Hierarchical B-Spline Refinement

David R. Forsey and Richard H. Bartels

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

ABSTRACT

Refinement is usually advocated as a means of gaining finer control over a spline curve or surface during editing. For curves, refinement is a local process. It permits the change of control vertices and subsequent editing of fine detail in one region of the curve while leaving control vertices in other regions unaffected. For tensor-product surfaces, however, refinement is not local in the sense that it causes control vertices far from a region of interest to change as well as changing the control vertices that influence the region. However, with some care and understanding it is possible to restrict the influence of refinement to the locality at which an editing effect is desired. We present a method of localizing the effect of refinement through the use of *overlays*, which are hierarchically controlled subdivisions. We also introduce two editing techniques that are effective when using overlays: one is direct surface manipulation through the use of *edit points* and the other is *offset referencing* of control vertices.

CR Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling — Curve, surface, solid, and object representations; Geometric algorithms, languages, and systems; Hierarchy and geometric transformations; Modeling packages.

Key Words and Phrases: Splines, free-form surface editing, refinement, subdivision.

1. Introduction

The material in this paper derives from experience gained during the design of a prototype B-spline surface editor intended for the construction of jointed bodies to be used in a realistic animation system [7] with which one of the authors has become involved. The two issues that influenced the design of the editor in this context were: the need to mix broad-scale surface manipulations with manipulations of a finer nature and the need to superimpose fine-scale details upon broad-scale surface movements and distortions. This has led to a hierarchy of surface refinements, which we refer to as *overlays*, to a representation of control vertices in terms of *offsets* relative to a hierarchy of local reference frames, and to the investigation of mechanisms for the *direct manipulation* of points on surfaces rather than manipulation indirectly through the movement of control vertices.

The techniques we employ are suitable for any tensor-product, parametric surface, defined in terms of control vertices and basis functions, provided that the surface supports a refinement algorithm; i.e. a re-representation process that replaces each basis function by an equivalent linear combination of one or more new basis functions. This includes, but is not restricted to, surfaces constructed from B-splines, Beta-splines, Bezier patches, and NURBS. This paper will develop the underlying ideas in the notation of B-splines, and assume that refinement is provided by the Oslo Algorithm [4].

Occasional discussions will be specific to the uniform, bicubic B-spline case, to streamline the presentation. The ideas have been implemented and tested using uniform cubic B-splines on a Silicon Graphics 4-D workstation. This simplification was made for reasons of the ease and speed with which such surfaces are supported on the hardware and in the graphics library of this workstation, not out of a limitation of the approach.

In Section 2 we will give a brief background for spline refinement. Section 3 follows with a description of overlays and their construction through the use of refinement. The representation of an overlay through the use of a local reference frame and offsets is covered in Section 4. Mechanisms for achieving direct manipulation of surface features are mentioned in Section 5. In Section 6 we touch upon some obvious ways to join our proposals with techniques introduced by Barr, Cobb, and Sederberg and Parry. Section 7 closes with some examples taken from our prototype editor.

2. Terminology and Notation

The material that we will be presenting, in its broadest formulation, relates to surfaces $\mathbf{S}(u,v)$ which are defined by *control vertices* $\mathbf{V}_{i,j}$ and *basis functions* $B_{i,k}(u)$, $B_{j,\ell}(v)$ of some polynomial order k and ℓ, respectively,

$$\mathbf{S}(u,v) = \sum_i \sum_j \mathbf{V}_{i,j} B_{i,k}(u) B_{j,\ell}(v) \ .$$

The basis functions, furthermore, should be *refinable* in the sense that each one can be re-expressed as a linear combination of one or more "smaller" basis functions

$$B_{i,k}(u) = \sum_r \alpha_{i,k}(r) N_{r,k}(u)$$

$$B_{j,\ell}(v) = \sum_s \alpha_{j,\ell}(s) N_{s,\ell}(v) \ .$$

Reflected in this property is a corresponding re-representation of the surface in terms of the smaller basis functions and a larger number of control vertices

$$\mathbf{S}(u,v) = \sum_r \sum_s \mathbf{W}_{r,s} N_{r,k}(u) N_{s,\ell}(v) \quad ,$$

where

$$\mathbf{W}_{r,s} = \sum_i \sum_s \alpha_{i,k}(r)\alpha_{j,\ell}(s)\mathbf{V}_{i,j} \quad .$$

Refinement is nonlocal in the sense that, if $\mathbf{V}_{i,j}$ is one control vertex that influences a (large) region, to which some fine detail is to be added, it is not possible merely to replace $\mathbf{V}_{i,j}$ by one or more control vertices $\mathbf{W}_{r,s}$. The conversion of V's to W's proceeds by way of the conversion of B's to N's, and this results in a large-scale replacement of V's by W's. V's that have no influence on the region to be edited may be changed along with V's that do have an influence.

The foregoing material is often presented in matrix terminology; e.g. as in [5,6,9], and we will follow this convention. The matrix notation derives from the fact that each of the functions B and N is nonzero in only a small region, where it is a piecewise composite of polynomials. For example, taking each polynomial piece of $B_{i,k}(u)$ and combining it with each polynomial piece of $B_{j,\ell}(v)$, a surface patch is produced that can be represented as

$$[\mathbf{u}][\mathbf{B}_u][\mathbf{V}][\mathbf{B}_v]^T[\mathbf{v}]^T \quad , \tag{2.1}$$

where the superscript T stands for the transpose of a matrix or vector. On each patch the parameters u and v vary over the unit interval, and

$$[\mathbf{u}] = \begin{bmatrix} 1\, u\, u^2 \cdots u^{k-1} \end{bmatrix} \quad ,$$

$$[\mathbf{v}] = \begin{bmatrix} 1\, v\, v^2 \cdots v^{\ell-1} \end{bmatrix} \quad .$$

The δ, γ^{th} patch is given by the control vertices

$$[\mathbf{V}] = \begin{bmatrix} \mathbf{V}_{\delta-k+1,\gamma-\ell+1} & \cdots & \mathbf{V}_{\delta-k+1,\gamma} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \mathbf{V}_{\delta,\gamma-\ell+1} & \cdots & \mathbf{V}_{\delta,\gamma} \end{bmatrix} \quad . \tag{2.2}$$

Finally $[\mathbf{B}_u]$ and $[\mathbf{B}_v]$ are the matrices formed from the polynomial coefficients appropriate for the basis pieces in the δ^{th} and γ^{th} parametric intervals respectively.

With an array of $(m+1)\times(n+1)$ control vertices

$$\begin{matrix} \mathbf{V}_{0,0} & \cdots & \mathbf{V}_{0,n} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \mathbf{V}_{m,0} & \cdots & \mathbf{V}_{m,n} \end{matrix} \quad . \tag{2.3}$$

and basis functions of order k and ℓ respectively, a tensor-product surface consisting of $(m-k+2)\times(n-\ell+2)$ patches is defined.

The common example is the uniform, bicubic, B-spline case, for which $k=\ell=4$ and both matrices $[\mathbf{B}]$ are equal to

$$\frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \tag{2.4}$$

for all polynomial segments.

The refinement process that produces basis functions N from basis functions B is derived, in the B-spline, Beta-spline, Bezier, and NURB case by breaking up one or more polynomial segments into a succession of smaller segments.

The re-expression of the surface defined by the $(m+1)\times(n+1)$ control vertices $[\mathbf{V}]$ as a surface of $(m+M+1)\times(n+N+1)$ new control vertices $[\mathbf{W}]$ is accomplished by two matrices composed of the α coefficients, $[\alpha_{left}]$ and $[\alpha_{right}]$:

$$[\mathbf{W}] = [\alpha_{left}][\mathbf{V}][\alpha_{right}]^T \quad , \tag{2.5}$$

where

$$[\mathbf{W}] = \begin{bmatrix} \mathbf{W}_{\mu-k+1,\lambda-\ell+1} & \cdots & \mathbf{W}_{\mu-k+1,\lambda} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \mathbf{W}_{\mu,\lambda-\ell+1} & \cdots & \mathbf{W}_{\mu,\lambda} \end{bmatrix} \quad ,$$

$$[\alpha_{left}] = \begin{bmatrix} \alpha_{\delta-k+1,k}(\mu-k+1) & \cdots & \alpha_{\delta,k}(\mu-k+1) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \alpha_{\delta-k+1,k}(\mu) & \cdots & \alpha_{\delta,k}(\mu) \end{bmatrix} \quad ,$$

$$[\alpha_{right}] = \begin{bmatrix} \alpha_{\gamma-\ell+1,\ell}(\lambda-\ell+1) & \cdots & \alpha_{\gamma,\ell}(\lambda-\ell+1) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \alpha_{\gamma-\ell+1,\ell}(\lambda) & \cdots & \alpha_{\gamma,\ell}(\lambda) \end{bmatrix} \quad ,$$

and $[\mathbf{V}]$ is as is given in (2.2).

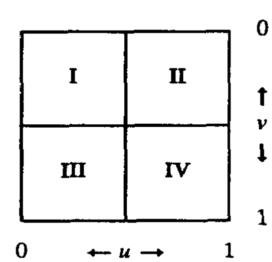
This matrix formulation is to be understood in the context of the δ, γ^{th} patch and the μ, λ^{th} subpatch that result from breaking the parametric ranges $0 \le u,v \le 1$ into a number of subranges

$$0 \le \cdots \le u_\mu \le u < u_{\mu+1} \le \cdots \le 1$$

and

$$0 \le \cdots \le v_\lambda \le v < v_{\lambda+1} \le \cdots \le 1 \quad .$$

The simplest example derives from the uniform cubic case where each parametric range in u and v is broken at its midpoint. This converts each patch determined by the control vertices $[\mathbf{V}]$ into four equal patches according to the diagram



Formula (2.1), in turn, is converted into

$$[\mathbf{u}][\mathbf{B}_u][\mathbf{W}][\mathbf{B}_v]^T[\mathbf{v}]^T \quad , \tag{2.6}$$

where $[\mathbf{B}_u]$ and $[\mathbf{B}_v]$ are given by (2.4), $[\mathbf{W}]$ is given by (2.5), and

$$[\alpha_{left}] = \begin{cases} [A_1] & \text{for regions I and III} \\ [A_2] & \text{for regions II and IV} \end{cases},$$

$$[\alpha_{right}] = \begin{cases} [A_1] & \text{for regions I and II} \\ [A_2] & \text{for regions III and IV} \end{cases}.$$

The matrices $[A]$ are given by

$$[A_1] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \end{bmatrix}$$

and

$$[A_2] = \begin{bmatrix} \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

A more thorough treatment of this material, and computational algorithms for determining the matrices $[\mathbf{B}]$ and $[\alpha]$ in the case of B-spline/Bezier surfaces (and by extension, NURBS) can be found in [2]. The computational algorithms presented there have their origin in the Oslo Algorithm first presented in [4]. For the case of Beta-splines the material in [8] is applicable.

3. Overlays

The refinement process described in Section 2 is the standard mechanism for reproducing an existing, V-defined surface using a W definition. The major complaint to be made about the process is that it frequently generates more W control vertices than we have any intention of moving. We would like to retain only those W's that interest us for editing purposes and discard the rest, retaining the unedited portion of the surface in its V definition.

It is important to remind ourselves that the refinement process produces an exact re-representation. The W-defined surface is the same as its V-defined parent. Figure 1 shows a small portion of a uniform, bicubic, V-defined surface in cross section (with circles indicating the V's), and Figure 2 shows a view of the same surface in a W definition (with black dots indicating the W's and with the V's included as circles for comparison). Refinement has been applied to the middle portion of the surface (centered about the topmost V). The right and left margins of the surface have not been included in the refinement.

If one of the W control vertices is moved, then the W surface departs from its V parent, but only in the area influenced by the W control vertex that has been changed. Outside of this area, parent and child are identical. Figure 3 shows the V surface superimposed on the W surface after one W has been moved. Because of this correspondence, we may continue to use the V definition as our basic description of the surface, save for the replacement of a small piece of the W definition. Generally more than just one W control vertex must be retained to define this small piece, but the number is small relative to the total number of W control vertices that the refinement produces. In the bicubic case, for example, 49 W control vertices must be retained in a 7×7 pattern centered around the control vertex that has been moved. In the general B-spline case, $[1+2(k-1)]\times[1+2(\ell-1)]$ W's must be retained centered about the control vertex that has been moved.

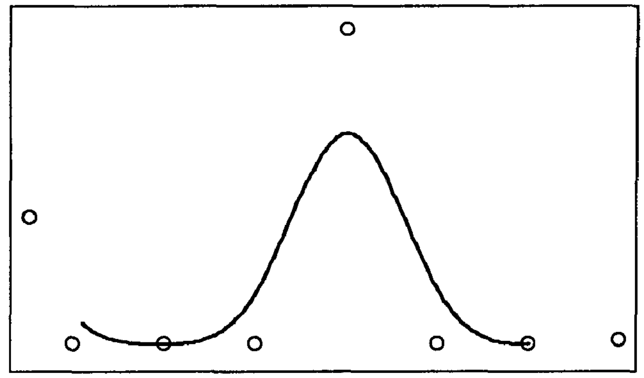


Figure 1. V surface in cross section.

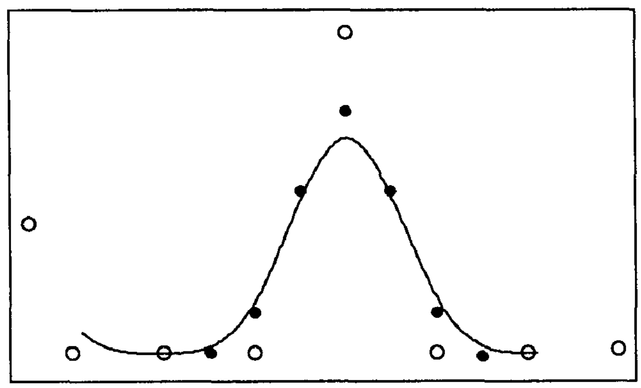


Figure 2. W surface in cross section.

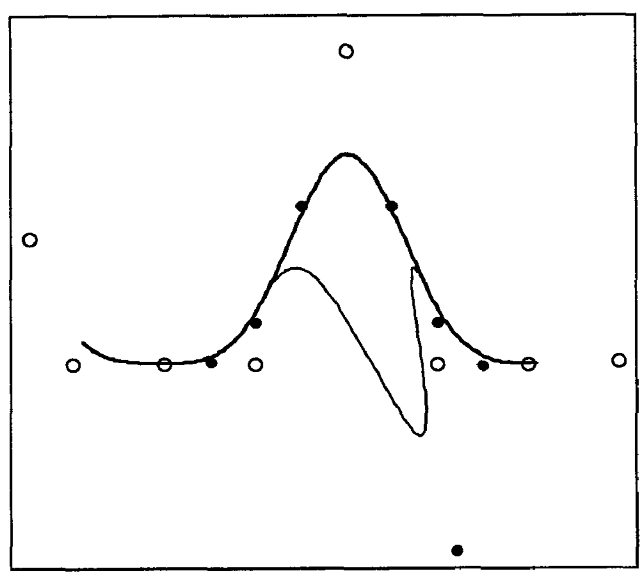


Figure 3. V surface and altered W surface.

We can regard the retained portion of the **W** definition as a separate surface to be manipulated. If we are careful to manipulate only the central **W** vertex and keep the peripheral **W** vertices static, we can localize editing/refinement processes to restricted patches of the surface. Each such localized patch subjected to such restricted editing and used as a replacement for a portion of the parent-level substrate constitutes an *overlay*.

We can repeat this approach on the interior of an overlay, regarding it, in its turn, to be the parent surface to be subjected to refinement for the creation of further overlays. The basic operation of creating an overlay consists of designating a patch on the surface at any level of refinement and executing a new refinement step to re-represent this patch. Also, we refine a surrounding number of patches sufficient to include the area influenced by any refined control vertex that we will manipulate. If this causes overlays to cross each other at some level of refinement, the overlays concerned are made into a composite overlay by combining their respective control vertices into a single collection.

In order to get a feel for the creation of an overlay, it is instructive to look at the simplest refinement step that could be carried out on a uniform, bicubic surface. Figure 4 shows a schematic plan of 7×7 control vertices, along with the 16 patches making up the surface they define. This constitutes the minimal surface portion that would change due to any movement of the central control vertex, $\mathbf{V}_{r,s}$. If all vertices save $\mathbf{V}_{r,s}$ are held fixed, the depicted region can be regarded as an independent surface, yet its boundary will always coincide with the larger surface from which it was derived.

If this area of surface is too large for the scale of detail we wish to introduce, the simplest thing we might do is refine each of the central four patches by halving, as was presented at the end of Section 2, Figure 5 diagrams the overlay that would result. The black dots represent the **W** control vertices, and the dashed boxes outline the smaller patches that the central **W** will influence. If only this central vertex is moved, and all the others are held fixed, the patches given by the dashed boxes can be considered an integral unit of surface whose boundary will always coincide with the 12 surrounding patches defined by the circles.

This is the style in which overlays are to be created. For editing involving the movement of several control vertices, modifying a larger area of surface, the overlay to be created must enclose the union of the separate single-vertex overlays. For editing that is to influence a smaller region, the refinement must break each patch into a greater number of subpatches. The creation of an overlay in the general B-spline case will involve other numbers of patches and **V** and **W** control vertices.

The net effect of refining and subdividing the surface, over time, will be that the surface will be broken into a collection of overlays at different levels of refinement. The obvious storage mechanism for managing overlays is a tree-structure with each level of depth in the tree corresponding to a level of refinement. The root node of the tree defines the basic **V**-defined surface, and every other node in the tree stores information that defines an overlay surface. Every node in the tree points to overlays that are derived entirely within the portion. Sibling pointers are used to access information in adjoining patches for ease in processing portions of the surface at a common level of refinement.

Successive application of refinement defines a composite surface recursively. The composite surface is given by the totality of information contained at the root and leaves of the data structure.

4. Offset Referencing

Up to this point, the formulas have been presented assuming that a single coordinate frame has provided the description of each portion of the surface at each level of refinement. This is not suitable in the context of hierarchical refinements. When editing takes place at one level of surface definition, any overlays resting within the edited area are expected to remain embedded in that area. They will follow editing changes only if they can be dynamically tied to that area, which amounts to saying that their control vertices must move in accord with the movement of the section of surface undergoing edit. One method of achieving this is to represent the control vertices of any overlay relative to a frame of reference fixed upon the surface being edited rather than relative to the fixed frame of reference defined by some external coordinate system. This brings into consideration a "reference-plus-offset" notation for the control vertices. By this we mean that we write each control vertex $\mathbf{W}_{i,j}$ of any part of the surface, whether root-level parent surface or overlay at any level of refinement, in the form

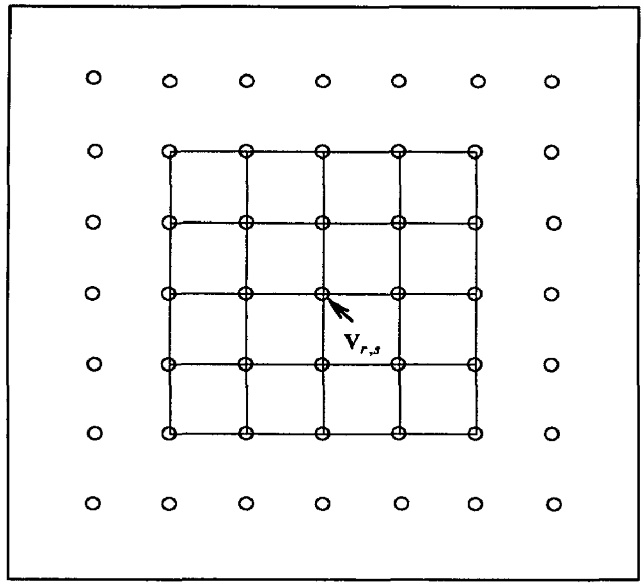


Figure 4. A 16-patch surface described by 49 control vertices.

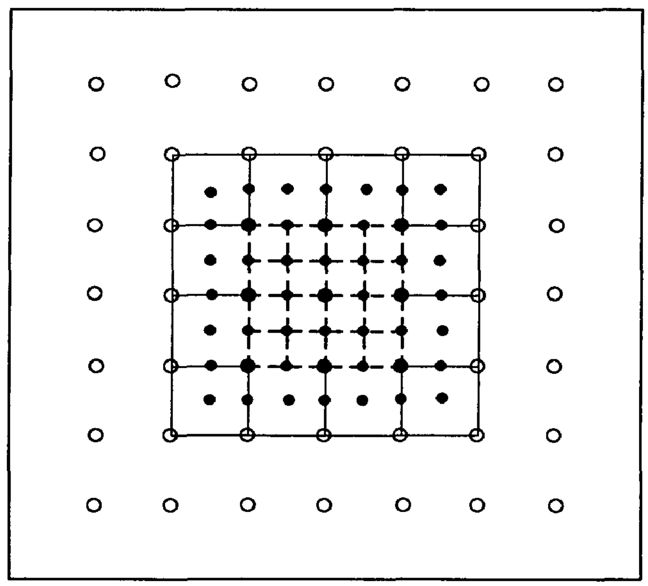


Figure 5. A 16-patch surface with refinement and overlay control vertices.

Figure 6. A surface created using local refinement.

$$\mathbf{W}_{i,j} = \mathbf{R}_{i,j} + \mathbf{O}_{i,j} \; ; \text{ i.e. } [\mathbf{W}] = [\mathbf{R}] + [\mathbf{O}] \; .$$

The point $\mathbf{R}_{i,j}$ is the reference, and it is to be specified from the parent surface. In our prototype editor $\mathbf{R}_{i,j}$, the reference for $\mathbf{W}_{i,j}$, is derived from the point on the surface maximally influenced by $\mathbf{W}_{i,j}$. The components of $\mathbf{R}_{i,j}$ are calculated from the tangent plane and the normal to the surface at this maximally influenced point. This is, however, only one of many possibilities, and an investigation of other choices is in order.

Editing changes to level **V** of a surface automatically cause revisions in the **R**'s, and through them revisions are made dynamically to the **W**'s. Editing changes to the **W**-level surface are to be recorded entirely as changes to the **O**'s. In the case of the root (original-level) surface, the **R**'s can be derived from points on a reference exterior to the surface; e.g. on the jointed member of a skeleton.

The implications of this on (2.1) are that the surface will be given as an offset component from a reference surface:

$$[\mathbf{u}][\mathbf{B}_u][\mathbf{W}][\mathbf{B}_v]^T[\mathbf{v}]^T \qquad (4.1)$$

$$= [\mathbf{u}][\mathbf{B}_u]\left\{[\mathbf{R}]+[\mathbf{O}]\right\}[\mathbf{B}_v]^T[\mathbf{v}]^T$$

$$= \left\{[\mathbf{u}][\mathbf{B}_u][\mathbf{R}][\mathbf{B}_v]^T[\mathbf{v}]^T\right\}$$

$$+ \left\{[\mathbf{u}][\mathbf{B}_u][\mathbf{O}][\mathbf{B}_v]^T[\mathbf{v}]^T\right\} \; .$$

Changes to the surface at any level of refinement closer to the root change **R**, and these appear as global changes to the overlay surface at the current level of refinement. The current level of surface, of course, has its control vertices changed in accord with changes in the **R** information, and this, in turn, changes the reference information for finer levels of surface. Likewise, changes to **O** affect the surface at the current level of refinement and all fine levels.

Rendering proceeds for any point on the surface using reference and offset information at the lowest level of the tree containing that point. Thus, offset information accumulates editing changes local to an overlay, and references reflect editing change at all higher levels; i.e., all global changes to the offset (See Figures 7 and 8).

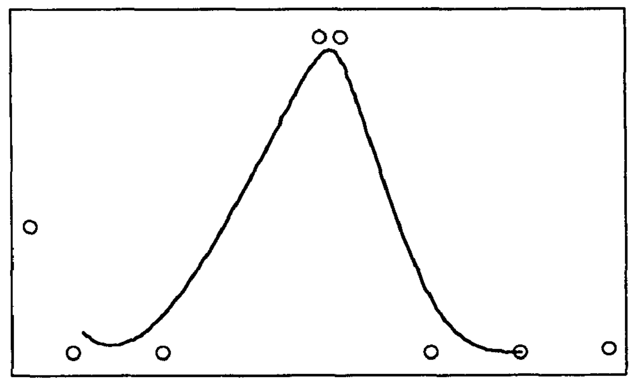


Figure 7. The V-level surface after moving one control vertex.

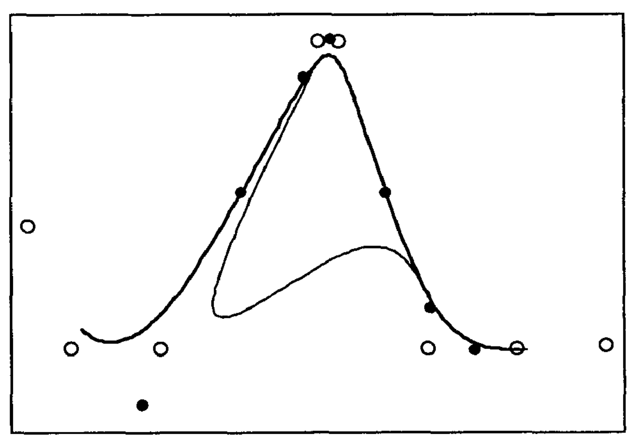


Figure 8. The W-level surface dragged along on the V surface due to offset referencing.

5. Direct Surface Manipulation and Edit Points

A composite surface of the nature indicated has a highly complicated structure of control vertices. It becomes more sensible to allow direct picking and manipulation of points on the surface than to hope that the user can make any sense of the maze of the control graph. The most fundamental relationship between surface and control vertices that can be provided is the one that relates each control vertex to the point on the surface over which it has maximal influence. A movement of that surface point can be converted, transparently to the user, to a corresponding change in that one control vertex. This generalizes to areas on the surface influenced by multiple control vertices.

The complete implementation of the pick operation to achieve these ideas would require a root-finding algorithm to convert a point **P** on the surface to u,v values on some patch. However, it is easy to determine the u,v associated with the point **P** maximally influenced by any given control vertex $\mathbf{V}_{i,j}$. For example, in the uniform cubic case, the point maximally influenced by any control vertex $\mathbf{V}_{i,j}$ is given by the formula:

$$\begin{aligned}\mathbf{P} = &\tfrac{1}{6}(\tfrac{1}{6}\mathbf{V}_{i-1,j-1}+\tfrac{4}{6}\mathbf{V}_{i,j-1}+\tfrac{1}{6}\mathbf{V}_{i+1,j-1}) \qquad (5.1)\\ &+\tfrac{4}{6}(\tfrac{1}{6}\mathbf{V}_{i-1,j}+\tfrac{4}{6}\mathbf{V}_{i,j}+\tfrac{1}{6}\mathbf{V}_{i+1,j})\\ &+\tfrac{1}{6}(\tfrac{1}{6}\mathbf{V}_{i-1,j+1}+\tfrac{4}{6}\mathbf{V}_{i,j+1}+\tfrac{1}{6}\mathbf{V}_{i+1,j+1}) \ ,\end{aligned}$$

We have carried out our preliminary studies by offering only these points of maximal influence on the surface for picking and manipulation.

The displacement of the above surface point **P** to a new position **Q** is carried out by adjusting the control vertex of maximal influence. For example, in the uniform cubic case, the control vertex $\mathbf{V}_{i,j}$ is to be replaced by the new control vertex $\overline{\mathbf{V}}_{i,j}$ according to the formula

$$\overline{\mathbf{V}}_{i,j} = \tfrac{36}{16}\left(\mathbf{Q}-\mathbf{P}\right) + \mathbf{V}_{i,j} \ .$$

Of course, $\overline{\mathbf{V}}_{i,j}$ is to be represented as the fixed reference $\mathbf{R}_{i,j}$ suitable for the **V**'s and an offset $\mathbf{O}_{i,j}=\overline{\mathbf{V}}_{i,j}-\mathbf{R}_{i,j}$. Further, as we have pointed out, all overlays depending on the surface just changed must have their reference information updated.

In the Section 7 we present some examples taken from our prototype surface editor. This editor was written in C to run on a Silicon Graphics 4-D workstation. The software uses the Silicon Graphics library for its spline computations; consequently all example pictures represent uniform bicubic B-spline surfaces.

6. Other Manipulation Techniques

Each overlay, together with the originally defined surface on which the overlays were constructed, may be viewed as an individual spline surface, independently defined by control vertices and basis functions. The only conditions required for the integrity of the hierarchical composite is that manipulations carried out on any overlay be restricted to the interior of the overlay, and that the control-vertex information be stored in reference-plus-offset format. Within this context, however, manipulations that are traditionally performed on control vertices may be performed on offset information instead. We have presented manipulations that concentrated on the use of edit points, but many things are possible. In particular, the composite forms of manipulation suggested by Cobb [3] may be used in this context.

The result of any hierarchical editing session is a composite surface, whose individual components are simply tensor-product spline surfaces occupying known locations in space. Transformational techniques that build new surfaces from existing ones can use the composites produced by our suggestions. In particular, Barr [1] and Sederberg and Parry [10] have proposed methods of transforming existing surfaces into deformed surfaces. Barr shows how space curves can be continuously deformed, and then shows how such deformations can be applied, approximately, to bivariate, parametric surfaces, by a process of point sampling. Sederberg and Parry embed surfaces in a trivariate spline volume, distort the volume by control-vertex manipulation, and then compute, approximately, the corresponding deformation of the embedded surface, also by a process of point sampling. Either of these techniques would work on the surfaces we have produced.

7. Examples

The surface in Figure 9 was constructed through repeated local refinement around the center of the 16-patch surface illustrated in Figure 5. The dark orange portion spans the region created in the first step of refinement as illustrated in Figure 6. The light orange portion spans a region created by a second step of refinement around the same edit point.

Parts b, c, and d of Figure 9 illustrate the effect of moving the central edit point at different levels of refinement. If the central edit point is moved at the finest level of refinement, the entire surface is affected (Figure 9b). In Figures 9c and 9d, by moving the edit point at coarser levels of refinement, more restricted regions of the surface are modified without affecting the integrity of the entire surface. Figure 10a-c illustrates how the modifications of the surface at a fine level of refinement affects edits performed at coarser levels of refinement. Edits performed at the coarsest level (Figure 10a), are retained when larger regions of the surface are modified (Figure 10b-c). After editing at a fine level, the surface can be still be modified at the coarsest level of refinement without affecting finer levels (Figure 10d). In Figure 11 lighter colored patches denote regions of the surface that have undergone greater refinement.

8. Conclusions

Tensor product B-splines are quite flexible, but they possess a deficiency when it comes to refinement. Refinement may change more control vertices than we wish to manipulate. We present a solution to this problem that uses a hierarchical data structure to localize the refinement operations. This data structure also supports, by means of offset referencing, a means of allowing a designer to manipulate a surface conveniently at various levels of detail.

Local refinement and offset referencing provide a flexible and powerful new editing tool. Local refinement controls the extent of any modification to the surface, and offset referencing allows localized edits to be retained over global changes to the surface.

9. Acknowledgements

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada and in part by the Studio d'Animation Français of the National Film Board of Canada. We wish to thank Robert Forget for arranging the latter funding. Thanks are also due to Silicon Graphics for the donation of the 4-D workstation on which the system was implemented, to Alan Paeth for his IM imaging tools, Victor Klassen for his sky and image resizing software, and Stewart Kingdon for his solid texturing software. Robert Dickinson of Visual Edge Software participated in many of our formative discussions on this material.

Comments by the reviewers were most helpful, and we appreciate the care with which they read the manuscript.

10. References

1. Barr, Alan, "Global and Local Deformations of Solid Primitives", Proceedings of SIGGRAPH'84 (Minneapolis, Minnesota, July 23-27, 1984). In *Computer Graphics, 18*,3 (July, 1984) 21-30.

2. Bartels, Richard, Beatty, John, and Barsky, Brian, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann Publishers, Palo Alto, California (1987).

3. Cobb, Elizabeth, *Design of Sculptured Surfaces Using the B-Spline Representation*, University of Utah PhD Thesis, Salt Lake City, Utah (1984).

4. Cohen, Elaine, Lyche, Tom, and Riesenfeld, Richard, "Discrete B-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics," *Computer Graphics and Image Processing, 14*,2 (October, 1980) 87-111.

5. Faux, Ivor and Pratt, Michael, *Computational Geometry for Design and Manufacture*, John Wiley & Sons (1979).

6. Foley, James and van Dam, Andries, *Fundamentals of Interactive Computer Graphics*, Addison Wesley (1982).

7. Forsey, David and Wilhelms, Jane, "Techniques for Interactive Manipulation of Articulated Bodies Using Dynamic Analysis," Proceedings of Graphics Interface '88 [to appear] (1988).

8. Joe, Barry, "Discrete Beta-Splines," Proceedings of SIGGRAPH'87 (Anaheim, California, July 27-31, 1987). In *Computer Graphics, 21*,4 (July, 1987) 137-144.

9. Mortenson, Michael, *Geometric Modeling*, Wiley, New York (1985).

10. Sederberg, Tom and Parry, Scott, "Free-Form Deformation of Solid Geometric Models," Proceedings of SIGGRAPH'86 (Dallas, Texas, August 18-22, 1986). In *Computer Graphics, 20*,4 (August, 1986) 151-160.

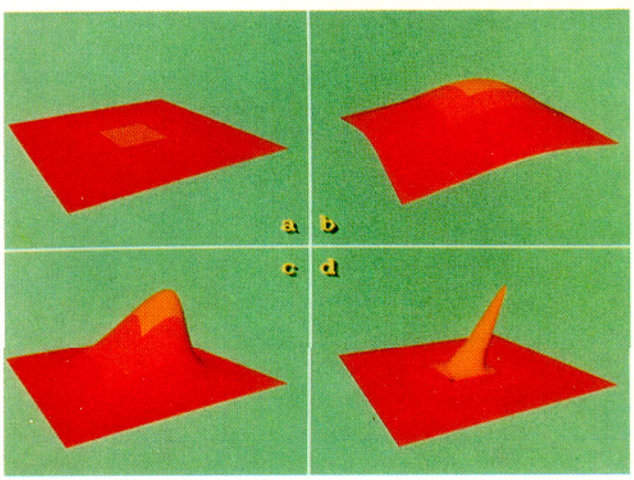


Figure 9. The effect of refinement level on surface modifications.

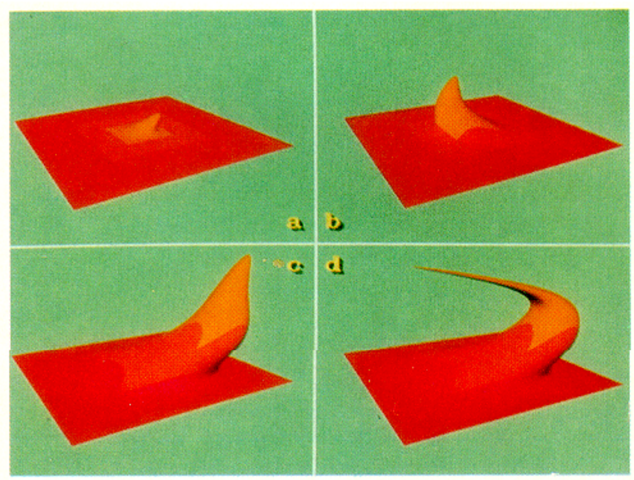


Figure 10. The use of offset referencing for surface editing

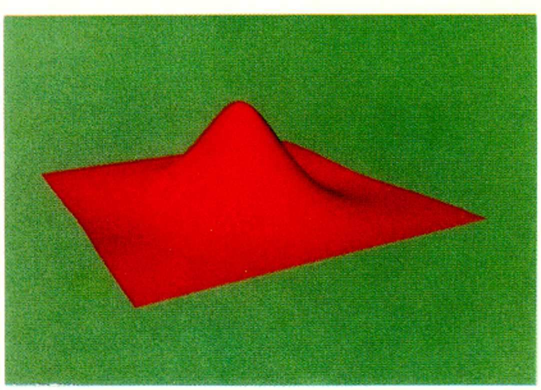


Figure 11a. A 16-patch surface is drawn into a hump by moving one edit point.

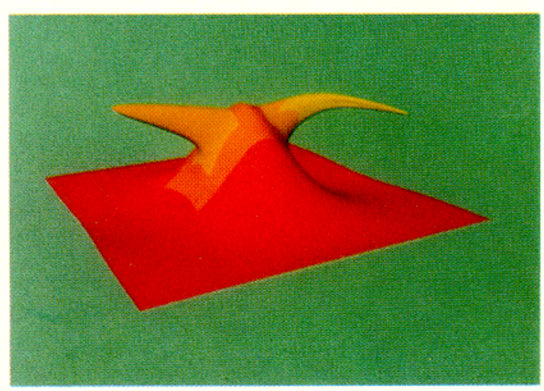


Figure 11b. Two refinement stages are used to create regions that can be drawn into horns. The horn on the right has been further refined to allow a local increase in curvature for a sharp point.

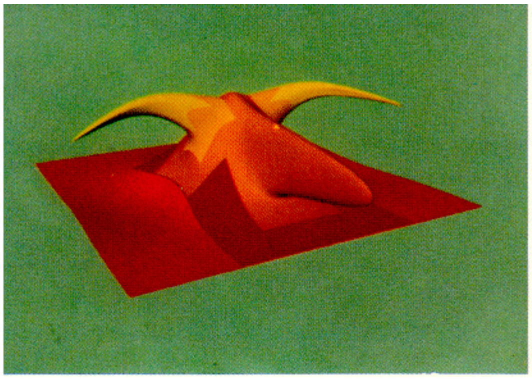


Figure 11c. The forward part of the hump is refined and drawn forward to create a snout.

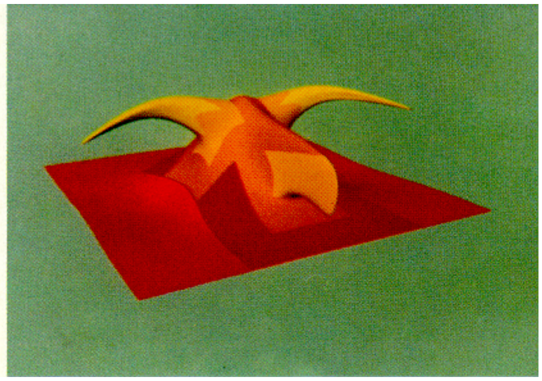


Figure 11d. The tip of the snout is refined and pulled down into a beak.

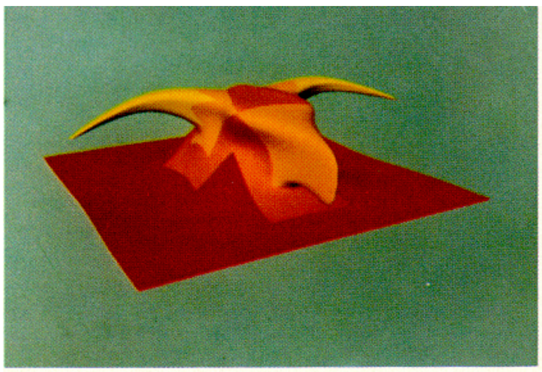


Figure 11e. Brow ridges are brought forward and down.

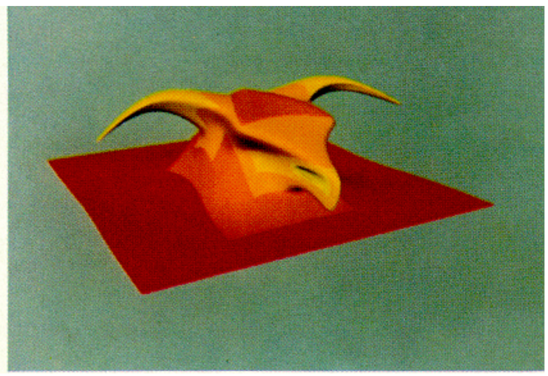


Figure 11f. The tip of the snout is refined twice and nostrils are constructed.

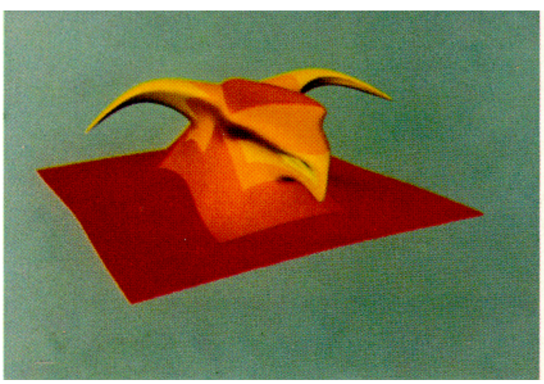


Figure 11g. The front of the snout is spread and the head is raised.

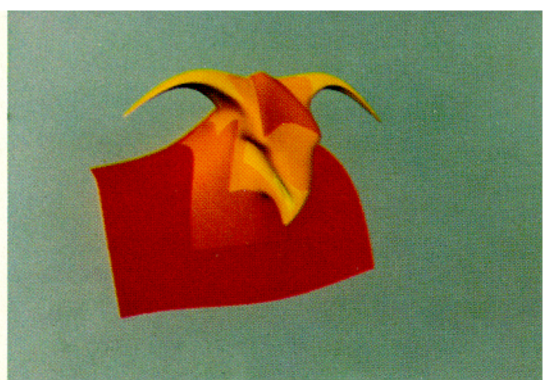


Figure 11h. The base-level surface is folded over.