



HAL
open science

Shape Interrogation

Stefanie Hahmann, Alexander Belyaev, Laurent Busé, Gershon Elber, Bernard Murrain, Christian Roessl

► **To cite this version:**

Stefanie Hahmann, Alexander Belyaev, Laurent Busé, Gershon Elber, Bernard Murrain, et al.. Shape Interrogation. Leila De Floriani, Michela Spagnuolo. Shape Analysis and Structuring, Springer, pp.1-51, 2008, Mathematics and Visualization, 978-3-540-33265-7. <10.1007/978-3-540-33265-7_1>. <inria-00193551>

HAL Id: inria-00193551

<https://inria.hal.science/inria-00193551v1>

Submitted on 7 Dec 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Shape Interrogation

Stefanie Hahmann¹, Alexander Belyaev², Laurent Busé³, Gershon Elber⁴,
Bernard Mourrain³, and Christian Rössl³

¹ Laboratoire Jean Kuntzmann, Institut National Polytechnique de Grenoble,
France

`Stefanie.Hahmann@imag.fr`

² MPII, Max Planck Institut für Informatik, Saarbrücken, Germany

`belyaev@mpi-sb.mpg.de`

³ INRIA Sophia-Antipolis, France

`lbuse@sophia.inria.fr`, `Bernard.Mourrain@sophia.inria.fr`,
`Christian.Roessler@sophia.inria.fr`

⁴ Technion - Israel Institute of Technology, Haifa 32000, Israel

`gershon@cs.technion.ac.il`

Summary. Shape interrogation methods are of increasing interest in geometric modeling as well as in computer graphics. Originating 20 years ago from CAD/CAM applications where "class A" surfaces are required and no surface imperfections are allowed, shape interrogation has become recently an important tool for various other types of surface representations such as triangulated or polygonal surfaces, subdivision surface, and algebraic surfaces. In this paper we present the state-of-the-art of shape interrogation methods including methods for detecting surface imperfections, surface analysis tools and methods for visualizing intrinsic surface properties. Furthermore we focus on stable numerical and symbolic solving of algebraic systems of equations, a problem that arises in most shape interrogation methods.

1 Introduction

Shape interrogation is the process of extraction of information from a geometric model. Surface interrogation is of central importance in modern Computer Graphics and Computer Aided Design (CAD) systems. Wherever geometrical models are used, they often need to be analyzed with respect to different aspects like, for example, visual pleasantness, technical smoothness, geometric constraints or surface intrinsic properties. The various methods, which are presented in this survey can be used to detect surface imperfections, to analyze shapes or to visualize different forms. We not only restrict the shapes to be investigated to free-form surfaces, but include polygonal meshes as well as algebraic surfaces. Artefacts of subdivision surfaces are subject of Chapter 4 of this book [23]. Particular attention is paid to stable numerical and symbolic

solving of algebraic systems of equations, a problem that arises in most shape interrogation methods.

In Section 2, fundamental notions of differential geometry are briefly recalled. Interrogation methods for polygonal meshes are discussed in Section 3. First and second order shape interrogation and visualization techniques are discussed in Sections 4, 5, focusing mainly on free-form curves and surfaces. The computation and visualization of characteristic curves on surfaces is subject of Section 6. Section 7 discusses the use of robust symbolic computation methods for shape interrogation. Interrogation of algebraic curves and surfaces is finally discussed in Section 8, in particular the transversal problem of solving of algebraic systems of equations is described.

2 Differential Geometry of curves and surfaces

Fundamental notions of differential geometry of curves and surfaces that are needed in the following of the paper will briefly be reviewed in this section. For a complete bibliography on differential geometry the reader is referred to standard literature [39, 102, 79, 176].

2.1 Curves

A **parametric curve** is a mapping x from $I = [a, b] \subset \mathbb{R}$ into \mathbb{R}^n of class C^r ($r \geq 1$). x is called **regular**, if $\frac{dx}{dt}(t) \neq 0$ for all $t \in I$. If L is the length of $x([a, b])$, there exists a unique parameter transformation s from I into $[0, L]$ such that for all $t_0, t_1 \in [a, b]$ the length of the arc $x([t_0, t_1])$ is equal to $s(t_1) - s(t_0)$. For all $t \in [a, b]$ $s(t) = \int_a^t \|\frac{dx}{dt}\| dt$. s is called the **arc length parameterization**. It is a geometric invariant of a curve and is therefore also called **natural parameterization**.

Let $x : [0, L] \rightarrow \mathbb{R}^3$, $s \mapsto x(s)$ be a regular and naturally parameterized curve of class C^3 , such that $\|x''(s)\| \neq 0$ for all $s \in]0, L[$, then

- $v_1(s) := x'(s)$ is called **tangent vector** of x in s .
- $v_2(s) := \frac{x''(s)}{\|x''(s)\|}$ is called **unit normal vector** of x in s .
- $v_3(s) := v_1(s) \times v_2(s)$ is called **binormal vector** of x in s ,

where \times denotes the vector product (cross product) in \mathbb{R}^3 . $\{v_1(s), v_2(s), v_3(s)\}$ form an orthonormal basis of \mathbb{R}^3 called the **Frenet frame** of x in s .

The following holds: v_1, v_2, v_3 are mappings of class C^1 , and

$$\begin{aligned} v_1' &= \kappa_1 v_2 \\ v_2' &= -\kappa_1 v_1 + \kappa_2 v_3 \\ v_3' &= -\kappa_2 v_2 \end{aligned}$$

where

$$\kappa(s) = \|x''\|, \quad \tau(s) = \frac{|x', x'', x''|}{\|x''\|}$$

are mappings of class C^1 and C^0 respectively. $|\cdot, \cdot, \cdot|$ denotes the determinant of the matrix formed by the three vector arguments. κ and τ are called **curvature** and **torsion** of the curve x . The curvature measures the deviation of a curve from a straight line, and the torsion measures the deviation of a curve from being planar.

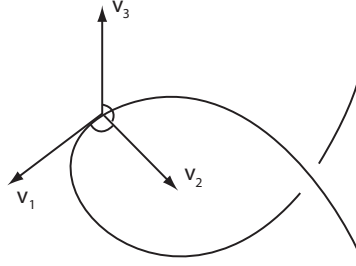


Fig. 1. Frenet frame.

2.2 Surfaces

A **parametric surface** is a mapping X from $\Omega \subset \mathbb{R}^2$ into \mathbb{R}^3 of class C^r ($r \geq 1$). X is called **regular** if for all $\mathbf{u} = (u, v) \in \Omega$, $dX_{\mathbf{u}}$ is an invertible linear mapping. The two partial derivatives of X in \mathbf{u} are denoted by $X_u(\mathbf{u})$ and $X_v(\mathbf{u})$. The affine subspace $T_{\mathbf{u}}X := \{X(\mathbf{u}) + \lambda X_u(\mathbf{u}) + \mu X_v(\mathbf{u}) \mid (\lambda, \mu) \in \mathbb{R}^2\}$ is called **tangent plane** to X in \mathbf{u} .

The **unit normal vector field** N is given by

$$N := \frac{X_u \times X_v}{\|X_u \times X_v\|}.$$

The moving frame $\{X_u, X_v, N\}$ is the **Gauss frame**. The Gauss frame is in general not an orthogonal frame.

The bilinear form on $T_{\mathbf{u}}X$ induced by the inner product of \mathbb{R}^3 is called the first fundamental form of the surface. The matrix representation of the first fundamental form $I_{\mathbf{u}}$ with respect to the basis $\{X_u, X_v\}$ of $T_{\mathbf{u}}X$ is given by $G = (g_{ij})$ with $i, j = 1, 2$:

$$\begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} = \begin{pmatrix} \langle X_u, X_u \rangle & \langle X_u, X_v \rangle \\ \langle X_v, X_u \rangle & \langle X_v, X_v \rangle \end{pmatrix}$$

where \langle, \rangle denotes the scalar product. The first fundamental form $I_{\mathbf{u}}$ is symmetric, positive definite and geometrically invariant. The first fundamental

form allows measurements on the surface (length of curves, angles of tangent vectors, areas of regions) without referring back to the space \mathbb{R}^3 , in which the surface lies.

The linear mapping $L_{\mathbf{u}}$

$$L_{\mathbf{u}} : T_{\mathbf{u}}X \rightarrow T_{\mathbf{u}}X \\ x \mapsto dN_{\mathbf{u}} \circ dX_{\mathbf{u}}^{-1}(x)$$

is called the **Weingarten map**.

The bilinear symmetric form $II_{\mathbf{u}}$ defined on $T_{\mathbf{u}}X$ by

$$II_{\mathbf{u}}(x, y) = \langle L_{\mathbf{u}}(x), y \rangle$$

is called the **second fundamental form** of the surface X .

Its matrix in the basis $\{X_u, X_v\}$ of $T_{\mathbf{u}}X$ is denoted $H = (h_{ij})$ with $i, j = 1, 2$:

$$\begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} = \begin{pmatrix} \langle N, X_{uu} \rangle & \langle N, X_{uv} \rangle \\ \langle N, X_{vu} \rangle & \langle N, X_{vv} \rangle \end{pmatrix}.$$

The matrix HG^{-1} of the Weingarten map $L_{\mathbf{u}}$ is symmetric and real and therefore it has two real eigenvalues κ_1, κ_2 with corresponding orthogonal eigenvectors. κ_1, κ_2 are called **principle curvatures** of the surface X , also labeled as $\kappa_{max}, \kappa_{min}$. The product of the principle curvatures $K = \kappa_1 \cdot \kappa_2 = \det(L_{\mathbf{u}}) = \frac{\det(H)}{\det(G)}$ is called the **Gaussian curvature** and its mean $M = \frac{1}{2}(\kappa_1 + \kappa_2) = \text{trace}(L_{\mathbf{u}})$ is called the **mean curvature**.

Another approach for the principle curvatures is the following: Let $A := \Delta u \cdot X_u + \Delta v \cdot X_v$ be a tangent vector with $\|A\| = 1$. If we intersect the surface with the plane given by N and A , we get an intersection curve y with the following properties:

$$\dot{y}(s) = A \quad \text{and} \quad e_2 = \pm N$$

where e_2 is the principal normal vector of the space curve y . The implicit function theorem implies the existence of this normal section curve. To calculate the extreme values of the curvature of a normal section curve (the normal section curvature) we can use the method of Lagrange multipliers because we are looking for the extreme values of the normal section curvature κ_N with the condition $\|\dot{y}(s)\| = 1$.

As a result of these considerations we obtain the following. Unless the normal section curvature is the same for all directions there are two perpendicular directions A_1 and A_2 in which κ_N attains its absolute maximum and its absolute minimum values. These directions are the *principal directions* with the corresponding normal section curvatures κ_1 and κ_2 .

For $A = A_1 \cos \varphi + A_2 \sin \varphi$ we get Euler's formula:

$$\kappa_N = \kappa_1 \cos^2 \varphi + \kappa_2 \sin^2 \varphi,$$

If the principal directions are taken as coordinate axes, Euler's formula implies the so-called **Dupin indicatrix**:

$$\kappa_1(u)^2 + \kappa_2(u)^2 = \pm 1. \quad (1)$$

We use the Dupin indicatrices as a tool to visualize curvature situations on surfaces. The Dupin indicatrices at elliptic points ($K > 0$) are ellipses, at hyperbolic points ($K < 0$) pairs of hyperbolas, and at parabolic points ($K = 0$) pairs of parallel lines. **Flat points** ($\kappa_1 = \kappa_2 = 0$) are degenerated parabolic cases. Points with $\kappa_1 = \kappa_2$ are called **umbilical points**.

3 Interrogation of discrete shapes

Polygonal meshes constitute the primary tool for 3D surface representation and are frequently used in a wide range of scientific applications, including computer graphics, visualization, and numerical simulations. Two fundamental questions of surface approximation by polygonal meshes concern approximation quality (accuracy) [60] and the relation between the accuracy and size of the approximation [61]. Recently both of these questions were also addressed in [29] where a variational approach for surface approximation by polygonal meshes was developed. Shape approximation with polygonal meshes is discussed in more detail in Chapter 2 of this book [1].

Accurate estimation of geometric properties of a surface from its discrete approximation is important for many applications. Nevertheless there is no consensus on how to achieve accurate estimations of simple surface attributes such as the normal vector and curvatures [122]. An accurate polygonal approximation of surface geometry in a least-squares sense [60, 29] does not guarantee accurate approximations of surface normals and curvatures by their discrete counterparts [121, 132, 119, 14]. Thus, deriving accurate, consistent, and numerically robust estimates for the surface normal vector and curvature tensor remains an area of active and creative research today.

3.1 Surface Normal Estimation

Given a smooth surface approximated by a dense triangle mesh, an accurate and robust estimation of vertex normals is important for a number of tasks including smooth shading [66, 156], curvature estimation (see, e.g., [180]), and feature extraction (see, e.g., [87]).

Usually the normal vector at a vertex of a triangle mesh is estimated as the normalized weighted sum of normals of the incident facets (triangles). A survey of various methods to estimate the normal vector can be found in [174]. Uniform (equal) weights are justified in [63] via finite difference approximations. In [180] the weights are chosen to be equal to the areas of the

incident triangles. Weighting by the inverse areas was considered in [174, 87], and weights equal to the facet angles at the vertex are proposed in [185]. A weighting scheme assuming that the mesh locally approximates a sphere was developed in [120]. The vertex normal vector can be also obtained from the mean curvature vector and, therefore, mean curvature vector estimates proposed in [37, 122] lead to approximations of the vertex normals. A standard approach for testing and comparing various methods to estimate surface normals and curvatures consists of tessellating known (analytical) surfaces and comparing the estimates from the resulting mesh and from the original surface [73, 180, 104, 32, 122]. An interesting statistical approach was recently proposed in [125, 126]. First steps towards a rigorous mathematical analysis and comprehensive comparison of various weighting schemes are made in [106].

3.2 Curvature Tensor Estimation

Estimates of the curvature tensor on polygonal meshes are applied in a variety of applications ranging from the detection of surface defects to the detection of features. Many techniques have been proposed (see, e.g., [153] for a recent survey), in this section we provide an overview of different approaches.

In order to estimate the curvature tensor at a vertex a certain neighborhood of this vertex is considered, typically its 1-ring. A common approach is to first discretize the normal curvature along edges. Given is an edge (i, j) , vertex positions X_i, X_j , and the normal N_i , then

$$\kappa_{ij} = 2 \frac{\langle (X_j - X_i), N_i \rangle}{\|X_j - X_i\|^2} \quad (2)$$

provides an approximation of the normal curvature at X_i in the tangent direction which results from projecting X_i and X_j into the tangent plane defined by N_i . This expression can be interpreted geometrically as fitting the osculating circle interpolating X_i and X_j with normal N_i at X_i (cf. [130]). Alternatively, the equation can be derived from discretizing the curvature of a smooth planar curve (cf. [180]). With estimates κ_{ij} of the normal curvature for all edges incident to vertex i , Euler's formula can be applied to relate the κ_{ij} to the unknown principal curvatures (and principal directions). Then approximates to the principal curvatures can be obtained either directly as functions of the eigenvalues of a symmetric matrix ([180, 147]) or from solving a least-squares problem ([130, 122]). Alternatively, the trapezoid rule is applied in [188] to get a discrete approximation of the mean curvature M expressed as the integral over the normal curvatures κ_N , the Gaussian curvature K is obtained from a similar integral over κ_N^2 , then M and K define the principal curvatures. Exact quadrature formulas for curvature estimation are provided in [107].

Another class of techniques for curvature tensor estimation locally fits a smooth parametric surface patch and then derives the differential quantities from that. This leaves the choice for the surface – typically polynomials of

low degree – the geometric quantities to interpolate or approximate – e.g., the vertex positions in a 1-ring neighborhood – and a projection operator to obtain a parameterization – in general the projection into the tangent plane. A straightforward choice is to consider the quadratic height surface

$$z(x, y) = \frac{1}{2}a_{20}x^2 + a_{11}xy + \frac{1}{2}a_{02}y^2 ,$$

for a local coordinate system spanned by the normal N_i (in z -direction) and two orthogonal tangent vectors (in x - and y -direction) and with origin $X_i = 0$ [64]. Then the parameters a_{20} , a_{11} , and a_{02} obtained as a least-squares solution are the elements of the symmetric matrix defining the Weingarten map. This can be interpreted as estimating the normal curvature from parabolas rather than circles (as with (2)) and then solving a least-squares system like in [122].

In [189] a quadratic Taylor polynomial of different form is applied, namely

$$X(u, v) = X_u u + X_v v + \frac{1}{2}u^2 X_{uu} + X_{uv} uv + \frac{1}{2}v^2 X_{vv} .$$

The coefficients of the local least-squares approximating polynomial are the first and second order partials and hence define the fundamental forms. For robustness reasons, an exponential map is used as projection operator rather than a simple projection to the tangent plane.

The use of a cubic approximation scheme which takes into account vertex normals in the 1-ring is proposed in [64]. As the normals themselves are local estimates, this effectively enlarges the neighborhood. Again, a least-squares problem is solved to find the coefficients of a cubic height surface, where the Weingarten matrix is obtained entirely from the quadratic terms in the same way as before.

In general, least-squares methods may suffer from degenerate cases – even for reasonable geometric configurations – which lead to ill-conditioned system matrices. In [189] the polynomial basis is successively reduced in such cases. An alternative is to provide more samples e.g. from linear interpolation. In [24] the patch fitting approach is discussed from an approximation theory point of view including robustness and numerical issues. For high-quality and consistent estimation of curvatures and their derivatives, [145] applies a (rather expensive) global fitting of an implicit surface to the surface mesh.

In contrast to the previously mentioned techniques, tensor averaging methods estimate the curvature tensor as an average over a certain region of a polyhedral mesh. In [30] the curvature tensor is derived building upon the theory of normal cycles. This work includes a proof of convergence under certain sampling conditions based on geometric measure theory. The curvature tensor is defined at each point along an edge, and all contributions are integrated over a small region, see also [2]. A similar discrete curvature measure is applied in [80].

Alternative approaches locally consider a triangle with given vertex normals. In [167], the directional derivatives of the normal are expressed as finite

differences for every edge of a triangle. The resulting system of six equations is set up from the vertex positions (in parameter space) and normals and then solved for the three unknowns of the Weingarten matrix in least-squares sense. The tensors which are obtained per triangle are transformed to a common coordinate system to get a per-vertex average over the 1-ring. The algorithm can be applied with only slight modifications to compute curvature derivatives from the prior result.

In [181] the curvature tensor is estimated as smooth function (rather than a constant value) per triangle. This technique is inspired by Phong shading [156], where the vertex normals are linearly interpolated over the triangle. These interpolated normals are used to define the first and second order partials of the unit normal. This yields a piecewise smooth function defining the curvature tensor and elegant expressions for the Gaussian and mean curvature. Although this function is in general not continuous over edges of the triangulation, the approximation error is comparable to other approaches. For the estimation at vertices, the error is reduced by taking averages from all incident triangles.

3.3 Applications to Discrete Shape Analysis

The techniques reviewed in the previous section enable the estimation of curvature on discrete shapes: curvature estimates such as principal curvatures, Gaussian curvature and mean curvature are available at every vertex. These values can then be linearly interpolated in triangles. This is illustrated in Figure 2(a) and (b) where M and K are color coded. For efficient visualization (scaled) curvature values are used as 1D texture coordinates such that linear interpolation is done by the graphics hardware. Principal curvature directions

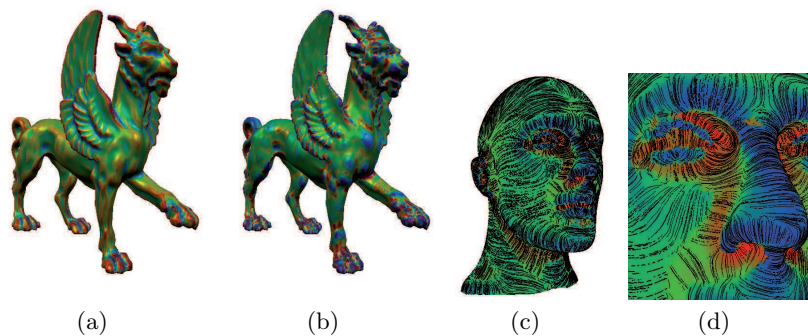


Fig. 2. Visualization of mean curvature M (a) and Gaussian curvature K (b) estimated on the Feline triangle mesh. Here, red, green and blue denote positive, zero and negative values, respectively, and lighting is enabled. (c) and (d) show the maximum curvature with lines of curvature on the Mannequin mesh.

define a vector field on the surface. Figure 2(c) and (d) shows lines of curvature obtained from stream line integration.

In addition to these examples, many surface interrogation methods which were initially developed for smooth surfaces can be adapted easily to work in the discrete setting. This applies to first order analysis (Section 4) using estimates of the surface normal: reflection lines can be simulated by environment mapping techniques, highlight lines and isophotes can be emulated similarly. With curvature estimates being available, second order analysis (Section 5) can be applied. For the computation of discrete characteristic lines (Section 6), curvature derivatives are approximated by appropriate differences.

The following sections discuss shape analysis of smooth surfaces. Interrogation of discrete shapes follows the general ideas closely and applies estimates of surface normals and curvature.

4 First-Order Shape Analysis

First-order surface interrogation methods make generally use of the surface normal vector by simulation of particular light reflecting behavior of the surface. The light reflection methods all simulate the special reflection behavior of light sources or light lines on the surface. Due to the intuitive understanding that everybody has when he observes light reflections, these methods are very effective in detecting surface irregularities. They are therefore very well suitable for testing the fairness of surfaces. Because the surface normals are involved in the computation of these lines, they also can be used to visualize first order discontinuities, like tangent discontinuities.

4.1 Reflection lines

The reflection line method determines unwanted dents by emphasizing irregularities in the reflection line pattern of parallel light lines. Let $X(u, v)$ be a representation of the surface to investigate, and let $N(u, v)$ be the unit normal vector of the surface. Furthermore a light line L is given in parameter form:

$$L(t) = L_0 + t \cdot \mathbf{s}$$

where L_0 is a point on L , \mathbf{s} is a vector defining the direction of L , $t \in \mathbb{R}$. The reflection line is the projection of the line L on the surface X , which can be seen from the fixed eye point A , if the light line L is reflected on the surface, see Figure 3(a). From geometric dependencies the following reflection condition is derived:

$$\mathbf{b} + \lambda \mathbf{a} = 2 \left(N(u, v) \cdot \mathbf{b} \right) N(u, v) \quad \text{with} \quad \lambda := \frac{\|\mathbf{b}\|}{\|\mathbf{a}\|}, \quad (3)$$

where $\mathbf{a} = P - A$, $\mathbf{b} = L - P$. Equation (3) has to be solved for the unknown parameters u and v of the reflection point P . These three non-linear equations

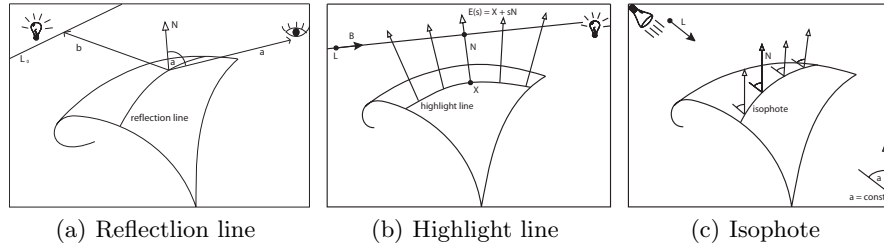


Fig. 3. First order shape analysis by simulating light reflection.

can be reduced to two equations by eliminating λ ; they can then be solved by numerical methods, but the existence and uniqueness of solutions has to be ensured by an appropriate choice of the eye point A [94, 98]. To analyze visually the surface one uses a set of parallel reflection lines with direction s , a fixed eye point A , and one steps along each curve of the set. Figure 4(a) shows a reflection line pattern on a part of a hair dryer and visualizes some surface irregularities.

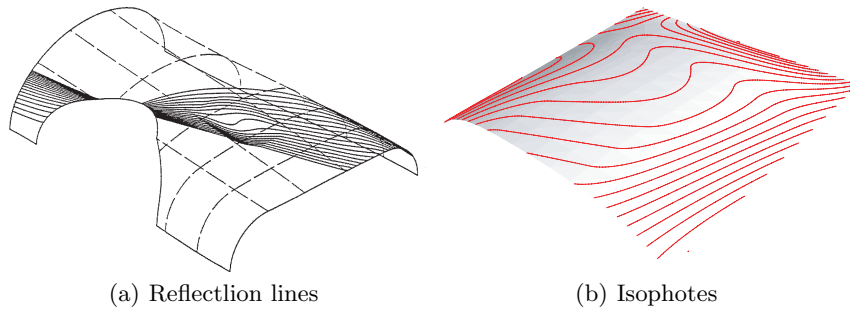


Fig. 4. Pattern of computed reflection lines and isophotes on NURBS surfaces.

4.2 Highlight lines

A highlight line is defined as the loci of all points on the surface where the distance between the surface normal and the light line is zero. The linear light source idealized by a straight line with an infinite extension

$$L(t) = L_0 + Bt$$

(L_0 is a point on L , B is a vector defining the direction of L , $t \in \mathbb{R}$), is positioned above the surface under consideration, see Figure 3(b). The highlight line method also detects surface irregularities and tangent discontinuities by

visualizing special light reflections on the surface. In comparison with the reflection line method, the highlight lines are calculated independently from any observers view point. For a given surface point $X(u, v)$ let $N(u, v)$ be the unit normal vector. The surface point $X(u, v)$ belongs to the highlight line if both lines, $L(t)$ and the extended surface normal

$$E(s) = X(u, v) + s \cdot N(u, v), \quad s \in \mathbb{R}$$

intersect, i.e. if the perpendicular distance

$$d = \frac{\| [B \times N] \cdot [L_0 - X] \|}{\| [B \times N] \|}$$

between these lines is zero, see Figure 3(b). This method can be extended to highlight bands, lines where $d \leq r$ (r fixed) is verified. For details on the algorithms to compute highlight lines see [7].

4.3 Isophotes

Isophotes are lines of equal light intensity. If $X(u, v)$ is a parameterization of the surface and L the direction of a parallel lighting, then the isophote condition is given by:

$$N(u, v) \cdot L = c,$$

where $c \in \mathbb{R}$ is fixed, see Figure 3(c). Note that silhouettes are special isophotes ($c = 0$) with respect to the light source. **Isoclines** are lines of equal normal inclination with respect to some direction V . If $X(u, v)$ is a parameterization of the surface, then the isocline condition is given by:

$$N(u, v) \cdot V = c$$

where $N(u, v)$ is the unit normal field of X and $c \in \mathbb{R}$ is fixed. In other words, isophotes are isoclines with respect to the light source direction. Similar to reflection lines and highlight lines, the isophotes provide a powerful tool to visualize small surface irregularities, which can not be seen with a simple wire-frame or a shaded surface image. In Figure 4(b) we use 20 different values for c in order to get an isophote pattern on a NURBS test surface.

Now, as stated out in the introduction of this section, the light reflection methods can be used to visualize first and second order discontinuities, because the surface normal vector is always involved in the line definitions. In fact, *if the surface is C^r -continuous, then the isophotes are C^{r-1} -continuous curves* (see [157] for more details). A curvature discontinuity can be recognized, where the isophotes possess tangent discontinuities (breaks). One should nevertheless be careful by using isophotes for this purpose, because sometimes the break points of the isophotes at curvature discontinuities may not be clearly recognized, because of an ill-conditioned light direction. This special case occurs if the orthogonal projection of the light direction L in the tangent plane at a

boundary point $X(u, v)$ is parallel to the tangent of the isophote at this point.

Isophotes for curvature discontinuity:

There is another isophote method, which on one hand is an automatic method (independent of a special light direction), but which on the other hand only visualizes curvature discontinuities across the boundaries of a patch work. It makes use of the fact that along a common boundary curve y between two surface patches that join only with tangent plane continuity the Dupin indicatrices i_1 and i_2 on both sides are different. In general there are two conjugate diameters of the Dupin indicatrix. This relation degenerates at parabolic points, because the asymptotic direction (the direction in which the normal section curvature vanishes) is the conjugate to itself, but also conjugate to all other directions. At planar points, we have this degeneration for each (tangent) direction. Since both patches have a common boundary curve, and the tangent planes along that curve are unique, the Dupin indicatrices i_1, i_2 have a common diameter, but differ in the other.

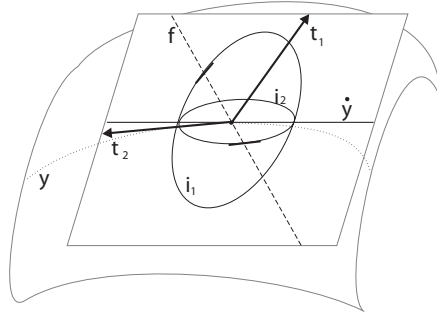


Fig. 5. Isophotes for curvature discontinuity.

We now consider an isophote c passing through P . The tangent t_i of c at P with respect to X_i is conjugate to the orthogonal projection f of the light ray onto the tangent plane ($i = 1, 2$), see Figure 5. In general the isophote c shows a tangent discontinuity at P if the Dupin indicatrices of X_1 and X_2 are not equal, but we have to avoid the situations $f = \dot{y} = t$ and $f = t'$. More details can be found in [161].

4.4 Detection of inflections

Orthotomics and the polarity method are both interactive interrogation tools capable to detect only one particular type of surface “imperfection”: the change of the sign in the Gaussian curvature. For example, surface with only convex iso-parameter lines are not necessarily convex, i.e. their Gaussian cur-

vature is not required to be positive at all surface points. Such surface imperfections are difficult to detect visually in this case and therefore a curvature based surface analysis is needed like color maps or generalized focal surfaces, see Section 5. The following methods in contrast can visualize a change of sign in the Gaussian curvature without computing second order derivatives of the surface.

Orthotomics

In [85] it has been shown that for a regular surface $X(u, v)$ and for a point P that does not lie on the surface or on any tangential plane of the surface the k -orthotomic surface $Y_k(u, v)$ with respect to P defined by

$$Y_k(u, v) = P + k \left((X(u, v) - P) \cdot N(u, v) \right) N(u, v),$$

where $N(u, v)$ is the unit normal vector of the surface has a singularity in (u_0, v_0) , if and only if the Gaussian curvature of X vanishes, or changes its sign at this point. To illustrate this method we consider a Bézier surface with completely convex parameter lines, see in Figure 6(left). But this surface is not convex: as shown in Figure 6(right), the orthotomic analysis emphasizes the change of sign of the Gaussian curvature in the corner region.



Fig. 6. Bicubic surface patch with line of vanishing Gaussian curvature (left). Orthotomic analysis (right).

Polarity method

The polarity method is a further method able to detect unwanted changes in the sign of the Gaussian curvature without computing second order derivatives of the surface. It works for curves as well. It uses the polar image of a curve or surface, where the singularities (cusps, edge of regression) of this image indicate the existence of points with vanishing Gaussian curvature. The polar surface looks similar to the orthotomic surface, because the center of polarity is chosen to be equal to the projection point of the orthotomic analysis. For more information about the polarity method and on how removing the inflections see [86].

4.5 Geodesic paths on surfaces and meshes

Geodesic paths, or simply geodesics, on a surface are surface curves which connect two surface points with minimum path length. A thorough study

of geodesics and their role of in surface interrogation requires much more attention than the present overview can provide. So below we give only a brief literature survey.

Geodesics deliver rich information about surface geometry and, therefore, have various theoretical and practical applications. In particular, detecting geodesic paths on surfaces approximated by triangles meshes is a common operation for many graphics and modeling tasks such as mesh parameterization [103], mesh segmentation [93], skinning [175], mesh watermarking [162], and mesh editing [100].

A rigorous mathematical treatment of geodesics can be found in [102, 42]. Some numerical aspects are presented in [56]. An algorithm for approximate computation of geodesic paths on smooth parametric surfaces has been explored in [155, 154]. Various algorithms exist for computing geodesic paths and distances. The so-called MMP algorithm [124] computes an exact solution for the "single point, all distances" shortest path problem by partitioning each mesh edge into a set of intervals over which the exact distance can be computed. In [179] an accelerated implementation of this algorithm is presented. An algorithm to solve the "single source, single distance" geodesic problem is given in [91]. See also [123] for a broad survey of algorithms for computing shortest paths on graphs.

5 Second-order shape analysis

Surface curvature is of central importance for surface design. Often the result is required to be mathematically smooth (continuous in the 2nd derivative) and aesthetically pleasing, i.e. have smooth flowing highlights and shadows. To obtain an aesthetically pleasing shape, the designer works with the curvature. A color map (see Section 5.5) can be used to visualize curvature (Gaussian, principal curvatures) over the surface. The problem is the good choice of the color scale, which depends on the curvature function and therefore on the underlying surface.

The surface interrogation methods presented in this section are therefore **curvature analysis** tools which are able to detect all surface imperfections related to curvature, like bumps, curvature discontinuity, convexity, and so on.

5.1 Local shape analysis with Gaussian curvature

Let us look at a smooth surface in a neighborhood of one of its point.

The simplest classification of local surface shapes is given by the the sign of the Gaussian curvature $K = \kappa_1 \cdot \kappa_2$.

$K > 0$. The normal curvatures $\kappa_N(\varphi)$ has the same sign in all directions, so the tangent plane touches the surface at one point. The usual convex or

concave regions corresponding to this, as demonstrated by the left image of Figure,7 and the left images of Figure 8.

$K < 0$. The normal curvature becomes zero twice during the half rotation of the normal plane around the normal. The tangent plane intersects with the surface in these directions of zero curvature. The surface is locally saddle-shaped, as seen in middle images of Figure 7 and Figure 8.

$K = 0$. At least one principal curvature is zero. It produces a parabolic point. See the right image of Figure 7 and the middle-right image of Figure 8. A set parabolic points may form a parabolic region shown in the right image of Figure 8.

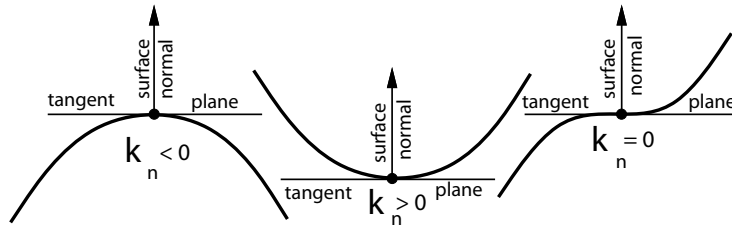


Fig. 7. Local shape of normal section curve is defined by curvature.

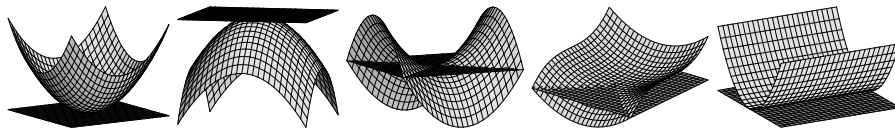


Fig. 8. Gaussian curvature determines local shape of surface. Left images: convex and concave regions ($K > 0$). Middle: saddle-shaped region ($K < 0$). Middle-right: a parabolic point ($K = 0$) Right: a region consisting of parabolic points.

The Gaussian curvature of a surface can be expressed through the coefficients of the first fundamental form. Thus we arrive at the following famous result called Gauss’s Theorema Egregium: the Gaussian curvature of a surface is a bending invariant.

Now let us consider a simple geometrical interpretation of the Gaussian curvature, by means of which Gauss originally introduced it.

Consider a two-sided surface in three-dimensional space. Let us transport the positive unit normal vector from each point of the surface to the origin. The ends of these vectors lie on the unit sphere. We obtain the mapping of the surface into the unit sphere, see Figure 9. It is called the **Gauss map**.

The Gauss mapping takes areas on surfaces to areas on the unit sphere. Consider the unit surface normals at the surface points within the area ΔS

on the surface. Let us denote the area on the unit sphere (solid angle) corresponding to ΔS by ΔA . It turns out that the Gaussian curvature at the point is the limit of the ratio of these areas:

$$K = \lim_{\Delta S \rightarrow 0} \frac{\Delta A}{\Delta S}.$$

This remarkable formula resembles the definition of the curvature of the plane curves: $\kappa = d\varphi/ds$.

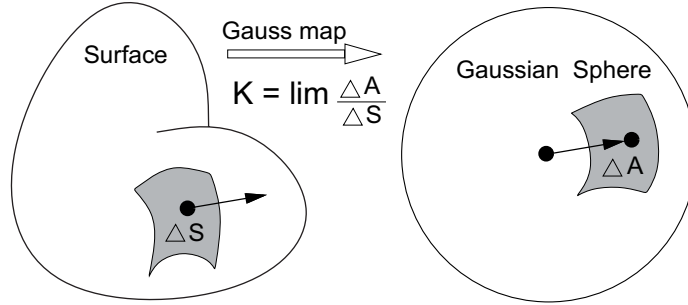


Fig. 9. Gauss map and geometric meaning of Gaussian curvature.

The Gauss map can be used for detecting spherical, cylindrical, and conical regions on a surface [12].

5.2 Focal Surface and Corresponding Surface Features

For a smooth surface $X = X(u, v)$ its **focal surface** is given by

$$X_F(u, v) = X(u, v) + \frac{N(u, v)}{\kappa(u, v)}, \quad \kappa = \kappa_1, \kappa_2,$$

where $N(u, v)$ is the oriented normal. The focal surface is formed by the principal centers of curvature and consists of two sheets corresponding to the maximal and minimal principal curvatures κ_1 and κ_2 . One can show that the focal surface is the envelope of the surface normals. In geometrical optics [77], a **caustic** generated by a family of rays is defined as the envelope of the family. Thus the focal surface is the caustic of the family of surface normals. The focal surface can be also defined as a surface swept by the singularities of the offset surfaces $O_d(u, v) = X(u, v) + dN(u, v)$.

The focal surface is the 3D analogue of the evolute of a planar curve and has singularities. The singularities of the focal surface consist of space curves called **focal ribs**.

Ridges, the surface curves corresponding to the focal ribs are natural generalization of the curve vertices for surfaces. The ridges can be defined

as sets of surface points where the principal curvatures have extremes along their associated principal directions and points where the principal curvatures are equal (**umbilics**). A thorough study of the ridges and their properties is conducted by Porteous [158]. See also [72] where a detail classification of the ridges is presented. Below we briefly discuss the ridges from a singularity theory point of view.

Near a point on a focal rib the focal surface can be locally represented in the parametric form $(c_1 t^3, c_2 t^2, s)$, where $c_1 \neq 0$ and $c_2 \neq 0$, in well chosen coordinates (s, t) . The focal ribs themselves have singularities at points corresponding to the umbilics and those ridge points where one of the principal curvatures has an inflection along its corresponding curvature line. Generic (typical) singularities of the focal surface are shown in Figure 10.

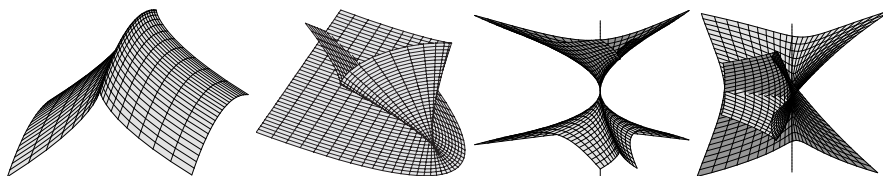


Fig. 10. Typical singularities of the focal surface. From left to right: cuspidal edge (rib), swallowtail, pyramid, purse. At the swallowtail singularity the rib has a cusp. The pyramid and purse correspond to the umbilical point on the surface. The vertical lines at the bottom images are the surface normals at the corresponding umbilics.

The umbilics and ridge points can be also characterized as surface points where the osculating spheres (spheres of curvature) have high-order contacts with the surface. Therefore the umbilics and ridges are invariant under inversion of the surface with respect to any sphere.

The focal surface points can be also described in terms of degenerate singular points of distance functions. Given a surface and a point in 3D, let us consider the distance function from the point and restrict the function onto the surface. This gives a three-dimensional family of distance functions defined on the surface and parameterized by points in 3D. Now the focal surface is generated by those point-parameters for which the distance function has degenerate critical points. A typical degenerate critical point has one of the following two forms $\pm s^2 + t^3$ in proper coordinates s and t . If the point-parameter is a typical point on a focal rib, the distance function has a critical point in one of the following four forms: $\pm s^2 \pm t^4$. More degenerate critical points occur when the point-parameter is located either at a swallowtail singularity of the focal surface or at an umbilical points. It is interesting that the **cut locus** of the surface [190] (skeleton or medial axis of a figure bounded by the surface) consists of those point-parameters which define the distance functions with two equal global minima. Thus, as illustrated in Figure 11, the edges of the skeleton are located at focal ribs.

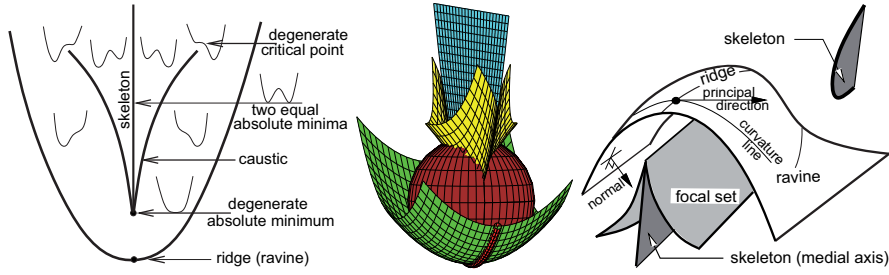


Fig. 11. Left: zoo of distance functions; thin lines are used to sketch typical profiles of the surface functions defined by the distance from a given point to the surface points. Center: the skeleton (blue), caustic (yellow), ridge (red) and an osculating sphere (brown) at a ridge point of the elliptic paraboloid. Right: schematic illustration of relationships between the cut locus, focal surface and ridges.

The focal surface possesses many interesting properties. For example, for each line of curvature on a surface there is a corresponding line on the corresponding sheet of the focal surface. It can be shown that those raised lines of curvature are geodesics on the focal surface [159, 131].

In [118, 99] umbilics are used for shape interrogation and shape matching purposes. Statistics of various types of umbilics on random surfaces computed and analyzed in [11] may have many potential applications for inspecting and interrogating surface properties.

5.3 Hedgehog diagrams and curvature plots

The hedgehog diagrams and curvature plots are well known interrogation tools for planar curves [6, 54]. A hedgehog diagram for planar curves visualizes the curve normals proportional to the curvature value at some curve points. A new curve is obtained by $\tilde{X}_{hedgehog}(t) = X(t) + \kappa N(t)$ thus visualizing curvature distribution and discontinuity. The inspection of surfaces with these methods can be done by applying them to planar curves on the surface (intersections of the surface with planes). [97] shows an example of application. Hedgehog diagrams for entire surfaces are nevertheless difficult to interpret and are therefore not to be recommended.

5.4 Generalized focal surfaces

Although the idea of generalized focal surfaces is quite similar to hedgehog diagrams, their application area is much larger. Instead of drawing surface normals proportional to a function value, only the point on the surface normal proportional to the function is drawn. The loci of all these points is the **generalized focal surface**. This method was introduced by [71], and is based on the concept of focal surfaces which are known from line geometry, introduced in Section 5.2. The generalization of this classical concept leads to the

generalized focal surfaces:

$$F(u, v) = X(u, v) + s \cdot f(\kappa_1, \kappa_2) \cdot N(u, v), \quad \text{with } s \in \mathbb{R}$$

where N is the unit normal vector of the surface X . f is a real valued function of the parameter values (u, v) .

The variable offset function f can be any arbitrary scalar function, but in the context of surface interrogation it is quite natural to take f as a function depending on the principal curvatures κ_1, κ_2 of X , f.ex. $f = \kappa_1 \kappa_2$ Gaussian curvature, $f = \frac{1}{2}(\kappa_1 + \kappa_2)$ mean curvature, $f = (\kappa_1^2 + \kappa_2^2)$ energy functional, $f = |\kappa_1| + |\kappa_2|$ absolute curvature, $f = \kappa_i$ principal curvatures, $f = \frac{1}{\kappa_i}$ focal points, $f = \text{const}$ offset surfaces. This not only enables to visualize a particular curvature behavior, but it can interrogate and visualize surfaces with respect to various criteria: A **convexity test** can be performed using the Gaussian curvature offset $f = \kappa_1 \cdot \kappa_2 = K$. A surface is locally convex at $X(u, v)$, if the Gaussian curvature is positive at this point. Often a surface is called non-convex, if there is a change in the sign of the Gaussian curvature. the two surfaces $X(u, v)$ and $F(u, v)$ intersect at the parabolic points, see Figure 12(a). The generalized focal surface therefore pin points directly on the area where the sign of K changes in contrast to orthotomics (Section 4) which are also used to test the convexity. **Flat points** which are special umbilic points with $\kappa_1 = \kappa_2 = 0$ can be detected using $f = |\kappa_1| + |\kappa_2|$ as well as $f = \kappa_1^2 + \kappa_2^2$. Flat points are undesired surface points because they make the surface bumpy. **Curvature discontinuity** can be visualized through gaps in the surface F with $f = \kappa_1^2 + \kappa_2^2$ since it is a second order surface analysis tool, see Figure 12(b). **Visualizing surface irregularities:** Surfaces are aesthetically pleasing if they have “nice” light reflections. Thus similar to reflection lines the generalized focal surfaces are also a tool for visualizing such surface imperfections because they are very sensitive to small irregularities in the shape. In Figure 12(b) part of a hair dryer is shown. It consists of biquintic C^1 -continuous patches. The iso-parametric lines do not reflect the bump in the surface, which is however emphasized by the focal analysis. Another aspect of surface analysis is the visualization of **technical aspects**. A surface which should be treated by a spherical cutter is not allowed to have a curvature radius smaller than the radius of the cutter R_{cutter} . The generalized focal surfaces are able to detect such undesired regions by intersection with the surface X . The offset function to choose in this special case, is $f = \frac{1}{R_{cutter}} - \kappa_{max}$. Figure 12(c) shows such a surface which is not allowed to be cut. Generalized focal surfaces not only visualize surface imperfections, they also give the user a 3D impression of the relative amount of the offset function over the surface, what color maps can't do.

5.5 Color mappings

Color is used to emphasize features on the surface. Texturing can emphasize the spatial perception of an 2D image of the surface. A **color-coded map** is

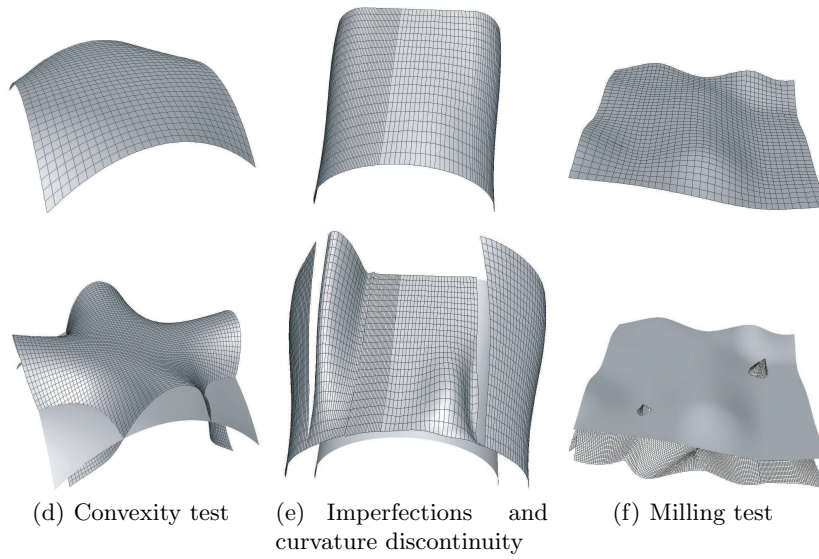


Fig. 12. Second order surface analysis with generalized focal surfaces.

an application, which associates to a scalar function value a specific color. The color scale presents an even gradation of color corresponding to the range of function values. Colors are principally used to visualize either continuously or discontinuously any scalar function over a surface [38, 5, 4, 59], like pressure, temperature, or curvature, see Figure 13. Colors are used as a fourth dimension and show the user immediately and quantitatively how the function varies over the surface.

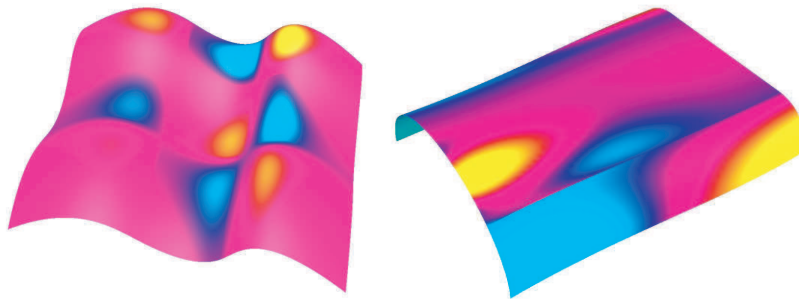


Fig. 13. Color codings of Gaussian curvature.

An even gradation of the linear or cyclic color coding is important to visualize the rapid curvature variation by the presence of color fringes. Beck et al. [5] propose to use the HSI (hue, saturation, intensity) model and to perform

transformations between this space and the three primary colors RGB. See [58] for more details on color spaces and transformations. An example of discrete color-coding of the interval $[0,1]$ is the following one:

Interval	Red	Green	Blue	Color
0.0 - 0.2	1	0	0	red
0.2 - 0.4	1	1	0	yellow
0.4 - 0.6	0	1	0	green
0.6 - 0.8	0	1	1	turquoise
0.8 - 1.0	0	0	1	blue

The main difficult of this simple interrogation method is the choice of a convenient color scale, which obviously depends on the function values to be visualized.

Pseudo texture

The use of colors for displaying a surface helps to emphasize the 3D understanding of an 2D image by simulating shadows, perspective and depth of the object. An artificial texturing is an aid for visualizing rendered surfaces. Isoparametric lines are commonly used, but they are in some situations ambiguous. Schweitzer [170] projects equally spaced dots of equal size over the surface in order to increase the visual perception of the form.

6 Characteristic lines

Drawing lines on surfaces is a powerful and widely used tool for analysis and visualization of surface features. The techniques of **isolines**, **lines of curvature**, **geodesic paths** and **ridges** are presented. Numerous graphical examples are illustrated in [159, 56]. In the last three cases a set of lines on the surface can be created, and should be interpreted with the knowledge of differential geometry. They are the most sophisticated tools from the mathematician's point of view. The user should interpret the lines of curvature or the geodesic paths.

6.1 Isolines

Isolines are lines of a constant characteristic value on the surface. They provide an interrogation tool with a wide variety of applications. They help analyzing surface characteristics, and they are used to visualize the distribution of scalar quantities over the surface. The visualization of a certain number of isolines, with respect to an even distribution of the characteristic values allows to study the behavior of these values.

Contour lines are planar lines on the surface which are all parallel to a fixed reference plane. Closed contour lines indicate maxima and minima of the surface with respect to the direction given by the plane's normal vector [76, 5]. Saddle points appear as “passes”. The contour lines only cross in the exceptional case of a contour at the precise level of a saddle point. [141] describes systematically the distribution of other critical points on a surface. A disadvantage of contour lines is the fact that they are costly to compute. Several surface contouring methods exist, which are sometimes depending of the specific surface formulation [152, 169, 108]. Hartwig and Nowacki [76] propose to subdivide the surface into sufficient small pieces which are then approximated by bilinear surfaces. Then the contour lines can easily be computed.

Iso-contouring is the technique of extracting constant valued curves and surfaces from 2D and 3D scalar fields. Interactive display and quantitative interrogation helps understanding the overall structure of a scalar field and its evolution over time. Traditional iso-contouring techniques examine each cell of a mesh to test for intersection with the iso-contour of interest. For an overview see [168]. Extraction of isosurfaces from 3D scalar field is generally be done by the Marching Cubes algorithm and its variants [111, 143, 27]

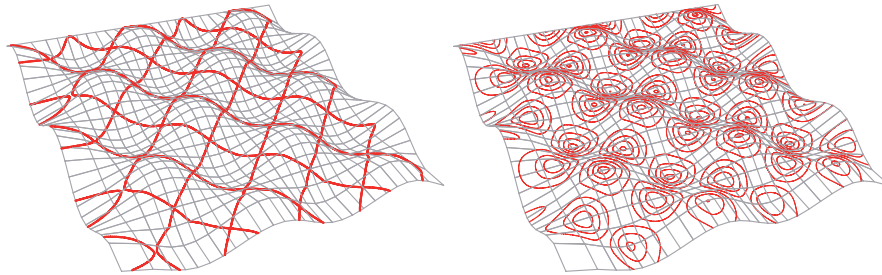


Fig. 14. Gaussian curvature isoline. Left: parabolic lines. Right: isolines corresponding to different constant Gaussian curvature values.

Parabolic lines are isolines of zero Gaussian curvature on the surface. They are of particular interest for intrinsic surface interrogation, since they divide the surface into elliptic and hyperbolic regions and they reflect therefore the local curvature behavior of a surface. Parabolic lines are special Gaussian curvature lines, see Figure 14. In [79] a more complex example with the statue Apollo Belvedere is drawn.

6.2 Lines of curvature, umbilics

Lines of curvature are curves whose tangent directions coincide with those of the principal directions, which are orthogonal. They form therefore an orthogonal net on the surface.

The net of lines of curvature becomes singular at an umbilical point where κ_1 and κ_2 are identical and the principal directions are indeterminate. Some numerical integration method is used to calculate the lines of curvature. But the integration process becomes unstable near an umbilic. Unfortunately umbilics appear frequently on free-form surfaces. A recent work about umbilics [117], destined for use in CAGD (Computer Aided Geometric Design), presents a procedure to compute the lines of curvature near an umbilic. And in [116] a computational method to locate all isolated umbilics on parametric polynomial surfaces is described. The discrete field of principle curvature directions computed on a surface mesh has been used for remeshing [2]. More details about umbilics and lines of curvature figures are found in classical differential geometry literature [35], or in a more recent book [159].

6.3 Curvature Extrema for Shape Interrogation

Surface features invariant under rotations, translations, and scaling are important for studying shapes of 3D objects. The ridge curves discussed briefly in Section 5.2 are among the most important view- and scale-invariant features of a smooth surface.

The ridges are defined as the extremes of the principal curvatures along their corresponding curvature lines and constitute powerful surface descriptors. They have been intensively studied in connection with research on the accommodation of the eye lens [69], structural geology [163] and geomorphology [109], human perception [83], image analysis [191, 129, 127, 40, 110], quality control of free-form surfaces [84], reverse engineering [87], analysis and registration of anatomical structures [68, 67, 151], face recognition [72], and non-photorealistic surface rendering [89, 114, 36]. (See also references therein.)

An explanation of why some ridges are good for sketching complex 3D shapes can be found in [191]: given a grey-scale image of an illuminated 3D object, under general illumination and reflection conditions, the zero-crossings of the second directional derivative of the image intensity along the direction of the image intensity gradient occur near the extremes of the principal curvature along their principal directions. Thus the projections of ridges onto the image plane are usually located near edges, the most salient image features.

Some subsets of ridges play an important role in perceptual shape organization. Human perception experiments suggest the so-called minima rule [83] which sets region boundaries along lines divides shapes into parts at negative minima of the principal curvatures along their lines of curvature. The minima rule was employed in [146] for mesh segmentation purposes.

The ridges on a surface have interesting relations with the skeleton (medial axis) of a figure bounded by the surface and can be described via high-order contacts between the surface and its osculating spheres. See [158, 101, 192, 159, 8], [72, Chapter 6], and recent reviews in [26, 25] for rigorous mathematical treatments revealing beautiful properties of these curvature features. Surface landmarks associated with the ridges were considered

in [101, 131, 160]. Bifurcations of the ridges on dynamic shapes were studied in [159, 15, 16, 160, 112].

Recently the so-called **crest lines**, a subset of the ridges consisting of the extremes of the principal curvature maximal in absolute value along its corresponding curvature line, draw much attention because of their ability to represent surface creases [184, 127, 151, 177, 145, 81]. See also references therein. One motivation for describing surface creases as the crest lines is based upon the following analogy with edges of grey-scale images [145]. Consider a surface and its Gauss map which associates with every point \mathbf{p} of the surface the oriented normal vector $\mathbf{n}(\mathbf{p})$. The derivative $\nabla\mathbf{n}(\mathbf{p})$ (Jacobian matrix) of the Gauss map measures the variation of the normal vector near \mathbf{p} , i.e., how the surface bends near \mathbf{p} . It is easy to see that the eigenvalues and eigenvectors of $\nabla\mathbf{n}(\mathbf{p})$ are the principal curvatures and principal directions of the surface at \mathbf{p} , respectively. Thus the maximal variation the surface normal is achieved in the principal direction of the principal curvature maximal in absolute value. So it is natural to define surface creases as loci of points where the positive (negative) variation of the surface normal in the direction of its maximal change attains a local maximum (minimum). Figure 16 shows the crest lines detected on various models represented by dense triangle meshes.

Practical detection of the ridges and their subsets is a difficult computational task since it involves estimating of high-order surface derivatives. Various techniques were proposed for detecting the ridge lines and their subsets on

- surfaces in implicit form and isosurfaces of 3D images [158, 129, 128, 184, 182, 10, 13];
- surfaces approximated by polygonal meshes [113, 9, 188, 82, 177, 26, 25, 145, 81];
- height data [65, 96, 95, 109];
- surface given in parametric form [84–75]

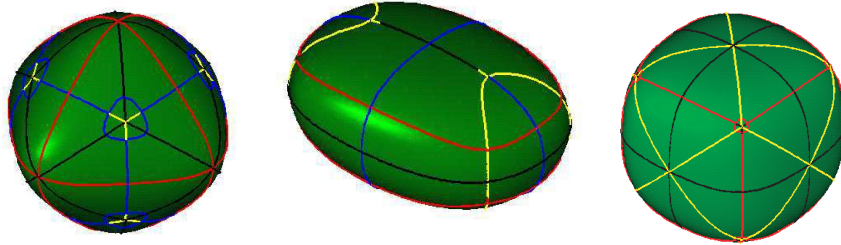


Fig. 15. Various types of ridges detected on smooth surfaces. The images are taken from [13].

For shape interrogation purposes (shape quality control and analysis of aesthetic free-form surfaces), the ridges were used in [84, 78]. Moreton and

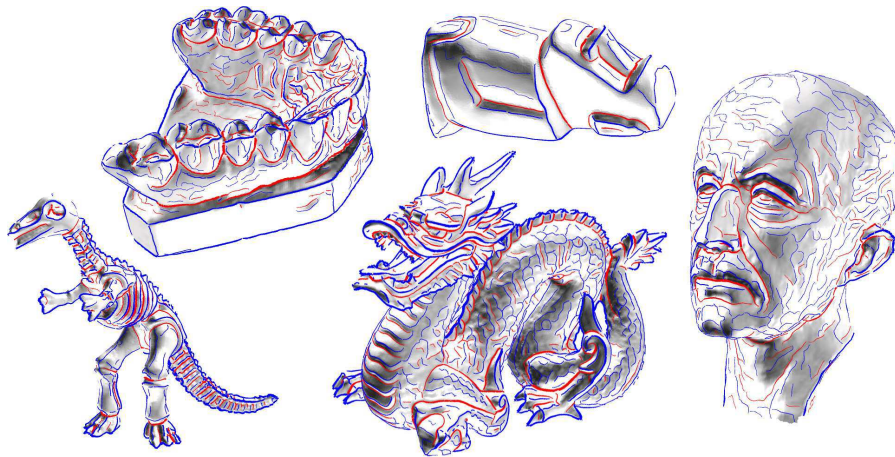


Fig. 16. The crest lines detected on various surfaces approximated by dense triangle meshes.

Sequin [130] used the sum of the squared derivatives of the principal curvatures along their corresponding curvature lines as a measure of surface fairness.

Often, instead of the ridges and their subsets defined via extremes of the principal curvatures, simpler surface features are detected. In geometric modeling, there has been considerable effort to develop robust methods for detecting surface creases, curves on a surface where the surface bends sharply. Interesting methods for crease detection on dense triangle meshes and point-sampled surfaces were proposed in [87, 166, 88, 70, 178, 148, 150]

Whereas the ridges were first studied one hundred years ago [69] and have rich history [159], the so-called **sub-parabolic lines**, the loci of points where one of the principal curvatures has an extreme value when moving along the curvature line corresponding to another principal curvature. The sub-parabolic lines were introduced in [17] and studied in [159, 131, 160]. They possess many remarkable properties: the sub-parabolic lines correspond to the parabolic lines on the focal surface, hence the name, and consist of geodesic inflections of the lines of curvature [131]. The sub-parabolic lines can be also detected by examining the profiles of surfaces [131].

6.4 Special Surface Points

In this section, following [131] we consider special surface points which lie on the ridges and sub-parabolic lines. We adapt the color scheme proposed by Porteous [158, 159]. Let us give the principal curvatures and corresponding principal directions, parabolic lines, and sheets of the focal surface a color (red or blue) in order to distinguish between them. The red (blue) sub-parabolic line consists of the extremes of the red (blue) principal curvature along the blue

(red) curvature line. The following surface landmarks are useful for surface interrogation purposes:

- **Umbilic points.** See [118, 149] for application of umbilics in surface matching and shape interrogation.
- **A ridge and sub-parabolic line of the same color cross.** The principal curvature of the same color takes an extreme value there (maximum, minimum, or saddle).
- **A ridge is tangent to the line of curvature of the same color.** These surface landmarks corresponds to the swallowtail singularities of the focal surface.
- **A ridge crosses a ridge of other color.** In [183] it was suggested to use these landmarks for 3D image registration.
- **A ridge crosses the parabolic line of the same color.** The Gauss map has the so-called pleat singularity at such a point [101].

Koenderink [101] introduced two curvature-based measures of surface curvature: the **curvedness**

$$C = \frac{2}{\pi} \ln(\kappa_1^2 + \kappa_2^2)$$

and the **shape index**

$$S = -\frac{2}{\pi} \arctan \frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}.$$

These measures are often more convenient for practical purposes than the standard curvature descriptors $\{\kappa_1, \kappa_2\}$ and $\{M, K\}$, where K and M are the Gaussian and mean curvatures, respectively. In [142] it was suggested to use local maxima of the curvedness to define surface corner points.

7 Robust Symbolic based Shape Interrogation and Analysis

Interrogation of polynomial and rational surfaces could be made with the aid of symbolic processing. The advantage of the symbolic approach over sampling of properties, like curvature, at a discrete set of point stems from the ability to analyze the properties globally and provide global (error) bounds. Many properties of free-form geometry are differential and can be derived after executing a few basic operations over the polynomial or rational representation of the original interrogated curve $C(t)$ or surface $S(u, v)$, namely: differentiations, summations and products. We also assume the availability of a zero set finding tool, an operation that is equivalent to intersecting a polynomial or a rational function with a line in R^2 (a plane in R^3). As a simple example, consider the curvature field of a planar regular curve $C(t) = (x(t), y(t))$ that is equal to:

$$\kappa(t) = \frac{x'(t)y''(t) - y'(t)x''(t)}{(x'^2(t) + y'^2(t))^{2/3}}.$$

$\kappa(t)$ is not rational due to the fractional power in the denominator, in the normalization factor. Nonetheless, if one only seeks the *inflection* points of $C(t)$, only the numerator of κ needs to be considered. Then, the solution of the constraint of

$$x'(t)y''(t) - y'(t)x''(t) = 0 \quad (4)$$

finds all the inflection points in the regular planar curve $C(t)$, if any. In Equation (4), the problem of finding all the inflection points of a planar regular curve was reduced to that of finding a zero set. Differentiation and products were used to compute the inflection points' constraints.

Differentiation of piecewise polynomials and rationals is well known [28, 53]. Similarly, the addition of two (piecewise) polynomials that share a function space (same order and knot sequence) is realized by simply adding the corresponding coefficients. Two polynomials could be elevated to the same function space via knot insertion and degree elevations; see [28, 53] for more details. Products are the last operator we seek, an operation also required because of the quotient rules over addition and differentiation of rationals. Products are more complex to compute (see [43, 53]) but, clearly, products of piecewise polynomials and/or rationals are piecewise polynomials and/or rationals as well.

In summary, the ability to form a closure and compute a differential property in the piecewise polynomial and/or rational domains, makes it far simpler and robust to analyze that property. While κ is not rational, its numerator is and so inflection points could be detected as a zero set of $x'(t)y''(t) - y'(t)x''(t)$. For similar reasons, the unit normal $N(t)$ of $C(t)$ is not rational but both $\kappa(t)N(t)$ and $N(t)/\kappa(t)$ are rational. Hence, x -extreme points and y -extreme points on $C(t)$ can be identified as

$$\langle \kappa(t)N(t), (0, 1) \rangle = 0, \quad \text{and} \quad \langle \kappa(t)N(t), (1, 0) \rangle = 0,$$

and the local maximum curvature locations in $C(t)$ are detectable [45] as the zeros of

$$\frac{d \langle \kappa(t)N(t), \kappa(t)N(t) \rangle}{dt},$$

yet another rational function.

In [45], points of extreme curvature, or alternatively, inflection points are detected using these schemes. In addition, a scheme to approximate an arc-length reparametrizations for piecewise polynomial and/or rational curves is presented.

In the next section, Section 7.1, we will demonstrate the power of symbolic based interrogation in geometric design, for curvature analysis. In Section 7.2, silhouette curves, isoclines and isophotes curves, and reflection curves are all shown to be reducible to zero set finding. Then, in Section 7.3, we consider the problem of symbolic recognition of simple primitive surface shapes.

7.1 Curvature Analysis

Reexamining the second order differential analysis of parametric surfaces (recall Section 2), it turns out that given a rational surface $S(u, v)$, the Gaussian curvature K is rational whereas the mean curvature M is not (while M^2 is). In [47], a rational form of (the numerator of) K is symbolically computed and its zeros are used to robustly extract the parabolic lines of the surface. Figure 17 presents one example of computing the parabolic curves for a bicubic surface patch as the zeros of K .

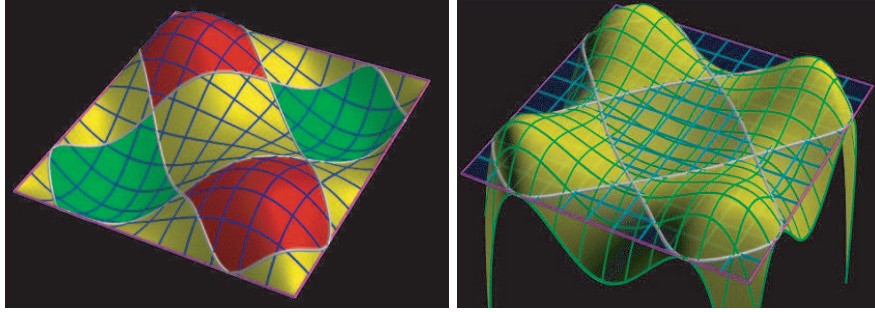


Fig. 17. Left: a free-form B-spline surface is presented, after being subdivided into convex (red), concave (green), and hyperbolic regions (yellow). The parabolic lines (white) separate the regions. Right: presents the function of $K(u, v)$ (in yellow) and its zero set (the parabolic lines).

While M is not rational, one can compute M^2 as a rational form. Similarly, the form of $\kappa_1^2 + \kappa_2^2$, where κ_i , $i = 1, 2$, are the two principle curvatures, is rational and can capture regions that are highly curved. By subdividing the original surface into regions that prescribe different values of $\kappa_1^2 + \kappa_2^2$, one can separate the surface into regions that could be NC-machined more efficiently with different sizes of ball- and flat-end cutters [44]. Let $\mathcal{K}_0 = \kappa_1^2 + \kappa_2^2$ at $S_0 = S(u_0, v_0)$. Then, the normal curvature at S_0 is bounded from above by $\sqrt{\mathcal{K}}$ or an NC ball end cutter of radius $1/\sqrt{\mathcal{K}}$ could be locally fitted to S_0 without (local) gouging. Figure 18 shows one such example where a surface is divided into regions of different values of extreme curvature, $\mathcal{K} = \kappa_1^2 + \kappa_2^2$. See also Equation (1).

7.2 Silhouette, Isoclines/Isophotes and Reflection lines

The extraction of silhouettes of a free-form surface could be easily reduced to a zero set finding problem. Looking at a rational surface $S(u, v)$ from direction vector V , the silhouettes of S are characterized as the rational constraints of

$$\langle N(u, v), V \rangle = 0,$$

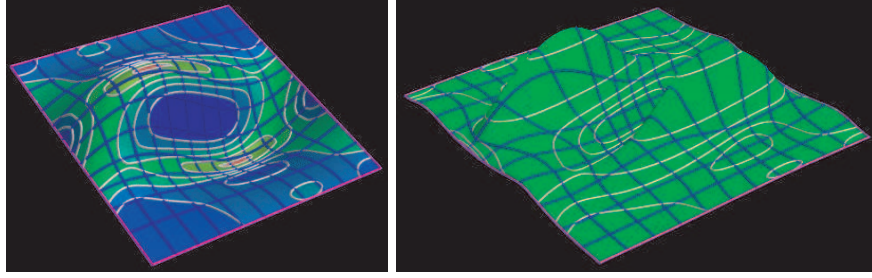


Fig. 18. Left: a free-form B-spline surface is presented, after being subdivided into regions of different levels of $\kappa_1^2 + \kappa_2^2$. Right: presents the rational surface $\kappa_1^2 + \kappa_2^2$ and its contouring (in white) at the different levels.

where $N(u, v) = \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}$. If the view is a perspective view through point P (the eye), the silhouettes could be derived as the rational form of

$$\langle N(u, v), S(u, v) - P \rangle = 0.$$

Interestingly enough, highlight lines [7] (see Section 4.2), isoclines and isophotes (see Section 4.3) could be similarly reduced to a zero set finding, using symbolic manipulation. Let the unit view direction vector for which isoclines are sought be V . Then, positions on surface $S(u, v)$ that present a normal with a constant inclination angle of α degrees could be characterized as

$$\left\langle \frac{N(u, v)}{\|N(u, v)\|}, V \right\rangle = \cos(\alpha),$$

which is not a rational but could be made into one by squaring both sides as,

$$\langle N(u, v), V \rangle^2 - \|N(u, v)\|^2 \cos^2(\alpha) = 0, \quad (5)$$

at the cost of extraction both the $+\cos(\alpha)$ and the $-\cos(\alpha)$ isoclines, simultaneously. Figure 19 shows an example of subdividing a free form surface into regions of steep slopes (more than 45 degrees) and shallow slopes, using isoclines' analysis. Such a dichotomy might be desired, for example, in layered manufacturing processing where support is to be added to the geometry only below a certain slope.

Reflection lines (see Section 4.1) can also be reduced to rational zero set constraints as follows. An incoming ray V that hits surface $S(u, v)$ will be reflected in direction $r(u, v)$,

$$r(u, v) = 2N(u, v) - V \frac{\langle N(u, v), N(u, v) \rangle}{\langle N(u, v), V \rangle}. \quad (6)$$

In practice, Equation (6) might be difficult to work with near silhouettes (where $\langle N(u, v), V \rangle$ vanish) and so, in [46], $2N(u, v)\langle N(u, v), V \rangle -$

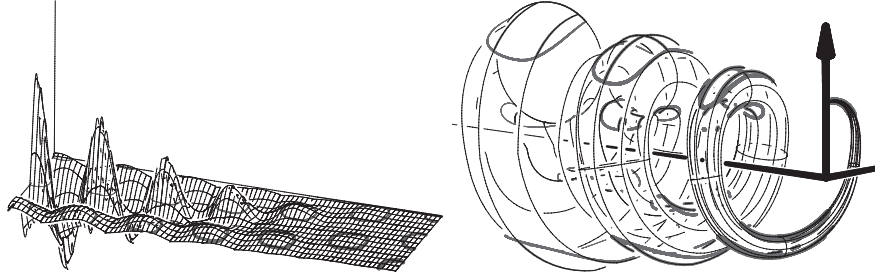


Fig. 19. Isoclines at 45 degrees from the vertical direction V . Left: the function whose zero set (see Equation (5)) prescribes the isoclines of the surface shown in the right figure is presented. Right: Isoclines also serve to split the surface into regions of slopes (normals) with more than 45 degrees (in thin lines) and regions of less than 45 degrees (in thick lines) with respect to the vertical direction V .

$V \langle N(u, v), N(u, v) \rangle$ was proposed as a better alternative. In [46], reflection ovals, or reflections of circular curves, were similarly reduced to zero set finding problems.

7.3 Surface Recognition

A fundamental question when analyzing free-form geometry is whether the given curve or surface is of a basic primitive nature. That is, a curve is tested if it is a circle, or a surface is tested if it is a cylinder, or alternatively, a surface of revolution. In [48], these questions are answered using symbolic differential analysis. A rational curve is a circle if its rational squared curvature field, $\kappa^2(t) = \langle \kappa(t)N(t), \kappa(t)N(t) \rangle$, is constant. In other words, given a B-spline curve $C(t)$, all its coefficient of the B-spline representation of $\kappa^2(t)$ should be the same and in fact equal to the square of the reciprocal of the radius of the circle. Alternatively, the evolute curve,

$$E(t) = C(t) + N(t)/\kappa(t),$$

which is also rational, should vanish (along with all its control points) at the circle's center locations.

A surface called the *mean evolute* surface,

$$E(u, v) = S(u, v) + \frac{N(u, v)}{2M(u, v)},$$

where M is the mean curvature (see Section 2.2) is also defined in [48] and was shown to be rational for rational surface $S(u, v)$. If $S(u, v)$ is a circular cone, $E(u, v)$ is reduced to a line, the cone's center axis. Figure 20 presents two such examples. In [48], the connection is made between rational surfaces

of revolution and rational pseudo-focal surfaces (see Section 5.4) $F_u(u, v) = S(u, v) + \frac{N(u, v)}{\kappa_u(u, v)}$, where κ_u is the normal curvature of $S(u, v)$ in the u iso-parametric direction. If the u iso-parametric directions are the latitude lines of the surface of revolution, then F_u reduces to the center axis line of the surface of revolution.

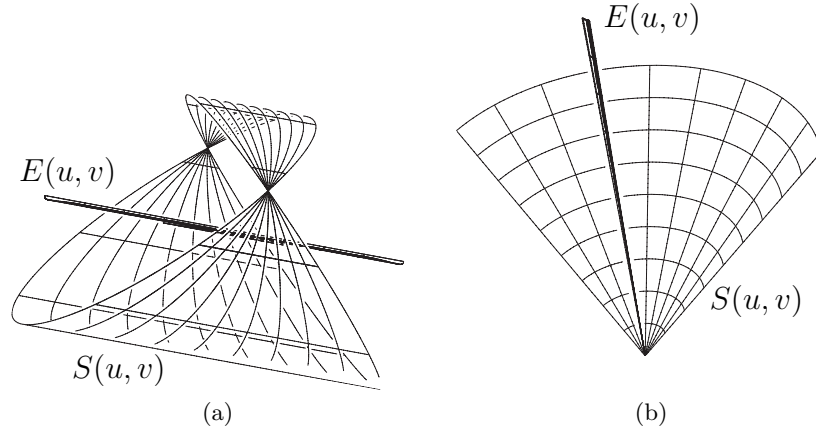


Fig. 20. The mean evolute surface reduces to the center axis line of a circular cone or cylinder. In (a), a polynomial approximation of a cylinder surface $S(u, v)$ with unconventional parameterization is presented along with its mean evolute $E(u, v)$. (b) presents a similar view of a portion of a polynomial approximation of a region of a circular cone along with its mean evolute.

For more information, see the recent book on shape interrogation in geometric design and manufacturing [149] that discusses many of the above topics as well as intersection problems, distance queries, curvature properties, and geodesics and offsets curves and surfaces.

8 Interrogation of algebraic curves and surfaces

In this section we will focus on particular geometric models: the algebraic curves and surfaces. We will show how to solve in this context some important shape interrogation problems as singularity detection, intersection problems and offset computation. It turns out that all these problems require at one point to solve an algebraic system of equations, this step being the crucial one. We thus articulate this section mainly on methods that can be applied on these algebraic systems.

Most of the curves and surfaces used in CAGD are given by parametric equations, as defined in Section 2. Planar rational curves in CAGD are typically defined as

$$x(t) = \frac{a(t)}{c(t)}, \quad y(t) = \frac{b(t)}{c(t)}$$

where $a(t)$, $b(t)$ and $c(t)$ are polynomials in the Bernstein basis for rational Bézier curves or in the B-spline basis for NURBS. Note that the algebraic methods most commonly use polynomials in the power basis and polynomials can be converted from Bernstein basis to power basis. Parametric rational surfaces in CAGD are defined by

$$x(u, v) = \frac{a(u, v)}{d(u, v)}, \quad y(u, v) = \frac{b(u, v)}{d(u, v)}, \quad z(u, v) = \frac{c(u, v)}{d(u, v)}$$

where $a(u, v)$, $b(u, v)$, $c(u, v)$ and $d(u, v)$ are polynomials.

Most of the shape interrogation problems for algebraic curves and surfaces can be translated in terms of a system of polynomial equations, as this has been widely illustrated in the previous sections (see also the extensive work of Thomas Sederberg on this topic, e.g. [172]). Consequently, methods for solving polynomial systems are required. The aim of this section is to give a quick overview of such methods. In order to be as much concrete as possible we mention the following two typical problems of shape interrogation that can easily be reduced to polynomial system solving:

Singularity detection. An important problem in CAGD is the detection of singularities of a 3D-surface. If an algebraic surface S is given implicitly by $P(x, y, z) = 0$ (that is $S = \{(x, y, z) \in \mathbb{R}^3; P(x, y, z) = 0\}$), a point (a, b, c) on S is singular if $\frac{\partial P}{\partial x}(a, b, c) = \frac{\partial P}{\partial y}(a, b, c) = \frac{\partial P}{\partial z}(a, b, c) = 0$. It is then clear that the singular points of S are the common roots of the polynomials $P, \frac{\partial P}{\partial x}, \frac{\partial P}{\partial y}, \frac{\partial P}{\partial z}$.

Computation of intersection points. Given two parametric curves, one would like to compute their intersection points. By implicitizing one of the two curves this problem is reduced to finding the real roots of a univariate polynomial which is obtained by substituting the parameterization of a curve into the implicit equation of the second one. Similar approaches can be used to compute curve/surface intersection points and more generally to compute a parameterization of an intersection surface/surface curve. Though we are manipulating objects in dimension 3, the polynomial systems that we consider might involve more than 3 variables. For instance, the intersection of 2 parametric surfaces involve 4 variables. Therefore, we are not going to restrict the number of variables in the methods that we are going to describe. Hereafter, the variables will be denoted x_1, \dots, x_n . However, since these systems come from real geometric modeling problems, we will consider only polynomials with real coefficients.

Since the problem of solving polynomial equations goes back to the ancient Greeks and Chinese, it is not surprising to see that a large number of methods exists to handle this task. Several families of solvers can however be identified:

- Analytic solvers exploit the value of a functional $f = (f_1, \dots, f_m)$ and its derivatives in order to converge to a solution or all the solutions of $f = 0$. Typical examples are Newton like methods, minimization methods, etc.
- Subdivision methods use an exclusion criterion to remove a domain if it does not contain a root of $f = 0$ or refine the search in sub-domains otherwise. These solvers are often used to isolate the real roots in a given domain.
- Algebraic solvers exploit the known relations between the unknowns. They are based on polynomial manipulations and involve effective algebraic geometry tools.

We are going to focus essentially on the two last families, which yield information on all the roots (resp. in a fixed domain).

8.1 Subdivision methods

The methods that we describe in this section, exploit the properties of Bernstein's basis for representing univariate and multivariate polynomials. The Bernstein polynomials are ubiquitous in geometric modeling. The representation of a polynomial in the Bernstein basis is known to be numerically more stable than the monomial basis representation [57, 55]. It has a direct geometric meaning, in terms of control points and useful properties such that the convex hull and the variation diminishing property. These properties in conjunction with the subdivision nature of Bernstein's polynomials explain the large variety of algorithms proposed until today for solving univariate polynomials, starting with Lane and Riesenfeld [105], up to the *Bezier clipping* methods initiated by Nishita and al [144]. They combine a global control on the domain where the roots are searched with local and efficient refinements. The situation in the multivariate case has not been studied so extensively. Two main sub-families coexist: a first family which are based on subdivision techniques like [49, 171]; a second family of solvers which are based on reduction techniques as [173]. We briefly describe these approaches, starting with univariate polynomials. For more details, see [136].

Univariate polynomials

Any polynomial $f(x) \in \mathbb{R}[x]$, of degree d , can be represented as $f(x) = \sum_{i=0}^d b_i B_i^d(x)$ where $B_i^d(x) = \binom{d}{i} (1-x)^{d-i} x^i$, $i = 0, \dots, d$ is the Bernstein basis associated to the interval $[0, 1]$. Similarly, we will say that a sequence \mathbf{b} represents the polynomial f on the interval $[r, s]$ if:

$$f(x) = \sum_{i=0}^d b_i \binom{d}{i} \frac{1}{(s-r)^n} (x-r)^i (s-x)^{d-i}.$$

The polynomials $B_d^i(x; r, s) := \binom{d}{i} \frac{1}{(s-r)^n} (x-r)^i (s-x)^{d-i}$ form the Bernstein basis on $[r, s]$. Hereafter, we are going to consider the sequence of values \mathbf{b} together with the corresponding interval $[r, s]$, as representing of our polynomial f .

A first property of this representation is that the derivative f' of f , is represented by the control coefficients: $d\Delta\mathbf{b} := d(b_{i+1} - b_i)_{0 \leq i \leq d-1}$. Another fundamental algorithm that we will use on such a representation is the de Casteljau algorithm [53]. It allows us to subdivide the representation of f into the two sub-representations on the intervals $[r, (1-x)r + xs]$ and $[(1-x)r + xs, s]$. It requires at most $2d(d+1)$ arithmetic operations. For a more detailed list of properties of this representation, we refer for instance to [53]. A simple but interesting property that we are going to use is the following:

Theorem 1 (Descartes rule). *The number of real roots of the polynomial $f(x) = \sum b_i B_d^i(x; r, s)$ in $]r, s[$ is bounded by the number $V(\mathbf{b})$ of sign changes of $\mathbf{b} = (b_i)_{i=0..n}$, and is equal modulo 2.*

As a consequence, if $V(\mathbf{b}) = 0$ there is no root in $]r, s[$ and if $V(\mathbf{b}) = 1$, there is one root in $]r, s[$. Another interesting property of this representation is the following (see e.g. [53], [164]):

Theorem 2 (Convex hull). *Let $\mathbf{b} = (b_i)_{i=0, \dots, d}$ be the control coefficients of $f(x)$ on the interval $[r, s]$ and $\mathbf{c} = (c_i)_{i=0, \dots, d}$ the corresponding control points. The graph $\{(x, f(x)); t \in [r, s]\}$ is in the convex hull of the control points \mathbf{c} .*

Multivariate polynomials

By a direct extension to the multivariate case, any polynomial $f(x_1, \dots, x_n) \in \mathbb{R}[x_1, \dots, x_n] = \mathbb{R}[\mathbf{x}]$ of degree d_i in the variable x_i , can be decomposed as:

$$f(x_1, \dots, x_n) = \sum_{i_1=0}^{d_1} \cdots \sum_{i_n=0}^{d_n} b_{i_1, \dots, i_n} B_{d_1}^{i_1}(x_1; r_1, s_1) \cdots B_{d_n}^{i_n}(x_n; r_n, s_n)$$

where $(B_{d_1}^{i_1}(x_1; r_1, s_1) \cdots B_{d_n}^{i_n}(x_n; r_n, s_n))_{0 \leq i_1 \leq d_1, \dots, 0 \leq i_n \leq d_n}$ is the tensor product Bernstein basis on the domain $\mathcal{D} := [a_1, b_1] \times \cdots \times [r_n, s_n] \subset \mathbb{R}^n$ and $\mathbf{b} = (b_{i_1, \dots, i_n})_{0 \leq i_1 \leq d_1, \dots, 0 \leq i_n \leq d_n}$ are the control coefficients of f on \mathcal{D} . The polynomial f is represented in this basis by the n^{th} order tensor of control coefficients $\mathbf{b} = (b_{i_1, \dots, i_n})_{0 \leq i \leq d_1, 0 \leq j \leq d_2, 0 \leq k \leq d_3}$. The size of \mathcal{D} , denoted by $|\mathcal{D}|$, is $|\mathcal{D}| = \max\{|s_i - r_i|; i = 1, \dots, n\}$.

De Casteljau's algorithm applies in each of the direction x_i , $i = 1, \dots, n$ so that we can split this representation in these directions. This algorithm can be used either to split the domain or to restrict the representation to

a sub-domain. For a univariate polynomial of degree d , this costs $2(d+1)d$ arithmetic operations. For a multivariate polynomial of degree d_i in x_i , we check that this restriction operation costs $2\sum_{i=1}^n d_i \prod_{i=1}^n (d_i+1) = \mathcal{O}(d^{n+1})$, where $d = \max\{d_1, \dots, d_n\}$. Thus, as the dimension and the degree increase, a good method to isolate the roots, should consider carefully when to apply this reduction operation, in order to save the computation time.

Notice that the univariate Bernstein representation also extends to the so-called triangular Bernstein basis representation. This representation can also be used in our approach, but we will concentrate on the tensor product one. For any $f \in \mathbb{R}[\mathbf{x}]$ and $j = 1, \dots, n$, let

$$m_j(f; x_j) = \sum_{i_j=0}^{d_j} \min_{\{0 \leq i_k \leq d_k, k \neq j\}} b_{i_1, \dots, i_n} B_{d_j}^{i_j}(x_j; r_j, s_j)$$

$$M_j(f; x_j) = \sum_{i_j=0}^{d_j} \max_{\{0 \leq i_k \leq d_k, k \neq j\}} b_{i_1, \dots, i_n} B_{d_j}^{i_j}(x_j; r_j, s_j).$$

We have the following property: for any $\mathbf{u} = (u_1, \dots, u_n) \in \mathcal{D}$, and any $j = 1, \dots, n$, we have

$$m(f; u_j) \leq f(\mathbf{u}) \leq M(f; u_j).$$

As a direct consequence, for any root $\mathbf{u} = (u_1, \dots, u_n)$ of the equation $f(\mathbf{x}) = 0$ in the domain \mathcal{D} , we have $\underline{\mu}_j \leq u_j \leq \bar{\mu}_j$ where

- $\underline{\mu}_j$ (resp. $\bar{\mu}_j$) is either a root of $m_j(f; x_j) = 0$ or $M_j(f; x_j) = 0$ in $[r_j, s_j]$ or r_j (resp. s_j) if $m_j(f; x_j) = 0$ (resp. $M_j(f; x_j) = 0$) has no root on $[r_j, s_j]$,
- $m_j(f; u) \leq 0 \leq M_j(f; u)$ for $u \in [\underline{\mu}_j, \bar{\mu}_j]$.

This transforms the problem of approximating the real roots of multivariate polynomials into problems on univariate polynomials.

Univariate Root Solver

Descartes rule (see theorem 1) yields a simple subdivision algorithm, which splits the domain when the number of sign variation of the control coefficients is bigger than 2. In the presence of a multiple root, the number of sign changes of a representation on any interval containing a multiple root is bigger than 2, and the algorithm splits the box until its size is smaller than a given ε . A detailed analysis of the behavior of the algorithm, has been carried out, using a partial inverse of Descartes rule given by the two circles theorem. See [140], [137], [52], [41].

This algorithm yields, in the presence of simple roots, an interval isolating the roots. But usually in practice, we are interested in approximating this root within a given precision ε . In order to approximate the isolated roots within ε , further steps of bisection may be required, either using de Casteljau's algorithm, Newton-like methods, or variants such as in [165].

Multivariate root finding

In this section, we consider a system of s equations in n variables

$$f_1(x_1, \dots, x_n) = 0, \dots, f_s(x_1, \dots, x_n) = 0$$

with coefficients in \mathbb{R} , that we will also denote by $\mathbf{f}(\mathbf{x}) = 0$. We are looking for an approximation of the real roots of $\mathbf{f}(\mathbf{x}) = 0$ in the domain $\mathcal{D} = [r_1, s_1] \times \dots \times [r_n, s_n]$, within a precision ϵ . The general framework of the families of algorithms that we will consider consists in (1) applying a preconditioning step on the equations, (2) in reducing the domain, and (3) if the reduction ratio is too small, in splitting the domain, until the size of the domain is smaller than a given epsilon. The solvers that we will consider are parameterized by the

- **Preconditioner**, that is, a transformation of the initial system $\mathbf{f} = 0$ into a system $M\mathbf{f} = 0$ (with an M invertible matrix), which has a better numerical behavior. We consider
 - **Global transformation** which aims at increasing the distance between the equations, considered as vectors of coefficients,
 - **Local straightening** which multiplies by the inverse of the Jacobian matrix at the center of the box, if it exists (it applies only for square systems).
- **Reduction strategy**, that is, the technique used to reduce the initial domain, for searching the roots of the system. We consider
 - **Convex hull reduction** as described in [173].
 - **Extreme root reduction**, which consists in computing the first (resp. last) root of the polynomial $m_j(f_k; u_j)$, (resp. $M_j(f_k; u_j)$), in the interval $[r_j, s_j]$. The improvement compared with the previous approach can be substantial (see Figure 8.1).

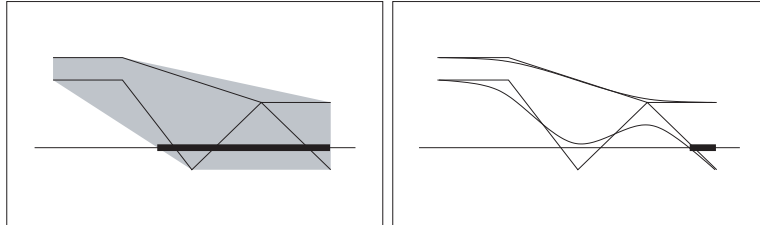


Fig. 21. Convex hull vs. extreme roots

- **Subdivision strategy**, that is, the technique used to subdivide the domain, in order to simplify the forthcoming steps, for searching of roots of the system. Here some simple rules that can be used to subdivide a domain, either in the parameter domain, or in the image.

This family of algorithms has been implemented in the C++ library SYNAPS⁵ and compared on several benchmarks. It appears that the strategy of local preconditioning with an emphasis on reduction is significantly better than the other strategies. The performances of such a solver are very good.

8.2 Algebraic methods

We now turn to algebraic methods for solving polynomial systems. We will mainly discuss the ones based on resultant matrix constructions, but also mention a method based on normal form computations which generalizes the well-known concept of Gröbner basis.

Resultant-based methods

Resultant theory. The theory of resultant is devoted to the study of conditions on the coefficients of an over-determined system, to have a solution in a fixed variety. The typical situation is the case of a system of $n + 1$ equations in a space of dimension n of the form:

$$\mathbf{f}_{\mathbf{c}} := \begin{cases} f_0(x) = \sum_{j=0}^{k_0} c_{0,j} \psi_{0,j}(x) \\ \vdots \\ f_n(x) = \sum_{j=0}^{k_n} c_{n,j} \psi_{n,j}(x) \end{cases}$$

where $\mathbf{c} = (c_{i,j})$ are parameters, x is a point of a variety X of dimension n , and the vector $\mathcal{L}_i = (\psi_{i,j})_{j=0,\dots,k_i}$ is a regular map [74] from X to the projective space \mathbb{P}^{k_i} independent of \mathbf{c} . The elimination problem consists, in this case, in finding necessary (and sufficient) conditions on \mathbf{c} such that the system $\mathbf{f}_{\mathbf{c}} = 0$ has a solution in X . From a geometric point of view, we look for the values of parameters $\mathbf{c} = (c_{i,j})$ such that there exists $x \in X$ with $\sum_{j=0}^{k_i} c_{i,j} \psi_{i,j}(x) = 0$ for $i = 0, \dots, n$. It turns out that these parameters are exactly the zero locus of a unique polynomial equation (defined up to the multiplication by a non-zero constant) in \mathbf{c} which is called the resultant of f_0, \dots, f_n and is denoted by $\text{Res}_X(\mathbf{f}_{\mathbf{c}})$. It is a quite attractive object because one can compute it through some matrix constructions.

Construction of resultant matrices. In order to construct a non-trivial multiple of $\text{Res}_X(\mathbf{f}_{\mathbf{c}})$, we apply the following strategy. A vector $L(\mathbf{c}, \mathbf{x})$ of polynomials in $\mathbb{Z}[\mathbf{c}][\mathbf{x}]$, where \mathbf{x} denotes a coordinate system of a projective space containing X , is constructed in such a way that

- (1) the polynomial entries of $L(\mathbf{c}, \mathbf{x})$ are generically independent,
- (2) the set $\mathbf{v}(\mathbf{x})$ of monomials (or polynomials) in \mathbf{x} needed to decompose all the polynomials of $L(\mathbf{c}, \mathbf{x})$ has the same size than $L(\mathbf{c}, \mathbf{x})$, and

⁵ <http://www-sop.inria.fr/galaad/software/synaps>

- (3) the polynomials in $L(\mathbf{c}, \mathbf{x})$ vanish when the input system has a common root in the variety X .

The coefficient matrix of the polynomials entries of $L(\mathbf{c}, \mathbf{x})$ with respect to the set $\mathbf{v}(\mathbf{x})$ yields a matrix $\mathbf{S}(\mathbf{c})$ whose entries are in $\mathbf{Z}[\mathbf{c}]$. Its determinant is nonzero, according to the point (1).

Above and hereafter, we use the term *generic*, which means that the property that we are considering is true on an open subset of the coefficient space for \mathbf{c} . Our aim is to construct matrices $\mathbf{S}(\mathbf{c})$, which can be used for generic systems of a certain class of polynomial equations. In practice, the problem is not posed in these terms. We are given a system of equations and it may happen that the construction we are considering yields a degenerate matrix $\mathbf{S}(\mathbf{c})$. In this case, the system is not generic for the resultant formulation and we have to chose another class of systems for which we are in a generic position. This explains why a lot of different types of resultant formulations have been studied; we will give a list in a moment.

By construction, we have $\mathbf{v}(\mathbf{x})^t \mathbf{S}(\mathbf{c}) = L(\mathbf{c}, \mathbf{x})^t$. Thus, according to the point (2), if ζ is a common root of a specialized system $\mathbf{f}_{\mathbf{c}_0} = \mathbf{0}$, we have $L(\mathbf{c}_0, \zeta) = \mathbf{0}$ and $\mathbf{v}(\zeta)^t \mathbf{S}(\mathbf{c}_0) = \mathbf{0}$. If (generically) $\mathbf{v}(\zeta)$ is not zero at a common root $\zeta \in X$ of $\mathbf{f}_{\mathbf{c}_0} = \mathbf{0}$, we deduce that $\det(\mathbf{S}(\mathbf{c}))$ vanishes when the system has a common root in X . Therefore $\det(\mathbf{S}(\mathbf{c}))$ is a non-trivial multiple of its equation, that is of the resultant $\text{Res}_X(\mathbf{f}_{\mathbf{c}})$.

A usual way to construct these resultant matrices (which extends Sylvester's construction for the classical resultant of two univariate polynomial), consists in choosing for L , a list of monomial multiples of the polynomials f_i . In this case, the matrix $\mathbf{S}(\mathbf{c})$ is the matrix of a map of the form

$$\begin{aligned} \mathcal{S} : \langle \mathbf{x}^{E_0} \rangle \times \cdots \times \langle \mathbf{x}^{E_n} \rangle &\rightarrow \langle \mathbf{x}^F \rangle \\ (q_0, \dots, q_n) &\mapsto g = \sum_{i=0}^n q_i f_i \end{aligned}$$

where $\langle \mathbf{x}^{E_i} \rangle$ is the vector subspace generated by a specific set of monomials \mathbf{x}^{E_i} . The entries of $\mathbf{S}(\mathbf{c})$ are filled as follows: every column of \mathbf{S} is indexed by an element of some $\{E_i\}_{i=0, \dots, n}$ and every row by an element of F ; equivalently, the columns and rows are indexed by the monomials of q_i and the monomials of g , respectively. The coefficient in the row corresponding to $\beta \in F$ and in the column corresponding to $\alpha \in E_j$ is the coefficient of \mathbf{x}^β in $\mathbf{x}^\alpha f_j$. The coefficients of monomials which do not explicitly appear in $\mathbf{x}^\alpha f_j$ have a zero entry. Thus the matrix $\mathbf{S}(\mathbf{c})$ is divided into blocks $(\mathbf{S}_0, \dots, \mathbf{S}_n)$, where each block \mathbf{S}_i depends linearly on the coefficients of f_i . Notice that such resultant matrices have a quasi-Toeplitz structure which can be exploited to accelerate many computations with them, by almost one order of magnitude in terms of the matrix dimension. This relies on FFT for fast multivariate polynomial and dense structured matrix arithmetic; a smaller acceleration is achieved by applying Karatsuba's divide-and-conquer arithmetic [135, 51].

Different resultant formulations. We recall briefly different resultant formulations which can be used in geometric problems. A detailed description is beyond the scope of this book. The formulation will be chosen according to the geometric properties of the problem to solve.

- **multivariate resultants:** They correspond to the classical case studied in [115, 187]. Here X is the projective space \mathbb{P}^n , \mathcal{L}_i is the vector of all monomials of a fixed degree d_i , and the function f_i is a generic homogeneous polynomial of degree d_i . The most used resultant matrix in this context is the Macaulay’s construction [115] which can be seen as an extension of the Sylvester’s method to the multivariate case. However, some other multivariate resultant matrices have been developed (see e.g. [90, 34]).

- **toric (or sparse) resultants:** They have been introduced in [92], then developed in [62]. It takes into account the monomial support of the input polynomials of a system, and not only their respective degree. Thus it is possible to work with polynomials having negative exponents, that is *Laurent polynomials*. Methods for constructing toric resultant matrices can be found in [22, 50, 33].

- **Residual resultants:** In many situations coming from practical problems, the equations have common zeroes which are independent of the parameters of the problems, and which we are not interested in. The residual resultant constructions has been designed to take into account these degenerate cases. It is described, as well as matrix construction, in [20, 18, 21]. A more general construction has been developed in [19] whose associated matrix construction is the so-called called Bezoutian matrix.

Solving polynomial systems via eigenvalues computations. Let $f_0(x)$, $f_1(x)$, \dots , $f_n(x)$ be polynomials in n variables $x = (x_1, \dots, x_n)$. By choosing an adapted resultant formulation one can construct a resultant matrix \mathcal{S} associated to this system. It turns out that this matrix can be divided into four blocs $\mathcal{S} = \begin{pmatrix} \mathcal{S}_{00} & \mathcal{S}_{01} \\ \mathcal{S}_{10} & \mathcal{S}_{11} \end{pmatrix}$ and that the Schur complement $\mathcal{S}_{00} - \mathcal{S}_{01}\mathcal{S}_{11}^{-1}\mathcal{S}_{10}$ is nothing but the matrix of the multiplication map by $f_0(x)$ in a canonical basis of the quotient ring $\mathbb{R}[x]/(f_1, \dots, f_n)$. The point is that the eigenvalues of such a multiplication matrix are particularly interesting, they are the evaluation of f_0 at the common root of f_1 and f_2 . If f_0 is a linear form one can thus easily solve the polynomial system $f_1(x) = f_2(x) = 0$.

Solving polynomial systems by hiding a variable. Another approach to solve a system of polynomial equations consists in *hiding* a variable (that is, in considering one of variables as a *parameter*), and in searching the value of this hidden variable for which the system has a solution. Typically, if we have n equations $f_1 = 0, \dots, f_n = 0$ in n variables, we “hide” a variable, say x_n , and apply one of resultant constructions described before to the over-determined system $f_1 = 0, \dots, f_n = 0$ in the $n - 1$ variables x_1, \dots, x_{n-1} and a parameter x_n . This leads to a resultant matrix $\mathcal{S}(x_n)$ with polynomial entries in x_n . It can be decomposed as

$$\mathbf{S}(x_n) = \mathbf{S}_d x_n^d + \mathbf{S}_{d-1} x_n^{d-1} + \cdots + \mathbf{S}_0,$$

where \mathbf{S}_i has coefficients in \mathbb{R} and the same size as $\mathbf{S}(x_n)$. We look for the values ζ_n of x_n for which the system has a *solution* $\zeta' = (\zeta_1, \dots, \zeta_{n-1})$ in the corresponding variety X' (of dimension $n - 1$) associated with the resultant formulation. This implies that

$$\mathbf{v}(\zeta')^t \mathbf{S}(\zeta_n) = \mathbf{0}, \quad (7)$$

where $\mathbf{v}(\zeta')$ is the vector of monomials indexing the rows of \mathbf{S} evaluated at ζ' . Conversely, for generic systems of the corresponding resultant formulation there is only one point ζ' above the value ζ_n . Thus the vectors \mathbf{v} satisfying $\mathbf{S}(\zeta_n)^t \mathbf{v} = \mathbf{0}$ are scalar multiples of $\mathbf{v}(\zeta')$. From the entries of these vectors, we can deduce the other coordinates of the point ζ' . This will be assumed hereafter⁶.

The relation (7) implies that $\mathbf{v}(\zeta')$ is a generalized eigenvector of $\mathbf{S}^t(x_n)$. Computing such vectors can be transformed into the following linear generalized eigenproblem

$$\left(\begin{bmatrix} 0 & \mathbb{I} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbb{I} \\ \mathbf{S}_0^t & \mathbf{S}_1^t & \cdots & \mathbf{S}_{d-1}^t \end{bmatrix} - \zeta_n \begin{bmatrix} \mathbb{I} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbb{I} & 0 \\ 0 & \cdots & 0 & -\mathbf{S}_d^t \end{bmatrix} \right) \mathbf{w} = \mathbf{0}. \quad (8)$$

The set of eigenvalues of (8) contains the values of ζ_n for which (7) has a solution. The corresponding eigenvectors \mathbf{w} are decomposed as $\mathbf{w} = (\mathbf{w}_0, \dots, \mathbf{w}_{d-1})$ so that the solution vector $\mathbf{v}(\zeta')$ of (7) is

$$\mathbf{v}(\zeta') = \mathbf{w}_0 + \zeta_n \mathbf{w}_1 + \cdots + \zeta_n^{d-1} \mathbf{w}_{d-1}.$$

Normal forms

Gröbner basis is a powerful tool to handle a lot of computations on polynomial systems. However their construction is not numerically stable, they may introduce artificial discontinuities due to the choice of a monomial order. A recent generalization of this notion has been proposed in [134, 138, 139]. It is based on a new criterion which gives a necessary and sufficient condition for a projection onto a vector subspace of R to be a normal form modulo the ideal I . More precisely we have:

Theorem 3. *Let B be a vector space in $R = \mathbb{R}[x_1, \dots, x_n]$ connected to the constant polynomial 1 ⁷. If B^+ is the vector subspace generated by $B \cup x_1 B \cup$*

⁶ Notice however that this genericity condition can be relaxed by using duality, in order to compute the points ζ' above ζ_n (when they form a zero-dimensional fiber) from the eigenspace of $\mathbf{S}(\zeta_n)$.

⁷ Any monomial $\mathbf{x}^\alpha \neq 1 \in B$ is of the form $x_i \mathbf{x}^\beta$ with $\mathbf{x}^\beta \in B$ and some i in $\{1, \dots, n\}$.

$\dots \cup x_n B$, $N : B^+ \rightarrow B$ is a linear map such that N is the identity on B , we define for $i = 1, \dots, n$, the maps

$$\begin{aligned} M_i : B &\rightarrow B \\ b &\mapsto M_i(b) := N(x_i b). \end{aligned}$$

The two following properties are equivalent:

1. For all $1 \leq i, j \leq n$, $M_i \circ M_j = M_j \circ M_i$.
2. $R = B \oplus I$, where I is the ideal generated by the kernel of N

If this holds, the B -reduction along $\ker(N)$ is canonical.

This leads to a completion-like algorithm which starts with the linear subspace K_0 generated by the polynomials f_1, \dots, f_m , that we want to solve and iterates the construction $K_{i+1} = K_i^+ \cap L$, where L is a fixed vector space. We stop when $K_{i+1} = K_i$. See [134, 138, 186, 139] for more details. This approach allows us to fix first the set of monomials on which we want to do linear operations and thus to treat more safely polynomials with approximate coefficients. It can be adapted very naturally to Laurent polynomials, which is not the case for Gröbner basis computation. Moreover it can be specialized very efficiently to systems of equations for which the basis of \mathcal{A} is known a priori, such as in the case of a complete projective intersection [138]. Let us see how we can deduce the roots from this normal form computation. For this purpose, we will use the properties of the operators of multiplication by elements of $\mathcal{A} = R(f_1, \dots, f_m)$. For any $a \in \mathcal{A}$, we define

$$\begin{aligned} M_a : \mathcal{A} &\rightarrow \mathcal{A} \\ b &\mapsto M_a(b) := a b. \end{aligned}$$

We also consider its transpose operator

$$\begin{aligned} M_a^t : \widehat{\mathcal{A}} &\rightarrow \widehat{\mathcal{A}} \\ \Lambda &\mapsto M_a^t(\Lambda) = \Lambda \circ M_a, \end{aligned}$$

where the dual space $\widehat{\mathcal{A}}$ is the set of \mathbb{R} -linear forms from \mathcal{A} to \mathbb{R} . The matrix of M_a^t in the dual basis of a basis B of \mathcal{A} is the transpose of the matrix of M_a in B . The multiplication operators can be computed using a normal form algorithm, as described above.

Hereafter, $\mathbf{x}^E = (\mathbf{x}^\alpha)_{\alpha \in E}$ denotes a monomial basis of \mathcal{A} (for instance obtained by a Gröbner basis). Then any polynomial can be reduced modulo (f_1, \dots, f_m) to a linear combination of monomials of \mathbf{x}^E .

The matrix approach to solve polynomial systems is based on the following fundamental theorem [3], [133]:

Theorem 4. Assume that $\mathcal{Z}(I) = \{\zeta_1, \dots, \zeta_d\}$. We have

1. Let $a \in \mathcal{A}$. The eigenvalues of the operator M_a (and its transpose M_a^t) are $a(\zeta_1), \dots, a(\zeta_d)$.

2. The common eigenvectors of $(M_a^t)_{a \in \mathcal{A}}$ are (up to a scalar) the evaluations $\mathbf{1}_{\zeta_1}, \dots, \mathbf{1}_{\zeta_d}$.

Since $\mathbf{x}^E = (\mathbf{x}^\alpha)_{\alpha \in E}$ is a basis of \mathcal{A} , the coordinates of $\mathbf{1}_{\zeta_i}$ in the dual basis of \mathbf{x}^E are $(\zeta_i^\alpha)_{\alpha \in E}$. Thus if \mathbf{x}^E contains $1, x_1, \dots, x_n$ (which is often the case), we can deduce directly all the coordinates of the roots. We have the following algorithm:

Algorithm 1 *Solving in the case of simple roots. Let $a \in \mathcal{A}$ such that $a(\zeta_i) \neq a(\zeta_j)$ for $i \neq j$ (which is generically the case) and M_a be the matrix of multiplication by a in the basis $\mathbf{x}^E = (1, x_1, \dots, x_n, \dots)$ of \mathcal{A} .*

1. Compute the eigenvectors $\Lambda = (\Lambda_1, \Lambda_{x_1}, \dots, \Lambda_{x_n}, \dots)$ of M_a^t .
2. For each eigenvector Λ with $\Lambda_1 \neq 0$, compute and output the point $\zeta = \left(\frac{\Lambda_{x_1}}{\Lambda_1}, \dots, \frac{\Lambda_{x_n}}{\Lambda_1} \right)$.

The set of output points ζ contains the simple roots (i.e. roots with multiplicity 1) of $f = 0$, since for such a root the eigenspace associated to the eigenvalue $a(\zeta)$ is one-dimensional and contains $\mathbf{1}_\zeta$. But as we will see in the next example, it can also yield in some cases the multiple roots.

In order to compute exactly the set of roots counted with their multiplicity, we use the following result. It is based on the fact that commuting matrices share common eigenspaces. [133, 135, 31].

Theorem 5. *There exists a basis of \mathcal{A} such that for all $a \in \mathcal{A}$, the matrix of M_a in this basis is of the form*

$$M_a = \begin{pmatrix} N_a^1 & \mathbf{0} \\ & \ddots \\ \mathbf{0} & N_a^d \end{pmatrix} \quad \text{with} \quad N_a^i = \begin{pmatrix} a(\zeta_i) & \star \\ & \ddots \\ \mathbf{0} & a(\zeta_i) \end{pmatrix}.$$

We deduce the algorithm:

Algorithm 2 *Solving by simultaneous triangulation.*

INPUT: Matrices of multiplication $M_{x_i}, i = 1, \dots, n$, in a basis of \mathcal{A} .

1. Compute a (Schur) decomposition P such that the matrices $T_i = PM_{x_i}P^{-1}$, $i = 1, \dots, n$, are upper-triangular.
2. Compute and output the diagonal vectors $\mathbf{t}_i = (t_{i,i}^1, \dots, t_{i,i}^n)$ of triangular matrices $T_k = (t_{i,j}^k)_{i,j}$.

OUTPUT: $\mathcal{Z}(I) = \{\mathbf{t}_i : i = 1, \dots, \dim_{\mathbb{R}}(\mathcal{A})\}$.

The first step in this algorithm is performed by computing a Schur decomposition of M_l (where l is a generic linear form) which yields a matrix P of bases change. Then we compute the triangular matrices $T_i = PM_{x_i}P^{-1}$, $i = 1, \dots, n$, since they commute with M_l . An implementation by Ph. Trébuchet of this algorithm is available in the SYNAPS library (see `solve(L,newmac<C>())`).

9 Conclusion

Shape interrogation methods are still of increasing interest in geometric modeling as well as in computer graphics. Originating 20 years ago from CAD/CAM applications where "class A" surfaces are required and no surface imperfections are allowed, shape interrogation has become recently an important tool for various other types of surface representations such as triangulated or polygonal surfaces, subdivision surfaces, and algebraic surfaces. In this chapter, we presented the state-of-the-art of shape interrogation methods including methods for detecting surface imperfections, surface analysis tools and methods for visualizing intrinsic surface properties. Furthermore we focused on stable numerical and symbolic solving of algebraic systems of equations, a problem that arises in most shape interrogation methods. Nevertheless, many issues are still open promising intensive research in various areas of shape interrogation. Let us focus on some of them now.

Discrete geometry representations are frequently used in many applications, especially for shapes acquired from real-world objects. Typically, surfaces are approximated by polygonal meshes, and we showed how to estimate differential properties for piecewise linear surfaces. Various methods exist so far, and recent approaches prove approximation and convergence properties. The design of robust methods coming with certain guarantees is still an area of active research.

In the area of algebraic and numerical polynomial system solvers, that provide one of the basic tools for shape interrogation methods, improvements are indispensable. Many critical problems in Computer Aided Geometric Design, such as shape interrogation, are reduced to finding the zero set of a system of polynomial equations. Several root-finding methods for polynomial systems exist, even if we mainly presented resultant-based methods and subdivision methods. A wide choice of techniques and algorithms to solve polynomial systems are thus now available, but as a main drawback, all of these methods have difficulties in handling roots with high multiplicities (or clusters of roots). They all have performance deterioration, lack of robustness in numerical computation and round-off errors during floating point arithmetic in such a situation. It is hence a crucial objective and an active research area to improve root-finding methods in this case which often occurs in practice.

In the area of symbolic curve and surface interrogation future/open problems can be addressed. Volumetric data sets are now used in many applications and serves as a prime candidate representation in medical applications. The extension of curve and surface interrogation methods to support volumetric representation, either as iso-surfaces in the volumes or direction analyze differential properties are highly desired. The degrees of many of these rational fields such as the Gaussian curvature, K , or the mean curvature square, M^2 , are high. Methods to robustly handle these fields, in a more stable way, could further improve the quality of the result.

References

1. P. Alliez, M. Attene, C. Gotsman, and G. Ucelli. Recent advances in remeshing surfaces. In *Shape Analysis and Structuring*. Springer, 2007.
2. P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun. Anisotropic polygonal remeshing. *ACM Transactions on Graphics*, 22(3):485–493, July 2003.
3. W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Proc. Intern. Conf. on Numerical Math.*, volume 86 of *Int. Series of Numerical Math.*, pages 12–30. Birkhäuser Verlag, 1988.
4. R. Barnhill, G. Farin, L. Fayard, and H. Hagen. Twists, curvature and surface interrogation. *CAD*, 20:314–346, 1988.
5. J. Beck, R. Farouki, and J. Hinds. Surface analysis methods. *IEEE CG & Appl.*, 6:19–35, 1986.
6. K.-P. Beier. The porcupine technique: principles, applications, and algorithms. Technical report, University of Michigan, 1987.
7. K.-P. Beier and Y. Chen. Highlight-line algorithm for realtime surface quality assessment. *CAD*, 26(4):268–277, 1994.
8. A. G. Belyaev, E. V. Anoshkina, and T. L. Kunii. Ridges, ravines, and singularities. In A. T. Fomenko, and T. L. Kunii, *Topological Modeling for Visualization*, pages 375–383. Springer, 1997. Chapter 18.
9. A. G. Belyaev and Y. Ohtake. An image processing approach to detection of ridges and ravines on polygonal surfaces. In *Eurographics 2000, Short Presentations*, pages 19–28, August 2000.
10. A. G. Belyaev, A. A. Pasko, and T. L. Kunii. Ridges and ravines on implicit surfaces. In *Proc. Computer Graphics International 1998*, pages 530–535, 1998.
11. M. V. Berry and J. H. Hannay. Umbilic points on gaussian random surfaces. *J. Phys. A*, 10:1809–21, 1977.
12. P. J. Besl and R. C. Jain. Invariant surface characteristics for 3D object recognition in range images. *Comput. Vision Graph. Image Process*, 33(1):33–80, 1986.
13. I. A. Bogaevski, V. Lang, A. G. Belyaev, and T. L. Kunii. Color ridges on implicit polynomial surfaces. In *GraphiCon 2003 Proceedings*, pages 161–164, September 2003.
14. V. Borrelli, F. Cazals, and J. M. Morvan. On the angular defect of triangulations and the pointwise approximation of curvatures. *Computer Aided Geometric Design*, 20(6):319–341, 2003.
15. J. W. Bruce, P. J. Giblin, and F. Tari. Ridges, crests and sub-parabolic lines of evolving surfaces. *International Journal of Computer Vision*, 18(3):195–210, 1996.
16. J. W. Bruce, P. J. Giblin, and F. Tari. Families of surfaces: focal sets, ridges and umbilics. *Math. Proc. Camb. Phil. Soc.*, 125:243–268, 1999.
17. J. W. Bruce and T. C. Wilkinson. Folding maps and focal sets. In M. Dæhlen, T. Lyche, and L. L. Schumaker, editors, *Proceedings of Warwick Symposium on Singularities, Springer Lecture Notes in Math., vol 1462*, pages 63–72, Berlin and New York, 1991. Springer-Verlag.
18. L. Busé. Residual resultant over the projective plane and the implicitization problem. In *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation*, pages 48–55 (electronic), New York, 2001. ACM.

19. L. Busé, M. Elkadi, and B. Mourrain. Generalized resultants over unirational algebraic varieties. *J. Symbolic Comput.*, 29(4-5):515–526, 2000. Symbolic computation in algebra, analysis, and geometry (Berkeley, CA, 1998).
20. L. Busé, M. Elkadi, and B. Mourrain. Resultant over the residual of a complete intersection. *J. Pure Appl. Algebra*, 164(1-2):35–57, 2001. Effective methods in algebraic geometry (Bath, 2000).
21. L. Busé, M. Elkadi, and B. Mourrain. Using projection operators in computer aided geometric design. In *Topics in algebraic geometry and geometric modeling*, volume 334 of *Contemp. Math.*, pages 321–342. Amer. Math. Soc., Providence, RI, 2003.
22. J. Canny and P. Pedersen. An algorithm for the Newton resultant. Technical Report 1394, Comp. Science Dept., Cornell University, 1993.
23. C. Catalano and I. Ivriissimtzis. Subdivision surfaces. In *Shape Analysis and Structuring*. Springer, 2007.
24. F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *Symposium on Geometry Processing*, pages 177–187, 2003.
25. F. Cazals and M. Pouget. Ridges and umbilics of a sampled smooth surface: a complete picture gearing toward topological coherence. Rapport de Recherche RR-5294, INRIA, September 2004.
26. F. Cazals and M. Pouget. Smooth surfaces, umbilics, lines of curvatures, foliations, ridges and the medial axis: a concise overview. Rapport de Recherche RR-5138, INRIA, March 2004.
27. S. Chan and E. Purisima. A new tetrahedral tessellation scheme for isosurface generation. *Computers & Graphics*, 22(1):83–90, 1998.
28. E. Cohen, R. Riesenfeld, and G. Elber. *Geometric Modeling with Splines: An Introduction*. AK Peters, 2001.
29. D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, 23(3):905–914, August 2004. Proceedings of SIGGRAPH 2004.
30. D. Cohen-Steiner and J.-M. Morvan. Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth Conference on Computational Geometry (SCG-03)*, pages 312–321, June 8–10 2003.
31. R. M. Corless, P. M. Gianni, and B. M. Trager. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In W.W. Küchlin, editor, *Proc. ISSAC*, pages 133–140, 1997.
32. P. Csákány and A. M. Wallace. Computation of local differential parameters on irregular meshes. In R. Cipola and R. Martin, editors, *The Mathematics of Surfaces IX*, pages 19–33. Springer, 2000.
33. C. D’Andrea. Macaulay style formulas for sparse resultants. *Trans. Amer. Math. Soc.*, 354(7):2595–2629 (electronic), 2002.
34. C. D’Andrea and A. Dickenstein. Explicit formulas for the multivariate resultant. *J. Pure Appl. Algebra*, 164(1-2):59–86, 2001. Effective methods in algebraic geometry (Bath, 2000).
35. G. Darboux. *Leçons sur la théorie générale des surfaces, Tome 4*. Gauthier-Villars, Paris, 1896.
36. D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *ACM Trans. on Graphics*, 22(3):848–855, 2003. Proc. ACM SIGGRAPH 2003.

37. M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Computer Graphics (Proceedings of SIGGRAPH 99)*, pages 317–324, 1999.
38. J. Dill. An application of color graphics to the display of surface curvature. *Computer Graphics*, 15(3):153–161, 1981.
39. P. M. Do Carmo. *Differential Geometry of curves and surfaces*. Prentice-Hall, Englewood Cliffs, 1976.
40. D. Eberly. *Ridges in Image and Data Analysis*. Kluwer, 1996.
41. Arno Eigenwillig, Vikram Sharma, and Chee K. Yap. Almost tight recursion tree bounds for the descartes method. In *ISSAC '06: Proceedings of the 2006 international symposium on Symbolic and algebraic computation*, pages 71–78, New York, NY, USA, 2006. ACM Press.
42. L. P. Eisenhart. *An introduction to differential geometry*. Princeton University Press, Princeton, N.J., 1976.
43. G. Elber. Free form surface analysis using a hybrid of symbolic and numerical computation. *PhD thesis, Department of Computer Science, The University of Utah*, 1992.
44. G. Elber. Freeform surface region optimization for three- and five-axis milling. *Computer Aided Design*, 27(6):465–470, June 1995.
45. G. Elber. Symbolic and numeric computation in curve interrogation. *Computer Graphics forum*, 14(1):25–34, March 1995.
46. G. Elber. Curve evaluation and interrogation on surfaces. *The Journal of Graphical Models*, 63(3):197–210, May 2001.
47. G. Elber and E. Cohen. Second-order surface analysis using hybrid symbolic and numeric operators. *ACM Trans. on Graphics*, 12(2):160–178, 1993.
48. G. Elber and M.-Soo Kim. Geometric shape recognition of freeform curves and surfaces. *Graphics Models and Image Processing*, 59(6):417–433, November 1997.
49. G. Elber and M.-Soo Kim. Geometric constraint solver using multivariate rational spline functions. In *Proceedings of the sixth ACM Symposium on Solid Modelling and Applications*, pages 1–10. ACM Press, 2001.
50. I. Emiris and J. Canny. A practical method for the sparse resultant. In M. Bronstein, editor, *Proc. Intern. Symp. on Symbolic and Algebraic Computation*, pages 183–192, Kiev, July 1993.
51. I. Z. Emiris and V. Y. Pan. Symbolic and numeric methods for exploiting structure in constructing resultant matrices. *J. Symbolic Comput.*, 33(4):393–413, 2002.
52. Ioannis Z. Emiris, Bernard Mourrain, and Elias P. Tsigaridas. Real algebraic numbers: Complexity analysis and experimentations. Research Report 5897, INRIA, Avril 2006.
53. G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, New York, 4th edition, 1996.
54. G. Farin and N. Sapidis. Curvature and the fairness of curves and surfaces. *IEEE CG & Appl.*, 9:52–57, 1989.
55. R. Farouki and V. Rajan. On the numerical condition of polynomials in bernstein form. *Computer Aided Geometric Design*, 4(3):191–216, 1987.
56. R. T. Farouki. Graphical methods for surface differential geometry. In R. Martin, editor, *in Mathematics of surfaces*, pages 363–385. IMA Series, 1987.
57. R. T. Farouki and T. N. T. Goodman. On the optimal stability of the bernstein basis. *Mathematics of computation*, 65(216):1553–1566, October 1996.

58. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics. Principles and Practice*. Addison-Wesley, 1990. 2nd edition.
59. A. Forrest. On the rendering of surfaces. *Computer Graphics*, pages 253–259, 1979.
60. P. J. Frey and H. Boroucraki. Surface mesh quality evaluation. *International Journal for Numerical Methods in Engineering*, 45:101–118, 1999.
61. I. Friedel, P. Schröder, and A. Khodakovsky. Variational normal meshes. *ACM Transactions on Graphics*, 23(4):1061–1073, 2004.
62. I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, resultants, and multidimensional determinants*. Mathematics: Theory & Applications. Birkhäuser Boston Inc., Boston, MA, 1994.
63. A. S. Glassner. Computing surface normals for 3D models. In A. S. Glassner, editor, *Graphics Gems*, pages 562–566. Academic Press, 1990.
64. J. Goldfeather and V. Interrante. A novel cubic-order algorithm for approximating principal directions vectors. *ACM Transactions on Graphics*, 23(1):45–63, 2004.
65. G. G. Gordon. Face recognition from depth maps and surface curvature. In *Geometric Methods in Computer Vision, Proc. SPIE 1570*, pages 234–247, 1991.
66. H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, C-20(6):623–629, 1971.
67. U. Grenader and M. I. Miller. Computational anatomy: An emerging discipline. *Quarterly of Applied Mathematics*, 56(4):617–694, 1998.
68. A. Guézic, X. Pennec, and N. Ayache. Medical image registration using geometric hashing. *IEEE Comput. Sci. Eng.*, 4(4):29–41, 1997.
69. A. Gullstrand. Zur Kenntnis der Kreispunkte. *Acta Mathematica*, 29:59–100, 1904.
70. S. Gumhold, X. Wang, and R. McLeod. Feature extraction from point clouds. In *Proc. 10th International Meshing Roundtable*, pages 293–305, Sandia National Laboratories, Newport Beach, CA, 2001.
71. H. Hagen and S. Hahmann. Generalized focal surfaces : A new method for surface interrogation. In *Proceedings Visualization'92*, pages 70–76. IEEE, 1992.
72. P. L. Hallinan, G. G. Gordon, A. L. Yuille, P. Giblin, and D. Mumford. *Two- and Tree-Dimensional Patterns of the Face*. A K Peters, 1999.
73. B. Hamann. Curvature approximation for triangulated surfaces. *Computing Suppl.*, 8:139–153, 1993.
74. J. Harris. *Algebraic geometry*, volume 133 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1992. A first course.
75. E. Hartmann. On the curvature of curves and surfaces defined by normalforms. *Computer Aided Geometric Design*, 16(5):355–376, 1999.
76. R. Hartwig and H. Nowacki. Isolinien und schnitte in coonschen flächen. *Geometrisches Modellieren 65*, Informatik Fachberichte der GI, 1982.
77. R. A. Herman. *A Treatise on Geometrical Optics*. Cambridge University Press, 1900.
78. M. Higashi, T. Saitoh, and Y.. Watanabe. Analysis of aesthetic free-form surfaces by surface edges. In *Pacific Graphics '95*, pages 294–305, 1995.
79. D. Hilbert and S. Cohn-Vossen. *Geometry and the imagination*. Chelsea Publishing Company, New York, 1952.
80. K. Hildebrandt and K. Polthier. Anisotropic filtering of non-linear surface features. In *Proc. Eurographics*, pages 391–400, 2004.

81. K. Hildebrandt, K. Polthier, and M. Wardetzky. Smooth feature lines on surface meshes. In *Third Eurographics Symposium on Geometry Processing*, pages 85–90, July 2005.
82. M. Hisada, A. G. Belyaev, and T. L. Kunii. A skeleton-based approach for detection of perceptually salient features on polygonal surfaces. *Computer Graphics Forum*, 21(4):689–700, 2002.
83. D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1985.
84. M. Hosaka. *Modeling of Curves and Surfaces in CAD/CAM*. Springer, Berlin, 1992.
85. J. Hoschek. Detecting regions with undesirable curvature. *CAGD*, 1:183–192, 1984.
86. J. Hoschek. Smoothing of curves and surfaces. *CAGD*, 2:97–105, 1985.
87. J. Hoschek, U. Dietz, and W. Wilke. A geometric concept of reverse engineering of shape: Approximation and feature lines. In M. Dæhlen, T. Lyche, and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces II*, pages 253–262. Vanderbilt Univ. Press, 1998.
88. A. Hubeli and M. Gross. Multiresolution feature extraction from unstructured meshes. In *Proc. IEEE Visualization 2001*, pages 287–294, 2001.
89. V. Interrante, H. Fuchs, and S. Pizer. Enhancing transparent skin surfaces with ridge and valley lines. In *Proc. IEEE Visualization 1995*, pages 52–59, 1995.
90. J. P. Jouanolou. Formes d’inertie et résultant: un formulaire. *Adv. Math.*, 126(2):119–250, 1997.
91. S. Kapoor. Efficient computation of geodesic shortest paths. In *STOC ’99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 770–779, New York, NY, USA, 1999. ACM Press.
92. M. M. Kapranov, B. Sturmfels, and A. V. Zelevinsky. Chow polytopes and general resultants. *Duke Math. J.*, 67(1):189–218, 1992.
93. S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph.*, 22(3):954–961, 2003.
94. E. Kaufmann and R. Klass. Smoothing surfaces using reflection lines for families of splines. *CAD*, 20:312–316, 1988.
95. J. T. Kent, D. Lee, Mardia K. V., and A. D. Linney. Using curvature information in shape analysis. In K. V. Mardia, G. A. Gill, and I. L. Dryden, editors, *Proc. Image Fusion and Shape Variability Techniques*, pages 88–99. Leeds University Press, 1996.
96. J. T. Kent, K. V. Mardia, and J. West. Ridge curves and shape analysis. In *The British Machine Vision Conference 1996*, pages 43–52, 1996.
97. J. Kjellander. Smoothing of bicubic parametric surfaces. *CAD*, 15:288–293, 1983.
98. R. Klass. Correction of local irregularities using reflection lines. *CAD*, 12:73–77, 1980.
99. K. H. Ko, T. Maekawa, N. M. Patrikalakis, H. Masuda, and F.-E. Wolter. Shape intrinsic fingerprints for free-form object matching. In *Proc. of 8th ACM Symposium on Solid Modeling and Applications*, pages 196 – 207, 2003.
100. L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of SIGGRAPH 98*, pages 105–114, 1998.

101. J. J. Koenderink. *Solid Shape*. MIT Press, 1990.
102. I. Kreyszig. *Differential Geometry*. Univ. of Toronto Press, Toronto, 1959.
103. V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *SIGGRAPH 96 Conference Proceedings*, pages 313–324, New York, NY, USA, 1996. ACM Press.
104. P. Krsek, G. Lukacs, and R. R. Martin. Algorithms for computing curvatures from range data. In R. Cripps, editor, *The Mathematics of Surfaces VIII*, pages 1–16. IMA, 1998.
105. J. Lane and R. Riesenfeld. Bounds on a polynomial. *BIT*, 21:112–117, 1981.
106. T. Langer, A. Belyaev, and H.-P. Seidel. Asymptotic analysis of discrete normals and curvatures of polylines. In *SCCG '05: Proceedings of the 21st spring conference on Computer graphics*, pages 229–232, 2005.
107. T. Langer, A. Belyaev, and H.-P. Seidel. Exact and approximate quadratures for curvature tensor estimation. In Gnther Greiner, Joachim Hornegger, Heinrich Niemann, and Marc Stamminger, editors, *Vision, Modeling, and Visualization 2005 (VMV'05)*, pages 421–428, 2005.
108. R. B. Lee and D. A. Fredericks. Intersection of parametric surfaces and a plane. *IEEE CG & Appl.*, 4(8):48–51, 1984.
109. J. J. Little and P. Shi. Structural lines, TINs and DEMs. *Algorithmica*, 30(2):243–263, 2001.
110. A. M. López, F. F. Lumbreras, and J. Serrat. Creaseness from level set extrinsic curvature. In *Proc. ECCV'98*, pages 156–169. Springer, 1998.
111. W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH 87 Conference Proceedings*, pages 163–169, New York, NY, USA, 1987. ACM Press.
112. C. Lu, Y. Cao, , and D. Mumford. Surface evolution under curvature flows. *Journal of Visual Communication and Image Representation*, 13(1/2):65–81, March/June 2002.
113. G. Lukács and L. Andor. Computing natural division lines on free-form surfaces based on measured data. In M. Dæhlen, T. Lyche, and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces II*, pages 319–326. Vanderbilt Univ. Press, 1998.
114. K.-L. Ma and V. Interrante. Extracting feature lines from 3D unstructured grids. In *Proc. IEEE Visualization 1997*, pages 285–292, 1997.
115. F. S. Macaulay. Some formulae in elimination. *Proc. London Math. Soc.*, 1(33):3–27, 1902.
116. T. Maekawa and Patrikalakis M. Interrogation of differential geometry properties for design and manufacture. *Visual Computer*, 10:216–237, 1994.
117. T. Maekawa, F.-E. Wolter, and N. Patrikalakis. Umbilics and lines of curvature for shape interrogation. *CAGD*, 13:133–161, 1996.
118. T. Maekawa, F.-E. Wolter, and N. M. Patrikalakis. Umbilics and lines of curvature for shape interrogation. *Computer Aided Geometric Design*, 13(2):133–161, 1996.
119. J.-L. Maltret and M. Daniel. Discrete curvatures and applications: a survey. Rapport de recherche 004.2002, Laboratoire des Sciences de l'Information et des Systèmes, 2002.
120. N. Max. Weights for computing vertex normals from facet normals. *Journal of Graphics Tools*, 4(2):1–6, 1999.

121. D. S. Meek and D. J. Walton. On surface normal and gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design*, 17:521–543, 2000.
122. M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *International Workshop on Visualization and Mathematics*, Berlin-Dahlem, Germany, May 2002.
123. J. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, volume 334, pages 633–702. Elsevier Science, 2000.
124. J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitrou. The discrete geodesic problem. *SIAM J. of Computing*, 16(4):647–668, 1987.
125. N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Symposium on Computational geometry*, pages 322–328. ACM Press, 2003.
126. N. J. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry and Applications*, 2004.
127. O. Monga, N. Armande, and P. Montesinos. Thin nets and crest lines: Application to satellite data and medical images. *Computer Vision and Image Understanding: CVIU*, 67(3):285–295, 1997.
128. O. Monga and S. Benayoun. Using partial derivatives of 3D images to extract typical surface features. *Computer Vision and Image Understanding: CVIU*, 61:171–195, 1995.
129. O. Monga, S. Benayoun, and O.D. Faugeras. From partial derivatives of 3-D density images to ridge lines. In *Proc. CVPR'92*, pages 354–359. IEEE, 1992.
130. H. P. Moreton and C. H. Sequin. Functional optimization for fair surface design. In *SIGGRAPH'92 Proceedings*, pages 167–176, August 1992.
131. R. Morris. The sub-parabolic lines of a surface. In G. Mullineux, editor, *Mathematics of Surfaces VI*, IMA new series 58, pages 253–262. Clarendon Press, 1996.
132. J. M. Morvan and B. Thibert. On the approximation of a smooth surface with a triangulated mesh. *Computational Geometry: Theory and Applications*, 33(3):337–352, 2002.
133. B. Mourrain. Computing isolated polynomial roots by matrix methods. *J. of Symbolic Computation, Special Issue on Symbolic-Numeric Algebra for Polynomials*, 26(6):715–738, Dec. 1998.
134. B. Mourrain. A new criterion for normal form algorithms. In M. Fossorier, H. Imai, Shu Lin, and A. Poli, editors, *Proc. AAEECC*, volume 1719 of *LNCS*, pages 430–443. Springer, Berlin, 1999.
135. B. Mourrain and V. Y. Pan. Asymptotic acceleration of solving multivariate polynomial systems of equations. In *STOC '98 (Dallas, TX)*, pages 488–496. ACM, New York, 1999.
136. B. Mourrain and J.-P. Pavone. Subdivision methods for solving polynomial equations. Technical Report 5658, INRIA Sophia-Antipolis, 2005.
137. B. Mourrain, F. Rouillier, and M.-F. Roy. *Bernstein's basis and real root isolation*, pages 459–478. Mathematical Sciences Research Institute Publications. Cambridge University Press, 2005.
138. B. Mourrain and P. Trébuchet. Solving projective complete intersection faster. In C. Traverso, editor, *Proc. Intern. Symp. on Symbolic and Algebraic Computation*, pages 231–238. New-York, ACM Press., 2000.

139. B. Mourrain and Ph. Trébuchet. Generalised normal forms and polynomial system solving. In M. Kauers, editor, *Proc. Intern. Symp. on Symbolic and Algebraic Computation*, pages 253–260. New-York, ACM Press., 2005.
140. B. Mourrain, M. Vrahatis, and J. C. Yakoubsohn. On the complexity of isolating real roots and computing with certainty the topological degree. *J. of Complexity*, 18(2):612–640, 2002.
141. L. R. Nackman. Two-dimensional critical point configuration graphs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(4):442–450, 1984.
142. M. Nielsen, O. F. Olsen, M. Sig, and M. Sigurd. Koenderink corner points. In *Proceedings of the 4th International Workshop on Visual Form*, pages 420–430. Springer-Verlag, 2001.
143. G. M. Nielson and B. Hamann. The asymptotic decider: resolving the ambiguity in marching cubes. In *VIS '91: Proceedings of the 2nd conference on Visualization '91*, pages 83–91, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.
144. T. Nishita, T. W. Sederberg, and M. Kakimoto. Ray tracing trimmed rational surface patches. *Computer Graphics*, 24(4 (Proc. ACM Siggraph 90)):337–345, August 1990.
145. Y. Ohtake, A. Belyaev, and H.-P. Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics*, 23(3):609–612, August 2004. Proc. ACM SIGGRAPH 2004.
146. D. L. Page, A. Koschan, and M. Abidi. Perception-based 3D triangle mesh segmentation using fast marching watersheds. In *Proc. Intl. Conf. on Computer Vision and Pattern Recognition, Vol. II*, pages 27–32, 2003.
147. D. L. Page, A. Koschan, Y. Sun, J. Paik, and A. Abidi. Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting. In *Proceedings on Computer Vision and Pattern Recognition*, 2001.
148. D. L. Page, Y. Sun, A. Koschan, J. Paik, and M. Abidi. Normal vector voting: Crease detection and curvature estimation on large, noisy meshes. *Journal of Graphical Models*, 64:1–31, 2002.
149. N. M. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, Berlin and Heidelberg, 2002.
150. M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled models. *Computer Graphics Forum*, 22(3):281–289, 2003. Eurographics 2003 issue.
151. X. Pennec, N. Ayache, and J. P. Thirion. Landmark-based registration using features identified through differential geometry. In I. N. Bankman, editor, *Handbook of Medical Imaging*, pages 499–513. Academic Press, 2000.
152. C. Petersen. Adaptive contouring of three-dimensional surfaces. *CAGD*, 1:61–74, 1984.
153. S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys*, 34(2), 2001.
154. G. Peyré and L. Cohen. Heuristically driven front propagation for geodesic paths extraction. In *Proceedings of VLSM'05*, pages 173–184. Springer LNCS, 2005.
155. V. Pham-Tron, N. Szafran, and L. Biard. Pseudo-geodesics on three-dimensional surfaces and pseudo-geodesic meshes. *Numerical Algorithms*, 26:305–315, 2001.

156. B. T. Phong. Illumination for computer generated pictures. *Communications of ACM*, 18(6):311–317, 1975.
157. T. Poeschl. Detecting surface irregularities using isophotes. *CAGD*, 1:163–168, 1984.
158. I. R. Porteous. Ridges and umbilics of surfaces. In R. R. Martin, editor, *The Mathematics of Surfaces II*, pages 447–458, Oxford, 1987. Clarendon Press.
159. I. R. Porteous. *Geometric Differentiation for the Intelligence of Curves and Surfaces*. Cambridge University Press, Cambridge, 1994.
160. I. R. Porteous and M. J. Puddephat. Landmarks of a surface. In R. Cipolla and R. R. Martin, editors, *Mathematics of Surfaces IX*, IMA new series, pages 114–125. Clarendon Press, 2000.
161. H. Pottmann. Visualizing curvature discontinuities of free-form surfaces. In *Proc. Eurographics'89*, pages 529–536, 1989.
162. E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. In *SIGGRAPH 99 Conference Proceedings*, pages 49–56, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
163. J. G. Ramsay. *Folding and Fracturing of Rocks*. McGraw Hill, 1967.
164. J. J. Risler. *Méthodes mathématiques pour la CAO*. Masson, 1991.
165. A. Rockwood. Accurate display of tensor product isosurfaces. In *IEEE Visualization '90 Conf.*, 1990.
166. C. Rössl, L. Kobbelt, and H.-P. Seidel. Extraction of feature lines on triangulated surfaces using morphological operators. In *Proceedings of the 2000 AAAI Symposium*, pages 71–75. AAAI Press, 2000.
167. S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Proc. of Second International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, Thessaloniki, Greece, 2004.
168. M. A. Sabin. Contouring - the state of the art. In Earnshaw R.A., editor, *Fundamental Algorithms for Computer Graphics*, pages 411–482. Springer Verlag, 1985.
169. S. G. Scatterfield and D. F. Rogers. Contour lines from a b-spline surface. *IEEE CG & Appl.*, 5(4), 1985.
170. D. Schweitzer. Artificial texturing: an aid to surface visualization. *Computer Graphics*, 17(3):23–29, 1983.
171. T. W. Sederberg and R. J. Meyers. Loop detection in surface patch intersections. *Computer Aided Geometric Design*, 5(2):161–171, 1988.
172. T. W. Sederberg and J. Zheng. Algebraic methods for computer aided geometric design. In *Handbook of computer aided geometric design*, pages 363–387. North-Holland, Amsterdam, 2002.
173. E. C. Sherbrooke and N. M. Patrikalakis. Computation of the solutions of nonlinear polynomial systems. *Comput. Aided Geom. Design*, 10(5):379–405, 1993.
174. K. Sloan. Surface normal (summary). In *Usenet comp.graphics article*, September 1991.
175. P.-P. J. Sloan, C. F. Rose, and M. F. Cohen. Shape by example. In *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 135–143, New York, NY, USA, 2001. ACM Press.
176. D. J. Struik. *Lectures on Classical Differential Geometry*. Dover Science, 1986.
177. G. Stylianou and G. Farin. Crest lines extraction from 3D triangulated meshes. In G. Farin, B. Hamann, and H. Hagen, editors, *Hierarchical and Geometrical Methods in Scientific Visualization*, pages 269–281. Springer, 2003.

178. Y. Sun, D. L. Page, J. K. Paik, A. Koschan, and M. A. Abidi. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. In *Proc. Int. Conf. Image Processing, Vol. 3*, pages 825–828, 2002.
179. V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe. Fast exact and approximate geodesics on meshes. In *Proceedings of SIGGRAPH'05*, New York, NY, USA, 2005. ACM Press/Addison-Wesley Publishing Co. to appear.
180. G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. ICCV'95*, pages 902–907, 1995.
181. H. Theisel, C. Rössl, R. Zayer, and H.-P. Seidel. Normal based estimation of the curvature tensor for triangular meshes. In *Proc. Pacific Graphics*, pages 288–297, Seoul, South Korea, 2004.
182. J.-P. Thirion. The extremal mesh and the understanding of 3D surfaces. *International Journal of Computer Vision*, 19(2):115–128, 1996.
183. J.-P. Thirion. New feature points based on geometric invariants for 3D image registration. *International Journal of Computer Vision*, 18(2):121–137, May 1996.
184. J.-P. Thirion and A. Gourdon. The 3D marching lines algorithm and its application to crest lines extraction. *Graphical Models and Image Processing*, 58(6):503–509, 1996.
185. G. Thürmer and C. A. Wüthrich. Computing vertex normals from polygonal facets. *Journal of Graphics Tools*, 3(1):42–46, 1998.
186. P. Trébuchet. *Vers une résolution stable et rapide des équations algébriques*. PhD thesis, Université Pierre et Marie Curie, 2002.
187. B. L. van der Waerden. *Modern Algebra*. F. Ungar Publishing Co., New York, 3rd edition, 1950.
188. K. Watanabe and A. G. Belyaev. Detection of salient curvature features on polygonal surfaces. *Computer Graphics Forum*, 20(3):385–392, 2001. Eurographics 2001.
189. W. Welch and A. Witkin. Free-Form shape design using triangulated surfaces. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94*, pages 247–256, 1994.
190. F.-E. Wolter. Cut locus and medial axis in global shape interrogation and representation. Technical Report memorandum 92-2, MIT, Department of Ocean Engineering, January 1992.
191. A. L. Yuille. Zero crossings on lines of curvature. *Graphical Models and Image Processing*, 45(1):68–87, 1989.
192. A. L. Yuille and M. Leyton. 3D symmetry-curvature duality theorems. *Graphical Models and Image Processing*, 52(1):124–140, 1990.