Computer Aided Geometric Design 7 (1990) 191–195
North-Holland

Local smooth surface interpolation: a classification *

Jörg PETERS

Center of Mathematical Sciences and Computer Sciences Department, University of Wisconsin–Madison, Madison, WI 53705, USA

Presented at Oberwolfach April 1989
Received August 1989; revised November 1989

Abstract. A classification of algorithms for local smooth surface interpolation with piecewise polynomials is given.

Keywords. Interpolation, C^1 surface, oriented tangent plane continuity, weight functions, blend functions, single patch approach, blending approach, splitting approach, vertex enclosure problem.

Table 1 below classifies some algorithms in the literature that construct smooth, interpolating surfaces with piecewise polynomials, i.e. algorithms that convert a 'mesh of 3D data' into the coefficients of a piecewise polynomial representation. The 'data' column describes the interpolation conditions in more detail: upper case letters indicate the highest order data matched by the interpolant; lower case letters signal important restrictions on the data. The 'sides' column specifies how many sides a facet of the mesh may have; if the entry is k, an arbitrary number of sides can be handled by the algorithm. Tensor product patches are prefixed 'bi' in the 'degree' column. The degree of a denominator polynomial (if any) is preceded by a '/'. The 'weights' column, finally, details how the algorithms sew the polynomial patches into a smooth quilt. All entries are explained in more detail below. The algorithms are grouped into blocks according to the three major approaches to the interpolation problem: the single patch approach, the blending approach and the splitting approach.

A regular parametric piecewise polynomial surface is *smooth* or C^1 $(= G^1 = V^1 = VC^1)$, if its pieces (patches) join with **regular oriented tangent plane continuity**; that is, if the surface normals of abutting patches are uniquely defined and agree at every point of the boundary. C^1-smoothness for (bivariate) functions differs from C^1-smoothness for surfaces, since the former labels the transition between maps from the (same) plane to 1-space while the latter characterizes maps from different copies of the unit square or unit triangle to 3-space, thus providing room for a change in parametrization. The following algebraic characterization of the necessary and sufficient constraints for oriented tangent plane continuity applies to patches with an arbitrary number of sides and in particular to any combination of tensor product and total degree patches. Consider a patch p, parametrized by u and v, and its neighbor patch q, parametrized by u and w and the definitions

$$p_u := \frac{\partial}{\partial u} p(u, 0), \qquad p_v := \frac{\partial}{\partial v} p(u, 0) \quad \text{and} \quad q_w := \frac{\partial}{\partial w} q(u, 0).$$

* This research was supported by NSF DMS-8701275.

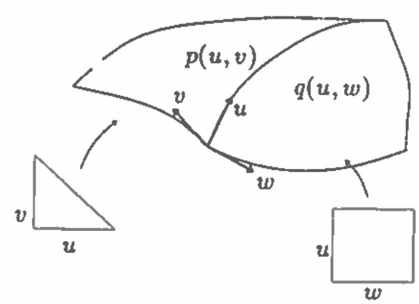
Table 1
Local smooth surface interpolation with piecewise polynomials
N = normal, T = tangent, II = curvature, M = curve mesh, D = transversal derivative on mesh, K = transversal curvature on mesh (surface is C^2), (K ⇒ D ⇒ M ⇒ T ⇒ N); f = cannot model closed surfaces, c = compatible $C^{2,2}$ tensor product spline data assumed, r = N or II is restricted, $_{1.5}$ = 1.5 patches per facet, $_6$ = 6 or 12 patches per facet

data	sides	degree	weights	reference
Pr	4	bi4	1, 0, 0	Beeker '86
M	4	bi6	4, 3, 0	Bézier '79, Sarraga '86
M	3, 4	4, bi4	3, 1, 1 or 2, 1, 0	Peters '89a
IIr	4	bi3	$p_u(N_0 \times N_1)$	Sabin '68
Nr	3, 4	3, bi3	$(1-t)N_0 + t\omega N_1$	Peters '88a
Dfc	4	bi3		Coons '67
Df	3	2(3+3)	(1/1)	Barnhill, Birkhoff, Gordon '73
Df	4	bi4/1		Gregory '74
M	4	bi4/1	1, 1, 1	Chiyokura, Kimura '83
M	3	1+2	[4/2] 1, 1, 1	Herron '85
Dc	5	3+2	[6/6] (1/1)	Charrot, Gregory '84
Kc	k	3+4/8	$[3(k-2)/3(k-2)]$ (1/1)	Gregory, Hahn '89
D	3	3+3	[2/2] (1/1)	Nielson '86
K	3	7+5	[4/4] (1/1)	Hagen, Pottmann '89
Tf	3	3	0, 0, 0	Clough, Tocher '65
Tf	3_6	2	0, 0, 0	Powell, Sabin '77
N	3	4	1, 0, 0	Farin '83
T	3	4	2, 1, 1	Piper '87
M	3, 4	4	1, 1, 1	Shirman, Séquin '87
M	3,4	3	2, 1, 1 and 1, 1, 1	Peters '88b
M	$k > 4$	4	3, 2, 2 and 1, 1, 1	Peters '88b
N	k	5	1, 0, 0	Jones '88
N	$3_{1.5}$	bi3	1, 0, 0	Peters '89b

Then p and q are part of a C^1-surface (see Figure) if and only if there exist scalar-valued *weight functions* λ, μ and ν such that, at each point of the boundary,

(E) $\lambda p_u = \mu p_v + \nu q_w$ [common tangent plane],

(I) $p_v \times p_u \neq 0$ [regularity],
$\mu\nu > 0$ [proper orientation].



(The particular characterization is taken from [Peters '88b]; formulations equivalent to (E) appear in [Liu '86, p. 437], [Beeker '86, p. 225], [Piper '87, p. 227, (4.1)], [Degen '89, p. 10] and [Liu & Hoschek, '89, Theorem 1].) The weight functions are, up to a common factor, polynomials in u; that is $(\lambda, \mu, \nu) = \alpha(p_1, p_2, p_3)$ for polynomials p_1, p_2 and p_3 and an arbitrary (nonzero) function α. Hence, 'without loss of flexibility', they may be chosen as polynomials. The degree of these polynomials can be bounded by the degree of p and q ([Peters '89a], cf. the proof in [Liu '86]). Once the degree of the interpolant is chosen, the triple

degree of λ, degree of μ, degree of ν

characterizes the transition (match) between the patches and motivates the entries in the 'weights' column. Much can be gained by allowing all three weight functions to be non-constant; a match characterized by entries i, j, 0 can be thought of as a match $q_w = \alpha p_u + \beta p_v$, where α and β are restricted to be polynomials instead of polynomials divided by polynomials (cf. [Liu & Hoschek '89, Table 1]).

To appreciate the twofold role of interpolation conditions and the meaning of the 'c' entries in the first column, assume for the moment that no data other than the location of the mesh nodes (data points) are prescribed. Then (E) and (I) form a *nonlinear system of equality and inequality constraints* in the coefficients of p, q, λ, μ and ν. If the patches are represented in Bernstein-Bézier (BB-)form (see [Farin '86], [de Boor '87]), the first layers of the BB-net along the patch boundaries and around each data point are connected via a *global* system. Hence, an important feature of any surface construction is the selection of (geometrically meaningful) coefficients to be fixed a priori (input or derived from the data), so as to arrive at a sufficient and consistent sequence of local and linear constraints. Such input can range from prescribing a patch complex around p [Gregory & Hahn '89], i.e. prescribing q, to just fixing the tangent plane at the data points. While additional interpolation requirements can reduce the algebraic complexity of the task, such data must be both available (or easy to generate) and consistent with the polynomial representation. Consistency is a major problem, since each mesh curve, whether input or constructed as a part of the algorithm, carries second-order data in the nontangential component of its second derivative. At the data (mesh) points such data from different curves comes together creating the **vertex enclosure problem**. If the data point has an odd number of neighbors, then the data can be dealt with (a certain circulant matrix has full rank); but if the number of neighbors is even and the impinging patches are at least second order smooth, then the data has to be constrained (now that matrix is rank deficient). This amounts to one scalar constraint per data point and can be enforced by making the mesh curves match second order data at the point [Peters '89a]. If transversal derivatives are prescribed in addition to the mesh curves as in the blending approach, then each patch boundary prescribes the mixed derivative at the data point independently thus leading to $3k$ constraints per point (one vector constraint for each patch corner). Each of the three major approaches below offers a different solution to the vertex enclosure problem.

An algorithm that constructs just one polynomial patch per mesh facet follows the **single patch approach**. This amounts to interpolation at the mesh nodes followed by a consistent construction of the mesh curves. For [Sabin '68] and [Peters '88a] the characterization of the weight function is replaced by the definition of the normal direction n along the mesh curves, since both algorithms are based on the following, slightly different, but equivalent scalar reformulation of (E): $np_u = np_v = nq_w = 0$. Once n is prescribed, the corresponding constraint system becomes linear and local.

The **blending approach** constructs k polynomial pieces for a k-sided mesh facet. Each piece matches a part of the data, so that a convex combination interpolates the combined data of the facet. The weights of the convex combination, the *blend(ing) functions*, limit the influence of the pieces to those edges where they match the data. Compared to the single patch approach, this *buys simpler pieces at the cost of more pieces*. Since surfaces generated by the blending approach typically interpolate a transversal derivative along with the mesh curves, the blend functions take on an additional role: they introduce a discontinuity in the second derivative at the data point, so that the mixed derivatives need not agree. A typical blend function is $b_1b_2/(b_1b_2 + b_2b_3 + b_3b_1)$ where b_1, b_2 and b_3 are the barycentric coordinates of a domain triangle (cf. [Nielson '86]). Discontinuities in higher order data can be introduced by taking powers of the products $b_l b_m$. This means, however, that consistency of lower order data has to be enforced by some other means (cf. [Hagen & Pottmann '89]). The degree of blended interpolants is difficult to capture. The table lists, in the 'degree' column, the degree of the data polynomials, q, + the degree of the (possibly rational) partial interpolant, and, in the 'weights' column, the degree of the blend functions (in brackets) and the degree of the rational parameters in terms of the original coordinates (in parentheses).

Only the third, the **splitting approach**, alters the original mesh of neighborhood relationships. Following the example of the Clough–Tocher split, k non-overlapping pieces are used to cover a k-sided mesh facet. The object is to *reduce the degree of the interpolant at the cost of more pieces*. Even though the splitting approach creates points with an even number of neighbors, it can interpolate (original) boundary curves. This is possible because the additional boundaries created by splitting create discontinuities in the higher derivatives just like the rational blend functions above.

Fixing the weight functions a priori as in [Goodmann '89] or [Höllig '89] is yet another way of removing the non-linearity in (E). This approach and approaches based on an implicit representation ([Sederberg '85], [Baja & Ihmsung '89], [Dahmen '89]) have not been included into the table. The reader should be weary that smooth surfaces constructed by any of the algorithms listed in the table can be a far cry from the taut, convex surface its designer may have in mind; the degree of the interpolants is only a first indicator.

The author welcomes comments and additions.

References

Bajaj, C.L. and Ihmsung, I. (1989), Hermite interpolation using real algebraic surfaces, *Proc. 5th ACM Symposium on Computational Geometry*, West Germany.

Barnhill, R.E., Birkhoff, G. and Gordon, W.J. (1973), Smooth interpolation in triangles, J. Approx. Theory 8, 114–128.

Beeker, E. (1986), Smoothing of shapes designed with free form surfaces, Computer Aided Design 18 (4), 224–232.

Bézier, P. (1977), Essai de définition numérique des courbes et des surfaces expérimentales, Doctoral dissertation, Université Pierre et Marie Curie, Paris.

de Boor, C. (1987), B-form basics, in: G. Farin, ed., *Geometric Modeling: Applications and New Trends*, SIAM, Philadelphia, PA.

Charrot, P. and Gregory, J.A. (1984), A pentagonal surface patch for computer aided geometric design, Computer Aided Geometric Design, 1, 87–94.

Chiyokura, H. and Kimura, F. (1983), Design of solids with free-form surfaces, Computer Graphics 17 (3), 289–298.

Clough, R.W. and Tocher, J.L. (1965), Finite element stiffness matrices for the analysis of plate bending, *Proc. 1st Conf. Matrix Methods in Struct. Mechanics*, Wright-Patterson AFB AFFDL TR 66-80, 515–545.

Coons, S.A. (1967), Surfaces for computer aided design of space forms, Report MAC-TR-41, Project MAC, MIT, Cambridge, MA.

Dahmen, W. (1989), Smooth piecewise quadric surfaces, Freie Universität Berlin, Preprint A-89-2.

Degen, W. (1989), Explic continuity conditions for adjacent Bézier surface patches, Preprint 5, Math. Inst. B., Universität Stuttgart.

Farin, G. (1983), Smooth interpolation to scattered 3D-data, in: R.E. Barnhill and W. Boehm, eds., *Surfaces in CAGD*, North-Holland, Amsterdam.

Farin, G. (1985), A modified Clough–Tocher interpolant, Computer Aided Geometric Design 2, 19–27.

Farin, G. (1986), Triangular Bernstein–Bézier patches, Computer Aided Geometric Design 3, 83–127.

Goodman, T.N.T. (1988), Closed surfaces defined from biquadratic splines, Report AA 886 University of Dundee.

Gregory, J.A. (1974), Smooth interpolation without twist constraints, in: R.E. Barnhill and R.F. Riesenfeld, eds., *Computer Aided Geometric Design*, Academic Press, New York, 71–88.

Gregory, J.A. and Hahn, J.M. (1989), A C^2 polygonal surface patch, Computer Aided Geometric Design 6, 69–75.

Hagen, H. and Pottmann, H. (1989), Curvature continuous triangular interpolants, Preprint.

Höllig, K. (1989), G-splines, Computer Aided Geometric Design 7, 197–207.

Jones, A.K. (1988), Nonrectangular surface patches with curvature continuity, Computer Aided Design 20 (6), 325.

Herron, G.J. (1985), Smooth closed surfaces with discrete triangular interpolants, Computer Aided Geometric Design 2, 297–306.

Liu, D. (1986), A geometric condition for smoothness between adjacent Bézier surface patches, Acta Math. Appl. Sinica 9 (4) (in Chinese).

Liu, D. and Hoschek, J. (1989), GC^1 continuity conditions between adjacent rectangular and triangular Bézier surface patches, Computer Aided Design 21, 194–200.

Nielson, G.M. (1979), The side-vertex method for interpolation in triangles, J. Approx. Theory 25, 318–336.

Nielson, G.M. (1986), A transfinite, visually continuous, triangular interpolant, in: G. Farin, ed., *Geometric Modeling: Applications and New Trends*, SIAM, Philadelphia, PA.

Peters, J. (1988a), Local cubic and bicubic C^1 surface interpolation with linearly varying boundary normal, Computer Aided Geometric Design, to appear.

Peters, J. (1988b), Smooth mesh interpolation with cubic patches, Computer Aided Design, to appear; see also: Local piecewise cubic C^1 surface interpolants via splitting and averaging, University of Wisconsin, CMS Tech. Report No. 89-11.

Peters, J. (1989a), Local interpolation of a cubic curve mesh by a piecewise [bi]quartic C^1 surface without splitting, Constructive Approximation, to appear.

Peters, J. (1989b), Rectangulation algorithms: smooth surface interpolation with bicubics, CMS Technical Report 90-1, Univ. of Wisconsin–Madison.

Piper, B. (1987), Visually smooth interpolation with triangular Bézier patches, in: G. Farin, ed., *Geometric Modeling: Applications and New Trends*, SIAM, Philadelphia, PA.

Powell, M.J.D. and Sabin, M.A. (1977), Piecewise quadratic approximations on triangles, ACM Trans. Math. Software 3 (4), 316–325.

Sabin, M.A. (1968), Conditions for continuity of surface normal between adjacent parametric surfaces, Tech. Rep., British Aircraft Corporation Ltd.

Sarraga, R.F. (1986), G^1 interpolation of generally unrestricted cubic Bézier curves, Tech. Report GMR-5424(A).

Sederberg, T.W. (1985), Piecewise algebraic surface patches, Computer Aided Geometric Design 2, 53–59.

Shirman, L.A. and Séquin, C.H. (1987), Local surface interpolation with Bézier patches, Computer Aided Geometric Design 4, 279–295.

van Wijk, J.J. (1984), Bicubic patches for approximating non-rectangular control-point meshes, Computer Aided Geometric Design 3 (1), 1–13.