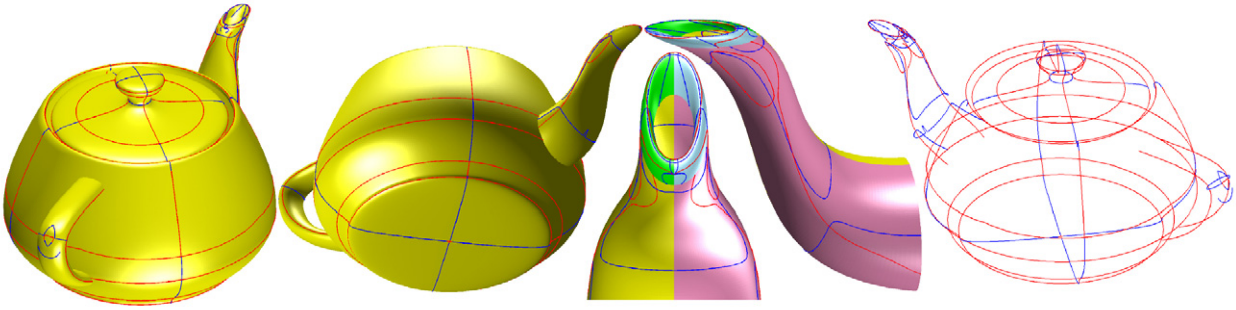


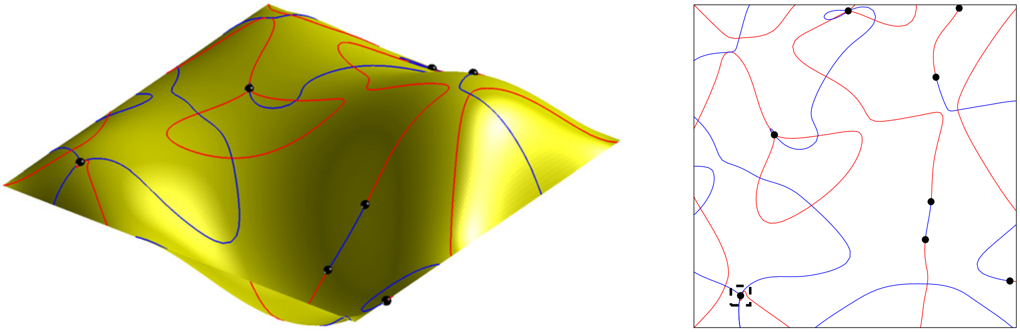
Fig. 9. Ridge graph on the Utah teapot. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

desired. We then examine what will happen for its ridges. A glance at the left two sub-figures gives a pleasing visual. Whereas a ridge is characterized by third-order derivatives, the discontinuity at the joins necessarily ruins the continuity of a ridge line when crossing a joint line. As illustrated with the spout part of the teapot of the 3rd and 4th sub-figures whose ridge patterns are a bit more complicated, ridge lines end at the joins of different patches and are displayed in different colors for the sake of clarity. This side effect is conveyed more clearly by flattening the graph onto the parametric plane as in Fig. 8(c): some ridge lines abruptly stop at the horizontal axis of the domain boundaries. This contrasts with the graph of the arm shape in Fig. 8(b) which has at least C^3-continuity of the blend, whose ridge lines smoothly cross through different patches. The phenomena tell us that ridges are sensitive to high-order continuity of surface and can serve as a measure for this.

It is also worth mentioning that there are four groups of ridge lines along the longitude lines of the teapot body and lip, implying that the body surface could not be a perfect surface of revolution. This proves to be true for it is impossible to represent a perfect circle using Bézier curves.

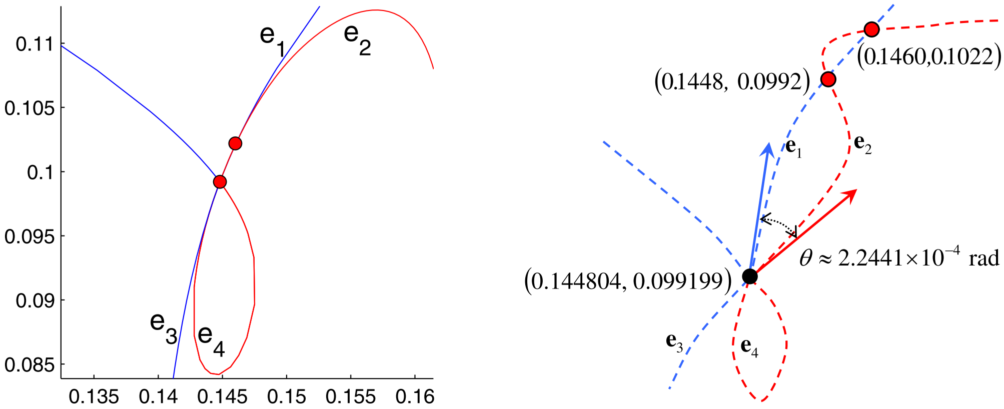
Table 2
Control points of a Bézier surface in the comparison example.

(0.00, 0.00, 0.00)	(0.25, 0.00, 0.00)	(0.50, 0.00, 0.00)	(0.75, 0.00, 0.00)	(1.00, 0.00, 0.00)
(0.00, 0.25, 0.00)	(0.25, 0.25, 1.00)	(0.50, 0.25, −1.0)	(0.75, 0.25, −1.0)	(1.00, 0.25, 0.00)
(0.00, 0.50, 0.00)	(0.25, 0.50, −1.0)	(0.50, 0.50, 1.00)	(0.75, 0.50, 1.00)	(1.00, 0.50, 0.00)
(0.00, 0.75, 0.00)	(0.25, 0.75, 1.00)	(0.50, 0.75, −1.0)	(0.75, 0.75, 1.00)	(1.00, 0.75, 0.00)
(0.00, 1.00, 0.00)	(0.25, 1.00, 0.00)	(0.50, 1.00, 0.00)	(0.75, 1.00, 0.00)	(1.00, 1.00, 0.00)



(a) Ridges on the surface (b) Flattened ridge graph

Fig. 10. The ridge graph on a Bézier patch. Its topology has been known. The local patten of an umbilical point (0.144804, 0.099199) inside the black dashed box is too complicated to observe.



(a) Local ridge graph (b) A readable sketch

Fig. 11. The ridge graph near an umbilical point (0.144804, 0.099199). (a) Ridge lines, $\mathbf{e}_1$ and $\mathbf{e}_2$ (resp. $\mathbf{e}_3$ and $\mathbf{e}_3$), close with each other such that their topology cannot be observed explicitly. (b) A topological sketch for clarity.

We have something more to say about the amusing face-like graph in Fig. 8(c): the specious critical points are actually normal ridges, as illustrated with Fig. 8(d). A generic critical point is determined by three conditions, meaning that it is more special than a generic umbilical point. Thus, it does not happen on a generic surface, as our experiments verify. This is one reason for existing methods to generally turn away from these points.

5.2. Comparison and computational cost

In this subsection, we exhibit the time taken in our method and demonstrate its reliability, in the sense of machine accuracy, on parameterized polynomial surfaces which are popular in CAD applications.

As a comparison, we select a topology-verified example considered by Cazals et al. (2005), whose control polyhedron is listed in Table 2. Its ridge graph is displayed in Fig. 10 and the topology is almost the same to that derived from Cazals et al. (2005). But the ridge pattern near an umbilical point of (0.144804, 0.099199) inside a dashed box is of ambiguity because two lines of $\mathbf{e}_1$ and $\mathbf{e}_2$ (resp. $\mathbf{e}_3$ and $\mathbf{e}_4$) almost coincide when approaching this umbilical point, as demonstrated in Fig. 11(a). To reveal topological patterns of these lines, we increased the desired accuracy of Runge–Kutta–Fehlberg to be 10^{-8} so that $\mathbf{e}_1$ and $\mathbf{e}_2$ (resp. $\mathbf{e}_3$ and $\mathbf{e}_4$) are approximated sufficiently by polylines. Their intersection operations show that $\mathbf{e}_3$ and $\mathbf{e}_4$ separate; $\mathbf{e}_1$ cuts across $\mathbf{e}_2$ two times, at approximately (0.1460, 0.1022) and (0.1448, 0.0992), before the lines arrive at (0.144804, 0.099199). It is best illustrated with the topological sketch in Fig. 11(b), which has been demonstrated

Table 3
Computational cost on smooth Bézier surface patches.

Bézier model	Number of umbilical points	Number of closed ridge lines	Time T_1 with projection (seconds)	Time T_2 without projection (seconds)	Ratio T_1/T_2
Bézier in Fig. 5	1	1	34.5	29.1	1.19
Arm shape (4 × 4)	0	7	243.1	208.1	1.17
Bézier in Fig. 10	8	0	51.6	43.3	1.19

by Cazals et al. (2005). Our further computation by Eq. (10) shows that the tangent vectors of the two lines $\mathbf{e}_1$ and $\mathbf{e}_2$ (resp. $\mathbf{e}_3$ and $\mathbf{e}_4$) travel outside in the angle as small as 2.2441×10^{-4} rad.

We implemented the algorithm in MATLAB with no optimization. The computational time of our experiments on sufficient smooth Bézier surfaces are listed in Table 3. Owing to the high accuracy of Runge–Kutta–Fehlberg, favorable resultant ridge graphs can be achieved even without a correction projection of solution points in integration except in some subtle scenarios such as in Figs. 3(b) and 11. In order to expose the influence of correction projection on the total computational cost, we performed experiments with and without correction to calculate ridge graphs. Our experiments show that the cost of correction projections roughly accounted for 20% of cost by Runge–Kutta–Fehlberg integration.

6. Discussion and conclusion

We have introduced a *ridge field* and, based on this concept, proposed a comprehensive but absolutely numerical scheme for ridge extraction on a smooth 2-manifold surface. The scheme incorporates an automatic correction procedure of solution points in integration via a novel projection, leading to a robust framework to extract the ridge graph in the ridge field. Furthermore, our scheme deals with singular points efficiently, especially critical points which are not described elsewhere in the literature. The capability of self-correction makes our method suitable for extracting a complete and global ridge graph with full topology without the aid of computer algebra; in particular subtle ridge structures which are difficult to be picked up by other existing methods. An interesting application of our work in CAGD and CG communities is to develop a robust algorithm for computing a feature-aligned polygonal approximation adapting to the geometry of an implicit surface. If ridge graph is considered in polygonization, high-quality mesh can be obtained.

We mainly focus on ridges from a general standpoint and therefore restrict computation within quantities described only by coefficients up to the fifth order in Monge form. Although we have tried to give a comprehensive scheme, an all-purpose proposal consideration still goes beyond the scope of this paper. The proposed framework is basic, and also open to further significant improvement. For instance, a generic critical point could be used as a starting point like an umbilical point if a higher-order analysis is performed on the point. Those potential improvements, we think, are technical and special measures can be taken in diverse applications. A drawback of our scheme is that we have resorted to a zero-crossing edge for a starting ridge. Although singular points can be found globally by a solver for polynomial systems for a polynomial surface, we still cannot find another way to compute a starting ridge on a generic surface rather than by a zero-crossing edge.

A ridge line is a geometrical feature line. Our experiments have shown that ridges on a smooth manifold surface can naturally yield its segmentation; unfortunately, they are high-order quantities and such a segmentation purpose cannot be realized if the underlying surface is less smooth than C^3-continuity, while a real world model is usually C^1-discontinuity mesh or a C^3-discontinuity parametric surface. Nevertheless, no work has been reported, to the best of our knowledge, that bridges the gap between the discontinuity of a real model and the high-order continuity of ridges.

The paper has laid the mathematical foundation of computing ridges on a smooth surface by the vector field. We plan to extend the proposed method to a discrete surface in our further work.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments.

This work is supported in part by National Natural Science Foundation of China under Grant Nos. 60970093, 60902078 and 60872120. WuJun Che was also funded jointly by French–Chinese Foundation for Sciences and Their Applications (FFCSA), and Chinese Government Scholarship of China Scholarship Council (CSC).

Appendix A. Practical formulae

There are two common forms to represent a geometric object mathematically, i.e. parametric and implicit. In this section we describe how to compute the concerned quantities in these two forms.

A.1. Monge form

The coefficient representations in Eq. (2) on a parametric surface or an implicit surface are proposed by Maekawa et al. (1996) and Zhang et al. (2009), respectively. However, only cubic coefficients are given. In this subsection, we will list these representations in a more systemic manner, as well as the quartic coefficients.

For this purpose, we assume the local reference is xyz originating at $\mathbf{p}_0$, and the corresponding unit axis vectors are $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$.

A.1.1. Parametric surface

The cubic coefficients are obtained by Taylor expanding the z-component of the surface (Maekawa et al., 1996):

$$b_0 = z'''\mathbf{u}_x^3 + 3z''\mathbf{u}_x\mathbf{u}_{xx} + z'\mathbf{u}_{xxx}, \qquad b_1 = z'''\mathbf{u}_x^2\mathbf{u}_y + z''\mathbf{u}_{xx}\mathbf{u}_y + 2z''\mathbf{u}_x\mathbf{u}_{xy} + z'\mathbf{u}_{xxy}$$
$$b_3 = z'''\mathbf{u}_y^3 + 3z''\mathbf{u}_y\mathbf{u}_{yy} + z'\mathbf{u}_{yyy}, \quad b_2 = z'''\mathbf{u}_x\mathbf{u}_y^2 + z''\mathbf{u}_x\mathbf{u}_{yy} + 2z''\mathbf{u}_y\mathbf{u}_{xy} + z'\mathbf{u}_{xyy}$$

Here $\mathbf{u}$ is $(u, v)^T$, as interpreted in Eq. (6).

The first two coefficients of the quartic are derived as follows:

$$c_0 = z^{(4)}\mathbf{u}_x^4 + 6z'''\mathbf{u}_x^2\mathbf{u}_{xx} + 4z''\mathbf{u}_x\mathbf{u}_{xxx} + 3z''\mathbf{u}_{xx}^2 + z'\mathbf{u}_{xxxx}$$
$$c_1 = z^{(4)}\mathbf{u}_x^3\mathbf{u}_y + 3z'''\left(\mathbf{u}_x^2\mathbf{u}_{xy} + \mathbf{u}_x\mathbf{u}_y\mathbf{u}_{xx}\right) + z''(3\mathbf{u}_x\mathbf{u}_{xxy} + \mathbf{u}_y\mathbf{u}_{xxx} + 3\mathbf{u}_{xx}\mathbf{u}_{xy}) + z'\mathbf{u}_{xxxy}$$

where $\mathbf{u}_x = \frac{1}{\frac{\partial(x,y)}{\partial(u,v)}}\binom{y_v}{-y_u}$, $\mathbf{u}_y = \frac{1}{\frac{\partial(x,y)}{\partial(u,v)}}\binom{-x_v}{x_u}$. Its higher-order derivatives of x and y, such as $\mathbf{u}_{xy}$, can be evaluated using the chain rule.

A.1.2. Implicit surface

A general implicit surface can be defined as a zero-level set of a scalar-valued mapping from three-dimensional uvw-space to $\mathbb{R}$:

$$H : \mathbb{R}^3 \rightarrowtail \mathbb{R}; \qquad \mathbf{p} = (u, v, w)^T \mapsto H(\mathbf{p})$$

The cubic coefficients are the following (Zhang et al., 2009):

$$b_0 = -\left.\frac{H'''\mathbf{x}^3 + 3\kappa_1 H''\mathbf{xz}}{H'\mathbf{z}}\right|_{\mathbf{p}_0}, \qquad b_1 = -\left.\frac{H'''\mathbf{x}^2\mathbf{y} + \kappa_1 H''\mathbf{yz}}{H'\mathbf{z}}\right|_{\mathbf{p}_0}$$
$$b_2 = -\left.\frac{H'''\mathbf{xy}^2 + \kappa_2 H''\mathbf{xz}}{H'\mathbf{z}}\right|_{\mathbf{p}_0}, \qquad b_3 = -\left.\frac{H'''\mathbf{y}^3 + 3\kappa_2 H''\mathbf{yz}}{H'\mathbf{z}}\right|_{\mathbf{p}_0}$$

The first two quartic coefficients are

$$c_0 = -\left.\frac{H^{(4)}\mathbf{x}^4 + 6\kappa_1 H'''\mathbf{x}^2\mathbf{z} + 3\kappa_1^2 H''\mathbf{z}^2 + 4b_0 H''\mathbf{xz}}{H'\mathbf{z}}\right|_{\mathbf{p}_0}$$
$$c_1 = -\left.\frac{H^{(4)}\mathbf{x}^3\mathbf{y} + 3\kappa_1 H'''\mathbf{xyz} + b_0 H''\mathbf{yz} + 3b_1 H''\mathbf{xz}}{H'\mathbf{z}}\right|_{\mathbf{p}_0}$$

A.2. $\dot{\kappa}_n$ and $\ddot{\kappa}_n$

A.2.1. Parametric surface

Most formulae expressing geometric quantities in terms of the derivatives of a parametric $\mathbf{r}$ are standard and can be found in textbooks on differential geometry. Here we will list these formulae from the viewpoint of computation.

The first and second fundamental forms are important geometric structures defined by the surface. They can be written on $\mathbb{R}^2$ as

$$I'' = \mathbf{r}' \cdot \mathbf{r}', \qquad II'' = \mathbf{r}'' \cdot \mathbf{n} \tag{11}$$

and their matrices are $\begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} \mathbf{r}_u\cdot\mathbf{r}_u & \mathbf{r}_u\cdot\mathbf{r}_v \\ \mathbf{r}_u\cdot\mathbf{r}_v & \mathbf{r}_v\cdot\mathbf{r}_v \end{pmatrix}$ and $\begin{pmatrix} L & M \\ M & N \end{pmatrix} = \begin{pmatrix} \mathbf{n}\cdot\mathbf{r}_{uu} & \mathbf{n}\cdot\mathbf{r}_{uv} \\ \mathbf{n}\cdot\mathbf{r}_{uv} & \mathbf{n}\cdot\mathbf{r}_{vv} \end{pmatrix}$, respectively.

Given $\kappa_G = (LN - M^2)/(EG - F^2)$, $\kappa_M = (LG - 2MF + NE)/2(EG - F^2)$ and $\alpha = \dot{\mathbf{u}}$ along $\mathbf{c}(s)$, we get

$$k_n = \kappa_M \pm \sqrt{\kappa_M^2 - \kappa_G}, \qquad \dot{k}_n = k_n'\alpha, \qquad \ddot{k}_n = k_n''\alpha^2 + k_n'\dot{\alpha}$$

We note that α is the system of differential equations governing lines of curvature

$$\alpha = \eta \begin{pmatrix} M - \kappa_n F \\ -L + \kappa_n E \end{pmatrix} \quad \text{or} \quad \alpha = \mu \begin{pmatrix} N - \kappa_n G \\ -M + \kappa_n F \end{pmatrix}$$

where $\eta = \frac{\pm 1}{\sqrt{E(M-\kappa_n F)^2 - 2F(M-\kappa_n F)(L-\kappa_n E) + G(L-\kappa_n E)^2}}$ and $\mu = \frac{\pm 1}{\sqrt{E(N-\kappa_n G)^2 - 2F(N-\kappa_n G)(M-\kappa_n F) + G(M-\kappa_n F)^2}}$. The signs of η and μ determine the directions in which s increases.

A.2.2. Implicit surface

A concise representation of the principal curvature is (Monga and Benayoun, 1995):

$$\kappa_n = -\frac{H''\alpha^2}{H'\mathbf{n}} \tag{12}$$

and the well-known formula for $\dot{\kappa}_n$ is Eq. (1).

According to the Frenet–Serret formulas and Rodrigues's Theorem in classical differential geometry, we have $\dot{\alpha} = \kappa\beta$ and $\dot{\mathbf{n}} = -\kappa_n\alpha$, where κ is the curvature of $\mathbf{c}$ as a space curve and β the unit (principal) normal vector.

The calculation with Eq. (1) leads to

$$\ddot{\kappa}_n = -\big(H^{(4)}\alpha^4 + 3\kappa H'''\alpha^2\beta + 3\kappa_n H'''\alpha^2\mathbf{n} + 4\dot{\kappa}_n H''\alpha\mathbf{n} + 3\kappa_n\kappa H''\beta\mathbf{n}\big)/H'\mathbf{n} - 3\kappa_n^3 \tag{13}$$

The computational difficulty of Eq. (13) rests with computability of κ and β, which are relevant to the differential quantities related to $\mathbf{c}(s)$. Both of κ and β can be computed as (Che et al., 2007):

$$\kappa = \frac{|\alpha \times \nabla\mathbf{T}\alpha|}{|\mathbf{T}|}, \qquad \begin{cases} \alpha = \frac{\mathbf{T}}{|\mathbf{T}|} \\ \beta = \gamma \times \alpha \\ \gamma = \frac{\alpha \times \nabla\mathbf{T}\alpha}{|\alpha \times \nabla\mathbf{T}\alpha|} \end{cases}$$

where $\mathbf{T}$ is the principal direction along $\mathbf{c}(s)$, α is the unit tangent vector, β is the unit (principal) normal vector, and γ is the unit binormal vector.

References

Belyaev, A.G. Pasko, A.A. Kunii, T.L., 1998. Ridges and ravines on implicit surfaces. In: Computer Graphics International, pp. 530–535.

Bogaevski, I.A. Lang, V. Belyaev, A.G. Kunii, T.L., 2003. Color ridges on implicit polynomial surfaces. In: Proceedings of GraphiCon, pp. 161–164.

Cazals, F., Pouget, M., 2004. Ridges and umbilics of a sampled smooth surface: a complete picture gearing toward topological coherence. INRIA, RR-5294.

Cazals, F., Pouget, M., 2005a. Differential topology and geometry of smooth embedded surfaces: selected topics. International Journal of Computational Geometry & Applications 15 (5), 511–536.

Cazals, F., Pouget, M., 2005b. Topology driven algorithms for ridge extraction on meshes. INRIA, RR-5526.

Cazals, F., Faugère, J.-C., Pouget, M., Rouillier, F., 2005. Topologically certified approximation of umbilics and ridges on polynomial parametric surface. INRIA, RR-5674.

Cazals, F., Faugère, J.-C., Pouget, M., Rouillier, F., 2006. The implicit structure of ridges of a smooth parametric surface. Computer Aided Geometric Design 23 (7), 582–598.

Che, W.J., Paul, J.-C., Zhang, X.P., 2007. Lines of curvature and umbilical points for implicit surfaces. Computer Aided Geometric Design 24 (7), 395–409.

Clémençon, B., Borouchaki, H., Laug, P., 2008. Ridge extraction and its application to surface meshing. Engineering with Computers 24 (3), 287–304.

Eberly, D., Gardner, R., Morse, B., Pizer, S., Scharlach, C., 1994. Ridges for image analysis. Journal of Mathematical Imaging and Vision 4 (4), 353–373.

Gravesen, J., 2005. Third order invariants of surfaces. In: Dokken, T., Juttler, B. (Eds.), Computational Methods for Algebraic Spline Surfaces. Springer.

Gueziec, A., 1993. Large deformable splines, crest lines and matching. In: Proceedings of Fourth International Conference on Computer Vision, pp. 650–657.

Hallinan, P.W., Gordon, G., Yuille, A.L., Giblin, P., Mumford, D., 1999. Two- and Three-Dimensional Patterns of the Face. A.K. Peters.

Hildebrandt, K., Polthier, K., Wardetzky, M., 2005. Smooth feature lines on surface meshes. In: Eurographics Symposium on Geometry Processing.

Hisada, M., Belyaev, A.G., Kunii, T.L., 2002. A skeleton-based approach for detection of perceptually salient features on polygonal surfaces. Computer Graphics Forum 21 (4), 689–700.

Hosaka, M., 1992. Modeling of Curves and Surfaces in CAD/CAM. Springer, Berlin.

Kent, J.T., Mardia, K.V., West, J.M., 1996. Ridge curves and shape analysis. In: The British Machine Vision Conference, pp. 43–52.

Koenderink, J., 1990. Solid Shape. MIT Press.

Maekawa, T., Wolter, F.-E., Patrikalakis, N.M., 1996. Umbilics and lines of curvature for shape interrogation. Computer Aided Geometric Design 13 (2), 133–161.

Monga, O., Benayoun, S., 1995. Using partial derivatives of 3D images to extract typical surface features. Computer Vision and Image Understanding 61 (2), 171–189.

Morris, R., 1996. The sub-parabolic lines of a surface. In: Mullineux, G. (Ed.), Proceedings of the 6th IMA Conference on the Mathematics of Surfaces, 1994. Mathematics of Surfaces VI. In: IMA New Series, vol. 58. Oxford University Press, pp. 79–102.

Musuvathy, S., Cohen, E., 2009. Extracting principal curvature ridges from B-spline surfaces with deficient smoothness. In: Proceedings of 5th International Symposium on Visual Computing Advances in Visual Computing. In: Lecture Notes in Computer Science, vol. 5875. Springer.

Musuvathy, S., Cohen, E., Seong, J.-K., Damon, J., 2009. Tracing ridges on B-spline surfaces. In: SIAM/ACM Joint Conference on Geometric and Physical Modeling, pp. 55–66.

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. Computer Journal 7, 308–313.

Ohtake, Y., Belyaev, A.G., Seidel, H.-P., 2004. Ridge-valley lines on meshes via implicit surface fitting. In: Proceedings of ACM SIGGRAPH, pp. 609–612.

Patrikalakis, N.M., Maekawa, T., 2002. Shape Interrogation for Computer Aided Design and Manufacturing. Springer-Verlag.

Porteous, I.R., 2001. Geometric Differentiation: For the Intelligence of Curves and Surfaces, 2nd edition. Cambridge University Press.

Stylianou, G., Farin, G., 2004. Crest lines for surface segmentation and flattening. IEEE Transactions on Visualization and Computer Graphics 10 (5), 536–544.

Thirion, J.-P., 1996a. New feature points based on geometric invariants for 3D image registration. International Journal of Computer Vision 18 (2), 121–137.

Thirion, J.-P., 1996b. The extremal mesh and the understanding of 3D surfaces. International Journal of Computer Vision 19 (2), 115–128.

Thirion, J.-P., Gourdon, A., 1996. The 3D marching lines algorithm. Graphical Models and Image Processing 58 (6), 503–509.

Watanabe, K., Belyaev, A.G., 2001. Detection of salient curvature features on polygonal surfaces. Computer Graphics Forum 20 (3), 385–392.

Yoshizawa, S., Belyaev, A.G., Yokota, H., Seidel, H.-P., 2008. Fast, robust, and faithful methods for detecting crest lines on meshes. Computer Aided Geometric Design 25 (8), 545–560.

Zhang, X.P., Che, W.J., Paul, J.-C., 2009. Computing lines of curvature on implicit surfaces. Computer Aided Geometric Design 26 (9), 923–940.