Automatic terrain modeling using transfinite element analysis

N.O. Collier, V.M. Calo

4700 King Abdullah University of Science and Technology
Building UN1550-4, Office 4333
Thuwal 23955-6900, Kingdom of Saudi Arabia

Abstract

An automatic procedure for modeling terrain is developed based on $L_2$ projection-based interpolation of discrete terrain data onto transfinite function spaces. The function space is refined automatically by the use of image processing techniques to detect regions of high error and the flexibility of the transfinite interpolation to add degrees of freedom to these areas. Examples are shown of a section of the Palo Duro Canyon in northern Texas.

Keywords: transfinite interpolation, attribute-based modeling, terrain modeling, automatic refinement

1. Introduction

This paper is focused on the use of a surface modeling technique called Attribute-Based Modeling (AB Modeling) [1, 2] which is a generalization of transfinite interpolation. Transfinite interpolation has its beginning in 1973 where the term was coined by Gordon [3] where its use was to generate curvilinear grids in two- and three-dimensions. The method blends the interior of a surface or volume by information from the boundaries. This is done in such a way that the boundary curves or surfaces are interpolated exactly.

Gao and Rockwood’s generalization of this idea extends the notion of transfinite interpolation to allow contribution from unconnected interior curves as well as boundary curves. In fact, the construction does not strictly need boundary curves. Interior curves may be added in regions where more control is desired. This allows for an intuitive design of surfaces, adding feature curves in areas where more control is desired without regard for the topology of the resulting parameterization.

While the notion of design of terrain is an interesting topic, this paper proposes a technique for automatically representing terrain data whose source is elevation. The proposed method will determine locations of curves based on a projective error, specifically in this work the $L_2$ residual. Since the surface will conform to the curves, a one-dimensional regression analysis may be used to improve curve quality in target areas. This computation of the projections requires AB Modeling to be re-expressed as a sum of basis functions which we will detail here.

There are several techniques for the modeling of terrain, some of which are detailed in [4]. If the points are regularly sampled, then a regular grid can be used to simply obtain a surface approximation. A Delaunay triangulation [5] can be used to obtain a surface of irregularly spaced points. When more resolution is desired, subdivision surfaces

Email addresses: nathaniel.collier@kaust.edu.sa (N.O. Collier), victor.cal0@kaust.edu.sa (V.M. Calo)
[6, 7] may be used to obtain smooth approximations based on the original grid. Alternatively, the grid data may also be used to develop a series of contour lines common to topographical maps. There are also examples of the use of polynomials and B-splines to approximation the terrain data. Walton [8] uses curved knot lines to adapt tensor product B-splines to be more suitable to model the sharp discontinuities that often are present in modeling terrain.

2. Transfinite Basis Function Construction

To illustrate how the parameterizations are constructed, consider Fig. 1. On the left side of the figure there are two lines and a point which are called footprints, labeled by numbers inside of circles. Each footprint is the parametric preimage that is mapped to image or physical space by some function ($f_1$, $f_2$, and $f_3$ in this case). The mapped footprints are shown on the right and labeled with numbers in squares. Note that a footprint can be any entity of equal or lower dimension relative to the dimension of the parameter space being used (e.g. points or lines in $\mathbb{R}^2$). The surface can then be parameterized as

$$F(u) = f_1(u_1) \frac{1}{d_1} + f_2(u_2) \frac{1}{d_2} + f_3(u_3) \frac{1}{d_3}$$

(1)

where each of the footprint’s mapping is blended by a rational function of reciprocal distances. In [1] this technique is demonstrated to be equivalent to a weighted least squares procedure, which is what leads to minimum potential energy surfaces.

The procedure involves finding some point on each footprint, $u_i$, and a distance, $d_i$, to the point at which we wish to evaluate the surface, $u$. Then we find the image of each of these points, $f_i(u_i)$, where the subscript $i$ refers to the $i^{th}$ footprint. Each of these images are weighted by the reciprocal distance to the footprint and divided by the sum. This blends the mapped curves in such a way that a partition of unity is maintained. We will define the reciprocal of the distance to the footprint as a variable, $W_i := \frac{1}{d_i}$. Therefore we can rewrite the interpolation as

$$F(u) = \sum_{i}^{n_f} f_i(u_i) \widehat{W}_i(d)$$

(2)

where

$$\widehat{W}_i(d) := \frac{W_i}{\sum_j W_j}$$

(3)

$n_f$ is the number of footprints, and $d$ is a vector of all the distances to all the footprints.
Now if we assume that each curve can be parametrized as

\[ f_i(u) = \sum_{k} B_{i,k}(u)C_{i,k} \]  

(4)

where \( k \) loops over the number of bases of the curve, \( n_b \) and the functions \( B_{i,k}(u) \) are the \( k^{th} \) one-dimensional basis of the \( i^{th} \) footprint with degree of freedom, \( C_{i,k} \). Then the entire surface interpolation will become

\[ F(u) = \sum_{i} \sum_{k} B_{i,k}(u)\hat{W}_i(d)C_{i,k} \]  

(5)

or if we take an index \( j \) to loop over unique combinations of the indices \( i \) and \( k \),

\[ F(u) = \sum_{j} \left( B_j(u)\hat{W}_i(d) \right) C_j \]  

(6)

So the basis function of the \( i^{th} \) footprint and \( k^{th} \) local curve basis can be thought of as the curve basis extended to the whole domain via a weighting function, that is

\[ N_j(u) = B_{i,k}^j(u)\hat{W}_i \]

(7)

This enables the construction of bases on connected or unconnected networks of footprints which can be entities of lower dimensions (i.e. lines and points in two-dimensions).

2.1. Derivative Control

The surface as it crosses the footprint may be controlled by expanding the function \( f \) in (2) into a combination of the footprint curve \( f \) and a lofted curve \( g \), extending the equation

\[ F(u) = \sum_{i} \left[ (1-s_i)f_i(u) + (s_i)g_i(u) \right] \hat{W}_i(d) \]  

(8)

where in this case the lofted curve is blended linearly with the footprint curve. These lofted curves are called ribbons in the literature [1]. The blending variable \( s_i \) is a scaled distance, where the scaling is selected such that \( s_i < 1 \) for the whole domain. This involves a computation to determine the point in the domain furthest from the footprint and scaling the distance \( d_i \) that is already computed for the weighting function \( \hat{W}_i(d) \).

While here the footprint is blended with the ribbon via linear functions of \( s \), richer functions can be used to enrich the surface interpolation further from the footprint. In particular, the construction used in this work utilizes the first two terms of the quadratic Bernstein polynomials,

\[ F(u) = \sum_{i} \left[ (1-s_i)^2f_i(u) + s_i(1-s_i)g_i(u) \right] \hat{W}_i(d) \]  

(9)

2.2. Sample Basis Functions

This method of constructing basis functions can be used on footprint networks which form polygons by enforcing continuity at common points on joined footprints. These are types of functions typically seen in engineering applications. However, the freedom of topology which transfinite interpolation brings allows the generation of basis functions for \( n \)-sided polygons, a series of which may be shown in Fig. 2.

However more general networks may also be chosen, where footprints may or may not be connected. This is the feature which is exploited here to model terrain, where degrees of freedom may be added at will in areas where refinement is desired. Consider the following square domain with an added line along the diagonal, shown in Fig. 3. In this example, each footprint is a cubic Bezier curve whose influence is extended to the two-dimensional domain via transfinite interpolation. The resulting basis for the bottom and interior footprints are shown. Note that the this basis will interpolate the included curve with \( C^0 \) continuity.
Figure 2: Plots of single basis functions for footprint networks which form convex polygons. In this case each side is a linear Bezier curve, however the freedom is present to allow each side to be of different order.

3. Terrain Modeling Technique

The general notion of the method developed to model terrain is to determine unknown degrees of freedom via $L_2$ projections from discrete terrain data onto the transfinite function spaces used. This is expressed as

$$\arg\min_c \left( \int_\Omega (Z - \sum_j N_j(u)c_j)^2 d\Omega \right)$$

where the transfinite interpolation is shown as a sum of basis functions and nodal degrees of freedom, $N_j$ and $c_j$, respectively, and where $Z$ represents the elevation data. To find the minimum, the derivative is taken with respect to each parameter, $c_j$ and set to zero.

$$\int_\Omega 2\left( Z - \sum_j N_j(u)c_j \right)(-N_i(u)) d\Omega = 0$$

This results in the following matrix equations.

$$M \cdot c = f$$

where $M$ is the mass matrix and whose $(a, b)^{th}$ entry is computed by

$$M_{a,b} = \int_\Omega N_a(u)N_b(u)d\Omega$$

$f$ is called the load vector whose $a^{th}$ entry is given by

$$f_a = \int_\Omega Z \cdot N_a(u)d\Omega$$

and where $c$ is the vector of unknown degrees of freedom. In this work, the integrals will be approximated by cell integration. Each pixel is taken as a small square area of the domain which we multiply by the integrand evaluated at the cell center. The overall effect is that the degrees of freedom are chosen that generate the closest surface to the input terrain as possible, in the $L_2$ sense. Note that the process is a continuous version of a least squares fit using discrete points, discussed in more detail in Chapter 2 of [9].

3.1. Refinement of Footprints

The method features footprint insertion and automatic refinement. This is realized by first inserting the footprints in the desired area in the domain. For boundary footprints this is trivial and for future interior footprints a method will be shown in Sec. 3.2. In this work, each footprint is parameterized as a B-spline [10] and initially a single span linear B-spline is used. Knots are subsequently inserted by splitting the span with the highest $L_2$ residual. This is done while maintaining a minimum feature length to avoid small features which tend to add ripples to the surface interpolations. The implementation takes a user-specified tolerance for the residual which enables a uniform refinement of footprint curves.

As a test case, elevation data was interpolated from the Palo Duro Canyon [11] using only boundary footprints and refining each individually. Also, derivative control is added to each footprint which gives extra control to the interior. The original data and approximation can be seen in Figs. 4 and 5.

1http://sketchup.google.com/3dwarehouse/details?mid=edb44c07a55682f121fafe1eef0f95e
Figure 3: Example footprint network and basis functions used to interpolate terrain data where red indicates maximum values and blue represents zero.
Figure 4: Palo Duro Canyon data and interpolation using individually refined boundary footprints

Figure 5: Surface plots of boundary footprint representation of Palo Duro Canyon. Coloring indicates source terrain elevation values.
3.2. Insertion of Footprints

Transfinite Element Analysis is a powerful tool in interpolation not only because footprints forming convex polygons can be independently refined and blended but also because the footprints themselves can be added in any location. An interpolation can be improved by adding footprints in regions where the approximation is poor, economizing the use of degrees of freedom.

The following procedure is a method by which footprints are inserted into the domain.

1. The absolute error between the terrain data and its interpolation is computed (Fig. 6a) and treated like an image. This error is thresholded at a user-defined value to highlight regions of large error.
2. Each disjoint thresholded region is then uniquely identified and treated as a separate domain (Fig. 6b). The small regions (less than 5 pixels) are removed from consideration (Fig. 6c). For each region, compute the $L_2$ residual and select the region with the highest value. This aims at targeting not just the maximum error, but the region which is most poorly represented.
3. For this targeted region, place a linear footprint which is a best fit to the pixels involved in that region (Fig. 6d).
4. Recalculate the point error and continue the procedure

This process can be continued until some criteria is reached (e.g. maximum number of footprints reached, global residual level achieved).

In the above boundary footprint representations (Fig. 5), this procedure is used to add ten footprints to key areas of the surface. This resulting surface can be seen in Fig. 7. While the representation is improved in the sense that the error was reduced, there are now lines of $C^0$ continuity which are not desirable from the aesthetic point of view.

Figure 6: Steps of the footprint insertion process. Each row corresponds to the first, fifth, and tenth insertion
4. Conclusions

This paper presents a procedure for automatically adapting transfinitely interpolated function spaces to approximate terrain, economizing the use of degrees of freedom by adding the control to regions determined as poorly represented. While certain visual artifacts are present, the method is successful at adapting to represent the terrain.

The method is an effective compression technique for representing abruptly varying data, such as permeabilities in reservoir simulations. Alternative metrics for computing the optimality of the approximation may be used, such as, $H^1$, where both the function values and its gradients may be used to determine which are the best coefficients to represent the terrain.

The results presented are the first attempt of a larger effort to use this method to automatically model terrain from discrete data. The method of constructing function spaces provides the modeler with a lot of freedom, yet the challenge is to determine how to use this freedom optimally. This is currently a direction being pursued in this research effort.

5. References