Multi-resolution Surface Approximation for Animation

David Forsey and Lifeng Wang
Dept. of Computer Science
University of British Columbia, Vancouver, B.C. Canada V6T 1Z2
604-822-8158
e-mail address: {drforsey|wang}@cs.ubc.ca

ABSTRACT

This paper considers the problem of approximating a digitized surface in R^3 with a hierarchical bicubic B-spline to produce a manipulatable surface for further modelling or animation. The 3D data's original mapping from R^2 (multiple rows of cylindrical scans) is mapped into the parametric domain of the B-spline (also in R^2) using a modified chord-length parameterization. This mapping is used to produce a gridded sampling of the surface, and a modified full multigrid (FMG) technique is employed to obtain a high-resolution B-spline approximation. The intermediate results of the FMG calculations generate the component overlays of a hierarchical spline surface representation.

Storage requirements of the hierarchical representation are reduced by eliminating offsets where-ever their removal will not increase the error in the approximation by more than a given amount. The resulting hierarchical spline surface is interactively modifiable (modulo the size of the data set and computing power) using the editing capabilities of the hierarchical surface representation allowing either local or global changes to surface shape while retaining details of the scanned data.

RESUME

Cet article addresse le problème d'approximer une surface digitalisée en R^3 par une B-spline hiérarchique bi-cubique pour produire une surface manipulable pour la création ou la modélisation. Un mapping initial des données de R^2 dans le domaine paramétrique de la B-spline (aussi en R^2) est modifié pour produire une paramétrisation euclidienne. Ce mapping concentre la surface spline dans les régions de haute pente et est utilisé pour produire un échantillonnage régulier des données. La méthode numérique de multigrid complète (FMG) approxime la surface; elle ajuste une surface B-spline de haute résolution aux données. Les résultats immédiaires des calculs sont utilisés directement pour générer les surfaces de composantes overlay de la spline hiérarchique. Les requièrements en mémoire pour la définition de la surface sont diminués en réduisant à zéro les vecteurs de décalage qui sont à l'intérieur d'epsilon dans chaque niveau overlay. La surface spline hiérarchique résultante est modifiable intéractivement localement et globalement tout en retenant les détails de la surface des données importées.

KEYWORDS: Geometric modelling, surface approximation, multigrid methods, hierarchical B-splines, parameterization, animation.

INTRODUCTION

Problem Definition

Hierarchical splines are a multi-resolution approach to splines for use in interactive creation of free-form surfaces [8] [22]. The additional need often arises both in CAD and computer animation for a deformable surface that resembles some existing physical object. This paper addresses the issue of creating a hierarchical bicubic B-spline approximation to data obtained from systematic measuring devices such as laser rangers, CAT imagery systems, or optical scanners. This data is typically arranged in a rectangular array indexed by row and column number q, r, and describes either a height field or a cylindrical surface where each row in the array encodes a single cross-section of the cylinder (Figure 1).

Surface Approximation

We are interested in fitting a tensor-product B-spline surface

$$S(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} V_{i,j} B_{i,k}(u_i) B_{j,l}(v_j) \qquad (1)$$

to a given set of data points. The variables k and l refer to the order of the basis function in the u and v parametric directions respectively. The basis functions B_i and C_j will be left open throughout this discussion but must *partition unity*, be *variation diminishing, non-negative*, have *compact support* and be *refinable* [1] [6]. These properties are common to B-spline, Beta-splines and rational splines derived from either of these kinds of bases. Throughout this paper, the basis will be of order 4, but the techniques discussed generalize to any order.

To generate the equations for surface approximation, each data point $(x, y, z) = D_\lambda \in R^3$ is associated with a domain point $(u, v) = \delta_\lambda \in R^2$ of the spline. This forms the equation:

$$\sum_{i=0}^{m}\sum_{j=0}^{n} V_{i,j} B_{i,k}(u_i) B_{j,l}(v_j) = D_\lambda. \tag{2}$$

Data is *gridded* if

$$\begin{array}{l} u \in \{u_0, \cdots, u_M\} \\ v \in \{v_0, \cdots, v_N\} \end{array} \tag{3}$$

and if the δs consist of all points in $\{u_0, \cdots, u_M\} \times \{v_0, \cdots, v_N\}$ the surface approximation equations become

$$\sum_{i=0}^{m}\sum_{j=0}^{n} V_{i,j} B_{i,k}(u_s) B_{j,l}(v_t) = D_{q,r}. \tag{4}$$

for $s = 0, \cdots, M$ and $t = 0, \cdots, N$. Thus the number of equations is equal to the number of data points.

The system of equations is *overdetermined* if the number of equations (i.e. the number of data points) exceeds the number of unknowns (i.e. the number of control vertices). In this case Equation 4 is replaced by the *normal equations* for a least squares solution. If the number of equations is equal to the number of unknowns, the system is *simply determined* and all of the data points will be interpolated. If the number of equations is less than the number of unknowns, the system is *underdetermined*, and multiple solutions exist. In this case, equations can be added to the system to uniquely determine the solution, or some numerical methods [3] can be employed to select one of the many possible solutions.

Surface Approximation for Animation

The problem of applying these equations to approximate a digitized surface with a hierarchical spline is complicated by two factors: an unusual parameterization, and the multi-resolution nature of the hierarchical formulation itself.

For gridded data arranged as in Figure 1, the most straightforward parameterization would map the data onto a cylindrical spline surface. However, for our particular application, a cylinder is inappropriate because the data is destined to be used to determine the shape of a rectangular sub-region of a pre-existing spline figure where the edges of the sub-region match the opening of the neck and the remainder of the surface extends over the top of the head. Furthermore, because a hierarchical spline is composed of multiple spline surfaces (called *overlays*), any surface approximation must also have some mechanism to to create these overlays.

This paper addresses these issues. A review of related work is followed by a brief overview in Section 2 of the hierarchical surface formulation, parameterization methods, and the multigrid method for solving a linear system of equations. Section 3 describes the process of applying these techniques to the problem of approximating a non-trivial example of a digitized surface (Victor Hugo) with a hierarchical spline surface.

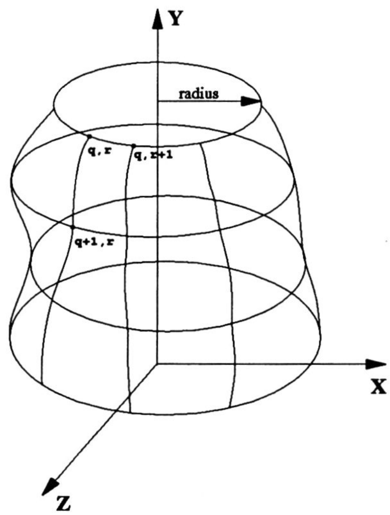

Figure 1: Digitized Data Topology

Related Work

The curve and surface fitting literature is extensive. This paper will not attempt a survey of the field but will briefly examine some related work relevant to our particular application.

Nahas et. al. [16] examine the problem of creating a B-spline facial model from digitized data. The raw mesh of digitized data points is used, unmodified, as the control vertices of a bicubic B-spline surface. Broad-scale changes in the shape of the surface are made by embedding the control vertices of the raw mesh in a coarser spline mesh created by selecting a relatively few *characteristic points* from the initial data points. Changes in the position of the characteristic points displace the vertices of the fine-mesh model.

In [18], an adaptive process was presented to fit surface data with a geometrically continuous collection of rectangular bicubic Bézier patches. The adaptivity is derived from fitting a portion of the data with a patch, testing the fit for satisfaction within a given tolerance, and subdividing the patch if the tolerance was not met. Geometric continuity was controlled by using a constrained least squares approximation method where the constraints imposed the continuity conditions. A considerable number of constraints are required to piece the Bézier patches together to form a continuous composite surface. The broader class of tensor product spline surfaces (e.g. B-spline, Beta-splines or their rational counterparts) provide continuity without the imposition of constraints but individual patches but typically cannot be refined to provide the localized approximation achieved in [18] with Bézier patches.

Hierarchical splines [8] [12] allow local refinement of uniform B-splines or Beta-splines and their rational counterparts. This property was exploited to provide an adaptive algorithm (similar to [18]), for approximating regular (gridded) data in three dimensions with a bicubic B-spline surface [11], and subsequently to multivariate B-splines in any dimension [10]. This approach iteratively applies least-squares to fit a series of successively finer spline surfaces (i.e. more patches) to the data. Eventually, regions of inadequate fit become separated and surrounded by regions within tolerance. These regions are individually accommodated using local refinement and constrained least squares surface approximation applied to each separable region.

The hierarchical approach makes a further contribution by providing an economical representation for the final composite surface. The algorithm produces reasonable results, but the resulting surface suffers from oscillations when presented with data with high frequency components. These oscillations were considerably reduced through the use of non-uniform hierarchical B-splines and improved parameterization [19]. However, hierarchical surfaces produced using least-squares are not amenable to further manipulation via the hierarchy because each level in the hierarchical surface is the result of a separate least squares approximation. Since the shape of each overlay level is unrelated to its parent, the results of broad-scale interactive changes in shape are unpredictable.

In contrast to this top-down approach, Lyche and Morken [15] work bottom up by first approximating the surface with a fine mesh of spline patches and then removing knots in those regions where knot removal will not cause the surface to move out of tolerance. This paper presents a bottom-up approach to surface approximation using hierarchical splines.

BACKGROUND

Hierarchical B-spline Surfaces

Tensor product B-splines, β-splines and their rational counterparts are widely used for free-form surface creation in computer aided design and animation. One characteristic of this family of surfaces is that subdivision (also referred to as refinement), the mechanism used to add more patches to the surface, is a non-local operation adding either an entire row or column of patches to the surface and thus splitting patches across the surface in places where that split is neither desired nor useful. Furthermore, once refinement has occurred, the local support property of the basis functions restricts the influence of any single control vertex making broad-scale changes to the shape of the surface more difficult.

The hierarchical spline formulation is a multi-resolution approach to the representation and manipulation of free-form surfaces [12] [8] that allows local refinement of a tensor-product surface with the choice of either local or global manipulations of surface shape.

A hierarchical B-spline is constructed from a base surface (*Level* 0) and a series of overlays derived from the immediate parent in the hierarchy. The levels are procedurally related, with each level defined as:

$$W^{\tau}_{i,j} = F^{\tau}_{i,j}(R^{\tau}_{i,j}, O^{\tau}_{i,j}). \tag{5}$$

where the $W^{\tau}_{i,j}$ are the control nodes[1] defining the shape of the level τ overlay, the $R^{\tau}_{i,j}$ are the positions calculated from refinement of parent surface (i.e. the $W^{\tau-1}_{i,j}$), the $O^{\tau}_{i,j}$ are the offsets, and $F^{\tau}_{i,j}$ are the functions that specify how to combine the offset and reference information to form the final position of a control node.

Modification of level $\tau-1$ changes the R^{τ}'s, and thus dynamically the W^{τ}'s. Modifications to the level τ surface are encoded entirely as changes to the O^{τ}s at that level. Modification to lower level overlays (larger τ) will cause fine-scale or local changes to the surface. Modification of high-level overlays (smaller τ) will result in broad-scale or global changes in surface shape.

The function $F^{\tau}_{i,j}$ determines precisely how the surface reacts to modifications to the levels of the hierar-

[1] a control node is distinguished from a control vertex by the addition of the overlay level superscript

chy. This approach has been used to force fine scale details to follow the normal of the parent surface or, in animation, to follow the changes in the pose of an underlying articulated figure [9]. Furthermore, because only the non-zero offsets must be stored, the final surface can have a very compact hierarchical representation. Surfaces composed of 3,000-4,000 bicubic patches have been represented with 500-800 data points [9].

Parameterization

To generate the equations for the surface approximation problem, each data point $D_{q,r} = (x, y, z) \in R^3$ must be associated with a corresponding parametric domain point (u_i, v_j). This process is generally referred to as *parameterization* of the surface. This is a *non-parametric* problem in cases (such as attempting to fit data from a function) where the association is known. It is a *parametric* problem in cases (such as in an approximation of digitized data) where the domain information is unknown and therefore must be estimated.

Parameterization is difficult to automate; different parameterizations of the same data set may result in surfaces that differ significantly in terms of their shape and continuity. However, a number of algorithms exist that have proven effective for univariate and bivariate problems [7][17]. The methods include uniform parameterization, Euclidean (chord-length) parameterization [6], centripetal parameterization [14], affine-invariant chord parameterization and affine-invariant angle parameterization [7].

For data scanned in a regular fashion, such as we are using here, a simple uniform parameterization does exist (namely the row and column indexes), but because of our particular application we cannot use this information directly and must re-parameterize. We have chosen to use a variation of chord length parameterization to concentrate the surface into those regions with a high gradient.

Multigrid Methods for Systems of Linear Equations

Multigrid methods [13] were originally applied to simple boundary value problems posed on spatial domains. Such problems are discretized by choosing a set of grid points in the domain of the problem and forming a system of algebraic equations associated with the chosen grid points. These grid points are then filtered to form multiple grids with different grid spacing. Multigrid methods have evolved as an efficient integrated algorithm for solving a system of algebraic equations with certain properties [2].

Typically each grid has twice the grid spacing of the next finer grid. (there seems to be no advantage in using grid spacings with ratios other than 2.) The domain of a fine grid with spacing h is denoted by Ω^h and the domain for the next coarser grid with spacing $2h$ by Ω^H.

The operator transferring error from the fine grid to the coarse grid is called *restriction*, denoted as I_h^H, and the operator transferring error from the coarse grid to fine the grid is called *prolongation*, denoted as I_H^h.

Full Multigrid V-Cycle

Let the matrix form of our system of linear equations be:

$$\mathbf{A}\mathbf{v} = \mathbf{D} \tag{6}$$

and $\mathbf{u}$ be the approximation to the exact solution $\mathbf{v}$. Then the error $\mathbf{e} = \mathbf{v} - \mathbf{u}$ satisfies the residual equation

$$\mathbf{A}\mathbf{e} = \mathbf{r} = \mathbf{D} - \mathbf{A}\mathbf{u} \tag{7}$$

where $\mathbf{r}$ is the residual.

Instead of starting at the finest level, the *full multigrid V-cycle* starts at the coarsest level and proceeds to the finest level as shown in figure 3. The exact solution calculated for the coarsest grid is interpolated onto the next finer grid as a good initial guess for the solution at that finer level. Compared with the regular multigrid V-cycle, each iteration costs more, but FMG gives a better overall performance because the improved initial guess (from the coarse grid solution) increases the rate of convergence.

The following is an outline of the FMG algorithm:

$$\mathbf{v}^h \leftarrow FMGV^h(\mathbf{u}^h, \mathbf{D}^h)$$

1. IF Ω^h = coarsest grid, THEN go to step 3.
   ELSE $\mathbf{D}^H \leftarrow I_h^H(\mathbf{D}^h - \mathbf{A}^h\mathbf{u}^h)$
   $\mathbf{u}^H \leftarrow 0$
   $\mathbf{u}^H \leftarrow FMGV^H(\mathbf{u}^H, \mathbf{D}^H)$.
2. Correct $\mathbf{u}^h \leftarrow \mathbf{u}^h + I_H^h\mathbf{u}^H$.
3. REPEAT $\mathbf{u}^h \leftarrow MR(\mathbf{u}^h, \mathbf{D}^h)$ ν_0 times.
4. END.

Figure 2: The Full Multigrid V-Cycle

where $FMGV^h$ is the full multigrid operator on the grid level with spacing h, MR is the coarsest level multiple relaxation operator, $\mathbf{v}^h$ is the final solution, $\mathbf{u}^h$ is the approximation to $\mathbf{v}^h$ at the current iteration, $\mathbf{D}^h$

is the right hand side of equation 6 (i.e. the raw data points), and ν_0 is the number of iterations needed to achieve the exact solution at the coarsest level.

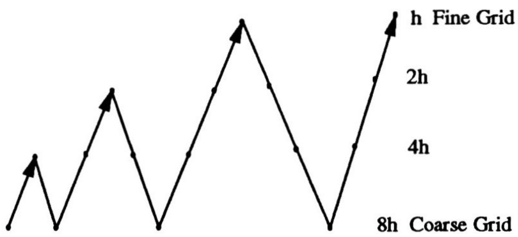

Figure 3: Full Multigrid V-cycle scheme, four level case.

Convergence rate analysis is a difficult area filled with dark corners and unsolved problems [13], but when $\nu_1 = \nu_2 = 1$, there is an error deduction rate of $\frac{1}{4}$ for each sweep. Due to the preliminary cycling through coarser grids, only $O(1)$ V-cycles are needed by the time the algorithm reaches the finest grid. Therefore, the computational cost of the FMG methods is still $O(MN)$.

MULTIGRID METHODS FOR HIERARCHICAL SURFACE APPROXIMATION

Parameterization

Recall that the digitized surfaces we wish to approximate are arranged cylindrically, but that the approach of using splines with a conforming cylindrical topology is inappropriate because of the need to approximate the surface using a portion of a pre-defined surface. Specifically, the approximating spline surface is a rectangular subregion, and it is the perimeter of this region that must approximate the base of the digitized data (i.e. the last row of data). It is also important to more closely approximate those regions of data with important features (eyes, nose, mouth etc.), and to provide more control over those regions for modeling and editing, and in the most efficient manner available.

Parameterization proceeds through a two stage mapping from parametric space onto the data, $(u,v) \rightarrow (q,r) \rightarrow (u,v)$, followed by a deformation of the mapping to concentrate more of the parametric domain into regions of high gradient. With this approach, the data does not map onto the regular grid in parametric space required by the multigrid methods. Instead, the data is resampled by evenly sampling parametric space to find the corresponding location in the digitized data. Linear interpolation is used to generate the sample point value when the mapping fails to correspond to a particular data point.

For the inital parameterization of the bust of Victor Hugo, the data is mapped onto an intermediate deformation space $ST(s,t) \in R^2$ row by row, spreading out from the center to the boundary (Figure 4). The parametric range forms a square which ranges from 0 to $2 \times M$ where M is the number of data rows. The mapping deformation occurs in this intermediate space.

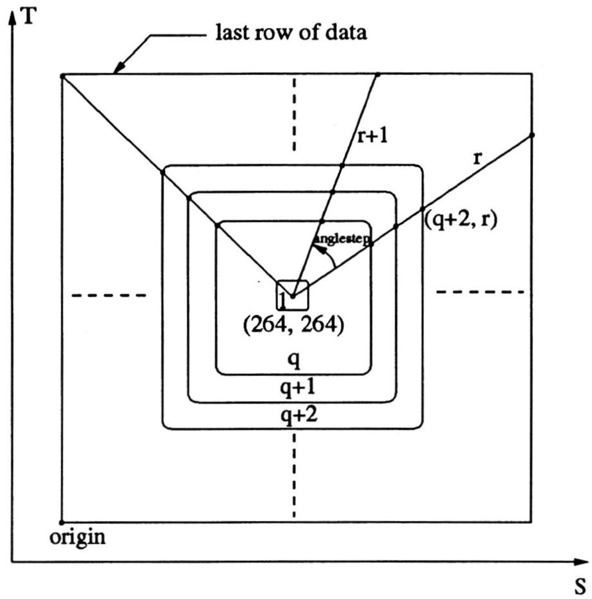

Figure 4: Initial Parameterization for Raw Data

The deformation is defined by a bivariate Bernstein polynomial and corresponds closely to two dimensional image warping [23]. For this particular surface a Bernstein basis of order 1 (corresponding to bilinear interpolation) provides sufficient control over the deformation.

Another appropriate approach to controlling the mapping deformation is presented in [21]. However, as we are more interested for the moment in constructing an approximating hierarchical spline from a given parameterization, we use a simple chord-length parameterization.

Parameterization begins with a single Bézier patch mapping the boundary of UV space to the boundary of ST space and thus to the last row of the data (Figure 5). Then the single patch is split into four sub patches. The control points lying on a boundary are constrained to remain there (V_7), as are those along the axis of symmetry of the data (V_5). Thus there is

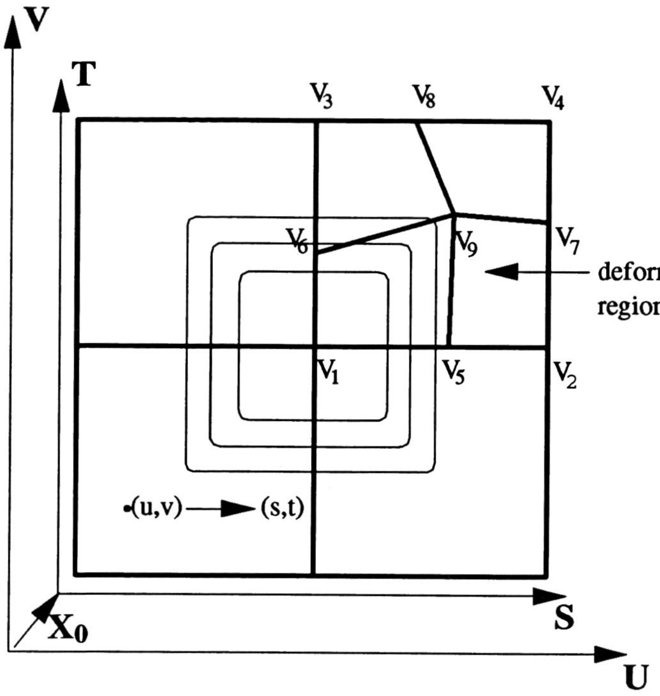

Figure 5: Local Coordinate System UV for deformation

control point displacement after the first subdivision and region S which is quadrilateral $V_1V_2V_4V_3$ is split into four quadrilaterals to meet the parameterization. Control points V_5, V_6 and V_9 which are free to move, are placed such that the gradient sum inside each of the four sub-quadrilaterals is equal. The same deformation is applied to the other three sub-quadrilaterals. The patches are subdivided and deformed until a discrete approximation of the surface area covered by a patch is smaller than a specified value. Plate 1 shows the final deformation of the parametric space with four levels of refinement.

Sampling the Data Points

After the mapping deformation is complete, the data must be sampled to derive the equations for surface approximation. The UV parametric space is evenly sampled on a grid and each (u_i, v_j) mapped directly into ST space to produce the corresponding (s_i, t_j). This point is mapped onto the data using the deformation spline (which is considered continuous), producing a non-integer index pair $(q+\delta_q, r+\delta_r)$. Because neither δ_q nor δ_r is likely to be zero, bilinear interpolation of $D_{q,r}, D_{q,r+1}, D_{q+1,r}$ and $D_{q+1,r+1}$ is used to produce a data point $D_{i,j} = (x, y, z)$ to be used for surface approximation.

Building the Equations

The $m \times n$ grid of data points will be approximated with an $(m-1)\times(n-1)$ array of patches. Given the set of $\gamma = \{(u_i, v_j)\}$ parametric positions sampled on an $h = m \times n$ grid, and the corresponding set of sampled data points $D_{i,j} = \{(x_{i,j}, y_{i,j}, z_{i,j})\}$, a system of $m \times n$ equations (of the form of Equation 2) in $(m+1) \times (n+1)$ unknowns is required. To uniquely determine this system of equations, an additional set of $2m + 2n$ equations are added to set the second derivative at the surface boundary to zero. The corresponding matrix is symmetric, positive definite, diagonally dominant and sparse.

SOLVING THE EQUATIONS

The system of equations defining the approximation is solved using the full multigrid method. Besides its excellent numerical behaviour, the FMG method provides a mechanism to directly create the levels in a hierarchical spline surface. To this end, the *full weighting method* [2] [13] is employed as the restriction operator, I_h^H. This operator gives a good error transfer of residuals from fine grids to coarser grids. The prolongation operator I_H^h employed is standard mid-point subdivision [4]. Refinement is suitable because the surface does not change shape after prolongation which reduces the high frequency error that is often introduced with bilinear interpolation or other prolongation schemes [13]. In our experience, midpoint subdivision requires fewer iterations to converge. Refinement is a particularly suitable operator because it mimics the structure of a hierarchical surface.

BUILDING A HIERARCHICAL B-SPLINE SURFACE

Building the Hierarchy

In solving the system of linear equations, the FMG defines multiple B-spline surfaces at multiple levels of resolution. The control vertices which approximate each level's solution are used to generate the hierarchical surface representation.

The coarsest level of the FMG solution becomes the definition for level 0 in the hierarchy, i.e. the $W_{i,j}^0$. Midpoint subdivision produces the reference points, $R_{i,j}^1$, for level 1. The corresponding offsets $O_{i,j}^1$ are calculated by the following equation:

$$O_{i,j}^1 = V_{i,j}^1 - R_{i,j}^1 \quad (8)$$

where control vertices $V_{i,j}^1$ are from FMG fit result at level 1. Repeated application of this process for each level in the FMG solution produces a hierarchical B-spline surface (Plates 2-8). This hierarchical surface is modifiable at all of the defined overlay levels. When the offsets are defined in terms of the tangent plane of the parent surface, the fine surface details follow any broad-scale changes in the shape of the overall surface.

Zeroing Offsets

Taken directly from the FMG solution, almost all the offsets (the $O^T_{i,j}$) would be non-zero, actually increasing the amount of storage required to define the surface. To reduce this requirement, each offset whose length is within a given ϵ is forced to zero before subdivision and the creation of the next level in the hierarchy. This reduces the storage requirement without altering the fit at the finest level of detail by more than the given tolerance (Figure 6). If a relatively large ϵ is used, a large number of zero offsets are created, but the resulting surface is a much poorer approximation of the data.

Rather than just forcing some offsets to zero, all offsets within a specified area are *smoothed* by reducing the magnitude of the offset (bounded by zero) by a given amount δ (usually δ is compatible with ϵ). Typically this is done at the leaves of the spline hierarchy and essentially "prunes" the tree so that it is the next lower resolution overlay that defines the shape of the surface. The transition between a smoothed region and the surrounding region of higher detail is itself smoothed by modulating the magnitude of the δ value over the given area by a Gaussian function.

RESULTS

The method outline above was tested on the data taken from a bust of Victor Hugo (courtesy F. Schmidt) with 264 rows of 361 data points (95,304 points: Figure 1, and Plate 9) with values ranging from 0 to 200. The value of each data point originally represented the radius from a central axis; however, for approximation this information was converted to points in R^3.

The scanned data was sampled in parametric space with a 257×257 grid (66049 data points) and approximated using the FMG method described above. On a Silicon Graphics Crimson workstation approximately 30 seconds are required to solve the equations.

A 9-level hierarchical surface interpolating all the sampled points of the Victor Hugo dataset contains 90,503 offsets. With a tolerance of 0.1, this number is reduced to 19,339 offsets, about 30% of the sample data set. Smoothing would further reduce storage, but the amount of reduction is highly dependent upon how much smoothing is required, which of course will differ from situation to situation.

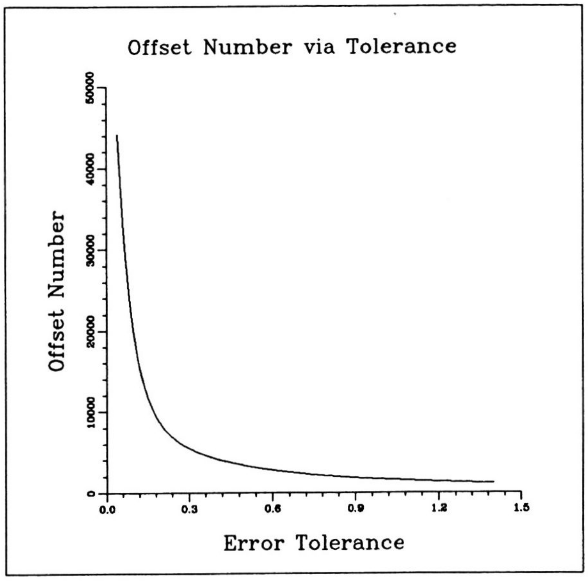

Figure 6: Offset number generated in hierarchical surface with given tolerance value. Raw data ranges from -100 to +100

Plates 10-12 show the results of interactive modification of surface hierarchy to make broad-scale shape changes while retaining fine surface features of the scanned data.

CONCLUSIONS AND FUTURE WORK

The full multigrid V-cycle method provides a fast and stable multi-resolution data fitting scheme that generates surfaces easily convertable into the hierarchical B-spline form. Oscillations still occur when the data has high-amplitude high-frequency regions. Future work will look into better methods of sampling the data for each resolution to reduce the effect of high-frequency components, and at the possibility of globally optimizing the number of nonzero-offsets in the surface definition.

The chosen parameterization scheme, though adequate for this particular application, is not general enough. Other methods, such as used in [5], will be investigated with the goal of using scanned data as a mold that can be applied to an existing surface to define its shape.

REFERENCES

[1] Bartels, R., Beatty, J. and Barsky, B. *An Introduction to Splines for Use in Computer Graphics and*

Geometric Modelling, Morgan Kaufmann Publishers Inc., Palo Alto, California, 1987.

[2] Briggs, W. *A Multigrid Tutorial*, SIAM, 1987.

[3] Birkhoff, C. and MacLane, S., *A Survey of Modern Algebra*, The Macmillan Company, New York, 1965.

[4] Cohen, E., Lyche, T., and Reidenfeld, R. "Discrete B-splines and Subdivision Techniques in Computer Aided Geometric Design and Computer Graphics", *Computer Graphics and Image Processing*, 14(2):87-111.

[5] Thingvold, J. and Cohen E., "Physical Modeling with B-spline Surfaces for Interactive Design and Animation", *Computer Graphics* 24(2):129-137, 1990.

[6] Farin, G *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press Inc., 1988.

[7] Foley, T. and Nielson G., "Knot Selection for Parametric Spline Interpolation", *Mathematical Methods in Computer Aided Geometric Design*:261-271, edited by T. Lyche and L. Schumaker, Academic Press, Boston, 1989.

[8] Forsey, D. *Motion Control and Surface Modeling of Articulated Figures in Computer Animation*, Univ. of Waterloo Technical Report CS-90-28, (Ph.D. Thesis) 1990.

[9] Forsey, D. *A Surface Model for Skeleton-Based Character Animation*, Second Eurographics Workshop on Animation and Simulation, Vienna, Sept. 1991.

[10] Forsey, D. and Bartels, R. "Tensor Products and Hierarchical Fitting", *Proceedings IEEE Curves and Surfaces in Computer Vision and Graphics II*, Boston, Nov. 1991.

[11] Forsey, D. and Bartels, R. "Surface Fitting with Hierarchical Splines" to appear in Transactions on Graphics.

[12] Forsey, D., and Bartels, R. H., "Hierarchical B-spline Refinement", *Proceedings of SIGGRAPH '88*:205-212, Atlanta, Georgia, 1988.

[13] Hackbusch, W., *Multigrid Methods and Applications*, Springer-Verlag, 1985.

[14] Lee, E., "On Choosing Nodes in Parametric curve Interpolation", *SIAM Conference on Applied Geometry*, Albany, New York, 1985.

[15] Lyche, T. and Morken, V., "Knot removal for parametric B-spline curves and surfaces", *Computer Aided Geometric Design*, 4(3):217-230, 1987.

[16] Nahas, M., Huitric, H. and Saintourens, M., "Animation of a B-Spline Figure", *The Visual Computer*, 3(4), 1987.

[17] Nielson, G. and Foley, T., "A Survey of Applications of an Affine Invariant Norm", *Mathematical Methods in Computer Aided Geometric Design*:445-467, edited by T. Lyche and L. Schumaker, Academic Press, Boston, 1989.

[18] Schmitt, F., Barsky, B. and Du, W. "An Adaptive Subdivision Method for Surface-Fitting from Sampled Data", *Proceedings of SIGGRAPH '86*:179-188, Dallas, Texas, 1986.

[19] Screckovic, M., *Adaptive Hierarchical Fitting of Curves and Surfaces*, Master's Thesis, University of Waterloo, 1991.

[20] Sederberg, T. and Parry, S. "Free-Form Deformation of Solid Geometric Models", *Proceedings of SIGGRAPH '86*:151-160, Dallas, Texas, 1986.

[21] Waters, K. and Terzopoulos, D., "Modelling and Animating Faces Using Scanned Data", *The Journal of Visualization and Computer Animation*, 2:123-128, 1991.

[22] Welch, W. and Witkin, A. "Variational Surface Modelling", *Proceedings of SIGGRAPH '92*:157-166, Chicago, Ill, 1992.

[23] Wolberg, G., *Digital Image Warping*, IEEE Computer Society Press, 1990.

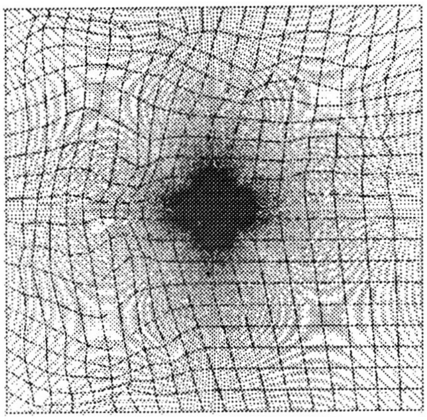

Plate 1: Deformed parametric space.

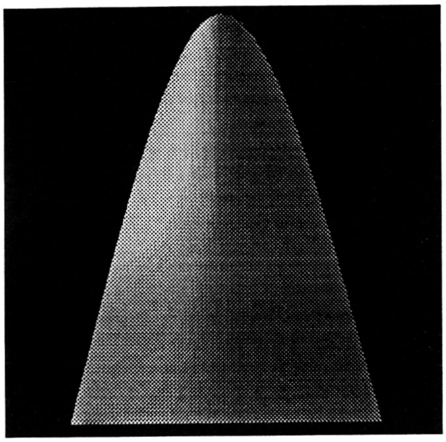

Plate 2: Surface fit at level 2.

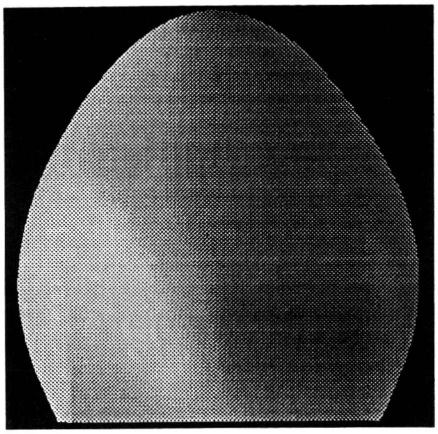

Plate 3: Surface fit at level 3.

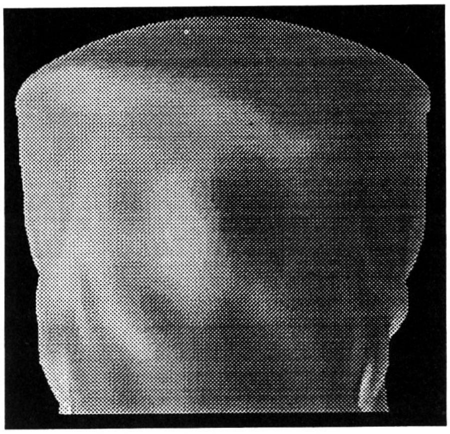

Plate 4: Surface fit at level 5.

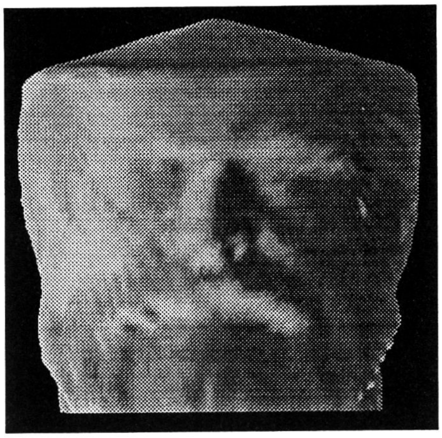

Plate 5: Surface fit at level 7.

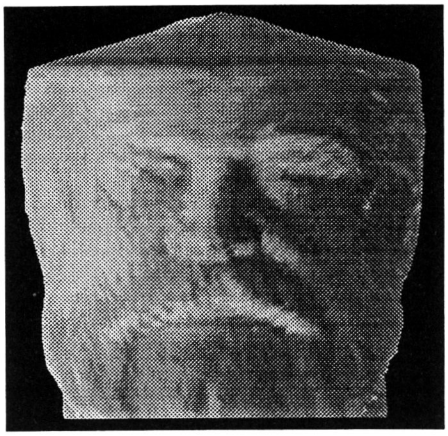

Plate 6: Surface fit at level 8.

Plate 7: Surface fit at level 9.

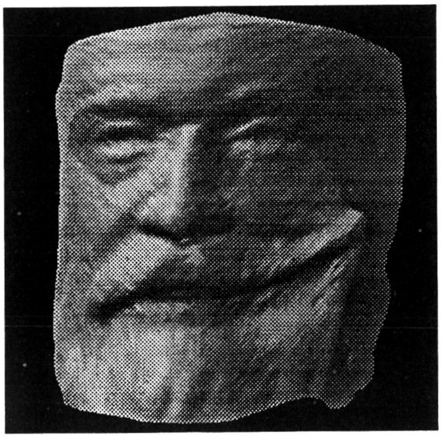

Plate 7: Smile 1.

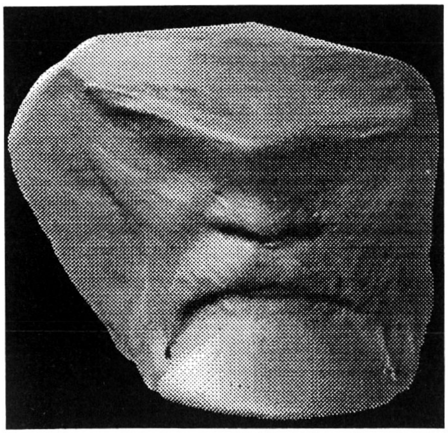

Plate 8: Neanderthal man.