Spline Overlay Surfaces

Richard H. Bartels

Computer Graphics Laboratory
Computer Science Department
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada

David R. Forsey

Computer Science Department
University of British Columbia
Vancouver, BC, V6T 1Z2, Canada

Abstract

We consider the construction of spline features on spline surfaces. [1] The approach taken is a generalization of the hierarchical surface introduced in [Forsey88]. Features are regarded as spline-defined vector displacement fields that are overlain on existing surfaces. No assumption is made that the overlays are derived from the base surface. They may be applied with any orientation in a non-hierarchical fashion.

In particular, we present a "cheap" version of the concept in which the displacement field is mapped to the base surface approximately, through the mapping of its control vectors alone. The result is a feature that occupies the appropriate position in space with respect to the base surface. It may be manipulated and rendered as an independent spline, thus avoiding the costs of a true displacement mapping. This approach is useful for prototyping and previewing during design. When a finished product is desired, of course, true displacement mapping is employed.

1 Introduction

The more tools one has with which to build free-form spline surfaces, the more extensive are the possibilities of imitating naturally-ocurring or industrial surfaces with spline models. Single, tensor-product surfaces are limited in what they can achieve. For this reason, publications appear regularly on *trimming*, *box-splines*, *multivariate splines*, *multi-sided patches*, *surface blending and fillets*, *hierarchical surfaces*, and *warped surfaces* [Casale87, Coquillart90, Dahmen91, Farin90, Forsey88, Faux79, Höllig89, Loop90, Sederberg86]. Here we investigate composite surfaces that are constructed by layering *displacement splines* onto a *root spline* in a manner that generalizes the hierarchical surfaces of [Forsey88].

Displacement mappings are standard tools in computer graphics [Foley90], and applying them in a stagewise fashion to layer features onto a surface substrate is an idea that requires little innovation to carry into implementation. Nevertheless, surface modelers are not usually built upon

[1]This work was supported by the Canadian Government under a Strategic NSERC grant, by the Province of Ontario under an ITRC grant, and by General Motors and Digital Equipment.

this principle because of the expense involved. Such surfaces are procedurally defined, and every manipulation must be made implicitly, modifying the displacement and tracking the effect through the construction process for every point to be rendered and evaluated. The hierarchical surfaces of [Forsey88] avoided this expense by mapping only the control vertices of each level of displacement onto their corresponding base surfaces. The resulting composite surfaces were limited in their forms by the requirement that each displacement region be derived from the root surface by a nested sequence of refinements. In this paper we investigate the possibility of approximating a true displacement construction by mapping only control vertices to form approximate displacements. There is to be no restriction on the regions, or on the orientations of the displacements. One uses these approximate displacements during an interactive process to design and preview the true displacement result.

In Section 2 we review true displacement maps to set the terminology and context of our proposals. In Section 3 we touch upon the local coordinate frames that are suitable for applying displacements on surfaces. The approximate displacements of our proposal are based upon the concept of representing the control vertices of displacement splines in scattered frames of reference derived from the displacement's domain. We discuss scattered reference frames in Section 4. Section 5 provides the material for mapping control vertices to approximate a true displacement map, and Section 6 provides a few words on the accuracy of this process. Section 7 closes with a mention of multiple overlays and a final example. We assume that the reader is familiar with the concepts and notations of splines; e.g. as presented in [Bartels87, Farin90].

2 Surface Displacement Maps

Let $\mathrm{S}(u,v)$ be a *surface*; that is, a collection of *points* in 3-D affine space [Foley90, Goldman85] generated by two *scalar variables*, u and v, that belong to a 2-D affine domain $\mathcal{D}_{\mathrm{S}}$. Let $\mathbf{d}(r,s)$ be a *displacement*; that is, a collection of *vectors* in the same 3-D affine space generated by two variables, r and s, belonging to another 2-D affine domain $\mathcal{D}_{\mathbf{d}}$. In order to keep the elements of the discussion distinct, points will be given in upper case Roman characters, scalars in lower case Italic or Greek, and vectors in lower case bold. Transformations will be given by upper case bold characters. Affine transformations are an important class, but we will not restrict ourselves exclusively to them. We will, however, assume that our transformations are invertible.

The process of applying the displacement $\mathbf{d}$ to the surface S begins with the mapping of the domain $\mathcal{D}_{\mathbf{d}}$ into the domain $\mathcal{D}_{\mathrm{S}}$ by a transformation $\mathbf{T}$. Then, for each, point (u,v) in the surface domain contained within $\mathbf{T}(\mathcal{D}_{\mathbf{d}})$, a pre-image point, $(r(u,v), s(u,v)) = \mathbf{T}^{-1}(u,v)$ must be found. The displaced version of the point $\mathrm{S}(u,v)$ is given by *some interpretation of* the affine, point-vector sum $\mathrm{S}(u,v) + \mathbf{d}(r(u,v), s(u,v))$. In the terminology of [Forsey88], S is the *base surface*, $\mathbf{d}$ is an *overlay*, and $\mathrm{S} + \mathbf{d}$ constitutes a new base surface to which further overlays may be applied. If this process is carried out repeatedly, the various domain images $\mathcal{D}_{\mathbf{d}}$ will form a layered composite of regions. Calculating the inverse transformation $\mathbf{T}^{-1}$ in such a situation takes on aspects of ray tracing, in which each point (u,v) in $\mathcal{D}_{\mathrm{S}}$ must be tested for inclusion in each overlay domain's image $\mathcal{D}_{\mathbf{d}}$. The collection of inclusion domains must be reported in the order in which the overlays were applied. Needless to say, this hampers the interactive use of displacements as a modeling tool. The restriction of strict domain nesting and alignment, provided by subdivision, made a restricted form of displacements interactively supportable in [Forsey88].

The reason for the qualifying phrase "some interpretation of" in the above description arises from the requirement that the overlay be sensitive to the local topography of the base. This requirement is well known in texture mapping, showing up most painfully when a flat pattern must be applied in reasonable way to a highly distorted surface. For this reason, representing $\mathbf{d}$ in the world coordinate frame $\{\mathrm{O}, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$

$$\mathbf{d} = \alpha(r,s)\mathbf{i} + \beta(r,s)\mathbf{j} + \gamma(r,s)\mathbf{k}$$

and applying straightforward, componentwise addition

$$\begin{aligned}
\mathrm{S}(u,v) + \mathbf{d}(r(u,v), s(u,v)) &= [\mathrm{O} + \kappa(u,v)\mathbf{i} + \lambda(u,v)\mathbf{j} + \mu(u,v)\mathbf{k}] \\
&\quad + [\alpha(r(u,v), s(u,v))\mathbf{i} + \beta(r(u,v), s(u,v))\mathbf{j} + \gamma(r(u,v), s(u,v))\mathbf{k}] \\
&= \mathrm{O} + [\kappa(u,v) + \alpha(r(u,v), s(u,v))]\mathbf{i} \\
&\quad + [\lambda(u,v) + \beta(r(u,v), s(u,v))]\mathbf{j} \\
&\quad + [\mu(u,v) + \gamma(r(u,v), s(u,v))]\mathbf{k}
\end{aligned}$$

is insufficient to achieve reasonable displacements.

A more satisfactory displacement is achieved when each displacement vector $\mathbf{d}(r,s)$ is formulated in a local coordinate frame $\{\mathrm{O}(r,s), \mathbf{i}(r,s), \mathbf{j}(r,s), \mathbf{k}(r,s)\}$

$$\mathbf{d} = \alpha(r,s)\mathbf{i}(r,s) + \beta(r,s)\mathbf{j}(r,s) + \gamma(r,s)\mathbf{k}(r,s)$$

The coordinatewise addition of $\mathbf{d}(\mathbf{T}^{-1}) = \mathbf{d}(r(u,v), s(u,v))$ to $\mathrm{S}(u,v)$ is interpreted in a compatible local coordinate frame on S, which is tied to the surface point $\mathrm{S}(u,v)$ as origin rather than to the world origin O, and which is tied to surface coordinate vectors $\{\mathbf{f}(u,v), \mathbf{g}(u,v), h(u,v)\}$ rather than global coordinate vectors. Thus, $\mathrm{S} + \mathbf{d}$ is interpreted as

$$\begin{aligned}
\mathrm{S}(u,v) + \mathbf{d}(r(u,v), s(u,v)) \quad \rightarrow \quad & \mathrm{S}(u,v) \\
& + \alpha(r(u,v), s(u,v))\mathbf{f}(u,v) \\
& + \beta(r(u,v), s(u,v))\mathbf{g}(u,v) \\
& + \gamma(r(u,v), s(u,v))\mathbf{h}(u,v)
\end{aligned}$$

This interpretation is presented in Figure 1.

Choosing local coordinates $\mathbf{i}, \mathbf{j}, \mathbf{k}$ and compatible coordinates $\mathbf{f}, \mathbf{g}, \mathbf{h}$ is open to many research possibilities. In Section 3 we present the choice we have been using.

3 Surface Frames of Reference

We compose local coordinates for $\mathbf{d}$ using the unit vectors in the r and s directions. These serve as the $\mathbf{i}$ and $\mathbf{j}$ coordinate vectors. We regard these vectors, and the r, s domain, as being an affine plane imbedded in the world coordinate frame. The third unit vector needed for a local coordinate frame is provided by the world-space cross product of $\mathbf{i}$ and $\mathbf{j}$, regarded in whichever order makes sense in the context of the model being constructed. One choice will map $\mathbf{d}(r,s)$ onto displacements for one side of S, and the other choice will map $\mathbf{d}(r,s)$ onto displacements for the reverse side of S. One choice is useful in providing displacements that raise elevations on the base surface; the other choice is useful in digging depressions into the base.

At a corresponding point $\mathrm{S}(u(r,s), v(r,s)) = \mathrm{S}(\mathbf{T}(r,s))$ on the surface, a local coordinate frame is constructed using the tangent plane and the surface normal. The unit direction vectors $\mathbf{f}$ and $\mathbf{g}$ in the tangent plane should convey the directions of the vectors $\mathbf{i}$ and $\mathbf{j}$ as mapped by the

Figure 1: Domain mapping and compatible local coordinate frames for displacements

transformation $\mathbf{T}$ and reflected onto the surface S at the image point (u, v). The closest differential geometric entities that meet this description are the partial derivatives vectors

$$\begin{aligned}\frac{\partial \mathrm{S}}{\partial r} &= \frac{\partial \mathrm{S}}{\partial u}\frac{\partial u}{\partial r} + \frac{\partial \mathrm{S}}{\partial v}\frac{\partial v}{\partial r} \\ \frac{\partial \mathrm{S}}{\partial s} &= \frac{\partial \mathrm{S}}{\partial u}\frac{\partial u}{\partial s} + \frac{\partial \mathrm{S}}{\partial v}\frac{\partial v}{\partial s}\end{aligned}$$

We have found it sufficient to approximate these vectors with difference quotients; e.g.,

$$\frac{\partial \mathrm{S}}{\partial \mathrm{r}} \approx \frac{\mathrm{S}(u(r+\delta, s), v(r+\delta, s)) - \mathrm{S}(u(r, s), v(r, s))}{\delta}$$

for a small value of δ depending on the magnitude of the components of S and the round-off precision of the computer. These approximate partial derivatives, normalized, provide $\mathbf{f}$ and $\mathbf{g}$ at (u, v). $\mathbf{h}$ is their cross product, normalized to provide the surface normal, whose sense is chosen to be meaningful in the context of the model being constructed.

4 Domain Points and a Scattered Control Vertex Representation

True displacement mapping would require that this construction of coordinate frames be undertaken at every point of interest for the surface. An approximation that involves mapping only the control vectors $\mathbf{d}_{k,\ell}$ of $\mathbf{d}$, however, together with a corresponding number of domain points (r, s), has proven useful in approximating the true displacement. This approximation can be used interactively for model design and rendering. Our experience in this regard has been limited to tensor products

$$\begin{aligned}\mathrm{S}(u, v) &= \sum_i \sum_j \mathrm{S}_{i,j} b_i(u) b_j(v) \\ \mathbf{d}(r, s) &= \sum_k \sum_\ell \mathbf{d}_{k,\ell} c_k(r) c_\ell(s)\end{aligned}$$

using nonuniform B-splines as basis functions, so we will present only this case here.

We associate each control vector $\mathbf{d}_{k,\ell}$ with the domain point $\mathrm{O}_{k,\ell} = (r_k, s_\ell)$ over which it has maximal influence; that is to say, the point at which its corresponding basis function $c_{k,\ell}(r, s) =$

$c_k(r)c_\ell(s)$ takes on a maximum. A suitable approximation to this, for practical purposes, is to take the midpoint of the basis support (or for odd-degree B-splines, the middle knot of the support). That is, for degree d, $r_k \leftarrow \frac{\overline{r}_k+\overline{r}_{k+d+1}}{2}$ (or $r_k \leftarrow \overline{r}_{k+\frac{d+1}{2}}$ for odd degrees), where c_k is positive over the range of knots $\overline{r}_k, \ldots, \overline{r}_{k+d+1}$, and similarly for s_ℓ.

The displacement spline $\mathbf{d}(r,s)$ must be a *vector* spline. As such, it is not sensitive to placement of the origins O or $\mathrm{O}_{k,\ell}$ However, designers have grown used to working with *spline surfaces* rather than *spline vector fields*, and consequently the design of an overlay feature is best done in the form

$$\begin{aligned} D(r,s) &= \sum_k \sum_\ell \mathrm{D}_{k,\ell} c_k(r) c_\ell(s) \\ &= \sum_k \sum_\ell \left[\mathrm{O}_{k,\ell} + \mathbf{d}_{k,\ell}\right] c_k(r) c_\ell(s) \end{aligned}$$

using the *control vertices* $\mathrm{D}_{k,\ell}$ instead of the *control vectors* $\mathbf{d}_{k,\ell}$.

The association of control vertex to domain point is used to provide an $\mathbf{i}, \mathbf{j}, \mathbf{k}$ coordinate frame, as has been described, and the coordinates α, β, γ of the control vertex in this local frame are found. Plate 1 shows an example, where the control vertices appear as cyan cubes, and the local frames appear as magenta gnomens. The control vectors can be visualized as the difference between each cyan cube, $\mathrm{D}_{k,\ell}$, and the origin, $\mathrm{O}_{k,\ell}$, of its corresponding gnomen.

As we have mentioned, the domain $\mathcal{D}_{\mathbf{d}}$ must be imbedded in the 3-D space of $\mathbf{d}(r,s)$ and $\mathrm{D}(r,s)$. This becomes particularly easy to arrange, since, as may be seen from Plate 1, control vertices around the margin of the displacement spline are not separated from the domain. In order for a displacement to make sense as a modification of the base surface that fits seamlessly on that base, the displacement spline must produce zero vectors around its margin. The design of such a spline would begin with control vertices residing on the "zero plane", and only interior control vertices would be moved. Thus, a displacement spline begins life, conceptially, as a displacement applied to its own domain plane.

5 Domain and Control Vertex Mapping

For any mapping $\mathbf{T}$ of the domain $\mathcal{D}_{\mathbf{d}}$ into $\mathcal{D}_{\mathrm{S}}$, we apply $\mathbf{T}$ only to the domain points $\mathrm{O}_{k,\ell}$ associated with the control vertices $\mathrm{D}_{k,\ell}$. The image points, $\mathbf{T}(\mathrm{O}_{k,\ell}) = \mathbf{T}(r_k, s_\ell) = (u_{k,\ell}, v_{k,\ell})$ are taken to produce surface points $\mathrm{S}(u_{k,\ell}, v_{k,\ell})$, and a local surface frame $\mathbf{f}_{k,\ell}, \mathbf{g}_{k,\ell}, \mathbf{h}_{k,\ell}$ is produced as for true displacement mapping. If

$$\mathrm{D}_{k,\ell} = \mathrm{O}_{k,\ell} + \alpha_{k,\ell}\mathbf{i}_{k,\ell} + \beta_{k,\ell}\mathbf{i}_{k,\ell} + \gamma_{k,\ell}\mathbf{i}_{k,\ell}$$

Then the point

$$S(\mathrm{O}_{k,\ell}) + \alpha_{k,\ell}\mathbf{f}_{k,\ell} + \beta_{k,\ell}\mathbf{g}_{k,\ell} + \gamma_{k,\ell}\mathbf{h}_{k,\ell}$$

is assigned as an image of $\mathrm{D}_{k,\ell}$ and $\alpha_{k,\ell}\mathbf{f}_{k,\ell} + \beta_{k,\ell}\mathbf{g}_{k,\ell} + \gamma_{k,\ell}\mathbf{h}_{k,\ell}$ is assigned as an image of $\mathbf{d}_{k,\ell}$. These control vector images serve to define a spline overlay surface that represents an approximation to the displaced portion of the base surface:

$$\mathrm{S}(u(r,s), v(r,s)) + \mathbf{d}(r,s) \rightsquigarrow \sum_k \sum_\ell \left[\mathrm{S}(u_{k,\ell}, v_{k,\ell}) + \alpha_{k,\ell}\mathbf{f}_{k,\ell} + \beta_{k,\ell}\mathbf{g}_{k,\ell} + \gamma_{k,\ell}\mathbf{h}_{k,\ell}\right] c_k(r) c_\ell(s)$$

for $(r,s) \in \mathcal{D}_{\mathbf{d}}$.

Plates 2–5 show this process applied to the displacement spline of Plate 1. Plate 2 shows an affine mapping of the $\mathrm{D}_{k,\ell}$-associated points, $\mathrm{O}_{k,\ell}$, in $\mathcal{D}_{\mathbf{d}}$, shown as yellow spots, into the domain $\mathcal{D}_{\mathrm{S}}$, shown as a white square. Plate 3 shows the corresponding view of the base surface, in white, the image control vertices, $\mathrm{S}(u_{k,\ell}, v_{k,\ell}) + \alpha_{k,\ell}\mathbf{f}_{k,\ell} + \beta_{k,\ell}\mathbf{g}_{k,\ell} + \gamma_{k,\ell}\mathbf{h}_{k,\ell}$ as cyan boxes, the surface coordinate frames, $\{\mathrm{S}(u_{k,\ell}, v_{k,\ell}), \mathbf{f}_{i,j}, \mathbf{g}_{i,j}, \mathbf{h}_{i,j}\}$ as magenta gnomens, and the approximation to the displaced surface, in yellow. Plate 4 shows the corresponding items in exploded view. Plate 5 shows a Phong shaded version of the two surfaces combined.

It should be emphasised that the approximation to the displaced portion of the base surface, shown in yellow on Plate 3, is a free-standing spline. It is no harder to render or manipulate than the base surface itself. In particular, any manipulation of this spline surface will only serve to change the quantities $\alpha_{k,\ell}$, $\beta_{k,\ell}$, and $\gamma_{k,\ell}$, which have a direct reflection back to the unmapped displacement $\mathbf{d}$. Thus, this approximate displacement mapping serves not only as a cheap means of rendering and previewing, it also serves as a mechanism for subsequent design editing. The editing may proceed on the composite surface, by adjusting the control coordinates $\alpha_{k,\ell}$, $\beta_{k,\ell}$, $\gamma_{k,\ell}$ to distort the yellow surface of Plate 3, and the corresponding adjustment of these control coordinates will adjust the shape of the feature surface D in Plate 1. Conversely, the editing may proceed directly on the feature surface in Plate 1, and the changes in α, β, γ may be transferred directly to the yellow surface of Plate 3. Once the entities $\mathrm{O}_{k,\ell} = (u_{k,\ell}, v_{k,\ell})$, $\mathbf{i}_{k,\ell}$, $\mathbf{j}_{k,\ell}$, $\mathbf{k}_{k,\ell}$, $\mathrm{S}(u_{k,\ell}, v_{k,\ell})$, $\mathbf{f}_{k,\ell}$, $\mathbf{g}_{k,\ell}$, and $\mathbf{h}_{k,\ell}$ have been produced, editing an approximate displacement is no more costly than editing any conventional spline. (In our usage, $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are constant, so the cost of the initial approximate displacement is even less than this list of entities might suggest.)

The example shown in Plates 1–5 involved an affine transformation $\mathbf{T}$. $\mathcal{D}_{\mathbf{d}}$ was mapped to $\mathcal{D}_{\mathrm{S}}$ using rotations and translations. Plates 6–8 show an overlay that is warped through a sine function before being applied to the base surface. Plate 6 presents the mapped domain $\mathbf{T}(\mathcal{D}_{\mathbf{d}})$; Plate 7 shows a wire-frame rendering of the base surface, in white, and the displacement surface, in yellow. Plate 8 presents a Phong-shaded rendering of the base and approximate displacement surfaces together.

6 Accuracy

We have stressed that the displacement surfaces that are produced are only approximations to a true displacement surface. It might appear that the yellow is pasted quite cleanly onto the white in Plate 3; however, its boundary is merely a very close approximation to the base surface upon which it is resting. The surfaces used in this example are bicubic. Except for the equiparametric lines, $u =$ const. and $v =$ const., a bicubic surface generally has degree six behavior. Thus, the margin of the yellow displacement surface in Plate 3 is varying cubically, while the white base surface on which it is resting is varying as a sixth degree surface, since the displacement margin is not oriented along a line of constant u or v.

The true displacement surface, for which the yellow surface of Plate 3 is an approximation, would be

$$\begin{aligned}
\overline{\mathrm{S}}(u,v) \quad = \quad & \sum_i \sum_j \mathrm{S}_{i,j} b_i(u) b_j(v) \\
& + \sum_k \sum_\ell \\
& \quad [\alpha(r(u,v), s(u,v))\mathbf{f}(u,v) \\
& \quad + \beta(r(u,v), s(u,v))\mathbf{g}(u,v)
\end{aligned}$$

$$+\gamma(r(u,v),s(u,v))\mathbf{h}(u,v)]$$
$$c_k(r(u,v))c_\ell(s(u,v))$$

the yellow surface is merely

$$\begin{aligned}
\mathrm{S}(u,v) &= \sum_i \sum_j \mathrm{S}_{i,j} b_i(u) b_j(v) \\
&\quad + \sum_k \sum_\ell \\
&\qquad [\alpha(r(u,v),s(u,v))\mathbf{f}(r_k,s_\ell) \\
&\qquad\quad +\beta(r(u,v),s(u,v))\mathbf{g}(r_k,s_\ell) \\
&\qquad\quad +\gamma(r(u,v),s(u,v))\mathbf{h}(r_k,s_\ell)] \\
&\qquad c_k(r(u,v))c_\ell(s(u,v))
\end{aligned}$$

The example of Plates 6–8 are more readily comprehended as an approximation, despite appearing to be quite cleanly joined. After all, the yellow surface is varying as the sine of a cubic function along one margin. This non-polynomial behavior cannot be matched by the base surface.

As a *reducto ad absurdum* example that approximate displacements can be pushed too far, consider Plates 9–11. Here, the vector spline that defines the displacement is the zero vector spline. The true displaced surface would coincide exactly with the base, but the approximate displacement is mapped onto the base as a polygon that meets the base surface only at its corners. Plate 9 shows an exploded picture. In Plate 10 the view is from above, and the base surface is clearly covering the approximate displacement. Plate 11 shows the situation from below in Phong-shaded rendering.

Further evidence of the fact that only an approximate displacement is being produced may be seen in Plates 17–19. In these plates the overlay surface is of a strongly contrasting color to the base surface. Some spotting of red overlay appears mixed with white base along the margin of the "flange" portion of the overlay (which extends quite far away from the central displacement hump, as can be seen in Plate 16). This is partly due to z-buffer resolution, but also due to the fact, as in earlier examples, that the displacement is varying as a cubic along its margin, while the base is varying as a sixth degree surface. The scalloped edge that the displacement appears to have, on the other hand, is not primarily an issue of accuracy. Both surfaces are rendered by being tessilated into rectangular facets. Since the surfaces are arranged at an angle to each other, so are these facets, and the scalloping is due to the skewed arrangement of these rectangles. Rendering displacement surfaces correctly requires the same skill as rendering trimmed and intersecting surfaces.

If the given approximation is unsatisfactory, two possibilities exist to achieve a closer approximation to the true displacement surface: degree elevation and subdivision [Bartels87, Farin90]. For example, the zero displacement shown is represented by a bi-constant spline consisting of a single patch; in short, a polygon. It could equally well be represented as a much higher degree spline consisting of many more patches. By recursively segmenting the yellow polygon of Plates 9–11, we could achieve a polygonal approximation to the displacement region on the base surface. Any prescribed accuracy can be obtained. By allowing the degree to be raised as well, even greater accuracy can be achieved for a given number of patches.

Considering the formulas for $\mathrm{S}(u,v)$ and $\overline{\mathrm{S}}(u,v)$ given above, if sudivision alone is used so that the maximum patch diagonal tends to zero, then the domain points $\mathrm{D}_{k,\ell}$ will become dense in $\mathcal{D}_{\mathbf{d}}$, their images $\mathrm{S}(u_{k,\ell},v_{k,\ell})$ will become dense on S, and if S is C^1 continuous, the vectors $\mathbf{f}_{k,\ell}$, $\mathbf{g}_{k,\ell}$, and $\mathbf{h}_{k,\ell}$ within the locality of $\mathrm{S}(u,v)$ will converge to $\mathbf{f}(u,v)$ resulting in the convergence of $\mathrm{S}(u,v)$ to $\overline{\mathrm{S}}(u,v)$. For further information on spline approximation, see [deBoor78].

7 Multiple Overlays and a Final Example

A base surface with an overlay; that is, with a displacement region, is suitable to serve as the base for further overlays. Plates 12–15 show two illustrations of this, with Plates 14–15 providing a cautionary note. In Plate 12, the yellow overlay has been applied to the white base, and a further, green overlay is applied that partially overlaps the yellow overlay's domain. The white-yellow base on which the green overlay is placed has low curvature and is quite simple. The result of the multiple overlays is well behaved, as is seen from the Phong-shaded rendering of Plate 13. In Plate 14, however, the second overlay has been placed across the first. Since the first overlay has resulted in an area of sharp curvature, the local coordinate frames produced by the approximate mappings vary significantly throughout the region. As a result, the basic shape of the green feature is distorted by a considerable amount. In fact, the original shape of the yellow displacement, seen in Plate 12, resulted in a green surface so distorted that it intersected itself in several places. Only after reducing the sharpness of the yellow ridge did a reasonable composite surface (Plate 15) result.

We close with a "practical" example. The overlay, shown in red in Plates 16–19, derives from a spline surface shaped as a single, straight ridge. It is mapped to the base surface, a bent tube, by a rotation and a translation. The result provides a spiral feature along the length of the base. The composite spline forms a familiar icon seen during that part of the year known as "SIGGRAPH deadline time." Plate 16 presents a wire-frame view, and Plate 17 provides a Phong-shaded view. In Plate 18 we have attempted to interest the Kollosal Kristmas Kandy Kompany of Hackensack, NJ, in the concept of a slimmer cane, for the calorie minded. We merely pinched the base surface; the overlay surface conformed automatically to the change. In Plate 19 we explored a specialty version of the cane to be marketed in movie theaters when Batman II is released. In this case, the change was edited into the cane by modifying the overlay surface directly in its position as a feature on the base surface.

8 Acknowledgements

All photographic examples in this paper were produced on a Silicon Graphics 4D/120GTX with a Focus Graphics Camera and using a set of C++ tools for building prototype spline editors. These tools have been under development at the University of Waterloo Computer Graphics Laboratory, with Canadian NSERC funding, since 1989.

The authors are grateful for the help of Greg Veres, who provided the C++ classes for coordinate frames, Milan Sreckovic, who provided rendering and display classes, Bruce Hickey and Alex Nicolaou, who provided window, device, and interaction classes, and Al Vermeulen, who provided the C++ classes for matrices, vectors, and spline bases functions.

References

[Bartels87] Bartels, Richard, Beatty, John, and Barsky, Brian. *An Introduction to Splines for use in Computer Graphics and Geometric Modelling.* Morgan Kaufmann, 1987.

[deBoor78] de Boor, Carl. *A Practical Guide to Splines.* Springer-Verlag (1978).

[Casale87] Casale, Malcolm. "Freeform Solid Modeling with Trimmed Patches." *IEEE Computer Graphics and Applications*, **7**, 1 (1987) 33–43.

[Coquillart90] Coquillart, Sabine. "Extended Free-Form Deformation: A Sculpting Tool for 3D Geometric Modeling." *Proceedings of SIGGRAPH '90*, in *Computer Graphics* **24**, 4 (1990) 187–196.

[Dahmen91] Dahmen, Wolfgang, Micchelli, Charles, and Seidel, Hans-Peter. "Blossoming Begets B-Splines Built Better by B-Patches." *Mathematics of Computation* [accepted].

[Farin90] Farin, Gerald. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, second edition, 1990.

[Faux79] Faux, Ivo, and Pratt, Michael. *Computational Geometry for Design and Manufacture.* Ellis Horwood Ltd. (John Wiley & Sons), 1979.

[Foley90] Foley, James, vanDam, Andries, Feiner, Steven, and Hughes, John. *Computer Graphics Principles and Practice.* Addison-Wesley, second edition, 1990.

[Forsey88] Forsey, David, and Bartels, Richard. "Hierarchical B-Spline Refinement." *Proceedings of SIGGRAPH '88*, in *Computer Graphics* **22**, 4 (1988) 205–212.

[Goldman85] Goldman, Ronald. "Illicit Expressions in Vector Algebra." *ACM Transactions on Graphics*, **4**, 3 (1985) 223–243.

[Höllig89] Höllig, Klaus. "Box-Spline Surfaces." in *Mathematical Methods in Computer Aided Geometric Design*, edited by Tom Lyche and Larry Schumaker. Academic Press (1989) 385–402.

[Loop90] Loop, Charles, and DeRose, Tony. "Generalized B-spline surfaces of arbitrary topology." *Proceedings of SIGGRAPH '90*, in *Computer Graphics* **24**, 4 (1990) 347–356.

[Sederberg86] Sederberg, Thomas, and Parry, Scott. "Free-Form Deformation of Solid Geometric Models." *Proceedings of SIGGRAPH '86*, in *Computer Graphics* **20**, 4 (1986) 151–160.