Pasting Spline Surfaces

Cristin Barghiel, Richard Bartels and David Forsey

Abstract. Details can be added to spline surfaces by applying displacement maps. However, the computational intensity of displacement mapping prevents its use for interactive design. In this paper we explore a form of simulated displacement that can be used for interactive design by providing a preview of true displacements at low computational cost.

§Introduction

Displacement mapping is a standard and useful tool for realistic rendering [3,8], and it has been suggested for application to surface approximation [6]. Because of its computational cost, however, it has not yet been fully explored as a design tool. In [4] a restricted form of displacement mapping was investigated for use in interactive design. Composite surfaces were constructed by layering a sequence of *detail surfaces* onto a *base surface*. Each detail was formulated as a vector displacement field and mapped onto a region of the base. The process was restricted in that, in order to achieve continuity between detail and base, it was necessary that the detail share a common parametric domain with the base and possess a knot structure derivable from the base by refinement, a setting studied recently by Weller and Hagen [9]. These restrictions permitted the computations for displacement mapping to be simplified to such an extent that interactive design was feasible. Here we extend the techniques of [4] to remove some of those restrictions.

True displacement mapping involves a vector-valued function $\mathbf{d}(r,s)$ that is nonzero over a compact domain $(r,s) \in \mathcal{D}$, a 1-1 transformation $\mathbf{T}$ of that domain into the domain $(u,v) \in \mathcal{S}$ of a surface (point-valued) $\mathrm{S}_0(u,v)$, and the resulting composition $\mathrm{S}_1(u,v) = \mathrm{S}_0(u,v) + \mathbf{d}(\mathbf{T}^{-1}(u,v))$. (Bold, lower case letters denote vectors; bold upper case letters are transformations; Roman and Greek lower case letters are scalars; Roman upper case letters are points, and calligraphic letters are domains.) The continuity of S_1 is controlled by the continuity of S_0, $\mathbf{T}$ and $\mathbf{d}$. In the surface design setting that we wish to consider, the intent of employing a displacement map is to achieve the "pasting" of a detail surface $\mathrm{D}(r,s)$ onto a base surface $\mathrm{S}_0(u,v)$. The function $\mathbf{d}$ is

Mathematical Methods in CAGD III
M. Dæhlen, T. Lyche, and L. L. Schumaker (eds.), pp. 1–3.

ISBN xxx.

constructed from the difference between D and a *reference surface*. For convenience, the domain of D may be embedded into the range of D; that is, the points on D are identified with points in the space $\{(r, s, t)\}$, and the domain of D is identified with the plane $(r, s, 0)$ by some convenient homeomorphism. This plane is often taken as the reference surface: $\mathbf{d}(r, s) = \mathrm{D}(r, s) - (r, s, 0)$. The composition yielding S_1 must be implemented with respect to a coordinate frame $\{(r, s, 0), \mathbf{i}, \mathbf{j}, \mathbf{k}\}$ appropriate to $\mathbf{d}$, D and some manifold coordinate frame $\{\mathrm{S}_0(u, v), \mathbf{l}(u, v), \mathbf{m}(u, v), \mathbf{n}(u, v)\}$ defined over S_0 (with $\mathbf{n}(u, v)$ taken as normal to $\mathrm{S}(u, v)$). To achieve this, the mapping $\mathbf{T}$ should not only map (u, v) smoothly into (r, s) but it should also provide a smooth mapping from $\{(r, s, 0), \mathbf{i}, \mathbf{j}, \mathbf{k}\}$ to $\{\mathrm{S}_0(u, v), \mathbf{l}(u, v), \mathbf{m}(u, v), \mathbf{n}(u, v)\}$. The elements of this setting are shown in Figure 1, and except for the third dimension, this is exactly the setting for texture mapping [7].

$$\mathbf{d} = \alpha\mathbf{i} + \beta\mathbf{j} + \gamma\mathbf{k} \qquad\qquad \mathrm{S}_1 = \mathrm{S}_0 + \alpha\mathbf{l} + \beta\mathbf{m} + \gamma\mathbf{n}$$
$$\mathrm{D} = (r, s, 0) + \mathbf{d}$$

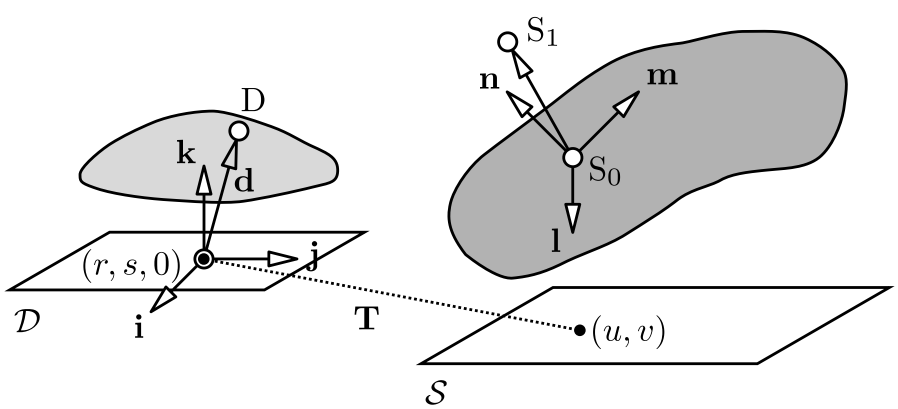

Fig. 1. Displacement mapping.

§Simulated Displacement

Since the process just outlined must be carried out at every surface point to be considered, a large amount of computation is involved. In high-quality image rendering, texture and displacement mappings are supportable because they are comparable to other computationally intensive aspects of the rendering process (e.g. ray tracing) and because we do not expect to produce high-quality images in real time. For interactive design, however, costs must be cut. We shall do this by limiting the mapping process to a selected, small number of sites, and choose the sites and the geometric entities to be mapped in a way that provides us with a reasonable preview of what a true displacement mapping will finally produce. The simulation must support such a preview in a dynamic as well as a static sense. That is; if the detail surface D is modified in its interior, that modification should be possible either before or after the pasting, and if the base surface S is modified after pasting, the pasted detail should conform to the change. Figure 2 illustrates this in a schematic way, where the detail is in grey, the base is in black, and the control points of the detail are shown as circles.

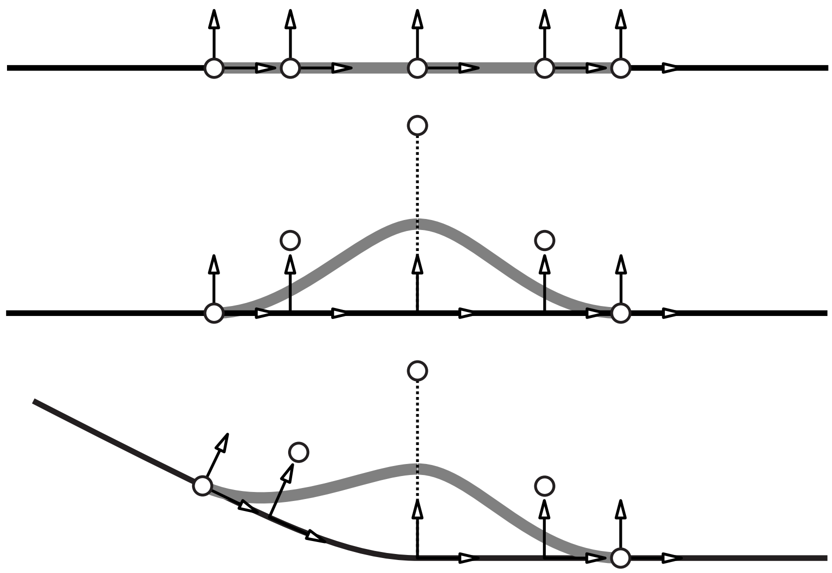

Fig. 2. Schema for displacement simulation.

Figure 2 also reveals further elements of the approach. The simulation will involve only mappings of the control points of the detail surface. We lock each control point of D to some individual origin site chosen on the base spline S_0, and the displacements $\mathbf{d}$ of these control points are measured with respect to coordinate vectors that are sensitive to the orientation of the base spline at the locality of each origin site. This is simpler to describe in terms of curves; having an extra variable adds nothing of substance to the ideas.

The detail spline is $\mathrm{D}(r) = \sum_\ell D_\ell c_\ell(r)$, $r \in \mathcal{D}$ for some basis splines c_ℓ. The displacement to be formed from $\mathrm{D}(r)$ is $\mathbf{d}(r) = \sum_\ell \left[D_\ell - (r_\ell, 0)\right] c_\ell(r)$ where $D_\ell - (r_\ell, 0)$ can be represented as $(r_\ell, 0) + \alpha_\ell \mathbf{i}_\ell + \beta_\ell \mathbf{j}_\ell$ in terms of coordinate frames $\{(r_\ell, 0), \mathbf{i}_\ell, \mathbf{j}_\ell\}$ (the orthonormal coordinate vectors $\mathbf{i}_\ell$,$\mathbf{j}_\ell$ are usually all chosen the same, for convenience). The base spline is $\mathrm{S}_0(u) = \sum_k \mathrm{S}_k b_k(u)$, $u \in \mathcal{S}$, and for all $u \notin \mathbf{T}(\mathcal{D})$ the composite spline $\mathrm{S}_1(u)$ is taken to be identical to $\mathrm{S}_0(u)$. For $\mathbf{T}(\mathcal{D}) \subset \mathcal{S}$ the composite spline $\mathrm{S}_1(u)$ is taken to be

$$\mathrm{S}_1(u) = \sum_\ell \left[\mathrm{S}_0(\mathbf{T}r_\ell) + \alpha_\ell \mathbf{m}(\mathbf{T}r_\ell) + \beta_\ell \mathbf{n}(\mathbf{T}r_\ell)\right] c_\ell(\mathbf{T}^{-1}u)$$

where $\mathbf{m}(u)$ and $\mathbf{n}(u)$ are unit tangent and normal, respectively, to $\mathrm{S}_0(u)$. Figure 3 provides an overview. (To avoid computing with $\mathbf{T}^{-1}$ one may regard this portion of S_1 to be a separate, free-standing spline in the variable r and display it juxtaposed to S_0 to provide an impression of S_1 in its entirety. On workstations that support *trimming*, S_0 can be trimmed against this free-standing spline for a better visual impression. Finally, note that if the *sense* of the normal $\mathbf{n}$ is changed, the result will be to map the displacement onto the opposite side of the curve. Thus, one may choose to map a displacement *positively* or *negatively*.)

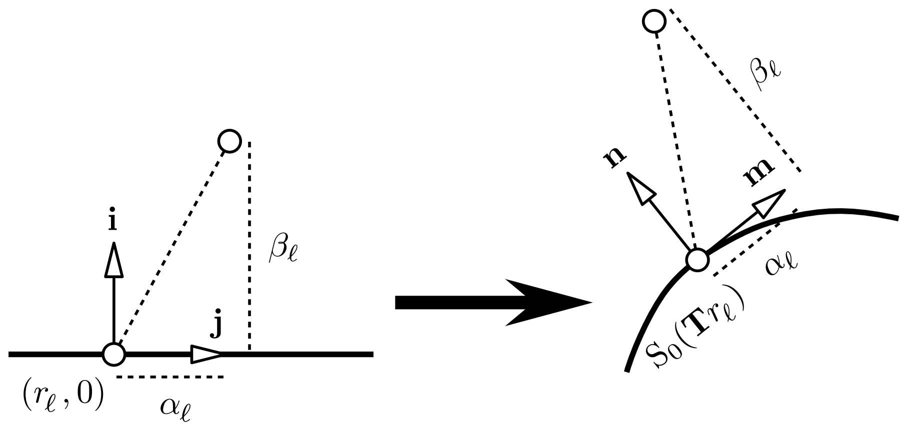

Fig. 3. Mapping displacement to base.

If modifications are made to D, these modifications are recorded through additional displacements for the control points of D

$$\mathbf{d}(r) \to \overline{\mathbf{d}}(r) = \sum_\ell \left[(r_\ell, 0) + (\alpha_\ell + \sigma_\ell)\mathbf{i}_\ell + (\beta_\ell + \tau_\ell)\mathbf{j}_\ell\right] c_\ell(r)$$

and these modifications may be applied to S_1 without the need to recompute the mapping

$$\mathrm{S}_1(u) = \sum_\ell \left[\mathrm{S}_0(\mathbf{T}r_\ell) + (\alpha_\ell + \sigma_\ell)\mathbf{m}(\mathbf{T}r_\ell) + (\beta_\ell + \tau_\ell)\mathbf{n}(\mathbf{T}r_\ell)\right] c_\ell(\mathbf{T}^{-1}u)$$

If any change is made retroactively to the base, $\mathrm{S}_0 \to \overline{\mathrm{S}}_0$, then the altered version of S_1 becomes

$$\mathrm{S}_1(u) \to \overline{\mathrm{S}}_1(u) = \sum_\ell \left[\overline{\mathrm{S}}_0(\mathbf{T}r_\ell) + \alpha_\ell \overline{\mathbf{m}}(\mathbf{T}r_\ell) + \beta_\ell \overline{\mathbf{n}}(\mathbf{T}r_\ell)\right] c_\ell(\mathbf{T}^{-1}u)$$

where, again, the mapping need not be recomputed. The only interactive design step that requires a recomputation of the mapping $\mathbf{T}$ is the step that revises the pasting of a detail onto a new base position.

The sites $(r_\ell, 0)$ chosen for this simulated displacement mapping are based on the *nodal points* (also called *Greville points*) for the basis $c_\ell(r)$; that is, the nodal point r_ℓ corresponding to c_ℓ is the value of r found by averaging all but the leftmost and rightmost knots on the support of c_ℓ. Several arguments make this a sensible choice. Under any reasonable refinement of D by knot insertion, each control point D_ℓ exhibits a higher order of convergence to the point $\mathrm{D}(r_\ell)$ than it does to any neighboring point $\mathrm{D}(r)$ [2], which provides this simulated version of displacement mapping a good limiting correspondence with true displacement mapping under refinement. Further, under the view that the simulated displacement mapping is designed to lay the domain $\mathcal{D}$ over the spline S_0 and thereby transmit the shape of S_0 in some meaningful way to the detail D, this association of nodal points with control points is known to be effective in providing such a geometric transmission of shape [5]. Finally, to take the most naive view, if D and S_0 both simply constituted flat planes, as suggested by the first part of Figure 2, and the mapping constituted the identity, $\mathbf{T}r \equiv r \equiv u$ then $\mathrm{S}_0(u) = \mathrm{S}_1$, and the most convenient values to take for α_ℓ and β_ℓ would be zero. In this setting our mapping would simplify to,

$$\mathrm{S}_0(u) = \mathrm{S}_1(u) = (u, 0) = \sum_\ell \left[\mathrm{S}_0(u_\ell)\right] c_\ell(u) = \sum_\ell (u_\ell, 0) c_\ell(u)$$

In order for this to be true, we must have $u = \sum_\ell u_\ell c_\ell(u)$ which is another property satisfied by the nodal points [2].

The above discussion extends to surfaces if (r) is replaced by (r, s), (u) is replaced by (u, v), the tangential curve coordinate vector $\mathbf{m}(u)$ is replaced by a suitable pair of tangential surface coordinate vectors $\mathbf{l}(u, v)$, $\mathbf{m}(u, v)$, a bivariate spline basis is used, and some equivalent of the nodal points can be found. For tensor product surfaces the extension is immediate and obvious.

§Hierarchical Application and Overlapping Domains

The use of displacement maps need not be restricted to a single detail $\mathrm{D} \equiv \mathrm{D}_0$. The surface S_1 can serve as the base for a further detail D_1, and the process can continue through a sequence of displacements

$$\langle \mathrm{S}_0, \mathrm{D}_0 \rangle \rightarrow \mathrm{S}_1 \,,\; \langle \mathrm{S}_1, \mathrm{D}_1 \rangle \rightarrow \mathrm{S}_2 \,,\; \langle \mathrm{S}_2, \mathrm{D}_2 \rangle \rightarrow \mathrm{S}_3 \,,\; \ldots$$

In this context it is more meaningful to speak of a *base composition*, consisting of an original base and zero or more details pasted upon it, and a *detail composition* consisting of another original base and zero or more details. The only thing distinguishing a base composition from a detail composition is the *pasting order*: the composite detail will be applied as a displacement upon the composite base.

Figure 5, for example, shows the domains of a base composition $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5\}$ and a detail composition $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2\}$. The components of each set are listed in pasted order. Domain $\mathcal{B}_1$ is the *root* of the composition, since it is not pasted onto any domain and is sufficiently large to accommodate all its pasted details. (*Every* surface composition is required to have a root domain – we have not investigated the possibility of having portions of a detail "hanging free" from any base.)

The domain dependencies are shown in Figure 6, where the nodes of the graph represent domains, and edges indicate portions of domains that overlap. Although both $\mathcal{B}_5$ and $\mathcal{D}_2$ are subdomains of $\mathcal{D}_1$, they lie on opposite sides: $\mathcal{B}_5$, a base component, lies *under* $\mathcal{D}_1$, while detail surface $\mathcal{D}_2$ is pasted *on* $\mathcal{D}_1$.

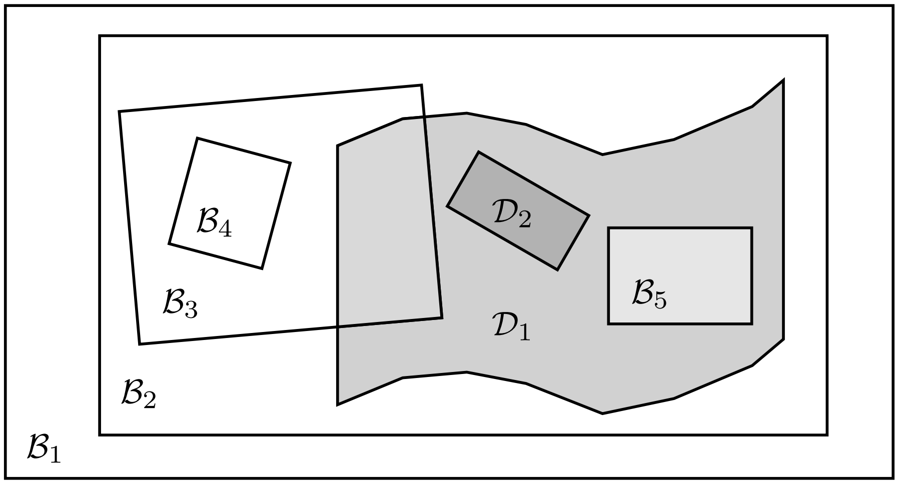

Fig. 5. Overlapping domains: the polygonal view.

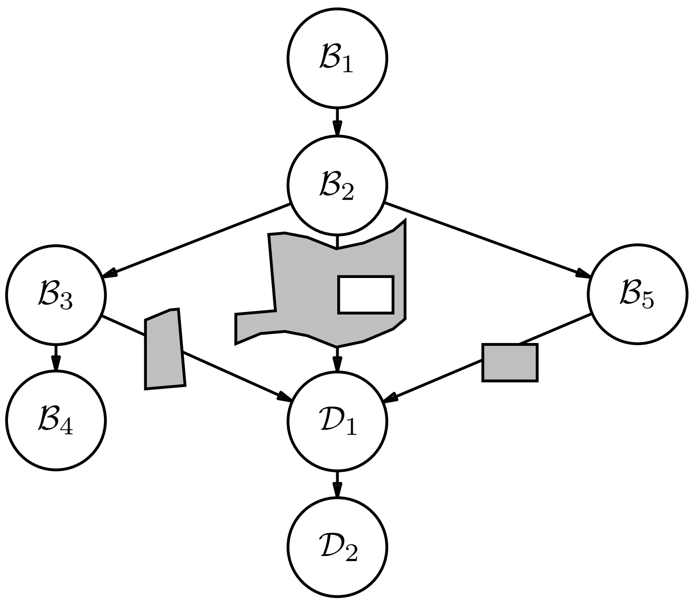

Fig. 6. Overlapping domains: the structural view.

This overlapping structure may change during interactive design as a detail composition is added to a base composition, moved about upon the base composition, or taken away from the base composition. When the structure of a composition changes, two steps precede the surface mapping operation: (1) recomputation of domain intersections, which requires a fast polygon-polygon clipping algorithm, and (2) reassociation of each Greville point on the detail domain composition with its underlying base-domain intersections, which requires a sequence of point-in-polygon operations, each one of which is followed by the construction of a surface-coordinate frame $\{\mathrm{S}, \mathbf{l}, \mathbf{m}, \mathbf{n}\}$ and the mapping of control vertices. This second step constitutes revising the mapping $\mathbf{T}$ between base composition and detail composition and represents the major expense for interactive updating. (If the domain dependency structure does

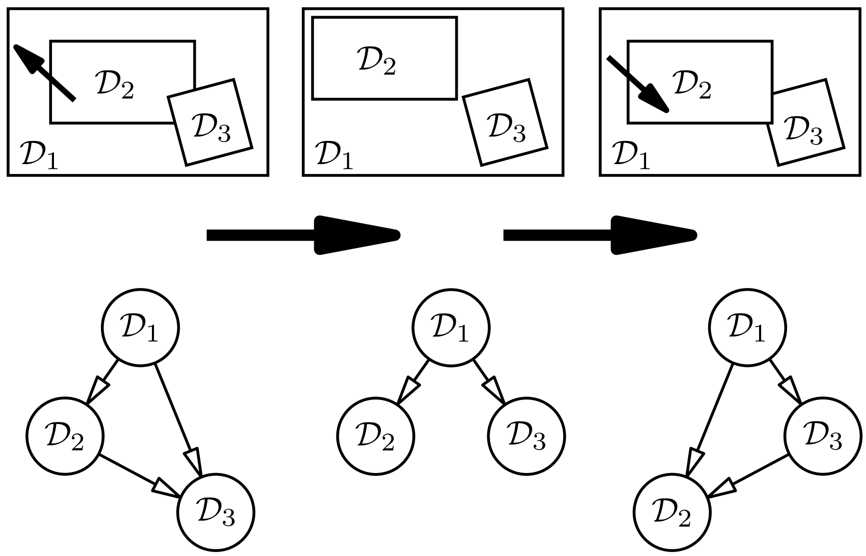

Fig. 7. Domain shuffling in a DAG.

not change but the *shape* of any component of surface – base or detail – is modified through the change of one or more control points, the two steps just mentioned are not necessary, but the control points for any detail surface affected must be re-mapped in pasted order, which requires evaluation of the affected surfaces at the nodal points, reconstruction of coordinate frames, and the transcription of control-point components into the new frames.)

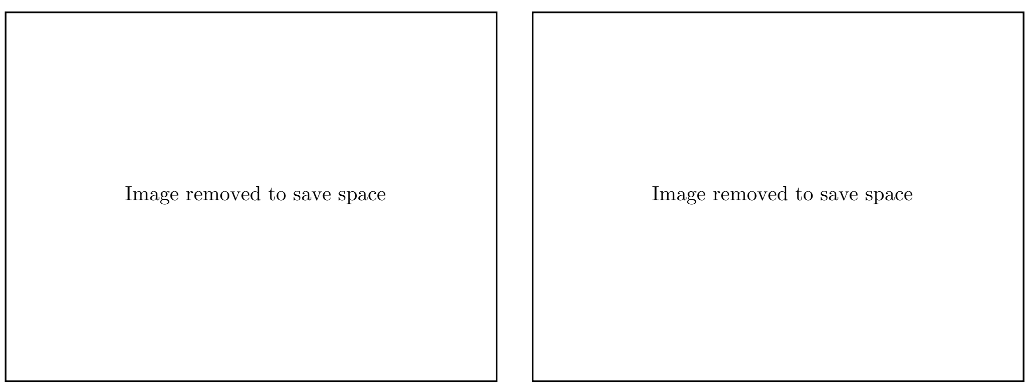

Fig. 8. Detail changes as base is modified.

The domain dependencies form a directed graph with interconnected nodes (Figure 6). Once some surface slides completely from under its detail surfaces, it ceases to influence the shape of these detail surfaces. If subsequently shifted back across any ex-detail surface domain, it is pasted *on top of* this ex-detail surface (Figure 7). This mechanism acts to prevent loop formation in the graph and provides an intuitive rule for structural motion. Thus, the layout of pasted domains may be maintained as a directed acyclic graph (DAG). The technical details of maintaining this DAG, together with some shortcuts involving the use and updating of a spanning tree for the DAG are covered in [1].

A surface editor, *PasteMaker*, has been written in C++ at the Computer

Image removed to save space

Image removed to save space

Fig. 9. Positive and negative pasting of the same detail.

Image removed to save space

Image removed to save space

Image removed to save space

Fig. 10. Pasting some features on a head.

Image removed to save space

Image removed to save space

Image removed to save space

Image removed to save space

Fig. 11. Simulated vs. true displacement mapping.

Graphics Laboratory in the University of Waterloo to illustrate the feasibility of interactive design using displacements. Editing may proceed at any level of a composition. The surfaces existing later in pasted order, thus serving as a detail composition, react in a natural way to the modification of their base composition (Figure 8). As we have mentioned previously, pasting may be *positive* or *negative*; that is, a detail may be attached so as to provide a bump or a depression by adjusting the sense of the surface normal (Figure 9).

Figure 10 provides an illustration of three stages in the design of a cartoon character for animation. The head surface was imported from another spline editor, while the beard, nose, and horns were designed within *PasteMaker*. Finally, Figure 11 shows a comparison of simulated displacement and true displacement. The upper left image in the figure shows a base and detail surface before pasting. The upper right image shows the result of pasting the detail at an angle across the base. The detail's 10 patches in the lengthwise direction are not quite able to "follow" the bulge in the base surface, resulting in an intrusion of the base into the detail. The lower left image shows the detail surface after refinement (naively, by midpoint insertion in the parameter corresponding to the lengthwise direction). The final image shows the situation after one more naive refinement. This image is indistinguishable from that produced by actual displacement mapping.

Acknowledgements. Funding was contributed by NSERC, ITRC, and GM. The authors are grateful for the help of Greg Veres, Milan Sreckovic, Bruce Hickey, Alex Nicolaou, and Al Vermeulen, who provided a number of C++ classes. Thanks are also due to Barry Fowler for helpful discussions.

References

1. Barghiel, Cristin, Feature oriented composition of B-spline surfaces, Master's thesis, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (Available as Computer Science Department Technical Report CS-94-13), March 1994.
2. de Boor, Carl, *A Practical Guide to Splines*. Springer-Verlag, New York, New York, 1978.
3. Crow, Frank, Texture, in *Computer Graphics Techniques: Theory and Practice*, D. F. Rogers and R. A. Earnshaw (eds.), Springer-Verlag, New York, New York, 1990, 159–187.
4. Forsey, David, and Richard Bartels, Hierarchical B-Spline Refinement, *Proceedings of SIGGRAPH '88*, Computer Graphics **22**, 4 (1988) 205–212.
5. Goodman, Tim, Bernstein-Schoenberg Operators, submitted to *Mathematical Methods in Computer Aided Geometric Design III*, T. Lyche and L. L. Schumaker (eds.), Academic Press, New York, New York, 1995.
6. Kaasa, Johannes, and Geir Westgaard, The 'Face Lift' Algorithm, in *Wavelets, Images, and Surface Fitting*, P-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A K Peters, Wellesley, Massachusetts, 1994, 303–310.
7. Rogers, David, *Procedural Elements for Computer Graphics*, McGraw-Hill, New York, New York, 1985.
8. Upstill, Steve, *The RenderMan Companion* Addison-Wesley, Reading, Massachusetts, 1990.
9. Weller, Frank, and Hans Hagen, Tensor Product Spline Spaces with Knot Segments, submitted to *Mathematical Methods in Computer Aided Geo-*

metric Design III, T. Lyche and L. L. Schumaker (eds.), Academic Press, New York, New York, 1995.

Cristin Barghiel
Side Effects Software Inc.
20 Maud Street, Suite 300
Toronto, Ontario CANADA M5V 2M5
cb@sidefx.com

Richard H. Bartels
Computer Science Department
University Of Waterloo
200 University Ave. W.
Waterloo, Ontario CANADA N2L 3G1
rhbartel@uwaterloo.ca

David R. Forsey
Department of Computer Science
University of British Columbia
Vancouver, BC CANADA V6T 1Z2
drforsey@cs.ubc.ca