Curvature Plots of Powell–Sabin Spline Surfaces

Jan Maes, Evelyne Vanraes
Paul Dierckx, Adhemar Bultheel

Report TW 358, May 2003



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Curvature Plots of Powell–Sabin Spline Surfaces

Jan Maes, Evelyne Vanraes
Paul Dierckx, Adhemar Bultheel

Report TW 358, May 2003

Department of Computer Science, K.U.Leuven

Abstract

In this paper we develop an algorithm for the construction of a curvature plot of a (rational) Powell–Sabin (PS) spline surface, where the PS–spline surface is given by its normalized B–spline representation. Second order surface properties are needed to visualize curvature, which are obtained by looking at the Bézier representation of the surface. As an example application, we compare the curvature plots of an initial smooth Uniform PS–spline surface with the curvature plots of the same spline surface after noise has been added and removed with a noise removal algorithm.

Keywords : Powell–Sabin splines, surface curvature, Bézier patch, CAGD
AMS(MOS) Classification : 65D07, 65D17, 65D18

Curvature Plots of Powell–Sabin Spline Surfaces

Jan Maes, Evelyne Vanraes
Paul Dierckx, Adhemar Bultheel

May 2003

Abstract

In this paper we develop an algorithm for the construction of a curvature plot of a (rational) Powell–Sabin (PS) spline surface, where the PS–spline surface is given by its normalized B–spline representation. Second order surface properties are needed to visualize curvature, which are obtained by looking at the Bézier representation of the surface. As an example application, we compare the curvature plots of an initial smooth Uniform PS–spline surface with the curvature plots of the same spline surface after noise has been added and removed with a noise removal algorithm.

Keywords: Powell–Sabin splines, surface curvature, Bézier patch, CAGD

AMS(MOS) classification: 65D07, 65D17, 65D18

1 Introduction

A spline surface can be obtained in many ways: for instance, as a surface that interpolates to given data points, or as the result of a surface fitting algorithm [2]. In both cases, the surfaces may look perfect on a computer screen, yet reveal significant imperfections on a full scale plot. Because the use of a large plotter is time–consuming and expensive, it is necessary to use other more convenient techniques which help us to decide if a surface is fair or not. Local flatness can be inspected by curvature. One can use the curvature of some surface curves defined on a grid in the domain plane, but this method neglects the curvature of the other surface curves. Therefore we use Gaussian curvature, mean curvature, the principal curvatures, or the absolute curvature.

In this paper we give an algorithm for the construction of a curvature plot of a Powell–Sabin (PS) spline [6] surface. PS–splines appear to be very valuable for CAGD applications. These piecewise quadratics can be defined on any triangulation and, moreover, Dierckx [3] proposed a stable algorithm to construct a normalized B–spline representation for PS–splines. Although PS–splines have only global C^1 continuity, this is sufficient for most geometric modeling problems. The interested reader is referred to [8] for a full framework on Powell–Sabin splines in CAGD applications.

Section 2 is devoted to some preliminaries. We recall some general concepts of quadratic polynomials on triangulations. Formulas for the derivatives of quadratic Bézier surfaces are given and Powell–Sabin splines are introduced. We also mention the relevant properties of a normalized B–spline representation for PS–splines. Section 3 deals with the basic concepts of surface curvature and applies this theory to the quadratic polynomials of section 2 and to the Powell–Sabin splines. As an example application, section 4 shows that curvature plots can be a useful tool when dealing with noisy surfaces.

2 Preliminary concepts

2.1 Quadratic polynomials on triangles

Consider a non–degenerated triangle $\mathcal{T}(V_1, V_2, V_3)$ in a plane, having vertices with coordinates $V_i(u_i, v_i),\ i = 1, 2, 3$. This triangle will be denoted as the domain triangle. Let Π_2 denote the linear space of bivariate polynomials of degree ≤ 2. Each polynomial $P_2(u, v) \in \Pi_2$ on $\mathcal{T}$ has a unique representation

$$P_2(u, v) := b(\tau) = \sum_{|\lambda|=2} b_\lambda B^2_\lambda(\tau), \tag{1}$$

with $\lambda = (\lambda_1, \lambda_2, \lambda_3),\ \lambda_i \geq 0$ a multi–index of length $|\lambda| = \lambda_1 + \lambda_2 + \lambda_3 = 2$, with $\tau = (\tau_1, \tau_2, \tau_3)$ the barycentric coordinates of a point (u, v) with respect to $\mathcal{T}$, and with

$$B^n_\lambda(\tau) = \frac{n!}{\lambda_1!\lambda_2!\lambda_3!}\tau_1^{\lambda_1}\tau_2^{\lambda_2}\tau_3^{\lambda_3} \tag{2}$$

the Bernstein–Bézier polynomials on the triangle [4].

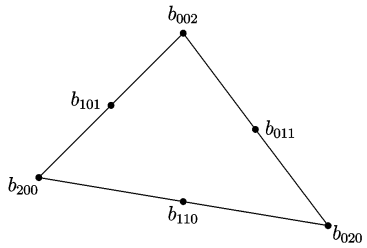


Figure 1: Positions of the Bézier ordinates.

The coefficients b_λ are called the Bézier ordinates of the polynomial $P_2(u, v)$ with respect to the triangle $\mathcal{T}$. By associating each ordinate b_λ with the Bézier domain point $(\frac{\lambda_1}{2}, \frac{\lambda_2}{2}, \frac{\lambda_3}{2})$ in the triangle $\mathcal{T}$ we can display this Bernstein–Bézier representation schematically, as in figure 1. The points $(\frac{\lambda}{2}, b_\lambda)$ are the control points for the surface $z = b(\tau)$ and the piecewise linear interpolant to these points is the Bézier net or control net. This Bézier net is tangent to the polynomial surface at the three vertices.

As a generalization a quadratic polynomial surface $\mathbf{P}_2(u, v)$ on $\mathcal{T}$ has a unique Bézier representation

$$\mathbf{P}_2(u, v) := \mathbf{b}(\tau) = \sum_{|\lambda|=2} \mathbf{b}_\lambda B^2_\lambda(\tau), \tag{3}$$

where $\mathbf{b}_\lambda = (b^x_\lambda, b^y_\lambda, b^z_\lambda)$ are the Bézier control points for the surface $\mathbf{b}(\tau)$.

2.2 Derivatives of a quadratic Bézier surface

The partial derivatives for tensor product patches are easily computed, but the situation is different for triangular patches. The appropriate derivative for a triangular Bézier patch is the directional derivative. Let $s = (s_1, s_2, s_3)$ and $t = (t_1, t_2, t_3)$ be the barycentric coordinates of two points in the domain with respect to a triangle $\mathcal{T}(V_1, V_2, V_3)$, having vertices with coordinates $V_i(u_i, v_i)$, $i = 1, 2, 3$. Their difference $d = s - t = (d_1, d_2, d_3)$ defines a vector with $|d| = 0$.

Definition 2.1 *The directional derivative of a surface (3) at* $\mathbf{b}(\tau)$ *with respect to* d *is given by*

$$D_d\mathbf{b}(\tau) = d_1\frac{\partial \mathbf{b}}{\partial \tau_1}(\tau) + d_2\frac{\partial \mathbf{b}}{\partial \tau_2}(\tau) + d_3\frac{\partial \mathbf{b}}{\partial \tau_3}(\tau). \tag{4}$$

The partial derivatives $\frac{\partial \mathbf{b}}{\partial \tau_i}(\tau)$ of a Bézier patch are calculated straight forward. We refer to [5] for the details. We find that the directional derivative of $\mathbf{b}(\tau)$ with respect to d is equal to

$$D_d\mathbf{b}(\tau) = 2\sum_{|\lambda|=1}(d_1\mathbf{b}_{\lambda+\vec{e}_1} + d_2\mathbf{b}_{\lambda+\vec{e}_2} + d_3\mathbf{b}_{\lambda+\vec{e}_3})B^1_\lambda(\tau), \tag{5}$$

with $\vec{e}_i = (\delta_{i1}, \delta_{i2}, \delta_{i3})$, $i = 1, 2, 3$, and δ_{ij} the Kronecker delta.

Analogously the second order directional derivative of a surface (3) at $\mathbf{b}(\tau)$ with respect to d and $\tilde{d}$ is given by

$$D_{\tilde{d}}D_d\mathbf{b}(\tau) = \tilde{d}_1\frac{\partial D_d\mathbf{b}}{\partial \tau_1}(\tau) + \tilde{d}_2\frac{\partial D_d\mathbf{b}}{\partial \tau_2}(\tau) + \tilde{d}_3\frac{\partial D_d\mathbf{b}}{\partial \tau_3}(\tau).$$

The computation of the second order partial derivatives $\frac{\partial^2 \mathbf{b}}{\partial \tau_i \partial \tau_j}(\tau)$ is again straight forward and we find that

$$D_{\tilde{d}}D_d\mathbf{b}(\tau) = 2\sum_{i=1}^{3}\sum_{j=1}^{3} d_i\tilde{d}_j\mathbf{b}_{\vec{e}_i+\vec{e}_j}. \tag{6}$$

The directional derivative of a surface with respect to d depends on the length of d. For our purposes we want d to be a unit vector. It is easily verified that $\|d\|^2 = (d_1u_1 + d_2u_2 + d_3u_3)^2 + (d_1v_1 + d_2v_2 + d_3v_3)^2$. The vector $\frac{d}{\|d\|} = (\frac{d_1}{\|d\|}, \frac{d_2}{\|d\|}, \frac{d_3}{\|d\|})$ defines a unit vector in the domain.

2.3 Derivatives of a rational quadratic Bézier surface

Consider a rational Bézier surface of degree 2 on a triangle $\mathcal{T}$, given by

$$\mathbf{b}(\tau) = \frac{\sum_{|\lambda|=2} w_\lambda \mathbf{b}_\lambda B^2_\lambda(\tau)}{\sum_{|\lambda|=2} w_\lambda B^2_\lambda(\tau)}. \tag{7}$$

Define a Bézier polynomial $w(\tau)$ and a Bézier surface $\mathbf{p}(\tau)$ as

$$w(\tau) = \sum_{|\lambda|=2} w_\lambda B^2_\lambda(\tau) \qquad , \qquad \mathbf{p}(\tau) = \sum_{|\lambda|=2} w_\lambda \mathbf{b}_\lambda B^2_\lambda(\tau). \tag{8}$$

By applying Leibniz' rule we find that

$$D^r_d\mathbf{p}(\tau) = D^r_d(w(\tau)\mathbf{b}(\tau)) = \sum_{j=0}^{r}\binom{r}{j} D^j_d w(\tau) D^{r-j}_d \mathbf{b}(\tau),$$

or

$$D^r_d\mathbf{b}(\tau) = \frac{1}{w(\tau)}\left(D^r_d\mathbf{p}(\tau) - \sum_{j=1}^{r}\binom{r}{j} D^j_d w(\tau) D^{r-j}_d \mathbf{b}(\tau)\right). \tag{9}$$

The second order directional derivative of a rational Bézier surface $\mathbf{b}(\tau)$ with respect to d and $\tilde{d}$ can also be computed. After some straight forward calculations we can write this as

$$D_{d\tilde{d}}\mathbf{b}(\tau) = \frac{1}{w(\tau)}\left[D_{d\tilde{d}}\mathbf{p}(\tau) - D_d w(\tau)\cdot D_{\tilde{d}}\mathbf{b}(\tau) - D_{\tilde{d}} w(\tau)\cdot D_d\mathbf{b}(\tau) - D_{d\tilde{d}} w(\tau)\cdot \mathbf{b}(\tau)\right]. \tag{10}$$

2.4 Powell–Sabin splines

Consider a simply connected subset $\Omega \subset \mathbb{R}^2$ with polygonal boundary $\delta\Omega$. Suppose we have a conforming triangulation Δ of Ω, being constituted of triangles ρ_j, $j = 1, \ldots, t$, and having vertices V_i with coordinates (u_i, v_i), $i = 1, \ldots, n$. The Powell–Sabin refinement Δ^* of Δ divides each triangle ρ_j into six smaller triangles with a common vertex. It can be constructed easily (see e.g. [6]). Powell–Sabin (PS) splines are piecewise quadratic polynomials with C^1 continuity on Ω, and they have a quadratic Bézier representation (1) on each PS–subtriangle $\mathcal{T}_{PS} \in \Delta^*$.

Definition 2.2 *A PS–spline surface has a normalized B–spline representation*

$$\mathbf{s}(u,v) = \sum_{i=1}^{n}\sum_{j=1}^{3} \mathbf{c_{ij}} B_i^j(u,v) \quad , \quad (u,v) \in \Omega, \tag{11}$$

where $\mathbf{c_{ij}} = (c_{ij}^x, c_{ij}^y, c_{ij}^z)$ *are the B–spline control points and* $B_i^j(u,v)$ *are the normalized B–splines.*

This representation has a lot of valuable properties which makes it a powerful tool for representing surfaces in CAGD. The remainder of this section summarizes the most important properties. For details we refer to the original paper [3] or to [8].

Property 2.1 *The B–spline basis functions* $\{B_i^j(u,v)\}_{i=1,\ldots,n}^{j=1,2,3}$ *form a convex partition of unity on* Ω, *i.e.*

$$B_i^j(u,v) \geq 0 \text{ for all } u,v \in \Omega, \tag{12}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{3} B_i^j(u,v) = 1 \text{ for all } u,v \in \Omega. \tag{13}$$

Furthermore these basis functions have local support: $B_i^j(x,y)$ vanishes outside the so–called molecule M_i of vertex V_i, which is the union of all triangles ρ_k containing V_i.

Property 2.2

$$B_i^j(V_l) = \frac{\partial B_i^j(V_l)}{\partial u} = \frac{\partial B_i^j(V_l)}{\partial v} = 0, \quad l \neq i. \tag{14}$$

Hence this representation is affine invariant and has the local control and convex hull properties. Linear functions can be represented exactly. More particularly let

$$u = \sum_{i=1}^{n}\sum_{j=1}^{3} U_{ij} B_i^j(u,v), \qquad v = \sum_{i=1}^{n}\sum_{j=1}^{3} V_{ij} B_i^j(u,v). \tag{15}$$

Definition 2.3 *The vertices* $Q_{ij}(U_{ij}, V_{ij})$, $i = 1,\ldots,n$, $j = 1,2,3$ *in the domain plane are the B–spline ordinates.*

Definition 2.4 *The triangles* $t_i(Q_{i1}, Q_{i2}, Q_{i3})$, $i = 1,\ldots,n$ *are called the PS–triangles .*

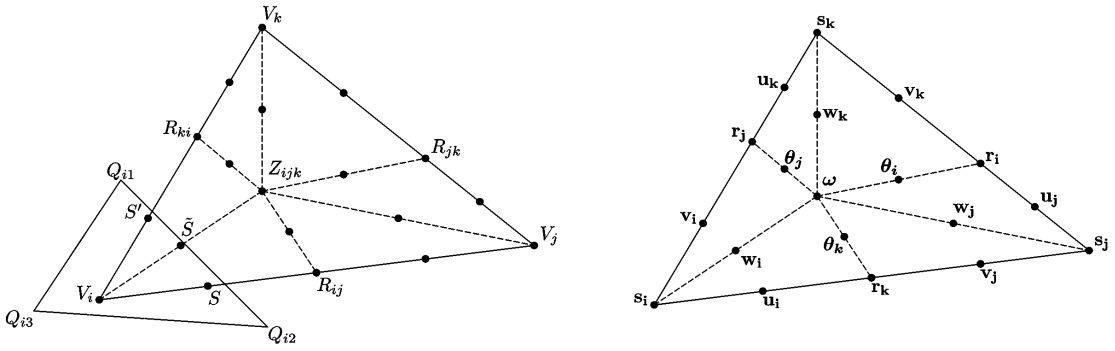


Figure 2: *Left:* PS–refinement of a triangle $\rho(V_i, V_j, V_k)$ with PS–points $S, \tilde{S}, S'$ for the vertex V_i and PS–triangle $t_i(Q_{i1}, Q_{i2}, Q_{i3})$. *Right:* Schematic representation of the Bézier control points.

Figure 2 shows a domain triangle with its PS–subdivision and a PS–triangle $t_i(Q_{i1}, Q_{i2}, Q_{i3})$.

Definition 2.5 *The PS–points of a vertex $V_i \in \Delta$ are a number of particular surrounding Bézier domain points and the vertex V_i itself. Figure 2 shows the PS–points $S, \tilde{S}, S'$ and V_i for the vertex V_i in the triangle $\rho(V_i, V_j, V_k)$.*

For a given PS–refinement Δ^*, the position of the Bézier ordinates is fixed. This is not the case for the B–spline ordinates. The following lemma however states that there is a restriction on the B–spline ordinates.

Lemma 2.1 *In order for the basis functions $\left\{B_i^j(u,v)\right\}_{i=1,\ldots,n}^{j=1,2,3}$ to constitute a convex partition of unity on Ω, it is required that for each vertex V_i, $i = 1, \ldots, n$ the PS–triangle $t_i(Q_{i1}, Q_{i2}, Q_{i3})$ contains the PS–points of V_i from all the triangles ρ_k in the molecule M_i.*

The Bézier representation of a PS–spline surface can be calculated from the B–spline representation. Consider a domain triangle $\rho(V_i, V_j, V_k) \in \Delta$ with its PS–refinement as on figure 2. Let

$$\begin{aligned} R_{ij} &= \lambda_{ij} V_i + (1 - \lambda_{ij}) V_j \\ Z_{ijk} &= a_{ijk} V_i + b_{ijk} V_j + c_{ijk} V_k. \end{aligned}$$

Denote the barycentric coordinates of the PS–points V_i, S, S' and $\tilde{S}$ with respect to the PS–triangle $t_i(Q_{i1}, Q_{i2}, Q_{i3})$ with $(\alpha_{i1}, \alpha_{i2}, \alpha_{i3})$, (L_{i1}, L_{i2}, L_{i3}), $(L'_{i1}, L'_{i2}, L'_{i3})$ and $(\tilde{L}_{i1}, \tilde{L}_{i2}, \tilde{L}_{i3})$. Then the Bézier control points can be written as the following unique convex barycentric combinations of the B–spline control points:

$$\begin{aligned} \mathbf{s_i} &= \alpha_{i1}\mathbf{c_{i1}} + \alpha_{i2}\mathbf{c_{i2}} + \alpha_{i3}\mathbf{c_{i3}} \\ \mathbf{u_i} &= L_{i1}\mathbf{c_{i1}} + L_{i2}\mathbf{c_{i2}} + L_{i3}\mathbf{c_{i3}} \\ \mathbf{v_i} &= L'_{i1}\mathbf{c_{i1}} + L'_{i2}\mathbf{c_{i2}} + L'_{i3}\mathbf{c_{i3}} \\ \mathbf{w_i} &= \tilde{L}_{i1}\mathbf{c_{i1}} + \tilde{L}_{i2}\mathbf{c_{i2}} + \tilde{L}_{i3}\mathbf{c_{i3}}. \end{aligned} \tag{16}$$

Similar expressions hold for $(\mathbf{s_j}, \mathbf{u_j}, \mathbf{v_j}, \mathbf{w_j})$ and $(\mathbf{s_k}, \mathbf{u_k}, \mathbf{v_k}, \mathbf{w_k})$. The other Bézier ordinates are obtained from the C^1–continuity conditions of the PS–spline:

$$\begin{aligned} \mathbf{r_k} &= \lambda_{ij}\mathbf{u_i} + (1 - \lambda_{ij})\mathbf{v_j} \\ \boldsymbol{\theta_k} &= \lambda_{ij}\mathbf{w_i} + (1 - \lambda_{ij})\mathbf{w_j} \\ \boldsymbol{\omega} &= a_{ijk}\mathbf{w_i} + b_{ijk}\mathbf{w_j} + c_{ijk}\mathbf{w_k}. \end{aligned} \tag{17}$$

Definition 2.6 *The control triangles are defined as $T_l(\mathbf{c_{l1}}, \mathbf{c_{l2}}, \mathbf{c_{l3}})$. The projection of each T_l onto the domain plane yields the PS–triangles t_l.*

Property 2.3 *Each control triangle $T_l(\mathbf{c_{l1}}, \mathbf{c_{l2}}, \mathbf{c_{l3}})$ is tangent to the PS–surface at $\mathbf{s}(V_l)$.*

Definition 2.7 *A Non Uniform Rational Powell–Sabin (NURPS) spline surface has the form*

$$\mathbf{s}(u,v) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{3} \mathbf{c_{ij}} w_{ij} B_i^j(u,v)}{\sum_{i=1}^{n}\sum_{j=1}^{3} w_{ij} B_i^j(u,v)}, \quad (u,v) \in \Omega \tag{18}$$

where we impose that $w_{ij} > 0$ in order for $\mathbf{s}(u,v)$ to be defined anywhere on Ω.

For the evaluation of this rational extension, calculating the corresponding rational Bézier representation, the geometric interpretation of the weights and the modeling of planar effects using relatively large weights we refer to [9].

3 Surface curvature

3.1 Introduction

The available methods to obtain derivatives can be used to visualize differential geometry properties of quadratic Bézier surfaces, such as curvature. Surfaces have two major kinds of curvature:

Gaussian curvature K and mean curvature H, which are both scalar–valued. They can be visualized by means of a color–coded map. Other types of curvature are, for instance, the principal curvatures κ_1 and κ_2 or the absolute curvature $\kappa_{\text{abs}} = |\kappa_1| + |\kappa_2|$.

The principal curvatures κ_1 and κ_2 have a nice geometric interpretation. They are scalar values corresponding to the axis directions of a conic section which is known as *Dupin's indicatrix* (see figure 3). When intersecting the surface with a plane parallel to the tangent plane at the considered point, we get a scaled picture of Dupin's indicatrix. If κ_1 and κ_2 are of the same sign,

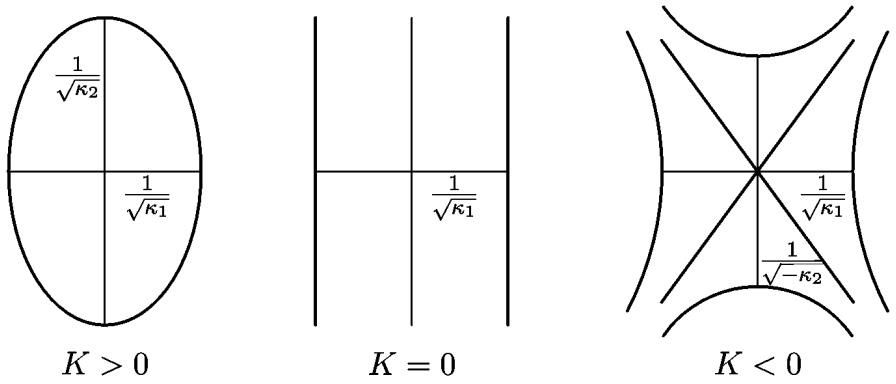


Figure 3: Dupin's indicatrix for an elliptic, a parabolic, and a hyperbolic point.

the considered point is called an elliptic point. If κ_1 and κ_2 have different signs, the considered point is called an hyperbolic point. And if either $\kappa_1 = 0$ or $\kappa_2 = 0$, then the considered point is called parabolic.

Gaussian and mean curvature are related to the principal curvatures κ_1 and κ_2 of the surface at the considered point:

$$K = \kappa_1 \kappa_2 \qquad \text{and} \qquad H = \frac{1}{2}(\kappa_1 + \kappa_2). \tag{19}$$

They both give important information about the smoothness of a surface.

The absolute curvature $\kappa_{\text{abs}} = |\kappa_1| + |\kappa_2|$ may be used in cases where Gaussian and mean curvature do not offer enough information, that is, for cylindrical surfaces, resp. minimal surfaces.

3.2 Basic concepts

Further details of the basic concepts of differential geometry required in this section can be found, for example, in [1].

As said before, the two major kinds of surface curvature are the Gaussian curvature K and the mean curvature H. For a given surface $\mathbf{x}(u, v)$, these can be calculated as

$$K = \frac{LN - M^2}{EG - F^2} \tag{20}$$

and

$$H = \frac{1}{2}\frac{NE - 2MF + LG}{EG - F^2}, \tag{21}$$

with

$$E = \frac{\partial \mathbf{x}(u,v)}{\partial u}\frac{\partial \mathbf{x}(u,v)}{\partial u} \tag{22}$$

$$F = \frac{\partial \mathbf{x}(u,v)}{\partial u}\frac{\partial \mathbf{x}(u,v)}{\partial v} \tag{23}$$

$$G = \frac{\partial \mathbf{x}(u,v)}{\partial v}\frac{\partial \mathbf{x}(u,v)}{\partial v} \tag{24}$$

$$L = \mathbf{n}(u,v)\frac{\partial^2 \mathbf{x}(u,v)}{\partial u^2} \tag{25}$$

$$M = \mathbf{n}(u,v)\frac{\partial^2 \mathbf{x}(u,v)}{\partial u \partial v} \tag{26}$$

$$N = \mathbf{n}(u,v)\frac{\partial^2 \mathbf{x}(u,v)}{\partial v^2}. \tag{27}$$

The normal vector at the surface point under consideration $\mathbf{n}(u,v)$ is computed as

$$\mathbf{n}(u,v) = \frac{\mathbf{x}_u(u,v) \wedge \mathbf{x}_v(u,v)}{\|\mathbf{x}_u(u,v) \wedge \mathbf{x}_v(u,v)\|} = \frac{\mathbf{x}_u(u,v) \wedge \mathbf{x}_v(u,v)}{\sqrt{EG - F^2}}. \tag{28}$$

Recall from the previous section that $K = \kappa_1\kappa_2$ and $H = \frac{1}{2}(\kappa_1 + \kappa_2)$. Hence κ_1 and κ_2 are the roots of

$$\kappa^2 - 2H\kappa + K = 0.$$

The variation of any of the quantities K, H, κ_1, κ_2 or κ_{abs} can be displayed using a color–coded curvature map. First, the curvature κ (which can lie anywhere between $-\infty$ and $+\infty$) has to be normalized to κ_{norm} in the interval $[-1,1]$ using

$$\kappa_{\text{norm}} = \text{sign}(\kappa)(1 - e^{-C\|\kappa\|}),$$

where C is a factor for controlling the contrast in the visualization. This simplifies mapping the curvature to a color code. For example, if we let the color vary linearly from green for $\kappa_{\text{norm}} = -1$ over black for $\kappa_{\text{norm}} = 0$ to red for $\kappa_{\text{norm}} = 1$, then red regions will indicate elliptic points, and green regions will indicate hyperbolic points for the Gaussian curvature plot. Figure 4 shows the Gaussian curvature plot of a Powell–Sabin B–spline basis function.

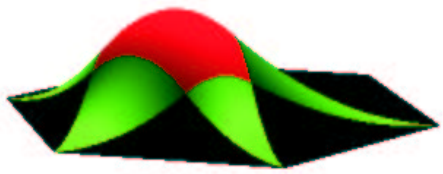


Figure 4: Gaussian curvature plot of a Powell–Sabin B–spline basis function.

3.3 Curvature of triangular quadratic Bézier patches

Be given a quadratic polynomial surface on a triangle $\mathcal{T}$

$$\mathbf{P}_2(u,v) = \begin{bmatrix} P_2^x(u,v) \\ P_2^y(u,v) \\ P_2^z(u,v) \end{bmatrix} = \begin{bmatrix} \sum_{|\lambda|=2} b_\lambda^x B_\lambda^2(\tau) \\ \sum_{|\lambda|=2} b_\lambda^y B_\lambda^2(\tau) \\ \sum_{|\lambda|=2} b_\lambda^z B_\lambda^2(\tau) \end{bmatrix}.$$

Its surface curvature depends on the quantities E, F, G, L, M and N, defined by the equations (22) – (27). Thus, we need the first and second order partial derivatives of $\mathbf{P}_2(u,v)$. These can be found by determining the directions d_u and d_v of the U and V axis with respect to the triangle $\mathcal{T}$, and taking the directional derivatives with respect to d_u and d_v as explained in section 2.2.

There is also an easier way. Consider the affine transformation (see figure 5)

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\theta_s & \cos\theta_t \\ \sin\theta_s & \sin\theta_t \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix},$$

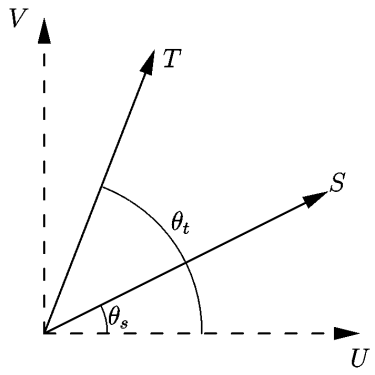


Figure 5: U–V axes versus S–T axes.

then the following relation is satisfied:

$$\mathbf{P}_2(u,v) = \begin{bmatrix} P_2^x(u,v) \\ P_2^y(u,v) \\ P_2^z(u,v) \end{bmatrix} = \begin{bmatrix} P_2^x(\phi(s,t)) \\ P_2^y(\phi(s,t)) \\ P_2^z(\phi(s,t)) \end{bmatrix} = \mathbf{Q}_2(s,t).$$

Instead of calculating E, F, G, L, M and N for the surface $\mathbf{P}_2(u,v)$ we may calculate E, F, G, L, M and N for the surface $\mathbf{Q}_2(s,t)$. The resulting curvature will not be affected.

Because Bézier triangles are affine invariant, the Bézier representation of a surface in the S–T space is exactly the same as the corresponding Bézier representation in the U–V space. Thus, we can choose two arbitrary directions s and t. Redefine

$$\begin{aligned} E &= D_s\mathbf{P}_2(u,v) \cdot D_s\mathbf{P}_2(u,v) && (29)\\ F &= D_s\mathbf{P}_2(u,v) \cdot D_t\mathbf{P}_2(u,v) && (30)\\ G &= D_t\mathbf{P}_2(u,v) \cdot D_t\mathbf{P}_2(u,v) && (31)\\ L &= \mathbf{n}(u,v) \cdot D_s^2\mathbf{P}_2(u,v) && (32)\\ M &= \mathbf{n}(u,v) \cdot D_sD_t\mathbf{P}_2(u,v) && (33)\\ N &= \mathbf{n}(u,v) \cdot D_t^2\mathbf{P}_2(u,v) && (34) \end{aligned}$$

and

$$\mathbf{n}(u,v) = \frac{D_s\mathbf{P}_2(u,v) \wedge D_t\mathbf{P}_2(u,v)}{\|D_s\mathbf{P}_2(u,v) \wedge D_t\mathbf{P}_2(u,v)\|} = \frac{D_s\mathbf{P}_2(u,v) \wedge D_t\mathbf{P}_2(u,v)}{\sqrt{EG-F^2}}, \qquad (35)$$

then equation (20) and (21) are still valid formulas for the Gaussian and mean curvature of the quadratic polynomial surface $\mathbf{P}_2(u,v)$ at (u,v).

3.4 An algorithm

In this section we give the preceding outline in an algorithmic form. The directional derivative of the Bézier patch $\mathbf{b}(\tau)$ with respect to d will be denoted as $\mathbf{b}_d$ and the second order directional derivative with respect to d and $\tilde{d}$ will be denoted as $\mathbf{b}_{d\tilde{d}}$. Consider a quadratic polynomial surface $\mathbf{P}_2(u,v)$ on a triangle $\mathcal{T}$ in its unique Bézier representation

$$\mathbf{b}(\tau) = \begin{bmatrix} \sum_{|\lambda|=2} b_\lambda^x B_\lambda^2(\tau) \\ \sum_{|\lambda|=2} b_\lambda^y B_\lambda^2(\tau) \\ \sum_{|\lambda|=2} b_\lambda^z B_\lambda^2(\tau) \end{bmatrix}.$$

Denote the vertices of $\mathcal{T}$ as V_i with coordinates (u_i, v_i), $i = 1, 2, 3$.

1. Choose

$$\begin{aligned} d &= (1,0,-1) \\ \tilde{d} &= (0,1,-1) \end{aligned}$$

2. Calculate

$$\begin{aligned} \|d\|^2 &= (u_1-u_3)^2+(v_1-v_3)^2 \\ \|\tilde{d}\|^2 &= (u_2-u_3)^2+(v_2-v_3)^2 \end{aligned}$$

3. Compute

$$\begin{aligned} \mathbf{b}_{dd} &= \frac{2}{\|d\|^2}(\mathbf{b}_{200}-2\mathbf{b}_{101}+\mathbf{b}_{002}) \\ \mathbf{b}_{\tilde{d}\tilde{d}} &= \frac{2}{\|\tilde{d}\|^2}(\mathbf{b}_{020}-2\mathbf{b}_{011}+\mathbf{b}_{002}) \\ \mathbf{b}_{d\tilde{d}} &= \frac{2}{\|d\|\|\tilde{d}\|}(\mathbf{b}_{110}-\mathbf{b}_{101}-\mathbf{b}_{011}+\mathbf{b}_{002}) \end{aligned}$$

4. Construct a grid for triangle $\mathcal{T}$ and for each point $\tau = (\tau_1,\tau_2,\tau_3)$ of the grid do

   (a) Compute

$$\begin{aligned} \mathbf{b}_d &= \frac{2}{\|d\|}((\mathbf{b}_{200}-\mathbf{b}_{101})\tau_1+(\mathbf{b}_{110}-\mathbf{b}_{011})\tau_2+(\mathbf{b}_{101}-\mathbf{b}_{002})\tau_3) \\ \mathbf{b}_{\tilde{d}} &= \frac{2}{\|\tilde{d}\|}((\mathbf{b}_{110}-\mathbf{b}_{101})\tau_1+(\mathbf{b}_{020}-\mathbf{b}_{011})\tau_2+(\mathbf{b}_{011}-\mathbf{b}_{002})\tau_3) \end{aligned}$$

   (b) Compute E,F,G:

$$\begin{aligned} E &= \mathbf{b}_d \cdot \mathbf{b}_d \\ F &= \mathbf{b}_d \cdot \mathbf{b}_{\tilde{d}} \\ G &= \mathbf{b}_{\tilde{d}} \cdot \mathbf{b}_{\tilde{d}} \end{aligned}$$

   (c) The normal $\mathbf{n}$ is equal to

$$\mathbf{n} = \frac{\mathbf{b}_d \wedge \mathbf{b}_{\tilde{d}}}{\sqrt{EF-G^2}}$$

   (d) Compute L,M,N:

$$\begin{aligned} L &= \mathbf{n} \cdot \mathbf{b}_{dd} \\ M &= \mathbf{n} \cdot \mathbf{b}_{d\tilde{d}} \\ N &= \mathbf{n} \cdot \mathbf{b}_{\tilde{d}\tilde{d}} \end{aligned}$$

   (e) Calculate the curvature in τ:

$$\begin{aligned} K_\tau &= \frac{LN-M^2}{EG-F^2} \\ H_\tau &= \frac{1}{2}\frac{NE-2MF+LG}{EG-F^2} \\ \kappa_{1,2_\tau} &= H_\tau \pm \sqrt{H_\tau^2-K_\tau} \end{aligned}$$

   end do-loop

5. Map the computed curvatures to the appropriate color and plot

This algorithm can be extended to rational Bézier surfaces very easily. We will sum up the changes briefly. In step (3) $\mathbf{p}_{dd}$, $\mathbf{p}_{d\tilde{d}}$, $\mathbf{p}_{\tilde{d}\tilde{d}}$, w_{dd}, $w_{d\tilde{d}}$ and $w_{\tilde{d}\tilde{d}}$ are computed, with $\mathbf{p}$ and w defined as in equation (8). For instance, $\mathbf{p}_{dd}$ is equal to

$$2(w_{200}\mathbf{b}_{200} - 2w_{101}\mathbf{b}_{101} + w_{002}\mathbf{b}_{002}).$$

Next, step (a) is best divided in three parts. The first part calculates $w = \sum_{|\lambda|=2} w_\lambda B^2_\lambda(\tau)$, $\mathbf{p} = \sum_{|\lambda|=2} w_\lambda \mathbf{b}_\lambda B^2_\lambda(\tau)$ and $\mathbf{b} = \frac{\mathbf{p}}{w}$. The second part obtains $\mathbf{p}_d$, $\mathbf{p}_{\tilde{d}}$, w_d and $w_{\tilde{d}}$, e.g.

$$\mathbf{p}_d = 2((w_{200}\mathbf{b}_{200} - w_{101}\mathbf{b}_{101})\tau_1 + (w_{110}\mathbf{b}_{110} - w_{011}\mathbf{b}_{011})\tau_2 + (w_{101}\mathbf{b}_{101} - w_{002}\mathbf{b}_{002})\tau_3).$$

And the third part computes $\mathbf{b}_d$, $\mathbf{b}_{\tilde{d}}$, $\mathbf{b}_{dd}$, $\mathbf{b}_{d\tilde{d}}$ and $\mathbf{b}_{\tilde{d}\tilde{d}}$,

$$\mathbf{b}_d = \frac{1}{\|d\|}\frac{1}{w}(\mathbf{p}_d - w_d\mathbf{b}), \quad \ldots, \quad \mathbf{b}_{\tilde{d}\tilde{d}} = \frac{1}{\|\tilde{d}\|^2}\frac{1}{w}(\mathbf{p}_{\tilde{d}\tilde{d}} - 2w_{\tilde{d}}\mathbf{b}_{\tilde{d}} - w_{\tilde{d}\tilde{d}}\mathbf{b}).$$

The other steps remain unchanged.

3.5 Powell-Sabin surface curvature

The algorithm from section 3.4 can be extended to an algorithm for generating a curvature plot of a Powell–Sabin spline surface. Consider a normalized B–spline representation for Powell–Sabin splines (11)

$$\mathbf{s}(u,v) = \sum_{i=1}^{n}\sum_{j=1}^{3} \mathbf{c_{ij}} B_i^j(u,v)$$

with $\mathbf{c_{ij}} = (c^x_{ij}, c^y_{ij}, c^z_{ij})$ the PS–control points.

For all triangles $\rho(V_i, V_j, V_k) \in \Delta$ do

1. Compute the Bézier control points as explained in section 2.4
2. For every triangle $\mathcal{T}$ in the PS-refinement of $\rho(V_i, V_j, V_k)$ with Bézier control points $(\mathbf{b}_{200}, \mathbf{b}_{110}, \mathbf{b}_{020}, \mathbf{b}_{011}, \mathbf{b}_{002}, \mathbf{b}_{101})$ do
   Run the algorithm from section 3.4
   end do-loop

end do-loop

In the inner most loop (see section 3.4) we must construct a grid for triangle $\mathcal{T}$. The curvature is calculated at each point τ of the grid. An alternative is to calculate curvature only at the points $\{(1,0,0),(0,1,0),(0,0,1)\}$. This yields the curvature at the vertices of the 6 triangles $\mathcal{T}$ resulting from the PS–refinement. If the resulting curvature plot is not satisfactory, more detail can be obtained by applying a subdivision algorithm as explained in [7].

Figures 6 and 7 demonstrate the use of curvature plots. At first sight the two spline surfaces look the same, but the curvature plots show that the surface in figure 7 is not smooth at all.

4 Application: noise removal

We now use the curvature plot algorithm to compare an initial smooth Uniform Powell-Sabin spline surface with the same spline after noise has been added and removed with a noise removal algorithm.

For the noise removal part we use the Uniform Powell–Sabin Wavelet transform which has been developed in [10]. Starting from a smooth Uniform PS–surface, noise is added. Next the



Figure 6: *Left:* A smooth PS–surface. *Middle:* Gaussian curvature plot of the PS–surface. *Right:* Mean curvature plot of the PS–surface.

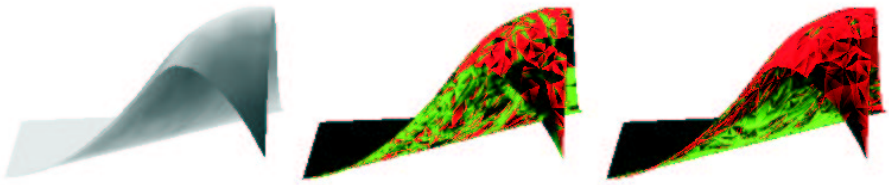


Figure 7: *Left:* A PS–surface with noise. *Middle:* Gaussian curvature plot of the PS–surface. *Right:* Mean curvature plot of the PS–surface.

noisy surface is smoothened again by using the Uniform Powell–Sabin Wavelet transform in a thresholding algorithm. A forward wavelet transform splits the PS–spline surface in a coarser part and a detail part. Next the same wavelet transform is applied to the coarser part, etc. The result is a PS–surface on a coarse level, together with f detail parts. These detail parts are nothing else but a series of wavelet coefficients.

A thresholding algorithm can be used to change the wavelet coefficients: if a wavelet coefficient has an absolute value smaller than a threshold value ϵ, then this wavelet coefficient is replaced by a zero. For the remaining wavelet coefficients we will consider two possibilities:

- **Hard thresholding** leaves the wavelet coefficients with an absolute value higher than the threshold ϵ unchanged,
- **Soft thresholding** decreases the absolute value of the remaining wavelet coefficients with ϵ, leaving their sign unchanged.

Applying the inverse wavelet transform f times results in the denoised surface.

We applied this algorithm to an initially smooth Uniform PS–spline surface (figure 8). The given smooth surface has been subdivided three times, after which the B–spline coefficients c_{ij} were replaced by $c_{ij} + \delta c_{ij}$ with δc_{ij} uniformly distributed on a small interval (figure 9). Figures 10 and 11 show the results of the thresholding algorithm with ϵ equal to 5% of the maximal wavelet coefficient in absolute value. Figure 12 shows the results of the thresholding algorithm with ϵ equal to 10% of the maximal wavelet coefficient in absolute value. It is difficult to see on sight which method gives the closest resemblance with the original surface. It is here that the curvature plots come in.

Comparing the curvature plots of figure 11 with those of figure 10 can give us a lot of information about which threshold method we should use. First, figure 11 is smoother than figure 10 because its curvature plots have less color variations. Second, the curvature plots of the denoised surfaces should resemble the curvature plots of the original surface as close as possible. Figure 11 has a closer resemblance with the original surface than figure 10. In this case one can say that soft thresholding yields the best results.

The same reasoning can be applied to determine a good threshold value ϵ. Figures 11 and 12 show the result of the same soft thresholding algorithm but with a different threshold ϵ. As expected, figure 12 is smoother than figure 11. More important here is the resemblance between

the curvature plots of the denoised surfaces and the curvature plots of the original surface. If the threshold ϵ is too high, the denoised surface will be very smooth, but the resemblance with the original surface disappears.

Of course, in a practical situation we only have the noisy surface. In such a case curvature plots can be used for displaying smoothness and local information.

Acknowledgements

This work is partially supported by the Flemish Fund for Scientific Research (FWO Vlaanderen) project MISS (G.0211.02), and by the Belgian Program on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with the authors.

References

[1] W. Boehm. Differential Geometry I & II. In G. Farin, editor, *Curves And Surfaces For Computer Aided Geometric Design: A Practical Guide*, pages 171–180, 348–362. Academic Press, Boston, 4th edition, 1997.

[2] P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford University Press, Oxford, 1993.

[3] P. Dierckx. On calculating normalized Powell–Sabin B–splines. *CAGD*, 15(3):61–78, 1997.

[4] G. Farin. Triangular Bernstein–Bézier patches. *CAGD*, 3(2):83–128, 1986.

[5] G. Farin. *Curves And Surfaces For Computer Aided Geometric Design: A Practical Guide*. Academic Press, Boston, 4th edition, 1997.

[6] M. J. D. Powell and M. A. Sabin. Piecewise quadratic approximations on triangles. *ACM Transactions on Mathematical Software*, 3:316–325, 1977.

[7] E. Vanraes, J. Windmolders, A. Bultheel, and P. Dierckx. Subdivision for Powell–Sabin spline surfaces. TW Report 345, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, September 2002.

[8] J. Windmolders. *Powell–Sabin splines for Computer Aided Geometric Design*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 2003.

[9] J. Windmolders and P. Dierckx. From PS–splines to NURPS. In Albert Cohen, Chrisophe Rabut, and Larry L. Schumaker, editors, *Curve and Surface Fitting: Saint–Malo 1999*. Vanderbilt University Press, Nashville, 2000.

[10] J. Windmolders, E. Vanraes, P. Dierckx, and A. Bultheel. Uniform Powell–Sabin wavelets. *Journal of Computational and Applied Mathematics*, 154(1):125–142, 2003.

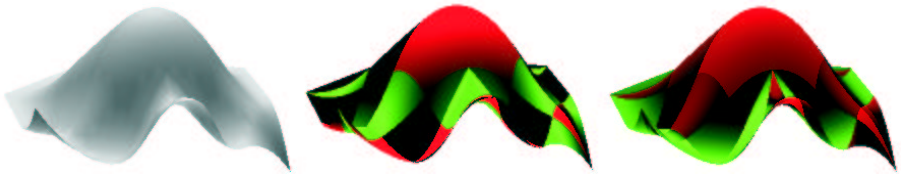


Figure 8: *Left:* The original surface. *Middle:* Gaussian curvature plot. *Right:* Mean curvature plot.

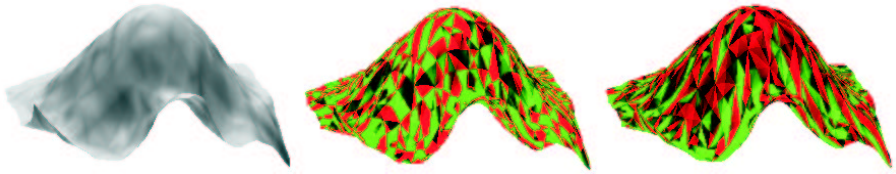


Figure 9: *Left:* Noise added to the smooth surface. *Middle:* Gaussian curvature plot. *Right:* Mean curvature plot.

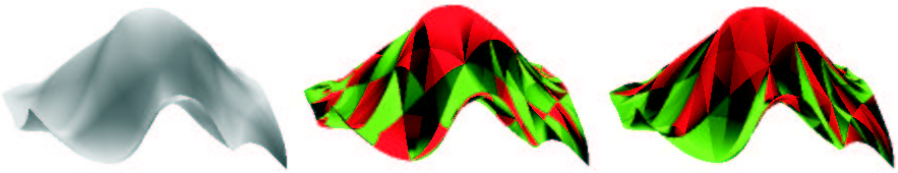


Figure 10: *Left:* Hard thresholding (5%). *Middle:* Gaussian curvature plot. *Right:* Mean curvature plot.

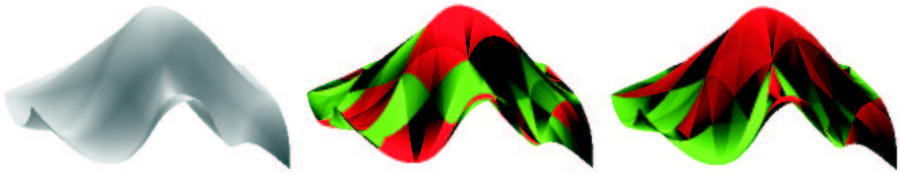


Figure 11: *Left:* Soft thresholding (5%). *Middle:* Gaussian curvature plot. *Right:* Mean curvature plot.

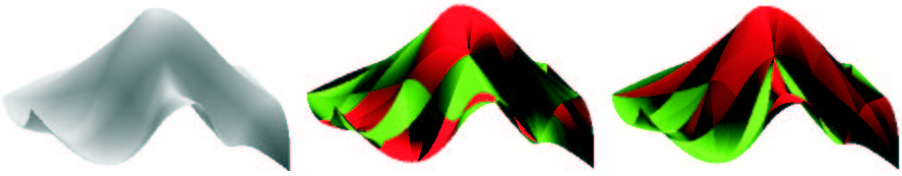


Figure 12: *Left:* Soft thresholding (10%). *Middle:* Gaussian curvature plot. *Right:* Mean curvature plot.