Linear perturbation methods for topologically consistent representations of free–form surface intersections

Xiaowen Song, Thomas W. Sederberg, and Jianmin Zheng
Department of Computer Science,
Brigham Young University, Provo, UT 84602

Rida T. Farouki[1] and Joel Hass[2]
[1]Department of Mechanical and Aeronautical Engineering,
[2]Department of Mathematics,
University of California, Davis, CA 95616

Abstract

By applying displacement maps to slightly perturb two free–form surfaces, one can ensure *exact* agreement between the images in $\mathbb{R}^3$ of parameter–domain approximations to their curve of intersection. Thus, at the expense of slightly altering the surfaces in the vicinity of their intersection, a perfect matching of the surface trimming curves is guaranteed. This exact agreement of contiguous trimmed surfaces is essential to achieving topologically consistent solid model constructions through Boolean operations, and has a profound impact on the efficiency and reliability of applications such as meshing, rendering, and computing volumetric properties. Moreover, the control point perturbations require only the solution of a linear system for their determination. The basic principles of this approach to topologically consistent surface trimming curves are described, and example results from the implementation of a simple instance of the method are presented.

Keywords: surface intersections, trimmed surfaces, control point perturbations, topological consistency, geometric continuity.

e–mail: songxw@cs.byu.edu, zheng@cs.byu.edu, tom@cs.byu.edu,
farouki@ucdavis.edu, hass@math.ucdavis.edu

1 Introduction and motivation

At the workshop on *Mathematical Foundations of Computer Aided Design*, held at the Mathematical Sciences Research Institute, Berkeley in June 1999, the problem of formulating algorithms for topologically consistent representations of trimmed surfaces emerged as a critical challenge confronting the computer aided geometric design research community. Although this problem is not new, enhancements in the efficiency of both hardware and downstream applications software have caused the lack of robustness and stagnation of trimmed surface algorithms to become a serious concern. Motivated by such considerations, this paper presents methods to achieve *exact* topological consistency of trimmed Bézier/B–spline surfaces.

Current computer–aided geometric design systems rely primarily on the use of (piecewise) polynomial and rational parametric surfaces to describe "free–form" shapes, such as automobile bodies, turbine or propellor blades, aircraft fuselages and wings, and ship hulls. Each "patch" of such a surface is the image of a simple (i.e., triangular or rectangular) parameter domain under a vector map to $\mathbb{R}^3$.

In certain applications, however, it may not be possible to precisely specify the desired portion of a surface as the image of a triangular or rectangular parameter domain under a polynomial or rational map. When solid models are constructed by Boolean operations, for example, it is necessary to *trim* surface patches along their curves of intersection. Such "trimmed surfaces" also arise in constructing offset surfaces, blend surfaces, swept volumes of moving objects, and many other contexts. It is, in general, impossible to describe the exact intersections of free–form surfaces by rational parameterizations, and algorithms for approximating trimmed surfaces [19] often incur "gaps" or "overlaps" between trimmed patches (see Figure 1). A common approach is to simply replace the given smooth analytic surfaces by polyhedral approximations: surface trimming can then be performed without the need for fundamentally new algorithms or representations. It is widely recognized, however, that such polyhedral models are unsatisfactory for precision engineering applications involving aerodynamic performance, contact kinematics or dynamics, detailed stress analyses, etc.

The absence of truly "robust" trimmed surface algorithms and representations poses a critical impasse for the efficient and systematic usage of CAD models in downstream applications. Computational fluid dynamics is a typical example [10] — while surface and volume meshing algorithms typically run in just a few hours, given topologically consistent input geometry, and the flow computations are also very fast, they must be preceded by a laborious "geometry preparation and repair" phase. This involves manual intervention to identify and remedy the topological

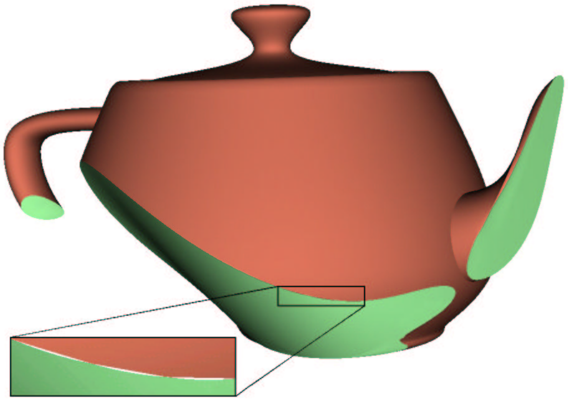

Figure 1: The "Utah teapot" cut with a bicubic patch. The intersection is approximated by surface trimming curves, creating a gap between the trimmed surfaces.

defects (i.e., unintended gaps or overlaps between adjacent trimmed surfaces) of the model, and may consume weeks or even months to render the CAD model in a sufficiently "water–tight" state for the meshing algorithm.

This imbalance of effort is especially intolerable in design optimization, where frequent alterations to the model geometry are invoked. Advances in computer hardware, meshing algorithms, and finite–element codes, in the face of a stagnant state–of–the–art for robust trimmed surfaces, have exacerbated this problem. The economic impact of a solution to this impasse, in terms of improved efficiency and accelerated product development cycles, has been estimated to run to hundreds of millions or even billions of dollars annually.

Tensor–product bicubic patches are perhaps the most commonly used surfaces in CAD systems. Unfortunately, the possibility of an *exact* trimming for such surfaces is precluded [21] by the remarkably high degree of their intersections — namely, 324 when regarded as algebraic space curves, and 54 in each variable when expressed in the surface parameter domains. Whereas an algebraic space curve must be of genus 0 to admit rational parameterization, the intersection curve of two bicubic surface patches is generically of genus 433 — see [15].

For many applications, exactitude of the surface trimming curve is not critical provided we can guarantee that an approximation is (i) "sufficiently close" to the true intersection, and (ii) *topologically consistent*. Property (ii) refers to the fact that, for surface trimming, we require *three* representations of the intersection — one in each of the two surface parameter domains, and one in $\mathbb{R}^3$. To avoid gaps and overlaps between the trimmed surfaces, these distinct representations of the intersection must define *exactly the same curve*.

We propose a surface trimming procedure that incorporates both the features (i) and (ii) above. Whereas the accuracy of intersection curve approximations has been addressed by previous authors (see [18] for a review) we believe that the scheme proposed herein offers the first explicit embodiment of the "consistency property" (ii) for the Bézier/B–spline surfaces[1] commonly used in computer aided design and geometric modeling. The key idea is that, given sufficiently accurate approximations of the intersection curve in the surface parameter domains, we take the liberty of *perturbing the surfaces* in the vicinity of their intersection, so that the mappings of the curves to $\mathbb{R}^3$ agree *exactly* — in terms of both geometry *and* parameterization. Prior work [2, 9, 13, 14] has failed to achieve this elusive "holy grail" of topologically consistent trimming for Bézier/B–spline surfaces.

This idea was first suggested at the Workshop on *Mathematical Foundations of Computer–Aided Design* — a follow–up to the SIAM Workshop on *Integration of CAD and CFD*, held at UC Davis [10] in April 1999, at which the impact of inconsistent trimmed surfaces on meshing and computational fluid dynamics was recognized. We show here that the perturbations required to achieve consistency can be obtained by solving *a system of linear constraints* on the surface control points in the vicinity of any smooth, monotone intersection segment. Moreover, we can use subdivision or degree elevation to ensure the availability of sufficient degrees of freedom to simultaneously satisfy the constraint equations and achieve an adequately accurate approximation of the true intersection.

Our plan for this paper is as follows. To apply the perturbation scheme, it is necessary to first perform a topological analysis of the intersection curve in the surface parameter domains, and to dissect it into "simple" smooth segments: this process is briefly summarized in §2. The basic principles governing the selection of control point perturbations, in order to achieve consistency of the trimming curves, are described in §3, and the approximation of intersection curve segments is discussed in §4. For brevity, we present results from only a simple illustrative implementation of the method in §5, in the context of linear intersection approximations in the parameter domains of bicubic surface patches. In this case, only five control points need to be perturbed, and the perturbations can be found in an uncoupled manner. Finally, §6 concludes with an assessment of the potential for practical use of this perturbation scheme and a brief discussion of topics for further study. The appendix summarizes some key algorithms for Bernstein–form polynomials, that are extensively used in the perturbation procedure.

[1] See [1, 17] for related methods on the trimming of free–form subdivision surfaces.

2 Pre–preprocessing the intersection

Before invoking the surface perturbation scheme, we subject the true intersection curve to certain pre–processing steps in order to resolve its topology and to split it into "simple" smooth monotone segments, amenable to approximation.

Suppose the given surfaces are tensor–product Bézier patches defined in terms of the Bernstein basis $b_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i$ in the form

$$\mathbf{p}(s,t) = \sum_{j=0}^{n}\sum_{k=0}^{n} \mathbf{p}_{jk} b_j^n(s) b_k^n(t) \,, \qquad (s,t) \in [0,1] \times [0,1] \,,$$

$$\mathbf{q}(u,v) = \sum_{j=0}^{n}\sum_{k=0}^{n} \mathbf{q}_{jk} b_j^n(u) b_k^n(v) \,, \qquad (u,v) \in [0,1] \times [0,1] \,.$$

For brevity, we focus[2] on tensor–product surface patches, although the method can be readily adapted to B–spline surfaces, triangular patches, etc. To identify segments that are monotone with respect to both the parameters on each surface, we dissect the intersection at all *turning points*, which may be identified as real solutions to the systems of four equations in the four unknowns $(s,t,u,v) \in [0,1]^4$ defined by the vector condition

$$\mathbf{p}(s,t) = \mathbf{q}(u,v)$$

and each in turn of the four scalar conditions

$$[\,(\mathbf{p}_s \times \mathbf{p}_t) \times (\mathbf{q}_u \times \mathbf{q}_v)\,] \cdot \mathbf{p}_s = 0 \,, \quad [\,(\mathbf{p}_s \times \mathbf{p}_t) \times (\mathbf{q}_u \times \mathbf{q}_v)\,] \cdot \mathbf{p}_t = 0 \,,$$

$$[\,(\mathbf{p}_s \times \mathbf{p}_t) \times (\mathbf{q}_u \times \mathbf{q}_v)\,] \cdot \mathbf{q}_u = 0 \,, \quad [\,(\mathbf{p}_s \times \mathbf{p}_t) \times (\mathbf{q}_u \times \mathbf{q}_v)\,] \cdot \mathbf{q}_v = 0 \,.$$

These may be solved by root–finding methods for Bernstein–form polynomials — e.g., [23]. Figure 2 gives a schematic illustration of the pre–processing procedure.

A topological analysis must also be performed, to establish how the monotone segments identified in this manner are connected — see, for example, [8, 11, 12]. It suffices, in the present context, to note that the output of the topology analysis is a set of paired rectangular sub–patches of the original surface patches, each pair being characterized by the property that their common intersection component is

[2] We shall also confine our attention to "generic" intersections: degenerate cases, such as the presence of singular points or coincidence with a surface isoparameter curve, can be handled by suitable special–case procedures.

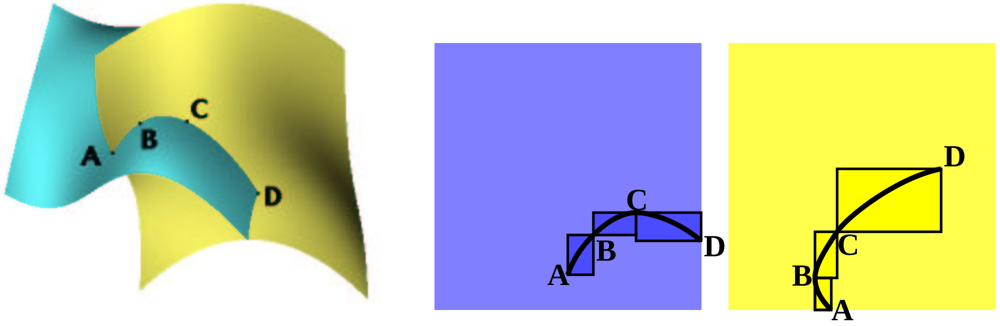

Figure 2: Left: two intersecting bicubic patches. Right: the pre–processing of their intersection in the two surface parameter domains. Point B is a vertical tangency in the parameter domain of the yellow surface, and C is a horizontal tangency in the parameter domain of the blue surface. A and D lie on patch boundaries. The subdivision into sub–patches, within which the intersection curve is monotone with respect to each surface parameter, is illustrated.

a smooth, monotone segment traversing these sub–patches between diametrically opposite corners.

For convenience we also assume that, if $\mathbf{p}(s,t)$ and $\mathbf{q}(u,v)$ are two sub–patches satisfying these criteria, they are parameterized on the domain $[0,1] \times [0,1]$ with the intersection segment connecting the opposite corners $(0,0)$ and $(1,1)$. Each sub–patch is represented by its own Bézier control net. Moreover, we assume that the portion of the sub–patch $\mathbf{p}(s,t)$ that is to be trimmed away is bounded by the intersection curve and the sides $s = 0$ and $t = 1$. Likewise, the portion of $\mathbf{q}(u,v)$ to be trimmed away is bounded by the intersection curve and the sides $u = 0$ and $v = 1$. These stipulations are easily satisfied by suitable re–parameterizations of the sub–patches.

3 Outline of perturbation scheme

Perturbation schemes have previously been invoked [22] to enforce topological consistency of geometric models, though primarily in the piecewise–linear realm. Recently, these methods have also been extended to subdivision surfaces. Our present intent is to describe their application to the ubiquitous Bézier/B–spline representations of computer aided geometric design.

Given two tensor–product Bézier patches, suppose we postulate polynomial parameterizations (with common parameter ξ) of their intersection segment in the surface parameter domains, of the form

$$s(\xi) = \sum_{i=0}^{m} s_i b_i^m(\xi)\,, \quad t(\xi) = \sum_{i=0}^{m} t_i b_i^m(\xi)\,, \tag{1}$$

and

$$u(\xi) = \sum_{i=0}^{m} u_i b_i^m(\xi)\,, \quad v(\xi) = \sum_{i=0}^{m} v_i b_i^m(\xi)\,, \tag{2}$$

for $\xi \in [0,1]$, with $s(0)=t(0)=u(0)=v(0)=0$ and $s(1)=t(1)=u(1)=v(1)=1$. The exact intersection does not ordinarily admit such a parameterization (since it is not, in general, a rational curve). However, by taking the liberty of perturbing the surfaces slightly, we can *force* such a parameterization to be exact for the intersection of the perturbed surfaces. To determine the required perturbations, we define an "error curve" by taking the vector difference of the images of the curves (1) and (2), defined in the surface parameter domains, under the mappings $\mathbf{p}(s,t)$ and $\mathbf{q}(u,v)$ — namely,

$$\mathbf{e}(\xi) = \sum_{j=0}^{n}\sum_{k=0}^{n} \mathbf{p}_{jk} b_j^n(s(\xi)) b_k^n(t(\xi)) - \mathbf{q}_{jk} b_j^n(u(\xi)) b_k^n(v(\xi))\,.$$

Using the arithmetic and composition algorithms for Bernstein–form polynomials (see the appendix), this can be expressed as a univariate vector polynomial

$$\mathbf{e}(\xi) = \sum_{i=0}^{2mn} \mathbf{e}_i b_i^{2mn}(\xi)$$

of degree $2mn$ in ξ. Our goal is to make $\mathbf{e}(\xi) \equiv \mathbf{0}$, for which we must have

$$\mathbf{e}_i = \mathbf{0}\,, \quad i = 0,\ldots,2mn\,. \tag{3}$$

Now the coefficients $\mathbf{e}_0,\ldots,\mathbf{e}_{2mn}$ depend linearly upon the $2(n+1)^2$ vector control points $\mathbf{p}_{jk}$ and $\mathbf{q}_{jk}$ of the two surfaces, and non–linearly on the $4(m+1)$ scalar coefficients of $s(\xi)$, $t(\xi)$, $u(\xi)$, $v(\xi)$. Our strategy is to compute the latter *a priori* to satisfy geometrical interpolation or smoothness criteria, and perturb only (a subset of) the surface control points to achieve the compatibility conditions (3) for the intersection curve in the spaces (s,t), (u,v), and $\mathbb{R}^3$.

The required perturbations are found by solving a linear system of constraint equations. Furthermore, we can maintain a prescribed order of continuity (C^1 or nearly–C^1) of the perturbed (trimmed) patch with contiguous patches of the parent surface. The feasibility of this scheme is predicated on ensuring a “sufficiently accurate” initial approximation of the intersection segment — as a measure of this, we may use the quantity

$$\Delta = \max_{\xi\in[0,1]} |\,\mathbf{p}(s(\xi),t(\xi)) - \mathbf{q}(u(\xi),v(\xi))\,|\,.$$

This can be computed with standard methods [16] by expressing the quantity on the right in Bernstein–Bézier form. Although formal solutions exist even for large Δ values, the surface perturbations they incur may be unreasonable for practical purposes. The need to subdue the error Δ below some prescribed tolerance is not an unduly onerous requirement — it can generally be achieved by subdividing the patch into smaller elements, over which the intersection is amenable to more accurate approximation.

To obtain the linear dependence of the compatibility conditions (3) on the surface control points explicitly, the composition of the polynomials $s(\xi)$, $t(\xi)$ and $u(\xi)$, $v(\xi)$ with each of the Bernstein basis functions is written as follows

$$b_j^n(s(\xi)) = \sum_{\alpha=0}^{mn} s_{j\alpha} b_\alpha^{mn}(\xi)\,, \qquad b_k^n(t(\xi)) = \sum_{\beta=0}^{mn} t_{k\beta} b_\beta^{mn}(\xi)\,,$$

and

$$b_j^n(u(\xi)) = \sum_{\alpha=0}^{mn} u_{j\alpha} b_\alpha^{mn}(\xi)\,, \qquad b_k^n(v(\xi)) = \sum_{\beta=0}^{mn} v_{k\beta} b_\beta^{mn}(\xi)\,.$$

Then, by the product rule [7] for polynomials in Bernstein form, we have

$$\mathbf{e}_i = \sum_{j=0}^{n}\sum_{k=0}^{n} \lambda_{ijk}\mathbf{p}_{jk} - \mu_{ijk}\mathbf{q}_{jk}\,, \quad i = 0,\ldots,2mn\,, \tag{4}$$

where

$$\lambda_{ijk} = \sum_{\gamma=\max(0,i-mn)}^{\min(i,mn)} \frac{\binom{mn}{\gamma}\binom{mn}{i-\gamma}}{\binom{2mn}{i}} s_{j\gamma} t_{k,i-\gamma}\,,$$

and

$$\mu_{ijk} = \sum_{\gamma=\max(0,i-mn)}^{\min(i,mn)} \frac{\binom{mn}{\gamma}\binom{mn}{i-\gamma}}{\binom{2mn}{i}} u_{j\gamma} v_{k,i-\gamma}\,.$$

After trimming, $\mathbf{p}(s,t)$ is bounded by the trimming curve and the original patch boundaries $s=1$, $t=0$. Since, in general, we wish to retain at least a C^0 or C^1

connection of the trimmed patch with the parent surface along these boundaries, we cannot perturb *all* the control points $\mathbf{p}_{jk}$ of $\mathbf{p}(s,t)$. For example, C^0 continuity will require that the $2n+1$ distinct control points $\mathbf{p}_{nk}$ for $k=0,\ldots,n$ and $\mathbf{p}_{j0}$ for $j=0,\ldots,n$ are fixed by the neighboring patch, and if we desire C^1 continuity, the control points $\mathbf{p}_{n-1,k}$ for $k=0,\ldots,n$ and $\mathbf{p}_{j1}$ for $j=0,\ldots,n$ are also fixed.

Thus, for C^0 continuity, only n^2 of the control points of $\mathbf{p}(s,t)$ are available for perturbation, while for C^1 continuity only $(n-1)^2$ of the control points can be perturbed. Since the same is true for $\mathbf{q}(u,v)$, we have $2n^2$ or $2(n-1)^2$ perturbable control points altogether, consistent with C^0 or C^1 continuity, respectively. To satisfy the compatibility conditions (3), the number of free control points must equal or exceed the number of these conditions, namely

$$m \le \frac{2n^2-1}{2n} \qquad \text{and} \qquad m \le \frac{2n^2-4n+1}{2n}$$

for C^0 and C^1 continuity, respectively. Table 1 indicates the largest feasible degree $m_{\max}$ for various values of n, and the number of left–over (vector) degrees of freedom ρ for this largest m.

n	$m_{\max}$	ρ
1	$*$	$*$
2	1	3
3	2	5
4	3	7
5	4	9

n	$m_{\max}$	ρ
1	$*$	$*$
2	$*$	$*$
3	1	1
4	2	1
5	3	1

Table 1: Maximum approximant degree and the number of residual freedoms, $\rho = 2(n-1)^2-(2m_{\max}n+1)$, for C^0 (left) and C^1 (right) continuity.

Thus, for example, bicubic patches ($n=3$) allow quadratic approximations to the intersection curve in the parameter domains with 5 residual vector freedoms if we want C^0 continuity with adjacent patches, and linear approximations with 1 residual vector freedom if we want C^1 continuity. If we want intersection approximations of higher order, it is not necessary that the original patches be greater than bicubic in degree, since we can use degree elevation to increase n (and thus the number of control points), and perturb the degree–elevated control points.

When $\mathbf{p}(s,t)$ is represented as a degree–$(n+1,n+1)$ rather than a degree–

(n,n) patch, the control points for this degree–elevated representation are

$$\begin{aligned}\mathbf{p}^{(1)}_{jk} &= \frac{j}{n+1}\frac{k}{n+1}\mathbf{p}_{j-1,k-1} + \frac{j}{n+1}\left[1-\frac{k}{n+1}\right]\mathbf{p}_{j-1,k} \\ &+ \left[1-\frac{j}{n+1}\right]\frac{k}{n+1}\mathbf{p}_{j,k-1} + \left[1-\frac{j}{n+1}\right]\left[1-\frac{k}{n+1}\right]\mathbf{p}_{j,k}\end{aligned}$$

for $0 \le j,k \le n+1$, where $\mathbf{p}_{jk} = \mathbf{0}$ if j or k is less than 0 or greater than n [4]. This process can be iterated, to elevate the degree of $\mathbf{p}(s,t)$ to $(n+r,n+r)$ for any $r \ge 1$ — the resulting control points may be expressed [7] as

$$\mathbf{p}^{(r)}_{\alpha\beta} = \sum_{j=\max(0,\alpha-r)}^{\min(\alpha,n)} \sum_{k=\max(0,\beta-r)}^{\min(\beta,n)} \frac{\binom{n}{j}\binom{r}{\alpha-j}}{\binom{n+r}{\alpha}} \frac{\binom{n}{k}\binom{r}{\beta-k}}{\binom{n+r}{\beta}} \mathbf{p}_{jk} \qquad (5)$$

for $0 \le \alpha,\beta \le n+r$. The control points

$$\mathbf{p}^{(r)}_{n+r,k}\,, \quad \mathbf{p}^{(r)}_{n+r-1,k} \qquad \text{for } k = 0,\ldots,n+r \qquad (6)$$

and

$$\mathbf{p}^{(r)}_{j0}\,, \quad \mathbf{p}^{(r)}_{j1} \qquad \text{for } j = 0,\ldots,n+r \qquad (7)$$

are then fixed if we require C^1 continuity with adjacent patches along the boundaries $s=1$ and $t=0$ — the remaining $(n+r-1)^2$ control points may be perturbed.

It is necessary, however, to first express the coefficients (4) in terms of the degree–elevated control points, in order to enforce the constraints (3). For this purpose, we need to revert the relation (5). Since the control points of a degree–elevated representation are not linearly independent, this reversion is not unique — the original control points can be recovered from just $(n+1)^2$ of the $(n+r+1)^2$ degree–elevated control points. The form appropriate to the present context is defined[3] for $0 \le j,k \le n$ by

$$\mathbf{p}_{jk} = \sum_{\alpha=j+r}^{n+r} \sum_{\beta=0}^{k} (-1)^{j+k+\alpha+\beta} \frac{\binom{\alpha-j-1}{r-1}\binom{n+r}{\alpha}}{\binom{n}{j}} \frac{\binom{k-\beta+r-1}{r-1}\binom{n+r}{\beta}}{\binom{n}{k}} \mathbf{p}^{(r)}_{\alpha\beta}\,.$$

Note that this makes no reference to the control points (6) and (7), which are fixed by the C^1 continuity requirment, and are not subject to perturbation. Thus, if $\Gamma^{(r)}_{jk\alpha\beta}$

[3] This holds *only* when $\mathbf{p}^{(r)}_{\alpha\beta}$ for $0 \le \alpha,\beta \le n+r$ are in fact the degree–elevated control points for a tensor–product surface of true degree (n,n).

denotes the coefficient[4] of $\mathbf{p}_{\alpha\beta}^{(r)}$ above, the constraints (3) can be written as

$$\mathbf{e}_i = \sum_{\alpha=r}^{n+r}\sum_{\beta=0}^{n} \lambda_{i\alpha\beta}^{(r)}\mathbf{p}_{\alpha\beta}^{(r)} - \mu_{i\alpha\beta}^{(r)}\mathbf{q}_{\alpha\beta}^{(r)} = \mathbf{0}\,, \quad i = 0,\ldots,2mn \tag{8}$$

where

$$\lambda_{i\alpha\beta}^{(r)} = \sum_{j=0}^{\alpha-r}\sum_{k=\beta}^{n} \lambda_{ijk}\Gamma_{jk\alpha\beta}^{(r)}\,, \qquad \mu_{i\alpha\beta}^{(r)} = \sum_{j=0}^{\alpha-r}\sum_{k=\beta}^{n} \mu_{ijk}\Gamma_{jk\alpha\beta}^{(r)}$$

(we invoke the same approach to degree–elevate $\mathbf{q}(u,v)$ and assume C^1 continuity is enforced along the sides $u = 1$ and $v = 0$). Thus, if we write the perturbed control points as

$$\mathbf{p}_{\alpha\beta}^{(r)} + \Delta\mathbf{p}_{\alpha\beta}^{(r)} \qquad \text{and} \qquad \mathbf{q}_{\alpha\beta}^{(r)} + \Delta\mathbf{q}_{\alpha\beta}^{(r)}$$

for $r \le \alpha \le n+r$ and $0 \le \beta \le n$, the perturbations must satisfy the linear system

$$\sum_{\alpha=r}^{n+r}\sum_{\beta=0}^{n} \lambda_{i\alpha\beta}^{(r)}\Delta\mathbf{p}_{\alpha\beta}^{(r)} - \mu_{i\alpha\beta}^{(r)}\Delta\mathbf{q}_{\alpha\beta}^{(r)} = \sum_{\alpha=r}^{n+r}\sum_{\beta=0}^{n} \mu_{i\alpha\beta}^{(r)}\mathbf{q}_{\alpha\beta}^{(r)} - \lambda_{i\alpha\beta}^{(r)}\mathbf{p}_{\alpha\beta}^{(r)}$$

for $i = 0,\ldots,2mn$. In the case $n = 3$, $m = 3$, $r = 2$, for example, the above amounts to 31 linear equations in 32 (vector) unknowns, and one more condition is needed to close the system — we may take

$$\sum_{\alpha=r}^{n+r}\sum_{\beta=0}^{n} \Delta\mathbf{p}_{\alpha\beta}^{(r)} = \sum_{\alpha=r}^{n+r}\sum_{\beta=0}^{n} \Delta\mathbf{q}_{\alpha\beta}^{(r)}\,,$$

i.e., the *mean* perturbation is the same for both surfaces.

The above method first approximates the intersection curve in the parameter domains of the two surfaces, and then seeks surface perturbations that will cause the images of these approximations in $\mathbb{R}^3$ to agree exactly. Another approach is to first approximate the intersection in $\mathbb{R}^3$, and then find surface perturbations that will cause curves of appropriate degree in the surface parameter domains to be mapped to this curve in $\mathbb{R}^3$. The latter method offers fewer degrees of freedom, but is easier to implement in simple cases — an example is described in §5.

[4] Note that, for given n and r, these quantitites are “universal constants” that may be pre–computed in rational arithmetic and stored for future use.

4 Approximating the intersection segment

As noted above, the coefficients that define the parameter–domain approximations (1) and (2) of the intersection segment should be determined prior to applying the surface perturbation scheme. The most basic requirement, which can be satisfied by linear $(m = 1)$ approximations in the surface parameter domains, is that of end–point interpolation:

$$s_0 = t_0 = u_0 = v_0 = 0 \qquad \text{and} \qquad s_m = t_m = u_m = v_m = 1\,.$$

To obtain a G^1 piecewise approximation of the intersection, we may interpolate end tangents. If $s(\xi)$, $t(\xi)$ and $u(\xi)$, $v(\xi)$ define the exact intersection, it has the spatial parameterization

$$\mathbf{r}(\xi) \,=\, \mathbf{p}(s(\xi),t(\xi)) \,=\, \mathbf{q}(u(\xi),v(\xi))\,, \qquad \xi \in [\,0,1\,]\,.$$

Denoting derivatives with respect to ξ by primes, we have

$$\mathbf{r}' \,=\, \mathbf{p}_s s' + \mathbf{p}_t t' \,=\, \mathbf{q}_u u' + \mathbf{q}_v v'\,, \tag{9}$$

and taking the dot products of the above expressions with $\mathbf{p}_s \times \mathbf{p}_t$ and $\mathbf{q}_u \times \mathbf{q}_v$, respectively, we obtain the ratios

$$\begin{aligned} s' : t' \;&=\; -\,(\mathbf{q}_u \times \mathbf{q}_v) \cdot \mathbf{p}_t \,:\, (\mathbf{q}_u \times \mathbf{q}_v) \cdot \mathbf{p}_s\,, \\ u' : v' \;&=\; -\,(\mathbf{p}_s \times \mathbf{p}_t) \cdot \mathbf{q}_v \,:\, (\mathbf{p}_s \times \mathbf{p}_t) \cdot \mathbf{q}_u\,. \end{aligned}$$

Substituting these expressions into (9) and using standard vector identities, we have

$$\mathbf{r}' \,=\, c\,(\mathbf{p}_s \times \mathbf{p}_t) \times (\mathbf{q}_u \times \mathbf{q}_v) \tag{10}$$

for $c \neq 0$, i.e., the direction of the curve tangent is the cross product of the surface normals. The parameter c may be used to fix the magnitudes of (s',t') and (u',v') as follows. Let

$$L \,=\, |\mathbf{r}(1) - \mathbf{r}(0)| \,=\, |\mathbf{p}(1,1) - \mathbf{p}(0,0)| \,=\, |\mathbf{q}(1,1) - \mathbf{q}(0,0)|$$

be the chord length of the intersection curve segment. The actual values of (s',t') and (u',v') at $\xi = 0$ and $\xi = 1$ are determined by requiring that

$$|\mathbf{r}'(0)| \,=\, |\mathbf{r}'(1)| \,=\, kL$$

where k is a factor, somewhat greater than 1, allowing for the greater length of the intersection curve segment than the chord between the end points. We obtain

$$c = \frac{kL}{|(\mathbf{p}_s \times \mathbf{p}_t) \times (\mathbf{q}_u \times \mathbf{q}_v)|}\,, \tag{11}$$

and hence the desired values of the end–derivatives are

$$(s',t') = kL\,\frac{(-(\mathbf{q}_u \times \mathbf{q}_v)\cdot\mathbf{p}_t, (\mathbf{q}_u \times \mathbf{q}_v)\cdot\mathbf{p}_s)}{|(\mathbf{p}_s \times \mathbf{p}_t) \times (\mathbf{q}_u \times \mathbf{q}_v)|}\,,$$

and

$$(u',v') = kL\,\frac{(-(\mathbf{p}_s \times \mathbf{p}_t)\cdot\mathbf{q}_v, (\mathbf{p}_s \times \mathbf{p}_t)\cdot\mathbf{q}_u)}{|(\mathbf{p}_s \times \mathbf{p}_t) \times (\mathbf{q}_u \times \mathbf{q}_v)|}\,,$$

where the surface derivatives are evaluated at $(s,t,u,v) = (0,0,0,0)$ if $\xi = 0$, and at $(s,t,u,v) = (1,1,1,1)$ if $\xi = 1$. When $m = 3$, for example, the coefficients of $s(\xi)$ in (1) are thus given by

$$s_0 = 0\,, \quad s_1 = 0 + \frac{s'(0)}{3}\,, \quad s_2 = 1 - \frac{s'(1)}{3}\,, \quad s_3 = 1\,.$$

Analogous expressions hold for the coefficients of $t(\xi)$, $u(\xi)$, $v(\xi)$.

One may, in principle, use higher–order approximations for (1) and (2), taking advantage of the additional degrees of freedom to interpolate further geometrical properties of the intersection curve. Second–order differential properties may be derived by considering the expressions

$$\begin{aligned} \mathbf{r}'' &= \mathbf{p}_s s'' + \mathbf{p}_t t'' + \mathbf{p}_{ss} s'^2 + 2\,\mathbf{p}_{st} s't' + \mathbf{p}_{tt} t'^2 \\ &= \mathbf{q}_u u'' + \mathbf{q}_v v'' + \mathbf{q}_{uu} u'^2 + 2\,\mathbf{q}_{uv} u'v' + \mathbf{q}_{vv} v'^2\,. \end{aligned}$$

Itermediate point data may also be interpolated or approximated by a least–squares method. See [20] for further details on approximating intersection segments.

5 Illustrative implementation

Implementation of the perturbation scheme, as described above, is a substantial task that should be supported by a careful analysis of the condition of the linear system and optimal strategies to best utilize the residual degrees of freedom. A thorough treatment of these issues is deferred to a future study: we confine our attention here to implementation of a simple instance of the perturbation scheme, using bicubic displacement maps. This example illustrates the basic principles, and admits a simple closed–form solution for the required surface perturbations.

The context for this implementation is the intersection of two bicubic surface patches, defined by

$$\mathbf{p}(s,t) = \sum_{j=0}^{3}\sum_{k=0}^{3}\mathbf{p}_{jk}b_j^3(s)b_k^3(t), \;\; \mathbf{q}(u,v) = \sum_{l=0}^{3}\sum_{m=0}^{3}\mathbf{q}_{lm}b_l^3(u)b_m^3(v)$$

for $(s,t) \in [0,1]\times[0,1]$ and $(u,v) \in [0,1]\times[0,1]$, respectively. The method can also be generalized to rational bicubics by working in projective 4–space.

We assume the pre–processing steps of §2 have been invoked, so that $\mathbf{p}(s,t)$ and $\mathbf{q}(u,v)$ intersect in a single curve segment, extending from $(0,0)$ to $(1,1)$ in the parameter domain of each surface. The portion of $\mathbf{p}(s,t)$ bounded by the intersection curve and the patch sides $s = 0$ and $t = 1$ is to be trimmed away. Similarly, the portion of $\mathbf{q}(u,v)$ bounded by the intersection and $u = 0$ and $v = 1$ is to be trimmed away. In the surface parameter domains, we shall use the simplest possible approximation for the intersection curve — namely, the diagonal line segment defined by

$$s(\xi) = t(\xi) = \xi \qquad \text{and} \qquad u(\xi) = v(\xi) = \xi. \tag{12}$$

Note that the images of these approximations in $\mathbb{R}^3$ under the vector mappings $\mathbf{p}(s,t)$ and $\mathbf{q}(u,v)$ are polynomial curves of degree 6.

Formally, we only require C^0 continuity of the retained portion of $\mathbf{p}(s,t)$ with adjacent patches along the sides $s = 1$ and $t = 0$, and likewise for the $\mathbf{q}(u,v)$ along the sides $u = 1$ and $v = 0$. In practice, however, we usually obtain near–G^1 continuity — discrepancies in the tangent–plane orientation are usually $< 1^\circ$, and this error can be further reduced by increasing the number (and decreasing the size) of the displacement maps.

To obtain a simple closed–form solution for the control–point perturbations, we use an approach slightly different from that of §3. Namely, we first construct an approximation of degree 6 to the true intersection segment in $\mathbb{R}^3$, and we then force the images of the parameter–domain approximations (12) to agree with this curve in $\mathbb{R}^3$ by perturbing the surface control points. This has the advantage of decoupling the perturbations for the two surfaces, and allows us to ensure that the approximation satisfies a prescribed geometrical tolerance. Suppose the approximate intersection is given by

$$\mathbf{r}(\xi) \,=\, \sum_{i=0}^{6} \mathbf{r}_i b_i^6(\xi)\,. \tag{13}$$

By construction, we have $\mathbf{r}(0) = \mathbf{r}_0 = \mathbf{p}_{00} = \mathbf{q}_{00}$ and $\mathbf{r}(1) = \mathbf{r}_6 = \mathbf{p}_{33} = \mathbf{q}_{33}$ as end points. For end tangents, we use (10) evaluated at $(s,t) = (u,v) = (0,0)$ and $(1,1)$. Thus, for suitable choices of k in (11), the first and fifth control points of $\mathbf{r}(\xi)$ are given by

$$\mathbf{r}_1 \,=\, \mathbf{r}_0 + \frac{\mathbf{r}'(0)}{6} \qquad \text{and} \qquad \mathbf{r}_5 \,=\, \mathbf{r}_6 - \frac{\mathbf{r}'(1)}{6}\,.$$

The intermediate control points $\mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4$ are typically fixed through a least–squares fit of $\mathbf{r}(\xi)$ to several (typically 10) points sampled along the true intersection curve. The method of [24] is used to assign parameter values in this fitting process.

Consider now the *error curves* defined for each surface by

$$\mathbf{e_p}(\xi) \,=\, \mathbf{r}(\xi) - \mathbf{p}(\xi,\xi)\,, \qquad \mathbf{e_q}(\xi) \,=\, \mathbf{r}(\xi) - \mathbf{q}(\xi,\xi) \tag{14}$$

— i.e., the differences between the approximate intersection curve (13) and the images of the parameter–domain curves (12) in $\mathbb{R}^3$. In order for the latter curves to coincide with the former, we must have $\mathbf{e_p}(\xi) \equiv \mathbf{0}$ and $\mathbf{e_q}(\xi) \equiv \mathbf{0}$. Note that the two error curves (14) are of degree 6 — we shall denote their control points by $\mathbf{e_p}_0, \ldots, \mathbf{e_p}_6$ and $\mathbf{e_q}_0, \ldots, \mathbf{e_q}_6$.

Now the control points of the error curve $\mathbf{e_p}(\xi)$ are given by

$$\mathbf{e_p}_i \,=\, \mathbf{r}_i - \sum_{j=\max(0,i-3)}^{min(i,3)} \frac{\binom{3}{j}\binom{3}{i-j}}{\binom{6}{i}}\, \mathbf{p}_{j,i-j}\,, \quad i = 0,\ldots,6\,,$$

and similarly for $\mathbf{e_q}(\xi)$, with $\mathbf{p}_{j,i-j}$ replaced by $\mathbf{q}_{j,i-j}$. In particular, $\mathbf{e_p}_0 = \mathbf{e_p}_6 = \mathbf{0}$ by construction, and the remaining control points are

$$\begin{aligned}
\mathbf{e_p}_1 &= \mathbf{r}_1 - \frac{\mathbf{p}_{01} + \mathbf{p}_{10}}{2}\,, \\
\mathbf{e_p}_2 &= \mathbf{r}_2 - \frac{\mathbf{p}_{02} + 3\mathbf{p}_{11} + \mathbf{p}_{20}}{5}\,, \\
\mathbf{e_p}_3 &= \mathbf{r}_3 - \frac{\mathbf{p}_{03} + 9(\mathbf{p}_{12} + \mathbf{p}_{21}) + \mathbf{p}_{30}}{20}\,, \\
\mathbf{e_p}_4 &= \mathbf{r}_4 - \frac{\mathbf{p}_{13} + 3\mathbf{p}_{22} + \mathbf{p}_{31}}{5}\,, \\
\mathbf{e_p}_5 &= \mathbf{r}_5 - \frac{\mathbf{p}_{23} + \mathbf{p}_{32}}{2}\,.
\end{aligned}$$

Similarly for the control points of $\mathbf{e_q}(\xi)$.

The image in $\mathbb{R}^3$ of the parameter–domain curve $(s(\xi),t(\xi))$ defined by (12) will match the approximate intersection curve (13) exactly if we can ensure that

$$\mathbf{e}_{\mathbf{p}1} = \mathbf{e}_{\mathbf{p}2} = \mathbf{e}_{\mathbf{p}3} = \mathbf{e}_{\mathbf{p}4} = \mathbf{e}_{\mathbf{p}5} = \mathbf{0}. \tag{15}$$

To satisfy these consistency conditions, we use perturbed control points $\tilde{\mathbf{p}}_{jk} = \mathbf{p}_{jk} + \Delta\mathbf{p}_{jk}$. The same approach is also used for the surface $\mathbf{q}(u,v)$. Under the appropriate perturbations, the perturbed surfaces will satisfy

$$\tilde{\mathbf{p}}(s(\xi),t(\xi)) \equiv \tilde{\mathbf{q}}(u(\xi),v(\xi)) \equiv \mathbf{r}(\xi)\,.$$

Now since the consistency conditions (15) amount to five (vector) constraints, we perturb only the five boundary control points

$$\mathbf{p}_{01},\,\mathbf{p}_{02},\,\mathbf{p}_{03},\,\mathbf{p}_{13},\,\mathbf{p}_{23}. \tag{16}$$

These points have least influence on the retained portion of the trimmed surface along the sides $s = 1$, $t = 0$ adjacent to other patches of the parent surface. Thus, although we formally impose only C^0 continuity along these sides, in practice the departure from C^1 continuity will typically be quite small. The control point perturbations for $\mathbf{q}(u,v)$ are treated in exactly the same manner.

On replacing $\mathbf{p}_{jk}$ by $\mathbf{p}_{jk} + \Delta\mathbf{p}_{jk}$ in equations (15), and assuming that the only non–zero perturbations are those associated with the boundary control points (16), we obtain the required perturbations explicitly as

$$\begin{aligned}
\Delta\mathbf{p}_{01} &= 2\mathbf{r}_1 - (\mathbf{p}_{01} + \mathbf{p}_{10})\,,\\
\Delta\mathbf{p}_{02} &= 5\mathbf{r}_2 - (\mathbf{p}_{02} + 3\mathbf{p}_{11} + \mathbf{p}_{20})\,,\\
\Delta\mathbf{p}_{03} &= 20\mathbf{r}_3 - (\mathbf{p}_{03} + 9\mathbf{p}_{12} + 9\mathbf{p}_{21} + \mathbf{p}_{30})\,,\\
\Delta\mathbf{p}_{13} &= 5\mathbf{r}_4 - (\mathbf{p}_{13} + 3\mathbf{p}_{22} + \mathbf{p}_{31})\,,\\
\Delta\mathbf{p}_{23} &= 2\mathbf{r}_5 - (\mathbf{p}_{23} + \mathbf{p}_{32})\,.
\end{aligned} \tag{17}$$

The required perturbations for the control points of $\mathbf{q}(u,v)$ are given by analogous expressions. Note that, if one surface is planar and the approximate intersection curve (13) is also planar, these perturbations will not compromise the planarity.

The strategy for determining the unknown coefficients of (13) and the control point perturbations is as follows. The perturbations $\Delta\mathbf{p}_{01}$ and $\Delta\mathbf{p}_{23}$ (and also $\Delta\mathbf{q}_{01}$ and $\Delta\mathbf{q}_{23}$) depend on the unknown scalar k in (10). Values are chosen for this parameter to minimize the discrepancy in surface normals along the sides $s =$

1 and $t = 0$, where the trimmed patch is contiguous with other patches of the parent surface (likewise along the sides $u = 1$ and $v = 0$ of the other patch). Once appropriate k values are found in this manner, $\mathbf{r}_1$ and $\mathbf{r}_5$ are known, and we can obtain $\Delta\mathbf{p}_{01}$ and $\Delta\mathbf{p}_{23}$ from (17); likewise $\Delta\mathbf{q}_{01}$ and $\Delta\mathbf{q}_{23}$ for the other surface. Knowing $\mathbf{r}_0$, $\mathbf{r}_1$ and $\mathbf{r}_5$, $\mathbf{r}_6$ the intermediate coefficients $\mathbf{r}_2$, $\mathbf{r}_3$, $\mathbf{r}_4$ are computed by the least–squares fit, and the remaining perturbations $\Delta\mathbf{p}_{02}$, $\Delta\mathbf{p}_{03}$, $\Delta\mathbf{p}_{13}$ can then be found from (17); likewise for $\Delta\mathbf{q}_{02}$, $\Delta\mathbf{q}_{03}$, $\Delta\mathbf{q}_{13}$.

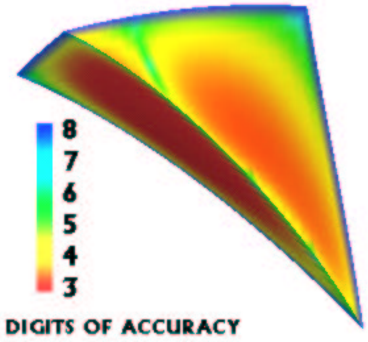

Figure 3: Results from applying the surface perturbation scheme to the patches in Figure 2 over the domain rectangle CD. Colors indicate the Hausdorff measure of the perturbation error.

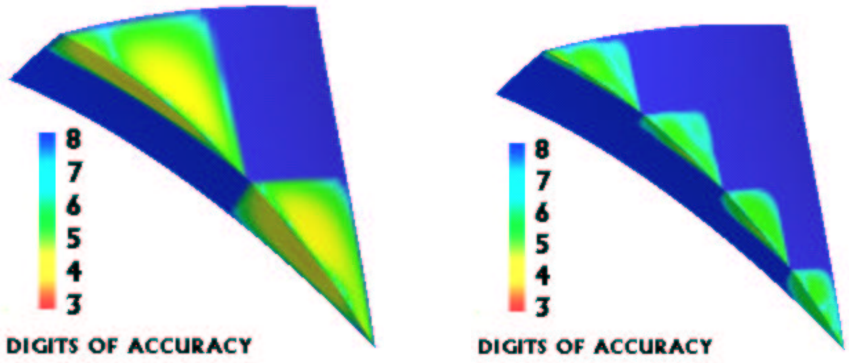

Figure 4: Results from splitting the surface perturbation domains of Figure 3 in half (left) and in quarters (right).

Results from an implementation of the above scheme are shown in Figures 3 and 4. This example employs the patches of Figure 2, over the domain rectangle

CD. Figure 4 illustrates how the approximation error can be be further subdued by splitting the perturbation domains. The solid blue regions are not perturbed at all. By using a greater number of smaller perturbation domains, the Hausdorff error diminishes rapidly. For computer graphics applications, the single perturbation shown in Figure 3 will probably suffice.

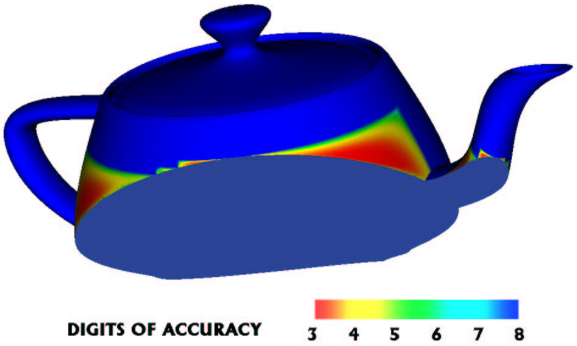

Figure 5: A “watertight” plane section of the Utah teapot.

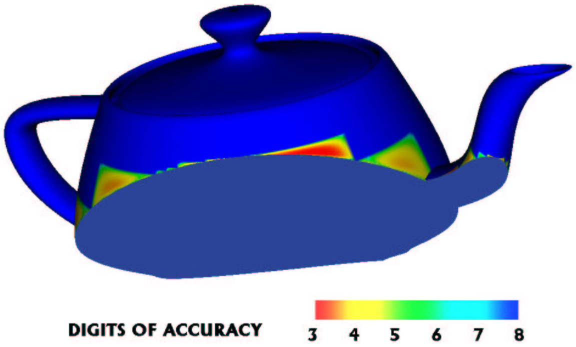

Figure 6: Results from splitting in half the perturbation domains in Figure 5.

The example in Figure 5 shows the teapot sliced by a plane, with the log of the Hausdorff error again painted on the patches. The error regions indicate the boundaries of the teapot patches that are intersected by the plane. Figure 6 shows what happens to the error when the perturbation domains are cut in half. Observe that there is no Hausdorff error on the plane: it is perturbed in plane. The example in Figure 7 shows the teapot sliced by a bicubic patch with the initial perturbation domains. The error can be reduced below any prescribed tolerance by refining the perturbation domains.

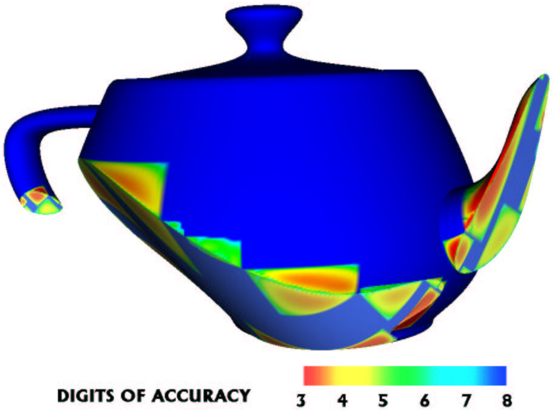

Figure 7: A “watertight” intersection between the Utah teapot and a bicubic patch.

The perturbation method can also be adapted to accommodate rational bicubic surface patches, with the intersection approximated by a rational curve of degree 6. Interpreting the control points and their perturbations as vectors in $\mathbb{R}^4$, one can avoid the degree–doubling that usually occurs in the subtraction of rational functions with different denominators. It is also necessary to match the weights of the common sub–patch corner control points; this can be accomplished by a suitable re–parameterization of the surfaces.

6 Closure

A general scheme for the topologically consistent representation of trimmed free–form surfaces has been presented, based on finding suitable perturbations of the surfaces that ensure exact matching in $\mathbb{R}^3$ of the images of the parameter–domain trim curves. These perturbations can be found by solving a linear system.

An implementation of a simple instance of the scheme, described in §5, offers very promising results. In this case, the displacement maps are bicubic, and can be derived in closed form. Furthermore, planes remain planes under this perturbation, and if one surface is ruled and the other surface is a plane, the intersection is exact and the Hausdorff error is zero. As can be seen in Figure 4, convergence under subdivision is relatively fast. Also, the algorithm runs quickly. The examples shown take less than a second of compute time.

Higher–order versions of the procedure have the potential to offer accurate approximations of the intersection curve with fewer patches. Perturbations that are C^1 and C^2 are also possible, using suitably high degrees. However, initial investigations suggest that careful pre–conditioning strategies may be needed to solve the linear system. We hope to report on these methods in due course.

Appendix: Bernstein basis algorithms

All our algorithms are formulated in terms of the Bernstein basis

$$b_k^n(t) \,=\, \binom{n}{k}(1-t)^{n-k}t^k\,, \qquad k=0,\ldots,n\,.$$

The Bernstein form is numerically very stable [6, 5] and offers geometrical insight into the graph of the function or locus defined in terms of it [4]. We review here a few basic algorithms for polynomials in Bernstein form that are used extensively in the surface trimming procedure — more complete details (and software) may be found in [7, 25].

Products of polynomials

The product of two Bernstein–form polynomials

$$p(t) \,=\, \sum_{i=0}^{m} p_i b_i^m(t) \quad \text{and} \quad q(t) \,=\, \sum_{j=0}^{n} q_j b_j^n(t)$$

can be written as a polynomial of degree $m+n$,

$$p(t)q(t) \,=\, \sum_{k=0}^{m+n} c_k b_k^{m+n}(t)$$

with Bernstein coefficients given by

$$c_k \,=\, \sum_{r=\max(0,k-n)}^{\min(m,k)} \frac{\binom{m}{r}\binom{n}{k-r}}{\binom{m+n}{k}}\, p_r q_{k-r}\,, \quad k=0,\ldots,m+n\,.$$

Composition of polynomials

We are concerned with computing the Bernstein coefficients of the polynomial $w(t) = q(p(t))$ defined by substituting a polynomial $u = p(t)$ into a polynomial $q(u)$. If $p(t)$ and $q(u)$ are of degree m and n, respectively

$$p(t) = \sum_{j=0}^{m} p_j b_j^m(t) \qquad \text{and} \qquad q(u) = \sum_{k=0}^{n} q_k b_k^n(u),$$

then $w(t)$ is evidently of degree mn,

$$w(t) = q(p(t)) = \sum_{i=0}^{mn} w_i b_i^{mn}(t).$$

The *product algorithm* [3] offers an elegant means of computing the Bernstein coefficients $w_0, \ldots, w_{mn}$. This algorithm populates a tetrahedral array of numbers $h_{i,j}^{(s)}$ as follows. In the first level, $s = 0$, we set

$$h_{i,0}^{(0)} = q_i \qquad \text{for } i = 0, \ldots, n.$$

The entries for successive levels, $s = 1, \ldots, n$, are then computed recursively using the formula

$$h_{i,j}^{(s)} = \sum_{k=\max(0,j-m)}^{\min(j,ms-m)} \frac{\binom{ms-m}{k}\binom{m}{j-k}}{\binom{ms}{j}} \left[(1 - p_{j-k}) h_{i,k}^{(s-1)} + p_{j-k} h_{i+1,k}^{(s-1)} \right]$$

for $i = 0, \ldots, n - s$ and $j = 0, \ldots, ms$. Once the array is populated, the last level $s = n$ contains the desired coefficients of $w(t)$:

$$w_j = h_{0,j}^{(n)} \quad \text{for } j = 0, \ldots, mn.$$

References

[1] Biermann, H., Kristjansson, D., and Zorin, D. (2001), Approximate Boolean operations on free–form solids, in *Proceedings of SIGGRAPH 2001*, ACM, 185–194.

[2] Cho, W. J., Maekawa, T., Patrikalakis, N. M., and Peraire, J. (1999), Topologically reliable approximation of trimmed polynomial surface patches, *Graphical Models and Image Processing* **61**, 84–109.

[3] DeRose, A. D. (1988), Composing Bézier simplexes, *ACM Transactions on Graphics* **7**, 198–221.

[4] Farin, G. (1997), *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press.

[5] Farouki, R. T., and Goodman, T. N. T. (1996), On the optimal stability of the Bernstein basis, *Mathematics of Computation* **65**, 1553–1566.

[6] Farouki, R. T., and Rajan, V. T. (1987), On the numerical condition of polynomials in Bernstein form, *Computer Aided Geometric Design* **4**, 191–216.

[7] Farouki, R. T., and Rajan, V. T. (1988), Algorithms for polynomials in Bernstein form, *Computer Aided Geometric Design* **5**, 1–26.

[8] Farouki, R. T. (1986), The characterization of parametric surface sections, *Computer Vision, Graphics, and Image Processing* **33**, 209–236.

[9] Farouki, R. T. (1987), Trimmed surface algorithms for the evaluation and interrogation of solid boundary representations, *IBM Journal of Research and Development* **31**, 314–334.

[10] Farouki, R. T. (1999), Closing the gap between CAD model and downstream application (report on the SIAM Workshop on Integration of CAD and CFD, UC Davis, April 12–13, 1999), *SIAM News* **32** (5), 1–3.

[11] Gonzalez–Vega, L., and Necula, I. (2002), Efficient topology determination of implicitly defined algebraic plane curves, *Computer Aided Geometric Design* **19**, 719–743.

[12] Grandine, T. A., and Klein, F. W. (1997), A new approach to the surface intersection problem, *Computer Aided Geometric Design* **14**, 111–134.

[13] Hamann, B., and Tsai, P. Y. (1996), A tessellation algorithm for the representation of trimmed NURBS surfaces with arbitrary trimming curves, *Computer Aided Design* **28**, 461–472.

[14] Hu, Y. P., and Sun, T. C. (1997), Moving a B–spline surface to a curve — a trimmed surface matching algorithm, *Computer Aided Design* **29**, 449–455.

[15] Katz, S., and Sederberg, T. W. (1988), Genus of the intersection curve of two rational surface patches, *Computer Aided Geometric Design* **5**, 253–258.

[16] Lane, J. M., and Riesenfeld, R. F. (1981), Bounds on a polynomial, *BIT* **21**, 112–117.

[17] Litke, N., Levin, A., and Schröder, P. (2001), Trimming for subdivision surfaces, *Computer Aided Geometric Design* **18**, 463–481.

[18] Patrikalakis, N. M., and Maekawa, T. (2002), Intersection problems, in *Handbook of Computer Aided Geometric Design*, (G. Farin, J. Hoschek, and M.-S. Kim, eds.), North Holland, 623–649.

[19] Rockwood, A. P., Heaton, K., and Davis, T. (1989), Real-time rendering of trimmed surfaces, in *Proceedings of SIGGRAPH 89*, ACM, 107–116.

[20] Sederberg, T. W., and Nishita, T. (1991), Geometric Hermite approximation of surface patch intersection curves, *Computer Aided Geometric Design* **8**, 97–114.

[21] Sederberg, T. W. (1983), *Implicit and Parametric Curves and Surfaces for Computer Aided Geometric Design*, PhD thesis, Purdue University.

[22] Seidel, R. (1998), The nature and meaning of perturbations in geometric computing, *Discrete and Computational Geometry* **19**, 1–17.

[23] Sherbrooke, E. C., and Patrikalakis, N. M. (1993), Computation of the solutions of nonlinear polynomial systems, *Computer Aided Geometric Design* **10**, 379–405.

[24] Speer, T., Kuppe, M., and Hoschek, J. (1998), Global reparameterization for curve approximation, *Computer Aided Geometric Design* **15**, 869–877.

[25] Tsai, Y.-F., and Farouki, R. T. (2001), Algorithm 812: BPOLY: An object–oriented library of numerical algorithms for polynomials in Bernstein form, *ACM Transactions on Mathematical Software* **27**, 267–296.