Annales Mathematicae et Informaticae
***36** (2009) pp. 77–84*
http://ami.ektf.hu

Surface interpolation with local control by linear blending

Imre Juhász[a], Miklós Hoffmann[b]

[a]University of Miskolc, Hungary

[b]Károly Eszterházy College, Hungary

Submitted 25 May 2009; Accepted 10 December 2009

Abstract

The purpose of this paper is to introduce an interactive surface interpolation method by spline surfaces, which is a generalization of the method presented in [2]. The technique is based on linear blending and works for a large class of surfaces including bicubic Bézier, B-spline, NURBS surfaces and the recently developed trigonometric surfaces as well. The interpolating surface can be interactively modified by control points, meanwhile the interpolation property is preserved.

Keywords: interpolation, spline surface, linear blending

MSC: 65D17, 68U07

1. Introduction

Interpolation of an ordered set of points is one of the most widely used methods in curve and surface modeling practice, hence there is a vast number of papers and book chapters dealing with this topic (cf. the books [6, 15] and references therein). Designers generally prefer splines, where most of the methods work globally. Even in the case of B-spline or NURBS surfaces, which are standard description methods in geometric design and have local control properties, the process of finding control points of an interpolating surface is global and the resulted surface cannot be locally controlled (see e.g. [1, 10, 13, 7]). To overcome this problem some methods have been developed by means of which the shape of the interpolating curve or surface can be adjusted by numerical techniques (see e.g. [8, 9, 16]). Shape parameters and other numerical techniques, however, do not provide intuitive shape control methods such as control point repositioning in approximation. Moreover, in these global methods a large system of equations has to be solved at a relatively high computational cost. Especially, for large set of data points, local methods have the advantage of solving smaller systems, since the computation of each curve segment is based on only a subset of data. Unfortunately, these local methods typically attain only C^1 continuity [11].

In the last couple of years a new local method has been developed for some types of spline curves and surfaces that requires only local computation and yields C^2 continuous spline curves. This technique - which is based on linear blending - has been implemented for NURBS in [17], for B-spline curve in [18] and for trigonometric C-B-spline curve in [14]. In this method the shape of the interpolating curve can also be adjusted numerically by some shape parameters.

In [2] the authors generalized the linear blending interpolation method for a large class of curves. The present contribution is the further generalization of the linear blending curve concept for surfaces. Since designers generally prefer geometric entities instead of numerical values, we provide intuitive, control point based modification of the interpolating surface. We let the designer alter the shape of the surface similarly to the approximating surfaces, meanwhile the interpolation property is continuously preserved.

2. Linear blending surfaces

At first we describe a surface generating method which we will call linear blending. Consider the points $\mathbf{p}_{k,l}$, $(k = 0,\dots,n; l = 0,\dots,m)$ and the piecewisely defined surface

$$\mathbf{b}(u,v) = \sum_{k=-1}^{n+1}\sum_{l=-1}^{m+1} F_k(u)G_l(v)\,\mathbf{p}_{kl}, \quad u \in [u_0,u_n], v \in [v_0,v_m], \tag{2.1}$$

where $\mathbf{p}_{kl}$ are called control points (they form the control net consisting of quadrilateral "faces"), $F_k(u)$ and $G_l(v)$ are basis functions of some space (not necessarily of the same). The number of faces must be equal to the number of patches. Thus, in case of open surfaces one has to define artificial control points, e.g. by doubling the control points on the boundary of the control net. The only restriction is that each patch of the surface has to be defined by 16 (4×4) neighboring control points, that is patches of this surface can be written as

$$\mathbf{b}_{i,j}(u,v) = \sum_{k=i-1}^{i+2}\sum_{l=j-1}^{j+2} F_k(u)G_l(v)\,\mathbf{p}_{kl}, \quad u \in [u_i,u_{i+1}], v \in [v_j,v_{j+1}]$$
$$i = 0,\dots,n-1, j = 0,\dots,m-1$$

where the values u_i, v_j are called knots.

Now, consider the (i,j)th face of the control net determined by the control points $\mathbf{p}_{i,j}, \mathbf{p}_{i,j+1}, \mathbf{p}_{i+1,j}, \mathbf{p}_{i+1,j+1}$ and interpolate this quadrilateral with the double ruled surface that we obtain from the four corner points by bilinear combination with some functions $f(u)$ and $g(v)$ in the form

$$\mathbf{h}_{i,j}(u,v) = \begin{bmatrix} (1-f(u)) & f(u) \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i,j} & \mathbf{p}_{i,j+1} \\ \mathbf{p}_{i+1,j} & \mathbf{p}_{i+1,j+1} \end{bmatrix} \begin{bmatrix} (1-g(v)) \\ g(v) \end{bmatrix},$$
$$u \in [u_i, u_{i+1}], v \in [v_j, v_{j+1}].$$

If the four points are coplanar this surface degenerates to a quadrilateral region in their plane, otherwise it is a patch of a hyperbolic paraboloid (saddle).

For the sake of interpolation, functions $f(u), g(v)$ have to fulfill conditions

$$f(u_i) = g(v_j) = 0, \quad f(u_{i+1}) = g(v_{j+1}) = 1. \tag{2.2}$$

Linearly blending the patches $\mathbf{b}_{i,j}(u,v)$ with the corresponding double ruled patches $\mathbf{h}_{i,j}(u,v)$ we obtain the linear blending surface consisting of the patches

$$\mathbf{c}_{i,j}(u,v,\alpha) = (1-\alpha)\mathbf{b}_{i,j}(u) + \alpha\mathbf{h}_{i,j}(u),$$

where α is a global shape parameter of the surface.

To achieve more flexibility in shape modification, the shape parameter α can also be the function of u and v. A natural way is to define them piecewisely by local shape parameters $\alpha^*_{i,j}$ associated to each point $\mathbf{p}_{i,j}$, and to use the same blending functions $f(u)$ and $g(v)$ like for the surface $\mathbf{h}_{i,j}(u,v)$, i.e.,

$$\alpha_{i,j}(u,v) = \begin{bmatrix} (1-f(u)) & f(u) \end{bmatrix} \begin{bmatrix} \alpha^*_{i,j} & \alpha^*_{i,j+1} \\ \alpha^*_{i+1,j} & \alpha^*_{i+1,j+1} \end{bmatrix} \begin{bmatrix} (1-g(v)) \\ g(v) \end{bmatrix}.$$

In this way each patch of the linear blending surface will have four local shape parameters (these parameters initially can be defined to be equal to 1 and will be modified in an interactive way)

$$\begin{gathered} \mathbf{c}_{i,j}(u,v,\alpha^*_{i,j},\alpha^*_{i,j+1},\alpha^*_{i+1,j},\alpha^*_{i+1,j+1}) = \\ (1-\alpha_{i,j}(u,v))\mathbf{b}_{i,j}(u,v) + \alpha_{i,j}(u,v)\mathbf{h}_{i,j}(u,v). \end{gathered} \tag{2.3}$$

In order to obtain $C^r (r > 0)$ continuity at joints of consecutive patches, functions $f(u), g(v)$ have to satisfy the conditions

$$\begin{array}{l} f^{(k)}(u_i) = f^{(k)}(u_{i+1}) = 0 \quad 1 \leqslant k \leqslant r, \quad j = 0, \ldots, n-1 \\ g^{(k)}(v_j) = g^{(k)}(v_{j+1}) = 0 \quad 1 \leqslant k \leqslant r, \quad j = 0, \ldots, m-1 \end{array}. \tag{2.4}$$

Beside these conditions, the choice of these functions highly depends on the base surface $\mathbf{b}(u,v)$, more precisely the type of its basis functions $F_k(u), G_l(v)$. The possible choice and generalizations of these functions can be found in [2].

3. Interpolation by linear blending surfaces

Now, we generalize the idea of curve interpolation by linear blending described in [17] for surfaces, and we modify the linear blending method defined above to interpolate a given grid of points. Surface patch $\mathbf{c}_{i,j}(u,v,\alpha^*_{i,j},\alpha^*_{i,j+1},\alpha^*_{i+1,j},\alpha^*_{i+1,j+1})$

is "between" the patches $\mathbf{b}_{i,j}(u,v)$ and $\mathbf{h}_{i,j}(u,v)$, that is the patch approximates the given points $\mathbf{p}_{i,j}, \mathbf{p}_{i,j+1}, \mathbf{p}_{i+1,j}, \mathbf{p}_{i+1,j+1}$. Note, that the method works in that cases as well when the double ruled surface is determined by any four points. We are going to specify such four corner points for the double ruled surface $\mathbf{h}_{i,j}(u,v)$ that the resulted blending patch $\mathbf{c}_{i,j}$ will interpolate the given points.

Let the points $\mathbf{p}_{i,j}$, associated parameter values (u_i, v_j) and shape parameters $\alpha^*_{i,j}$, $(i = 0, \ldots, n, j = 0, \ldots, m)$ be given. The problem is to find a linear blending patch that has the given shape parameters $\alpha^*_{i,j}$ and interpolates the given points $\mathbf{p}_{i,j}$ at the given parameter values (u_i, v_j).

Let us consider the approximating patch defined by the given points as control points

$$\mathbf{b}_{i,j}(u,v) = \sum_{k=i-1}^{i+2} \sum_{l=j-1}^{j+2} F_k(u) G_l(v)\, \mathbf{p}_{kl}, \quad u \in [u_i, u_{i+1}], v \in [v_j, v_{j+1}],$$

and the double ruled patch

$$\mathbf{h}_{i,j}(u,v) = \begin{bmatrix} (1-f(u)) & f(u) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{i,j} & \mathbf{v}_{i,j+1} \\ \mathbf{v}_{i+1,j} & \mathbf{v}_{i+1,j+1} \end{bmatrix} \begin{bmatrix} (1-g(v)) \\ g(v) \end{bmatrix},$$
$$u \in [u_i, u_{i+1}], v \in [v_j, v_{j+1}]$$

where the points $\mathbf{v}_{i,j}$ are unknown. Using the interpolation assumptions

$$\begin{aligned} \mathbf{c}_{i,j}(u_i, v_j, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1}) &= \mathbf{p}_{i,j} \\ \mathbf{c}_{i,j}(u_i, v_{j+1}, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1}) &= \mathbf{p}_{i,j+1} \\ \mathbf{c}_{i,j}(u_{i+1}, v_j, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1}) &= \mathbf{p}_{i+1,j} \\ \mathbf{c}_{i,j}(u_{i+1}, v_{j+1}, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1}) &= \mathbf{p}_{i+1,j+1} \end{aligned} \tag{3.1}$$

we obtain

$$\begin{aligned} \mathbf{v}_{i,j} &= \mathbf{p}_{i,j} + \frac{1-\alpha^*_{i,j}}{\alpha^*_{i,j}} (\mathbf{p}_{i,j} - \mathbf{b}_{i,j}(u_i, v_j)) \\ \mathbf{v}_{i,j+1} &= \mathbf{p}_{i,j+1} + \frac{1-\alpha^*_{i,j+1}}{\alpha^*_{i,j+1}} (\mathbf{p}_{i,j+1} - \mathbf{b}_{i,j}(u_i, v_{j+1})) \\ \mathbf{v}_{i+1,j} &= \mathbf{p}_{i+1,j} + \frac{1-\alpha^*_{i+1,j}}{\alpha^*_{i+1,j}} (\mathbf{p}_{i+1,j} - \mathbf{b}_{i,j}(u_{i+1}, v_j)) \\ \mathbf{v}_{i+1,j+1} &= \mathbf{p}_{i+1,j+1} + \frac{1-\alpha^*_{i+1,j+1}}{\alpha^*_{i+1,j+1}} (\mathbf{p}_{i+1,j+1} - \mathbf{b}_{i,j}(u_{i+1}, v_{j+1})). \end{aligned} \tag{3.2}$$

By means of these points, the corresponding linear blending patches will interpolate the given points at the given parameter values (see Fig.1).

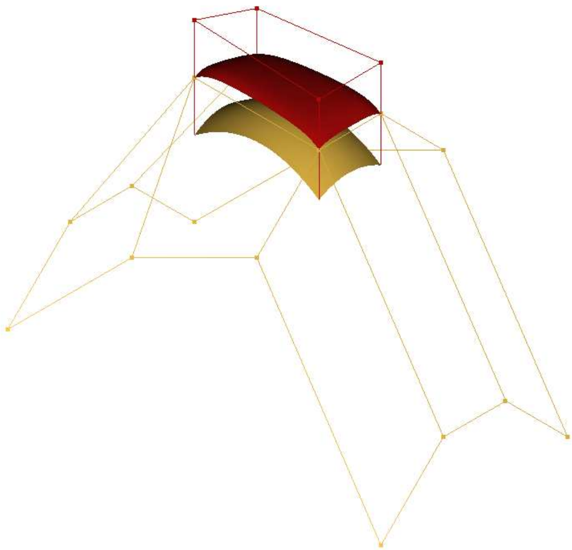

Figure 1: The original base surface (below) and the interpolating linear blending surface with the calculated four control points.

4. Interactive shape modification

Points $\mathbf{v}_{i,j}$ depend on three parameters: the corresponding parameter values u_i, v_j and the local shape parameter $\alpha^*_{i,j}$. Instead of manipulating these values numerically, calculating the points $\mathbf{v}_{i,j}$ and finally the interpolating surface patch $\mathbf{c}_{i,j}(u, v, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1})$, we intend to develop an interactive shape modification tool. In this tool points $\mathbf{v}_{i,j}$ shall be used analogously to the control points of an approximating surface, meanwhile the interpolating property of the surface is preserved. Although, these points are not "real" control points of the surface, the geometric effect of dragging these points is quite similar to the effect of control point repositioning.

When the position of the point $\mathbf{v}_{i,j}$ is modified, we have to recalculate the actual values of parameters u_i, v_j and $\alpha^*_{i,j}$ to preserve the interpolation. This problem leads us to the following questions: what happens to the surface (and especially to the point $\mathbf{v}_{i,j}$) if one of these parameters is changed? What are the possible positions of the point $\mathbf{v}_{i,j}$?

At first let us fix the parameters u_i, v_j and alter the shape parameter $\alpha^*_{i,j}$. It is obvious from Eq. (3.2) that preserving the interpolation property the point $\mathbf{v}_{i,j}$ will move along a straight line connecting the given point $\mathbf{p}_{i,j}$ and the point $\mathbf{b}_{i,j}(u_i, v_j)$ of the original surface patch.

Now, consider the case when the shape parameter $\alpha^*_{i,j}$ is fixed and the parameters u_i, v_j are altered. By Eqs. (3.1) the surface interpolates the point $\mathbf{p}_{i,j}$

at parameters (u_i, v_j), which may vary between u_{i-1}, u_{i+1} and v_{j-1}, v_{j+1}, respectively. These values, however also serve as knot values of the original base surface (2.1). Therefore, the alteration of these parameters changes the shape of the original surface patch $\mathbf{b}_{i,j}(u, v)$ as well. The geometric description of the effect of knot alteration is far from being trivial. For B-spline and NURBS surfaces it has been described in detail in [3], [4], [12] and [5]. Using the results of these studies we can conclude that the point $\mathbf{b}_{i,j}(u_i, v_j)$ of the base surface will move along a well-defined surface patch

$$\mathbf{e}(u_i, v_j) = \mathbf{b}_{i,j}(u_i, v_j) \quad u_i \in [u_{i-1}, u_{i+1}], v_j \in [v_{j-1}, v_{j+1}]. \tag{4.1}$$

E.g., in case of a B-spline surface of degree (k, l), the surface $\mathbf{e}(u_i, v_j)$ is a B-spline surface patch of degree $(k-1, l-1)$, defined by the same control points and knot values (except the knots u_i and v_j) as the original surface [3].

By means of Eqs. (3.2) it is easy to see that altering the parameters u_i, v_j the point $\mathbf{v}_{i,j}$ will move along a surface that can be obtained by a central similarity from surface (4.1), where the center of similitude is the given point $\mathbf{p}_{i,j}$ and the ratio is $\left(1 - \alpha^*_{i,j}\right) / \alpha^*_{i,j}$.

Summarizing the above results one can see that the permissible positions of $\mathbf{v}_{i,j}$ is a volume bounded by a cone-like surface the apex of which is the given point $\mathbf{p}_{i,j}$ and its base is composed of the four boundary curves of the envelope surface (4.1) (see Fig.2).

Figure 2: The original surface (below) and the interpolating linear blending surface. The permissible positions of the upmost control point is shown by a volume bounded by a cone-like surface.

For each actual position of $\mathbf{v}_{i,j}$ within this region one has to recalculate the parameters u_i, v_j and $\alpha^*_{i,j}$, and (by fixing the rest of the shape parameters) substitute them into $\mathbf{c}_{i,j}(u, v, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1})$ in order to obtain the interpolating surface.

5. Conclusions

An easy-to-compute interpolation method is presented in this paper, based on linear blending of a base surface and a computed control mesh. The resulted surface can interactively be modified by the points of this control mesh, meanwhile the interpolation property continuously holds. The method works for a large class of surfaces, including all the standard surface types (Bézier, B-spline, NURBS, C-B-spline, etc.) of computer aided geometric design.

Acknowledgements. The authors wish to thank the National Office of Research and Technology (Project CHN-37/2005) for their financial support of this research. The second author is also supported by János Bolyai Scholarship of Hungarian Academy of Sciences.

References

[1] Barsky, B.A., Greenberg, D.P., Determining a set of B-spline control vertices to generate an interpolating surface, *Computer Graphics and Image Processing*, 14 (1980), 203–226.

[2] Hoffmann, M., Juhász, I., On interpolation by spline curves with shape parameters, *Lecture Notes in Computer Science*, 4975 (2008), 205–214.

[3] Hoffmann M., Juhász I., Geometric aspects of knot modification of B-spline surfaces, *Journal for Geometry and Graphics*, 6 (2002), 141–149.

[4] Hoffmann, M., Juhász, I., On the family of B-spline surfaces obtained by knot modification, *Mathematical Communications*, 11 (2006), 9–16.

[5] Hoffmann, M., Juhász, I., Constrained shape control of bicubic B-spline surfaces by knots, in: Sarfraz, M., Banissi, E. (Eds.) Geometric Modeling and Imaging, London, IEEE CS Press, 41–47, 2006

[6] Hoschek, J., Lasser, D., Fundamentals of CAGD, AK Peters, Wellesley, MA, 1993.

[7] Juhász, I., Hoffmann, M., On parametrization of interpolating curves, *Journal of Computational and Applied Mathematics*, 216 (2008), 413–424.

[8] Kaklis, P.D., Sapidis, N.S., Convexity-preserving interpolatory parametric splines of non-uniform polynomial degree, *Computer Aided Geometric Design*, 12 (1995), 1–26.

[9] Kong, V.P., Ong, B.H., Shape preserving F3 curve interpolation, *Computer Aided Geometric Design*, 19 (2002), 239–256.

[10] Lavery, J.E., Univariate cubic Lp splines and shape-preserving multiscale interpolation by univariate cubic L1 splines, *Computer Aided Geometric Design*, 17 (2000), 319–336.

[11] Li, A., Convexity preserving interpolation, *Computer Aided Geometric Design*, 16 (1999), 127–147.

[12] Li, Y.-J., Wang, G.-Z., On knot modification of B-spline and NURBS surfaces. *J. of Computer-Aided Design and Computer Graphics*, 15 (2005), 986–989.

[13] Ma, W., Kruth, J.P., NURBS curve and surface fitting and interpolation, in: M. Daehlen, T. Lyche, L. Schumaker (Eds.), Mathematical Methods for Curves and Surfaces, Vanderbilt University Press, Nashville & London, 1995.

[14] Pan, Y.-J., Wang, G.-J., Convexity-preserving interpolation of trigonometric polynomial curves with shape parameter, *Journal of Zhejiang University Science*, 8 (2007), 1199–1209.

[15] Piegl, L., Tiller, W., The NURBS Book, Springer, Berlin. 1995

[16] Sapidis, N., Farin, G., Automatic fairing algorithm for B-spline curves, *Computer-Aided Design*, 22 (1990), 121–129.

[17] Tai, C.-L., Barsky, B.A., Loe, K.-F., An interpolation method with weights and relaxation parameters. In: Cohen, A., Rabut, C., Schumaker, L.L. (eds.): Curve and Surface Fitting: Saint-Malo 1999. Vanderbilt Univ. Press, Nashville, Tenessee (2000) 393–402.

[18] Tai, C.-L., Wang, G.-J., Interpolation with slackness and continuity control and convexity-preservation using singular blending, *J. Comp. Appl. Math.*, 172 (2004), 337–361.

Imre Juhász
Department of Descriptive Geometry, University of Miskolc,
H-3515 Miskolc, Hungary
e-mail: `agtji@uni-miskolc.hu`

Miklós Hoffmann
Institute of Mathematics and Computer Science,
Károly Eszterházy College, Eger, Hungary
e-mail: `hofi@ektf.hu`