1997

Computing Moments of Piecewise Polynomial Surfaces

Carlos Gonzalez-Ochoa

Scott McCammon

Jörg Peters

Report Number:
97-009

COMPUTING MOMENTS OF PIECEWISE POLYNOMIAL SURFACES

Carlos Gonzalez-Ochoa
Scott McCammon
Jorg Peters

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD-TR 97-009
January 1997

Computing moments of piecewise polynomial surfaces

Carlos Gonzalez-Ochoa Scott McCammon Jörg Peters

January 29, 1997

Abstract

Combining the advantages of a low-degree polynomial surface representation with Gauss' divergence theorem allows efficient and exact calculation of the moments of objects enclosed by a free-form surface. Volume, center of mass and the inertia tensor can be computed in seconds even for complex objects with 10^5 patches while changes due to local modification of the surface geometry can be computed in real time as feedback for animation or design. Speed and simplicity of the approach allow solving the inverse problem of modelling to match prescribed moments.

1 Introduction

Realistic animation and geometric design must both pay close attention to the physics implied by the first few moments, the volume, center of mass and inertia frame, of the objects they manipulate. A jug whose fill level is inconsistent with the amount of water filled into it, or that does not topple over when the projection of the center of mass moves outside the support, puzzles and distracts the viewer and can lead to disaster when unobserved in the design process. While animators and designers often have an excellent intuitive grasp of the approximate physics implied by shape, the immediate and exact feedback from a tool like the one developed below can save the user both time and energy. Real time visual feedback and visualization of moments and the solution of simple inverse problems help identifying critical points in the moment distribution.

The key to fast and exact computation of moments of smooth free-form surfaces is to combine the advantages of a low-degree polynomial surface representation with Gauss' divergence theorem. In Section 2, we show how Gauss' divergence theorem reduces the computation of the moments of a solid to the evaluation of an integral over the surface that can be evaluated explicitly if the surface has a piecewise polynomial parametrization. In Section 3, we select surface splines [10] as the currently most appropriate representation of smooth free-form surfaces for fast and exact moment computation. In Section 4, we discuss the details of the evaluation of the surface integrals as simple averages and bound the computation effort. In Section 5, we measure and visualize moments and in Section 6 we model geometry to match moments. Section 7 lists extensions and more applications ahead.

1.1 Prior work

A search through the list of standard text books on geometric modeling did not unearth any reference to the moments of the objects modeled. The engineering literature is rich in applications of moments but the computation is typically based on cellular, i.e. piecewise constant, or polyhedral, i.e. piecewise linear, approximations to the object (see e.g. [8], and the massprop command in the ACIS solid modeler). The direct and exact computation of the moments of smooth (and non-smooth) free-form surfaces as explained in this paper should in principle be familiar to anyone who studied advanced calculus but appears to be new in the context of free-form surfaces of arbitrary patch layout and topological genus.

2 The elegant divergence theorem

First mentioned by Gauss in 1813, the divergence theorem (see e.g. [11] 10.51), relates a volume integral to an integral over the surface. Given a map $\mathbf{f} : R^3 \mapsto R^3$, its divergence $\nabla \cdot \mathbf{f} = \sum \frac{\partial}{\partial x_i} f_i$ and its normal component $\mathbf{f} \cdot N = \sum f_i N_i$, the theorem states that

$$\int_V \nabla \cdot \mathbf{f} \; dV = \int_S \mathbf{f} \cdot N \; dS$$

i.e. that the integral of the divergence of $\mathbf{f}$ over the volume V equals the integral of the normal component over the surface S of V.

To apply the theorem to moment calculation denote the parametrization of the surface S by

$$\mathbf{x}(\mathbf{u}) = [x, y, z](u, v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix},$$

and the domain of $\mathbf{x}$ by U, i.e. $(u, v) \in U$. The surface normal is the vector

$$N = \mathbf{n}/|\mathbf{n}|, \quad \text{where} \quad \mathbf{n} = \frac{\partial \mathbf{x}}{\partial u} \wedge \frac{\partial \mathbf{x}}{\partial v}.$$

The crucial observation is that by change of variable

$$\int_S dS = \int_U |\mathbf{n}| \; dudv.$$

That is, the area element $|\mathbf{n}|$ is the inverse of the normalization factor of the normal direction so that

$$\begin{aligned} \int_V \nabla \cdot \mathbf{f} \; dV &= \int_S \mathbf{f} \cdot \mathbf{n}/|\mathbf{n}| \; dS \\ &= \int_U \mathbf{f} \cdot \mathbf{n} \; dudv \end{aligned}$$

Now if both $\mathbf{f}$ and $\mathbf{x}$ are polynomial the integrand is polynomial; and if U is a simple domain, say a triangle or square, then it is clear, and made precise in Section 4, that the integral can be determined efficiently, explicitly and exactly. This remains true if the parametrization $\mathbf{x}$ of S consists of patches $\mathbf{x}^i$ and U is the union of the patch domains U^i:

$$\int_U \mathbf{f} \cdot \mathbf{n} \; dudv = \sum_i \int_{U^i} \mathbf{f} \cdot \mathbf{n}^i \; du^i dv^i.$$

We now consider specific choices of $\mathbf{f}$ that allow us to determine the first three moments of a solid with uniform mass distribution. It is efficient to choose $\mathbf{f} = [0, 0, f_3]$ so that we need to compute only

$$n_3 = \frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u}.$$

The zeroth moment, $\mathcal{M}_0 = \int_V dV$, measures the volume enclosed by S. With $f_3 = z$,

$$\boxed{\mathcal{M}_0 \;\; = \int_U z \; n_3 \; dudv.}$$

The first moments $\mathcal{M}_i = \int_V x_i dV$ represent the components of the center of mass. Choosing f_3 as xz, yz and $z^2/2$ respectively yields

$$\boxed{\begin{array}{ll} \mathcal{M}_1 & = \int_U xz \;\; n_3 \; dudv \\ \mathcal{M}_2 & = \int_U yz \;\; n_3 \; dudv \\ \mathcal{M}_3 & = \int_U z^2/2 \; n_3 \; dudv \end{array}}$$

The second moments yield the components of the inertia tensor

$$\mathcal{I} = \begin{bmatrix} \int (x_2^2 + x_3^2) dV & \int x_1 x_2 dV & \int x_1 x_3 dV \\ \int x_2 x_1 dV & \int (x_1^2 + x_3^2) dV & \int x_2 x_3 dV \\ \int x_3 x_1 dV & \int x_3 x_2 dV & \int (x_1^2 + x_2^2) dV \end{bmatrix}.$$

We list $\mathcal{M}_{ij} = \int_V x_i x_j dV$.

$$\boxed{\begin{array}{ll} \mathcal{M}_{11} & = \int_U x^2 z \;\; n_3 \; dudv \\ \mathcal{M}_{12} & = \int_U xyz \;\; n_3 \; dudv \\ \mathcal{M}_{13} & = \int_U xz^2/2 \; n_3 \; dudv \\ \mathcal{M}_{22} & = \int_U y^2 z \;\; n_3 \; dudv \\ \mathcal{M}_{23} & = \int_U yz^2/2 \; n_3 \; dudv \\ \mathcal{M}_{33} & = \int_U z^3/3 \;\; n_3 \; dudv \end{array}}$$

The above formulas assume that the volume has homogeneous density. If this is not the case but the anti-derivative of the moment-weighted mass distribution $\rho(x, y, z)$ is known, we can easily adapt the f_3 term and work with similar formulas in the same framework.

3 Selecting an efficient representation

The previous section did not depend on a particular representation of the surface S. However, it pointed out that for efficient and exact computability of the moments the following properties of the surface patches are desirable.

1. polynomial patches
2. low degree patches
3. few patches
4. simple domains

These properties rule out several otherwise excellent surface representations. For example, S-patches [3] and the recent construction by Hughes and Grimm [7] are rational and of high degree so that integrals are best evaluated by numerical approximation. Simplex splines [6] use a large number of pieces and generalized subdivision surfaces, e.g. [4] [1] [12], lack a global analytic definition for application of the divergence theorem. Standard tensor-product B-splines match the requirements but fail at modeling smooth surfaces of arbitrary patch layout or topological genus. Trimmed NURBS patches on the other hand generically violate the requirement of simple domains.

Surface splines [10] seem to best fit the bill. Surface splines define a piecewise cubic manifold capable of modeling smooth and non-smooth surfaces of arbitrary patch layout and topological genus. The embedding of the smooth manifold into R^3 is locally governed by so-called cut ratios. These numbers allow for local distribution of curvature and fine-tuning of shape; where they are set to zero sharp edges and vertices result so that also purely polyhedral objects can be modeled in this frame work. Surface splines come in three flavors, with an underlying representation of exclusively 4-sided, exclusively 3-sided and a mixture of 3-and 4-sided patches, respectively. Exclusively 4-sided patches have recently been used for reconstruction [5]. However, the exclusively 3-sided and the mixed representation have better shape properties, satisfy the strong local convex hull property and have lower algebraic and parametric degree. According to [9] the 3-sided patches can be represented as standard linearly-trimmed NURBS quilts and hence stored, transmitted and rendered in standard form. The appendix has additional remarks and details of surface splines.

4 Computing surface integrals

To determine the moments, integrals of type

$$\int_U x^i y^j z^k n_3 \, dudv$$

are evaluated. For cubic surface splines, the integrands of the qth moment are scalar-valued polynomial pieces in Bernstein-Bézier form of total degree

$$d = 3(q+1)+4$$

in two variables, i.e. degree 7, 10 and 13 for the first three moments. These pieces are exactly integrated by taking the average of the Bernstein coefficients (see e.g. [2]); that is integration consists of summing the $k := (d+2)(d+1)/2$ coefficients and dividing by k.

To verify that the moment calculation fits into an interactive environment, we compute a bound on the number of operations. The polynomial n_3 is a product of derivatives. Algorithms for differentiation and multiplication of polynomials in the Bernstein-Bézier form are standard and can be found in most CAGD text books and in the short article [2]. Differentiation of a vector-valued polynomial of degree d requires $d(d-1)/2$ vector differences. Multiplication of two scalar-valued polynomials of degree d_1 and d_2 respectively, requires $(d_1+1)d_1(d_2+1)d_2/4$ multiplications and additions. Thus computing n_3 requires $3*4*6+2*2*9+15 = 123$ additions and $2*2*9 = 36$ multiplications. The additional work for each moment is tabulated below.

moment	d	+	*
n_3	4	123	36
volume	7	28+60	60
centroid	10	55+56*3	56*3
inertia	13	91+110*3	110*3

Hence, we can compute all components of the first three moments of a patch in less than 6K operations. On a 100 MHz processor this allows computing at least 10 patches per millisecond, enough for interactive feedback when updating the surface.

5 Measuring Moments

The principal inertia directions are axes of symmetry with respect to mass distribution and are computed as eigenvectors of the inertia tensor $\mathcal{I}$. The eigenvalues of the inertia tensor measure resistance to torque. Both sets of information are displayed in the inertia frame, a stencil attached to the center of mass whose legs are aligned with the principal inertia directions and stretched according to the corresponding eigenvalues. A smoothed brick shape as shown in Figure 1 is probably the simplest example illustrating the inertia frame. Figure 2 shows three snapshots of a real time design sequence that modifies the geometry of the brick. More complicated objects are shown in Figure 3. In Figure 3(a), the geometry of the object is changed by moving one arm of the robot gripper, the result of lifting the surface spline control net. In Figure 3(b), the geometry of the object is changed by smoothing, the result of changing the surface spline cut ratios from zero to 0.5. The shadow is generated as an identical object with all z-components collapsed to zero.

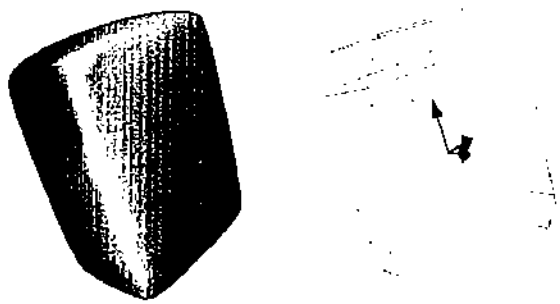

Figure 1: The inertia frame of a smoothed brick shape. The inertia frame is formed by the principle directions of the inertia tensor attached to the center of mass.

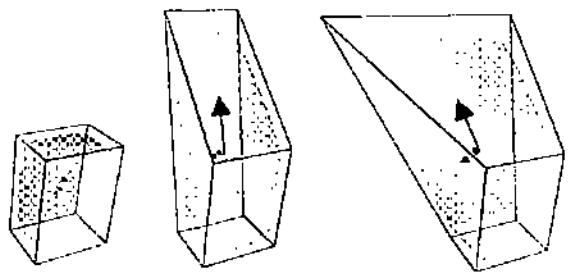

Figure 2: Change of the inertia frame under distortion of the transparently rendered brick.

6 Modeling to match moments

The ability to measure moments efficiently as demonstrated in the previous section is the key to solving the inverse problem of designing geometry to match a prescribed (set of) moments. In general this problem is underconstrained since the geometry offers many degrees of freedom. The standard approach to regularizing the problem is to add variational constraints. Rather than grazing on this fertile field, we look at some practically relevant problems where the geometry is restricted to change only in one parameter.

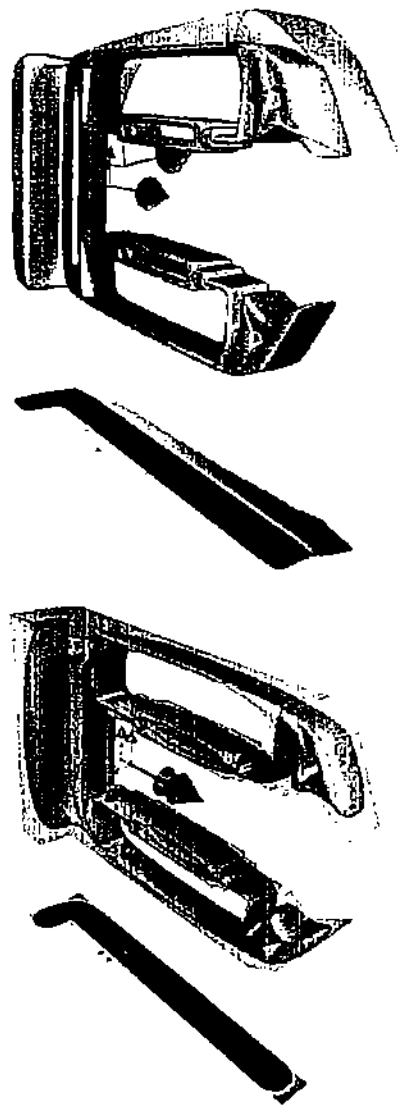

Figure 3: Change of the inertia frame under deformation and smoothing.

6.1 Volume matching

Consider filling water into a jug held in fixed position. Here, until overflow, the geometry of the water is fixed by the geometry of the container; the only degree of freedom is the height of the water level. A well-posed problem is to ask for the *fill height* $\bar{h}$ of the container $\mathbf{x}$ given a fixed water volume V_0. The problem is non-trivial, since the top surface of the water at rest is a plane that cuts the container patches in algebraic curves resulting in challenging boundary integrals when an exact solution is sought. Rotating the container if necessary so that the negative z-direction agrees with the direction of gravity the problem can be formally written as

$$\text{find } \bar{h}, \text{such that} \quad \text{volume } \mathbf{x}(\bar{h}, \mathbf{u}) = V_0.$$

The problem has the nice property that $\bar{h}$ is a monotone function of the volume so that bisection will not only succeed if the volume fits into the container but also yield a sequence of upper and lower bounds in the process. Newton's method guarantees no such bound and has to cope with the fact that $\bar{h}$ is generally not a differentiable function of V_0: consider the rate of change at the branch point when filling an extruded T-shape.

Each bisection step requires computing a volume bounded by the intersection of a piecewise cubic freeform surface and a half-space. To avoid the exact intersection computation, we subdivide the patches in the vicinity of the intersection such that we overestimate or underestimate the volume already enclosed. The computation is fast enough to allow the real time animation shown in Figure 4. Here the volume is kept constant and the fill height is recomputed while the containing shape is interactively deformed.

6.2 Mass matching

As a second example, we determine the point of instability of a gripper placed on a table top. Here the free parameter is the opening angle α and the condition is that the projection of the center of mass $m(\mathbf{x})$ comes to lie on the on the table edge E, i.e.

$$\text{find } \alpha \text{ such that} \quad m(\mathbf{x}((\alpha, \mathbf{u})) \subset E.$$

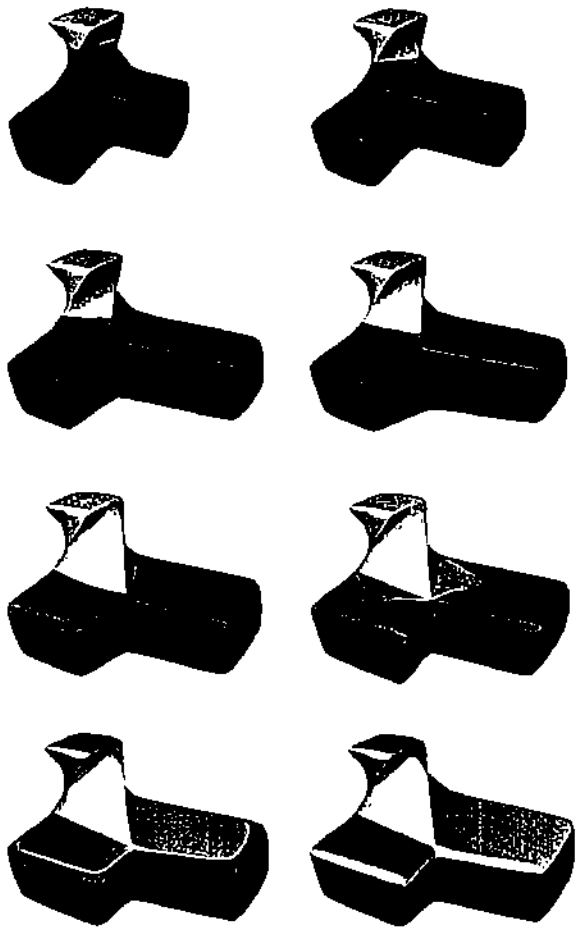

Figure 4: Deformation of an object alters the fill height for a fixed fill volume.

Bisection with a call to the mass computing module rather than the volume module solves the problem. The sequence of computations is animated in Figure 5. Note that as α varies the blend of the finger with the body is affected. Therefore the standard engineering approach of decomposing the gripper into a fixed and a moving part and adding their respective centers of mass to compute the center of mass of the whole object yields a less accurate result.

6.3 Inertia matching

The inertia tensor $\mathcal{I}$ is a positive definite, symmetric matrix. As pointed out in Section 5, its eigenvectors and eigenvalues can be used, together with the center of mass, to define a frame that is unique if the eigenvalues are distinct. We can use this inertia frame to efficiently find an unknown rigid motion by which an object is displaced with respect to a reference position. For example, in object registration, a cloud of measured points from a physical object and a surface model of this object (Figure 6) are to be brought in agreement by a rigid motion. This motion is efficiently computed as an affine transformation relating the inertia frames of the two entities. In practice, this method works very well even when the sampled points are not exactly on the given model and is vastly faster than sampling the surface model and attempting a least-squares fit of the two point clouds.

7 A sketch of further applications

Moment-based tools will give animators and designers a much better feel for the objects they are working with and undoubtedly inspire innovative uses. A likely application, hinted at in Figure 6, is physics-based animation. Mechanical engineers for their part may appreciate exact moment information for complex objects.

The authors of this paper, varying from undergraduate to postgraduate, learned and appreciated multivariate calculus anew while implementing and using the tool. Consequently we used it to generate non-trivial applications of Gauss' divergence theorem

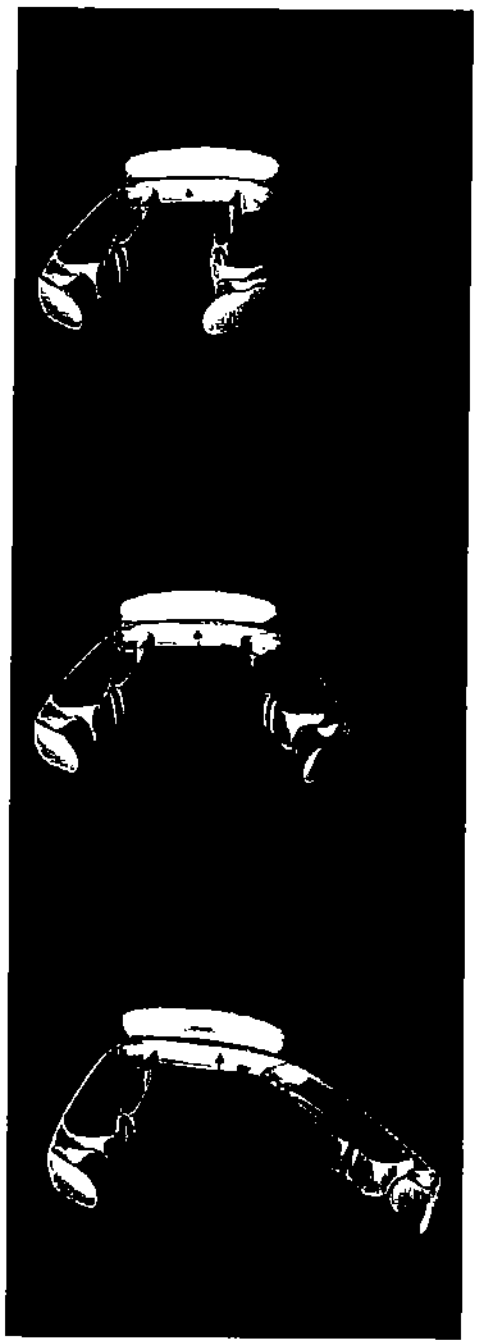

Figure 5: Determining the point of instability when varying the opening angle.

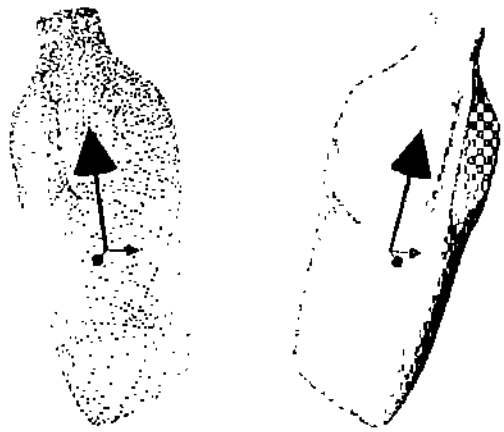

Figure 6: Using the inertia frame to match an object to a cloud of sample points.

for an undergraduate advanced calculus class. The examples kindled interest in the underlying mathematics and will be used in future lectures.

As mentioned in Section 6, automatic shape modification based on matching moments is an intriguing though underconstrained challenge. For a low degree representation like surface splines, it is actually possible to relate a change of the moments explicitly to a change of the spline parameters, extending the hope that variational criteria more complex than generic least-squares minimization will be well-defined and computationally tractable.

References

[1] CATMULL, E., AND CLARK, J. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Des. 10* (1978), 350 – 355.

[2] DE BOOR, C. B-form basics. In *Geometric Modeling: Algorithms and New Trends*, G. Farin, Ed. SIAM, Philadelphia, 1987, pp. 131–148.

[3] DEROSE, T. D., AND LOOP, C. T. S-patches: A class of representations for multi-sided surface

patches. Tech. rep., University of Washington, Department of Computer Science, Seattle, 98195, 1988.

[4] DOO, D., AND SABIN, M. Behaviour of recursive subdivision surfaces near extraordinary points. *CAD 10*, 6 (November 1978), 356–360.

[5] ECK, M., AND HOPPE, H. Automatic reconstruction of B-spline surfaces of arbitrary topological type. *Computer Graphics – Proceedings of Siggraph 95* (1995), 359–368.

[6] FONG, P., AND SEIDEL, H.-P. An implementation of triangular B-spline surface over arbitrary triangulations. *CAGD 10* (1993), 267–275.

[7] GRIMM, C. M., AND HUGHES, J. F. Modeling surfaces of arbitrary topology using manifolds. *Computer Graphics – Proceedings of Siggraph 95* (1995), 359–368.

[8] LEE, Y. T., AND REQUICHA, A. A. G. Algorithms for computing the volume and other integral properties of solids. II a family of algorithms based on representation conversion and cellular approximation. *Commun. ACM (USA) 25* (Sept. 1982), 642–650.

[9] PETERS, J. Personal Communication, Aug 1996.

[10] PETERS, J. C^1-surface splines. *SIAM Journal on Numerical Analysis 32*, 2 (1995), 645–666.

[11] RUDIN, W. *Real and complex analysis*. Series in Higher Mathematics. McGraw-Hill, 1987.

[12] SCHRODER, SWELDENS, AND ZORIN. Interpolating subdivision for meshes with arbitrary topology. *Computer Graphics – Proceedings of Siggraph 96* (1996), 189–192.

Appendix: Surface spline synopsis

Surfacing with surface splines consists of two stages. In the first, illustrated in Figure 7, an arbitrarily connected input polyhedron is transformed into a planar-cut polyhedron by (edge and) corner cutting (cf [10] pp 649–650). A planar-cut polyhedron is a polyhedron with every interior vertex surrounded by four facets. The first and third facet are four-sided, the other two must be planar if they have more than four edges. – Zero depth cuts, corresponding to a non-smooth embedding of the C^1 manifold into R^3, are permitted so that also purely polyhedral approximations can be modeled and measured in this frame work. – In the second step, the Bézier coefficients of the cubic patches are computed as simple convex combinations of the planar-cut polyhedron (cf [10] p 652).

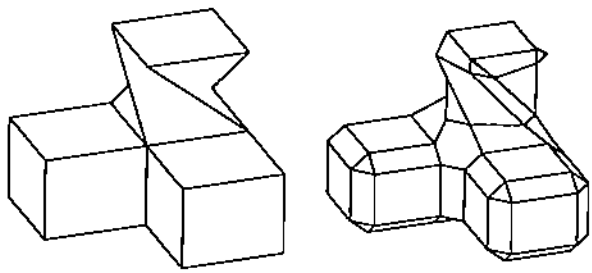

Figure 7: Input polyhedron and planar-cut polyhedron.