Spline-Blended Surfaces

Mária Kuklišová

7 April 1999

In many applications such as computer vision, medical imaging, and image generation, surface data is available in the form of set of points, planar or 3-dimensional contours and it is desired to reconstruct a surface from this data. In this paper we will assume that we are given two sets of curves, one set running fore and aft, the other transversely. This network of lines defines a number of topologically rectangular patches. We will construct a spline - blended surface interpolating the network of curves, known also as Gordon surface. We will construct new blending functions used by construction of Gordon surface. We will also use this blending functions as basal functions for construction of spline curves and tensor product surfaces. At the end we will generalize interpolating splines to the system of approximating splines containing interpolating splines and B-splines.

1 Gordon surface

We are given a network of curves as mentioned above. It is desired to construct a surface $G(u,v)$ interpolating the given curves. Let us denote given u-curves $G(u,v_j), j = 1,\dots,n$ and v-curves $G(u_i,v), i = 1,\dots,m$. Curves intersect in points $G(u_i,v_j), i = 1,\dots,m, j = 1,\dots,n$. Gordon surface is defined:

$$G(u,v) = G_1(u,v) + G_2(u,v) - G_{12}(u,v)$$

where

$$\begin{aligned} G_1(u,v) &= \sum_{i=1}^{m} G(u_i,v)L_i^m(u) \\ G_2(u,v) &= \sum_{j=1}^{n} G(u,v_j)L_j^n(v) \\ G_{12}(u,v) &= \sum_{i=1}^{m}\sum_{j=1}^{n} G(u_i,v_j)L_i^m(u)L_j^n(v) \end{aligned}$$

and $L_i^m(u)$ are blending functions satisfying:

$$L_i^m(u_i) = 1,\ L_i^m(u_k) = 0, i \neq k \tag{1}$$

We will use B_2-splines to construct appropriate blending functions.

2 B_2-splines

B_2-splajn is similar to B-spline. Instead of $2m+1$ B-spline control vertexes $D_0, \ldots, D_{2m}$, B_2-spline is determined by odd control vertexes $D_0, D_2, \ldots, D_{2m}$ and even control points are replaced by points $P_i,\ i = 1, \ldots, m$, joining segments Q_{2i-1} and Q_{2i}. We will consider special case of B_2 -splines, when knot set is equidistant, in other words knot $u_i = i$. In this case point P_i holds an equation:

$$P_i = \frac{1}{6}(D_{2i-2} + 4D_{2i-1} + D_{2i}) \tag{2}$$

To construct blending functions we need B_2-spline functions. It is easy to derive vertices of B_2-spline function: $D_{2i} = (i, d_{2i}),\ P_i = (i, p_i)$. We do not need to write first coordinate, so we will consider each control point to be identical to its second coordinate. Next equations show, how we can count points of segments Q_{2i-1} and Q_{2i}:

$$Q_{2i-1}(t) = (t^3, t^2, t, 1)\frac{1}{24}\begin{pmatrix} -7 & 18 & -16 & 6 & -1 \\ 18 & -36 & 18 & 0 & 0 \\ -12 & 0 & 12 & 0 & 0 \\ 0 & 24 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} D_{2i-2} \\ P_i \\ D_{2i} \\ P_{i+1} \\ D_{2i+2} \end{pmatrix} \tag{3}$$

$$Q_{2i}(t) = (t^3, t^2, t, 1)\frac{1}{24}\begin{pmatrix} 1 & -6 & 16 & -18 & 7 \\ -3 & 18 & -30 & 18 & -3 \\ 3 & -18 & 0 & 18 & -3 \\ -1 & 6 & 14 & 6 & -1 \end{pmatrix}\begin{pmatrix} D_{2i-2} \\ P_i \\ D_{2i} \\ P_{i+1} \\ D_{2i+2} \end{pmatrix} \tag{4}$$

where $t = u - (2i - 1)$ resp. $t = u - 2i$.

3 Construction of blending functions

In this section we will construct blending functions $L_i^m(u)$ as B_2-splajn. Let $P_1^i, \ldots, P_m^i$ be joining points and let $D_0^i, D_2^i, \ldots, D_{2m}^i$ be odd control vertexes of blending function $L_i^m(u)$. We have to change equation (1), because knot set is $-1, 0, 1, \ldots, 2m + 1$ and one set of curves contains just m curves:

$$L_i^m(2i - 1) = 1,\ L_i^m(2k - 1) = 0, i \neq k,\ i = 1, \ldots, m$$

This is equivalent to conditions:

$$P_i^i = 1,\ P_k^i = 0,\ i \neq k,\ i = 1, \ldots, m \tag{5}$$

Conditions (1) determine joining points $P_1^i, \ldots, P_m^i$. Odd control vertexes do not need to satisfy any conditions. We have to find appropriate odd control vertexes.

Firstly we want the points of surface $G_1(u,v)$ to be a barycentric combination of points of curves $G(u_i,v),\ i=1,\dots,m$. This is satisfied, if the next equation is satisfied:

$$\sum_{i=1}^{m} L_i^m(u) = 1\ \forall u \tag{6}$$

We can thing about surfaces $G_2(u,v)$ and $G_{12}(u,v)$ in the same way. We can easy get this conditions from equations (3), (4) and (6):

$$\sum_{i=1}^{m} P_k^i = 1 \quad \sum_{i=1}^{m} D_{2k}^i = 1 \tag{7}$$

Secondly, we want the given curves to influent the surface just locally. Therefore we want blending function to be non-zero just over few segments around point P_i^i and knot $2i-1$. The optimal number of non-zero segments appears to be 12. We can see from equations (3), (4), that segments Q_{2j-1}^i and Q_{2j}^i are controled by vertexes $D_{2j-2}^i, P_j^i, D_{2j}^i, P_{j+1}^i, D_{2j+2}^i$. If any of this two segments is identically equal to zero and $P_j^i = P_{j+1}^i = 0$, then also $D_{2j-2}^i = D_{2j}^i = D_{2j+2}^i = 0$. If we have just 12 non-zero segments, then just four odd control vertexes $D_{2i-4}^i, D_{2i-2}^i, D_{2i}^i, D_{2i+2}^i$ are non-zero. Obviously the only non-zero joining vertex is $P_i^i = 1$.

Thirdly we want each given curve to have the same influence to Gordon surface. This will be satisfied, if $D_{2i-4}^i = a, D_{2i-2}^i = b, D_{2i}^i = c, D_{2i+2}^i = d, \forall i$, where a,b,c,d are real numbers.

Finally we want each blending function to be symmetrical. That means $a=d$ and $b=c$. We did not satisfy conditions (7) yet. First one is obviously satisfied. Now we can replace second one by new simplier one: $2a+2b=1$. Let p be a real parameter. We will set $a=-p$ and $b=p+\frac{1}{2}$. This is the result of construction. Now we can assume all verexes of blending functions:

D_0	P_1	$\dots$	D_{2i-4}	P_{i-1}	D_{2i-2}	P_i	D_{2i}	P_{i+1}	D_{2i+2}	$\dots$	P_m	D_{2m}
0	0	$\dots$	$-p$	0	$\frac{1}{2}+p$	1	$\frac{1}{2}+p$	0	$-p$	$\dots$	0	0

Now we need to note, that condition (6) for odd vertexes is not satisfied for $k \in \{0,1,m-1,m\}$. We need to increase the number of blending functions. We need also L_{-1}^m, L_0^m, L_{m+1}^m and L_{m+2}^m. In definition of Gordon surface we can use for example $L_{-1}^m + L_0^m + L_1^m$ instead of L_1^m and $L_m^m + L_{m+1}^m + L_{m+2}^m$ instead of L_m^m.

4 Using blending functions to construct a spline curve

In this section we will construct spline curve using blending functions. We are given $m+3$ control points $W_{-1},\dots,W_{m+2}$. Spline curve $L(u)$ is defined:

$$L(u) = \sum_{i=0}^{m+1} W_i L_i^m(u), u \in [1, 2m-1] \tag{8}$$

Curve $L(u)$ interpolates control points $W_1, \dots, W_m$ because of conditions (5). Now we can count the B_2-spline control points of the curve $L(u)$. Let us denote joining points of the curve $R_1, \dots, R_m$ and odd control vertexes $C_0, C_2, \dots, C_{2m}$. It is obvious that joining points are identical with control points interpolated by the curve: $R_i = W_i,\ i = 1, \dots, m$. It is not difficult to count the odd control points: $C_{2i} = -pW_{i-1} + (p + \frac{1}{2})W_i + (p + \frac{1}{2})W_{i+1} - pW_{i+2},\ i = 0, \dots, m$.

In simillar way we can get also approximating spline curve. We just need to replace condition $R_i = W_i$ by new one

$$R_i = aW_{i-1} + (1 - 2a)W_i + aW_{i+1}$$

where a is real parameter. This will cause new values of joining points of blending functions: $P^i_{i-1} = a, P^i_i = 1 - 2a, P^i_{i+1} = a$. For $a = 0$ we have got the original blending functions. For $a = \frac{1}{6}$ and $p = 0$ we will get classical B-spline curve. Implementation of spline curve is a good way to find out the best value of parameter p. For interpolating curve it seems to be $p = \frac{1}{6}$ and for $a = \frac{1}{6}$ we will simply choose B-spline value $p = 0$. For other values of parameter a we can use linear interpolation of this two values of p. We will get: $p = \frac{1}{6} - a$.

5 Computing points of curves and surfaces

We have no problems to compute points of curves constructed in previous section. We have counted B_2-spline vertexes of the curve. From equation (2) we can count B-spline control vertexes of the curve and then well-known DeBoor algorithm can be used.

Using blending functions we can also construct a tensor-product surface (approximating or interpolating). Surface G_{12} from definition of Gordon surface is such tensor product. We can write:

$$G_{12}(u, v) = \sum_{j=1}^{n} [\sum_{i=1}^{m} G(u_i, v_j) L_i^m(u)] L_j^n(v).$$

In brackets we have got n spline curves. We can compute point of each of them in parameter u. Then we get n control points of other curve. If we compute point of this curve in parameter v, we get a point of the tensor-product surface in parameters u and v. So we can compute point of tensor-product surface using DeBoor algorithm $n + 1$ (or $m + 1$) times.

Now it is easy to compute a point of Gordon surface. We have already got algorithm to compute $G_{12}(u, v)$. To compute point $G_1(u, v)$ we just use DeBoor Algorithm, because parameter v is fixed and $G(u_i, v)$ are control points of a curve. We compute point $G_2(u, v)$ the same way. Finally we just count $G(u, v) = G_1(u, v) + G_2(u, v) - G_{12}(u, v)$.

6 Conclusion

We have constructed appropriate B_2-spline blending functions for construction of spline-blended (Gordon) surface. We have also constructed system of spline curves containing appropriate interpolation curve and B-spline curve. The interpolation curve is just special case of B_2- spline curve, but we have found appropriate odd controll points. User does not have to take care of them, but they can also be changed using parameter p.

The results of this paper were implemented and we represent them in several pictures below.

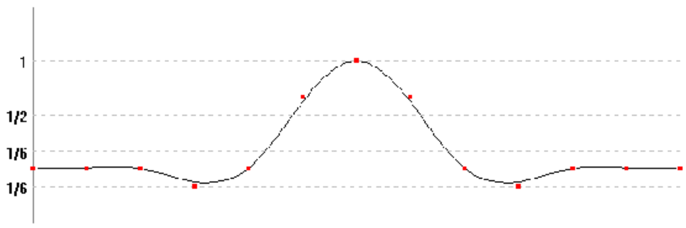


Figure 1: Blending function with parameters a=0 and p=1/6

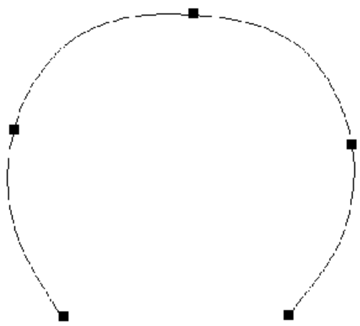


Figure 2: Interpolating curve

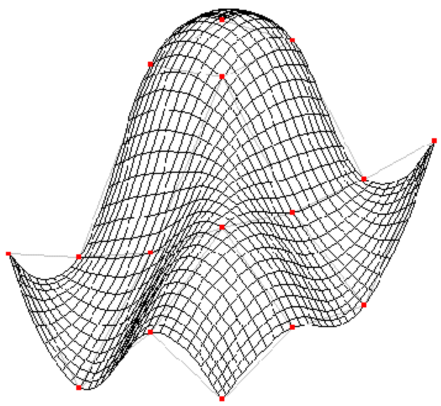


Figure 3: Interpolating tenzor-product surface

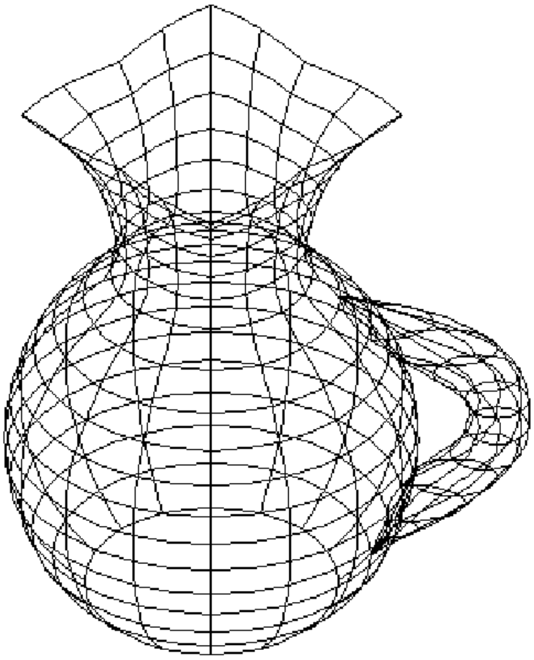
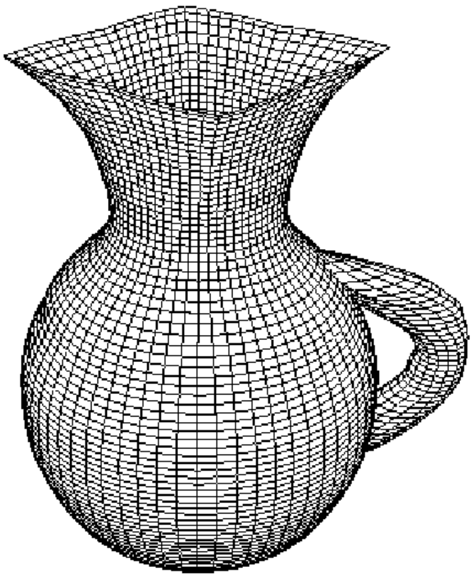


Figure 4: Network of given curves



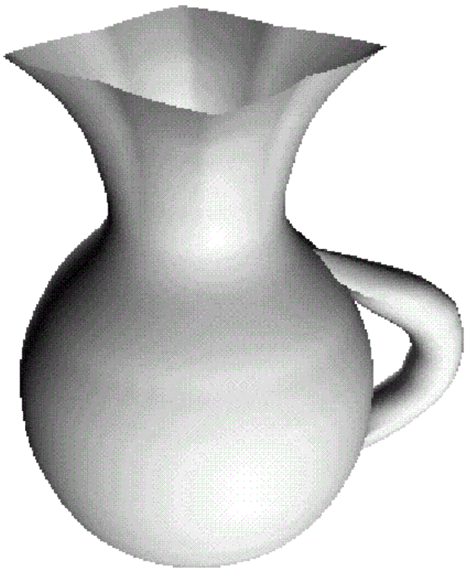


Figure 5: Wireframe model of Gordon surface

Figure 6: Shaded model of Gordon Surface