PROCEDURAL RECONSTRUCTION OF NURBS SURFACES

TSUNG-HSIEN WANG
Carnegie Mellon University
tsunghsw@cmu.edu

Abstract. A potential way to bridge the gap between complex form generation and models for physical manifestations is to panelize NURBS surfaces with polygonal faces. This paper investigates the transition from NURBS surface to a mesh solid through a procedural modeling approach, in the process, illustrating how a discretized planar surface can be reconstructed for form generation and further exploration. The paper promotes this approach as an efficient way to modeling complex forms using an example drawn from real life architecture to demonstrate a generative process with customized restructuring.

Keywords. Rule-based; surface reconstruction; procedural modelling; architectural exploration.

1. Introduction

Parametric modeling is bound to constraints that regulate geometry. Kilian (2006) takes a bidirectional perspective to exemplify the potential of modeling design with associative constraints. Moustapha (2005) proposes a formal expression to describe the transformative and recursive nature of constraint applications for form exploration. In general, these constraint-based approaches associate geometric components with prescriptive parameters. However, modeling a sophisticated form, such as a freeform solid, demands a technique typically of the kind derived from Non-Uniform Rational B-Splines (NURBS). For purposes of construction or fabrication, the NURBS surface is usually reconstructed as a triangular, quadrilateral or, in general, a multilateral polygonal mesh. The challenge of restructuring a NURBS surface into a meshed object requires processing the underlying geometry.

In this paper, we illustrate a procedural approach to reconstructing NURBS surfaces using a parametric subdivision scheme. The scheme encompasses formal expressions and operations on nodes to generate subdivision meshes, and also, on organizing successive meshes to form the ultimate surface structure.

The focus in this particular work is to make the subdivision scheme customizable to users for further geometric exploration. Ultimately, the goal is to extend this approach to architectural exploration, to make the modeling process generative and more flexible. We demonstrate this using a modeled surface, inspired by the work of Santiago Calatrava, on which the resulting generative reconstructions are illustrated.

1.1. PROBLEM STATEMENT

The problem is to create a surface structure, a mesh solid, from a given NURBS surface with zero depth. We propose a grammatically based procedural modeling approach coupled with a formal re-meshing process. There are precedents for both.

One of the earliest and better known formalisms is L-system developed by Aristid Lindenmayer in 1968, which has been successfully applied to plant modeling and visualization (Prusinkiewicz and Lindenmeyer, 1990). Basically, it is a string grammar, with rules specifying symbols, variables and transformations. The growth process of plants is simulated by recursively applying rules. More recently, Muller et al (2006) extend another grammar formalism, shape grammars (Stiny, 2006), to a technique aimed at urban modeling. In their approach, buildings are created grammatically, through representations from volumetric mass to building facades, so as to mimic the complex environment of a city.

Recently, there has been interest in employing re-meshing techniques for architectural applications to support complex form generation. In particular, quad meshing has been proposed for structural modeling (Pottmann et al, 2006). Liu et al (2006) developed PQ meshes, which are quad meshes with planar faces, to discretize curvilinear surfaces. A re-meshing application by Culter and Whiting (2007) addresses issues of fabrication and topological coherence via a clustering algorithm, and interactions between designer and the re-meshing process. Akleman et al (2004, 2005) implemented a topological mesh modeler to support innovative sculpture generations. A common theme underlying these various developments is the reconstruction of the original mesh, by certain geometrical operations, to propagate more sophisticated forms.

2. A Procedural Approach for Reconstructing Architectural Geometry

The general workflow for reconstruction starts by discretizing input NURBS surfaces as quadrilateral meshes. This first stage serves as the basis for subsequent geometric reconstruction and is controlled by the specified number of UV quadrilaterals. Each UV quadrilateral is a UV quad face derived by interpolating UV intervals with the given NURBS surface. The subsequent reconstruction scheme is defined by a number of parameters that reflect the inter-relationships from predecessor to successor shapes. The reconstruction stage is interactive, where, potentially, designers sketch out their intentions on structuring the underlying geometry. Lastly, the reconstruction procedures are applied to the discretized surface derived from first stage to produce alternative designs.

2.1. RECONSTRUCTION SCHEME

For the remainder of this paper, we assume that we are working with quadrilateral meshes. The following process is then basic: each quadrilateral face of a given discretized quad mesh surface is potentially replaced by a number of alternative quadrilateral faces. This process is dominated by a modeling procedure that specifies a hierarchical relationship from predecessor to successors. The modeling procedure, namely, a subdivision scheme with customized tuning parameters, is the reconstruction scheme.

Notation: Each quadrilateral face is represented as: face:$\{v_1, v_2, v_3, v_4\}$, where "face" represents the quadrilateral mesh and v_1, v_2, v_3, v_4 are the mesh vertices representing the face in counter-clockwise order.

Figure 1 illustrates three different quad faces (shaded areas). The original input, face:$\{A, B, C, D\}$, is shown on the left hand side. Face:$\{T_1, B, T_2, D\}$ and faceSet(2):$\{\{A, B, T_1, D\}, \{B, C, D, T_2\}\}$ are alternative meshed faces that can be produced. The transition from one face (the input mesh quad) to a single face or multiple faces specifies a reconstruction scheme. Note that the face can be either convex or concave.

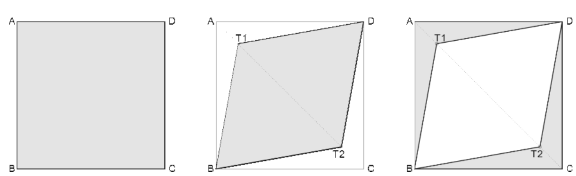


Figure 1. (Left) face:{A, B, C, D}; (Middle) face:{T_1, B, T_2, D}; (Right) faceSet(2):{{A, B, T_1, D}, {B, C, D, T_2}}

As an example, consider the reconstruction from *face*:$\{A, B, C, D\}$ to *face*:$\{T_1, B, T_2, D\}$, which represents a replacement, where nodes A and C are replaced by T_1 and T_2 respectively. To achieve this, two simple rules, the vertex and face rules are defined. The vertex rule creates a new vertex, and the face rule replaces the original face by one or more new faces.

2.1.1. Vertex Rule

The vertex rule, given by (1), takes, as the input, three parameters, vertices v_1 and v_2 and weight w and creates a new vertex, *newV*, as a linear combination of vertices v_1 and v_2, specified by the weight, w.

$$\text{newV}\ (v_1, v_2, w)\text{: } v_1 * w + v_2 * (1 - w) \tag{1}$$

2.1.2. Face Rule

This rule specifies the replacement of a quad face by a list of quad faces and takes by the following general form:

face:{vertices…} $\longrightarrow$

faceSet(num):{ face:{ vertices …}, …, face:{vertices …}}

Here the left most "face" is the quad face that is to be replaced by "faceSet(num)," a list of quad faces. "num" is the number of faces in this set. For example, consider the rule

face:{A, B, C, D} $\longrightarrow$

faceSet(3):{face:{A, B, T_1, D}, face:{B, T_2, D, T_1}, face:{C, D, T_2, B}} (2)

The original face, {A, B, C, D}, is replaced by three successive faces, {A, B, T_1, D}, {B, T_2, D, T_1}, {C, D, T_2, B}, by newly defined vertices T_1 and T_2, which are derived from vertices A and C. We can combine the two rules and reconstruct the face. Applying rule (2) to face:{A, B, C, D} generates the set, {face:{A, B, T_1, D}, face:{B, T_2, D, T_1}, face:{C, D, T_2, B}}, shown in the middle of Figure 2.

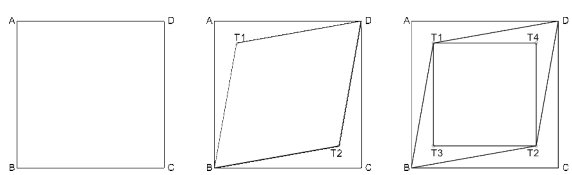


Figure 2. (Left) face:{A, B, C, D};
(Middle) faceSet(3):{{face:{A, B, T_1, D}, face:{ B, T_2, D, T_1}, face:{C, D, T_2, B}}};
(Right) faceSet(5):{ face:{A, B, T_1, D}, face:{B, T_2, T_3, T_1}, face:{T_2, T_4, T_1, T_3}, face:{D, T_1, T_4, T_3}, face:{C, D, T_2, B}}

If we then apply the same rule to face:{B, T_2, D, T_1} we generate three more faces, {face:{B, T_2, T_3, T_1}, face:{T_2, T_4, T_1, T_3}, face:{D, T_1, T_4, T_3}}. After two iterations, face:{A, B, C, D} has been replaced by five faces, face:{A, B, T_1, D}, face:{B, T_2, T_3, T_1}, face:{T_2, T_4, T_1, T_3}, face:{D, T_1, T_4, T_3}, and face:{C, D, T_2, B}, shown on the right of Figure 2.

2.1.3. Fenestration Function

Using the two rules above, we have a subdivision scheme for face reconstruction. We can add a function to control the visibility of every face so as to create openings from them. For example, consider the notation:

Fenes(Num):{Boolean, … , Boolean}

"*Fenes*" is the function identifier with "*Num*" number of Boolean values contained in the list. Each Boolean value maps to a corresponding face. The visibility of each face is turned on or off, according its value, 1 or 0, in the list. As Figure 3 illustrates, two different fenestration rules applied to the same set of faces generate different configurations.

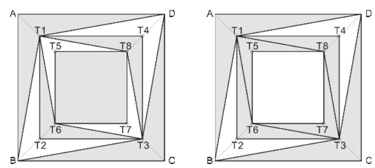


Figure 3. (Left) Fenes(9):{1, 0, 1, 0, 0, 1, 0 , 1, 1}; (Right) Fenes(9):{1, 0, 1, 0, 1, 1, 1, 1, 0}; Shaded regions are solid faces, others are empty.

By varying the visibility of each polygon face, we can generate alternative surface structures with ease. Moreover, surface planarity, under mesh reconstruction, is maintained to be the same as the original. By doing so, the topological structure of the surface remains unchanged and only regional geometrical details are added. Notwithstanding, the outcomes derived from the reconstruction schemes can make interesting changes to the final appearance.

2.2. POST PROCESSING: SOLID GEOMETRY EXTRUSION

After reconstructing the input surface, a post-processing procedure is suggested. This process involves a mesh offset operation on the polygonal faces and makes the surface with zero depth become a more realistic solid artifact, a panelized surface structure with thickness. We can create offset meshes along each vertex normal by a given distance "D", as shown in the Figure 4. In this way, a solid volumetric surface, or structural components, such as surface frames, can be created and meanwhile maintain the original geometry topology.

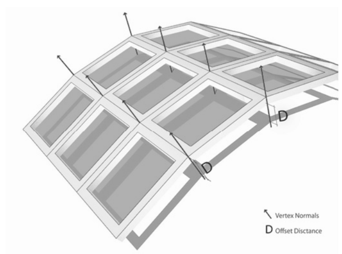


Figure 4. Offsetting the surface by a given distance D.

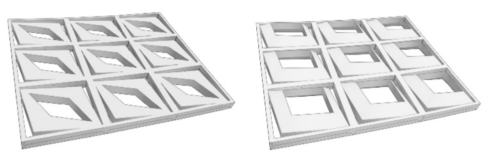
Despite its relative simplicity, this step can make a surface appear more like a piece of real architecture, as shown in the Figure 8.

2.2.1. Vertex Rule with Linear Vertex Normal Transformation

Except the vertex rule given in section 2.1.1, a vertex can be further modified along the vertex normal, which is calculated as a new vertex is generated. For example, consider the rule notation

$$newV_i\ (V_i, N_i, d): V_i + N_i * d$$

The original vertex, V_i, is transformed along the vertex normal direction, N_i, by a given distance, d, to derive a new vertex, $newV_i$. This rule is helpful while the generative process is extended to three-dimensional operations. Figure 5 shows the examples that vertex normal transformation is applied to generate patterns with 3D variations.



(a) (b)

Figure 5. (a) Reconstructed surface without applying vertex normal transformation; (b) Same surface with an additional vertex normal transformation on inner vertices

3. A Surface Reconstruction Example

An architecture example taken from one of Santiago Calatrava’s projects in 2000 was remodeled. The project, Windery Complex for the Bodega & Bebidas Group, features a curved roof composed of elementary rectangular tubes, as shown in Figure 6. We remodeled the curved surface of the roof, as an illustration, to explore potential variations using the procedural reconstructing schemes presented in this paper.

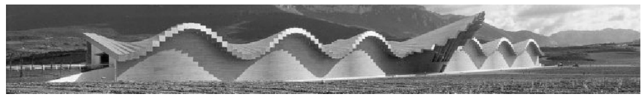
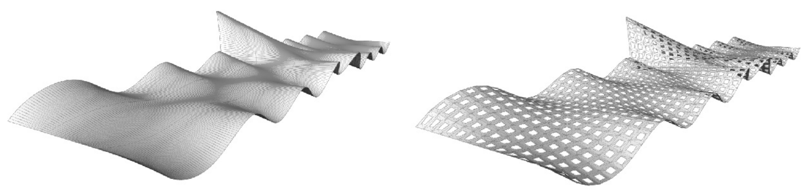


Figure 6: Image after Windery Complex for the Bodega & Bebidas Group by Santiago Calatrava, 2000

The initiated surface, created using NURBS, is illustrated in Figure 7a. Although NURBS model complex freeform objects as mathematical representations, for the practical purposes, such as manufacture and fabrication, NURBS-based objects are usually panelized. Motivated thus, we discretized the surface as quadrilateral meshes. Other mesh types are possible, for instance, triangular meshes possess the same kinds of geometric information, namely, nodes, edges, planes etc. However, in this work, we considered only quad faces. The number of quad meshes to replace the original NURBS surface controls the discretization. For this experiment, this number was set to 1000.



(a) (b)

Figure 7: a) The original NURBS surface; b) Surface derived from the reconstruction scheme comprising face and fenestration rules: (1) face:{A, B, C, D}$\longrightarrow$ faceSet(5):{face:{A, B, T_1, D}, face:{ B, T_2, T_3, T_1}, face:{T_2, T_4, T_1, T_3}, face:{D, T_1, T_4, T_3}, face:{C, D, T_2, B}}; and (2) Fenes(5):{1,1,0,1,1}

Initially, we applied the reconstruction scheme without any post face-offset operation and generated the final structure, shown in Figure 7b. The reconstruction scheme applied here is the second iteration shown in Figure 2, in which the original face is replaced by the set with five faces. Also, the fenestration is represented as Fenes(5):{1, 1, 0, 1, 1}.

For the last step, we took the reconstruction scheme illustrated on the right in Figure 3, which has a total of 9 faces in the set and we assigned an arbitrary distance for the face offset operation. The generated structure is shown in Figure 8. This reconstructed mesh solid demonstrates how a NURBS surface is faceted with a customized pattern. It should be noted, for purposes of implementation, that during the generative process, the input NURBS surface, its discretized meshed surfaces, and its final structured mesh solid are linked in a directed graph. Associative geometric relations between different generations are maintained through intermediate data nodes. In this way, up-to-date information can be maintained as changes occur.

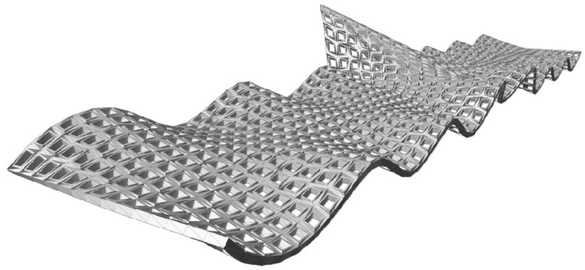


Figure 8. Surface derived from the reconstruction scheme and a post face-offset. Reconstruction scheme comprises the following face and fenestration rules: (1) face:{A, B, C, D} $\rightarrow$ faceSet(9):{face:{A, B, T_1, D}, face:{B, T_2, T_3, T_1}, face:{T_2, T_3, T_6, T_1}, face:{T_3, T_8, T_7, T_6}, face:{T_8, T_5, T_6, T_7}, face:{T_1, T_6, T_5, T_8}, face:{T_4, T_1, T_8, T_3}, face:{D, T_1, T_4, T_3}, face:{C, D, T_2, B}}; and (2) Fenes(9):{1, 0, 1, 1, 0, 1, 1, 0, 1}

In Figure 9, we demonstrate two variations using the schemes illustrated in Figures 5a and 5b. These patterns are generated, by tuning the parameter values associated with each specified reconstruction scheme. Obviously, the generated geometry can be very different, albeit, with the same parametric scheme, and this feature provides users with flexibility for architectural exploration.

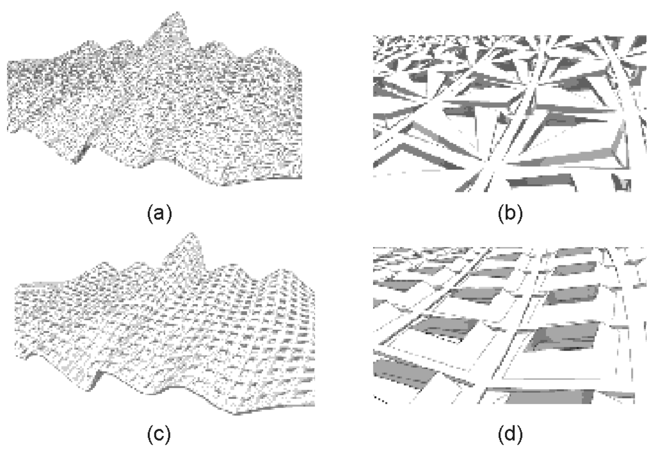


Figure 9: a, b) Surface derived from the reconstruction scheme shown in Figure 5a with an additional normal offsetting rule for vertex manipulation in 3D space; c, d) Surface derived from the reconstruction scheme shown in Figure 5b.

4. Discussion and Conclusion

Our current work focuses on formalizing an approach to geometry reconstruction. The example here uses an input NURBS surface with simple

UV discretization. We are currently experimenting with a similar reconstruction scheme that will extend the boundary of a single module and make the entire surface as an integrated whole. Our ultimate objective is to extend the generative process from surface initiation to detail reconstruction. This will benefit the representation of the entire design model using a constrained associative network and render generative control to designers for further formal exploration. Currently, the system is executed through scripting commands. We have plans to provide a graphical user interface to support visual and more responsive interactions for designing reconstruction schemes.

Admittedly, the approach in this paper is not without criticism from the perspective of real designs. For instance, it lacks any consideration of structural, material, or performance aspects. However, we expect that once an underlying geometry can be processed and organized coherently, other computational tasks become easier to handle. Another is that the approach is procedural, that is, programming-oriented, and this may distract designers from their initial design activity. Yet, the knowledge and ability to control modeling procedures and parameters must be regarded as the foundation by means of which the computational requirements of subsequent design applications can be addressed.

Acknowledgements

I wish to thank Professors Ramesh Krishnamurti and Kenji Shimada for their insightful comments and inspirational discussions. I would also like to thank my colleagues in Computational Design for their encouragement and support.

References

Akleman, E., Srinivasan, V., and Mandal, E.: 2005, Remeshing schemes for semi-regular tilings. *In Proceedings of the International Conference on Shape Modeling and Applications*, pp. 44–50.

Akleman, E., Srinivasan, V., Mandal, E., Chen, J., Melek, Z., and Lendreneau, E.: 2004, Topmod: topological mesh modeling system. Technical Report, Texas A&M University. Department of Architecture.

Culter, B. and Whiting, E.: 2007, Constrained planar remeshing for architecture. *Proceedings of Graphics Interface 2007*, pp. 11–18.

Kilian, A.: 2006, Design Exploration through Bidirectional Modeling of Constraints. PhD thesis, Massachusetts Institute of Technology.

Liu, Y., Pottmann, H., Wallner, J., Yang, Y.-L., and Wang, W.: 2006, Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics*, 25(3), 681–689.

Moustapha, H.: 2005, Architectural Explorations: A Formal Representation for the Generation and Transformation of Design Geometry. PhD Thesis, Carnegie Mellon University.

Muller, P., Wonka, P., Haegler, S., Ulmer, A., and Gool, L.: 2006, Procedural modeling of buildings. *Proceedings of SIGGRAPH 2006*, pp. 614–623.

Pottmann, H., Brell-Cokcan, S., and Wallner, J.: 2006, Discrete surfaces for architectural design. *Curves and Surface Design: Avignon 2006*, pp. 213–234.

Pottmann, H., Liu, Y., Wallner, J., and Wang, W.: 2007, Geometry of multi-layer freeform structures for architecture. *ACM Transactions on Graphics*, 26(3), Article No. 65.

Prusinkiewicz, P., and Lindenmeyer, A.: 1990, *The Algorithmic Beauty of Plants*, Spring-Verlag, New York.

Stiny, G.: 2006, *Shape: Talking about seeing and doing*, MIT Press, Cambridge.