Citation for published version:
Shepherd, P & Richens, P 2010, 'Subdivision surface modelling for architecture' Paper presented at International Association for Shell and Spatial Structures (IASS) Symposium 2009, Valencia, Spain, 28/09/09 - 2/10/09, pp. 1206-1217.

Publication date:
2010

Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2009, Valencia
Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures
28 September – 2 October 2009, Universidad Politecnica de Valencia, Spain
Alberto DOMINGO and Carlos LAZARO (eds.)

Subdivision surface modeling for architecture

Paul SHEPHERD*, Paul RICHENS[a]

* Department of Architecture & Civil Engineering, University of Bath
Bath BA2 7AY, UK
p.shepherd@bath.ac.uk

[a] Department of Architecture & Civil Engineering, University of Bath

Abstract

This paper introduces research currently under development in the Digital Architectonics Research Group of the Department of Architecture & Civil Engineering at the University of Bath, UK. This research aims to develop a software framework for the application of subdivision surface techniques to the modeling of complex 3D building forms. By incorporating both structural and environmental optimization algorithms into the framework, a "concurrent engineering" approach is adopted, whereby information on the consequences of different design decisions is provided up-front. This results in a tool through which practicing engineers and architects can explore concepts in an informed manner, helping to steer their creativity towards designs with efficiency built-in.

The paper introduces the basic software platform and goes on to describe the implementation of subdivision surface modeling, formfinding and optimization techniques. The benefits of combining these capabilities in a single tool are then demonstrated through a case-study, and the future direction of the research is discussed.

Keywords: Subdivision Surface, Modeling, Software, Digital Architectonics, Concurrent Engineering, Optimization, Formfinding, Environmental Performance.

1. Introduction

The Department of Architecture & Civil Engineering at the University of Bath, UK, has a long-standing reputation as a leader in the field of research into the design of innovative and efficient long-span structures. Its Digital Architectonics Research Group is currently focusing its investigations on answering two questions. Firstly, how can subdivision surface modeling be used most effectively in the building industry? And secondly, how can the benefits of concurrent engineering best be incorporated into the design process?

To answer these questions, a prototype piece of software has been developed which provides a basic three-dimensional modeling environment for the construction and manipulation of complex-geometries, as outlined in the following section. Into this software application, various subdivision surface modeling algorithms have been implemented, along with surface- and grid-formfinding methods, as described in sections 3 and 4 respectively. A number of these surface geometries and structural grids can be explored using the concurrent evaluation of performance in both environmental and structural terms as shown in section 5. An abstract example of the power of using such a combined approach to modeling is presented in section 6, followed by a brief outline of the future direction of the group's research.

2. Modeling Environment

2.1. Basic Tools

In order to explore advanced concepts such as subdivision surface modeling and formfinding it is imperative that a sound software framework is developed on which to build. The framework needs to be easy to use and debug, allow for collaborative software development and be future-proof. For these reasons a basic graphical interface program has been created using object-orientated programming in the C#.NET language in a Microsoft Windows environment with the TAO framework [1] implementation of OpenGL graphics.

A number of simple geometric objects such as *vertices*, *edges*, *faces* and *quads* have been introduced into the software. The hierarchical nature of these objects lends itself well to being manipulated in an object-orientated software framework. Vertices are used to represent positions in 3-dimensional space. Topological relationships such as other connected objects are also stored to speed up mesh traversal operations. Vertices are stored in a kd-trie (Orenstein [10]) to make searching for particular vertices in space a fast and efficient operation. Edges are line-elements used to define higher-order elements and can also be used to represent physical objects themselves, such as cables and struts, by association of physical properties such as radius, slack-length and material. Triangular face elements and quadrilateral elements are used for the basic representation of surfaces. They can be defined topologically by either their vertices or edges, and material properties can also be applied for physical modeling.

2.2. Interoperability

Since building design is a collaborative process involving many parties, each of whom may be using their own proprietary software, the ability to exchange data between different software applications is essential. The research software has been developed to load and save a wide variety of file-formats including the most common open-source CAD formats (dxf, 3dm), rapid prototyping files (stl, vrml, ply) as well as its own text-based or binary file format for full storage of model data.

More interestingly, the software also takes advantage of some COM interfaces between commercial packages. Both the commercial modeling program Rhinoceros (Rhino) and the structural analysis package Robot Millennium (Robot) allow external control by third-party software. In effect, anything a human user can do whilst running Rhino or Robot can be done automatically by the research software.

All the above capabilities, whilst not exactly revolutionary, are a prerequisite to providing a sound platform on which to experiment with the design of buildings. Without this basic set of modeling tools to build on, the practical generation, manipulation and optimization of complex geometry surfaces and structures would not be feasible.

3. Subdivision Surfaces

This section provides a very brief introduction to subdivision surfaces and the schemes used for this research and goes on to describe specific extensions which are required for use in building design. For a more in-depth introduction to subdivision surfaces the reader is directed to the seminal text by Zorin et al [14].

3.1. Motivation

The technique of subdivision surface modeling has been developed in the computer graphics, animation and gaming industries as a way of efficiently representing complex shapes for rendering. It is a recursive process whereby each single element in a mesh, for example each triangle in a triangular mesh, is subdivided into a number of smaller elements. Each original triangle in a mesh could be said to be defined by a set of three parent vertices. By subdividing the mesh a set of new child vertices are created. In the simplest subdivision scheme, the child vertices are placed at the midpoint of an edge and no real benefit is seen. In fact, since it can create a mesh with four times as many elements that represents exactly the same geometry, it could be said to be counter productive (see Figure 1b). However, many other schemes exist whereby the position of the child vertices is a carefully calculated weighted average of their surrounding vertices and this leads to a smoothing process whereby the subdivided mesh represents a new smoother surface than the original mesh (see Figure 1c). Such schemes are known as *interpolating* schemes if the child vertices are smoothed but the parent vertices remain in position. Alternative, *approximating*, schemes also move the position of the parent vertices, as shown in Figure 1d. Either way, subdivision schemes can be viewed as smoothing the coordinates of a mesh, they are recursive, and lead to finer and finer meshes with increasing numbers of vertices and elements.

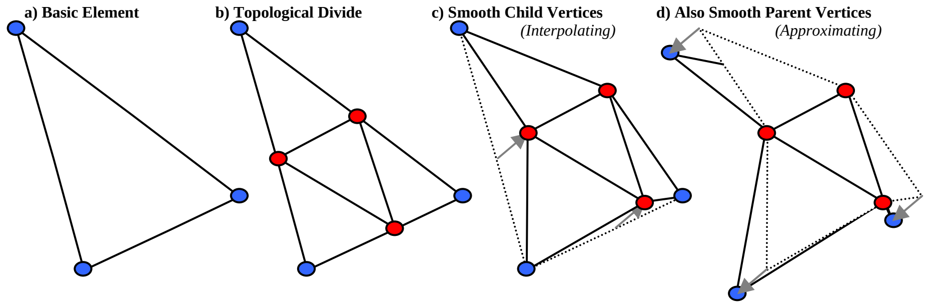

Figure 1: Schematic explanation of subdivision algorithms.

One important property of the subdivision process is that if it were to be repeated infinitely many times, the resulting surface would converge onto what is known as the limit-surface, which can be shown to have G2 continuity (Zorin [15]), a quality often desirable in architecture for its aesthetic. Mathematical operations exist whereby the position of this limit-surface can be calculated directly, without the need for infinite recursion.

Unlike other surface representations (such as NURBS), subdivision surfaces do not require a parameterization of the surface (such as a UV-parameterization). This allows them to represent complex topological shapes such as a three-legged pipe, which would require a discontinuous parameterization (seam) using other techniques.

Since subdivision surfaces can use a very coarse starting mesh to represent a smooth limit-surface, so the number of parameters (such as coordinates) defining any given geometry is small compared to the detail of the surface represented.

These properties together suggest that subdivision surfaces may be particularly well suited to use in building design. They can suggest aesthetically pleasing surfaces to represent a building form without the need to parameterize the surface. And their definition by few control points makes them particularly suitable for the application of optimization algorithms which can operate on a very coarse mesh with few degrees of freedom.

3.2. Schemes

Three different subdivision schemes have been implemented into the basic software framework described above. One interpolating scheme (Butterfly [6]) and one approximating scheme (Loop [9]) have been introduced for triangular mesh elements to explore the differences in behavior of the two types of scheme. In addition, a scheme which operates on quadrilateral elements (Catmull-Clarke [4]) has also been included to extend the range of problems that can be tackled using these tools. Each scheme has a different set of weightings for the position of the parent- and child-vertices relative to the surrounding vertices, and different rules for how the topology of the mesh changes between generations. The subdivision is implemented in the software using recursive programming with object orientated classes and inheritance, such that other schemes can be easily implemented in the future as required. This approach will also facilitate investigations into *non-stationary* subdivision schemes, where the subdivision rules scheme change at each level of subdivision.

3.3. Property Smoothing

If the subdivision algorithms are viewed as a method of generating new mesh vertices with a smooth transition of coordinates, then the same smoothing process can be applied to other quantities assigned to the vertices. The software currently stores extra information such as RGB color information, texture coordinates and velocities at each vertex, as well as the spatial coordinates. The subdivision schemes have been designed to work on a generic vector of such properties at each vertex. This allows for additional attributes to be assigned to vertices and for them to be automatically modified and weighted proportions passed on to child vertices. This provides great flexibility in possible applications of the software within the design process. For example, parameters such as louver-opening-angle or glazing fritting densities could be allocated to each vertex of a coarse control mesh for a proposed building façade surface. These values would then be smoothed as the mesh is subdivided, allowing for gradual transitions in appearance between different areas of louvers / glazing.

3.4. Constraints

Although the smoothing implicit in subdivision surfaces can be beneficial in the context of architectural design, there are some cases where it is not desirable. In particular, most building design schemes have a very definite idea about how the geometry should behave around the boundary. For example, when proposing a design for a roof to cover an existing courtyard or atrium, although there is considerable scope for creativity and experimentation within the interior of the roof structure, it is of course imperative that the roof meets the existing building around the edge to shield the occupants from rain. Subdivision surface methods might seem an attractive method of generating just such a smooth and complex geometry based simply on initial control points around the boundary and maybe a few more within the interior. However, as highlighted in Figure 2, such an approach can lead to problems around points of high curvature (such as the corners) where the vertices are smoothed and begin to move away from the boundary (shown in grey).

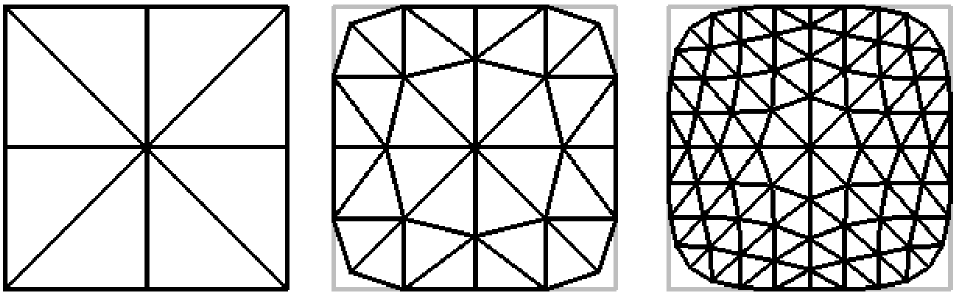

Figure 2: Simple square mesh showing unconstrained subdivision.

In order to provide the control needed for these tools to be of practical use in building design, the concept of constrained subdivision has been adopted for this research following Levin [8]. The concept of a *constraint* property has been introduced for each vertex, allowing the vertex to be defined as fixed in space or attached to a boundary curve (currently defined by line- and arc-segments). Furthermore, the child vertices inherit the constraint from their parents, with a Boolean algebra defining what is inherited if the parent vertices have different constraints. This allows the example above to be initiated with all boundary vertices constrained to the square boundary curve and the central vertex remains unconstrained. On successive subdivisions, child internal vertices do not inherit any constraints and are generated in a smooth manner. In contrast, child vertices around the boundary inherit the constraint and remain fixed to the boundary, as can be seen by Figure 3.

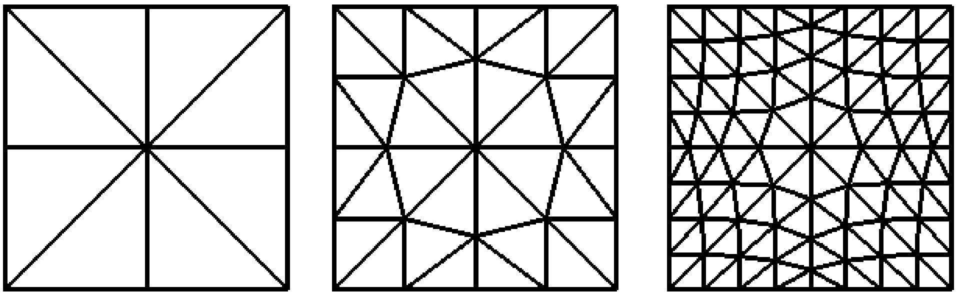

Figure 3: Simple square mesh showing constrained subdivision.

This gives the designer much more control over how the resulting surface geometry will behave around the boundary. The price paid for this added control is that the surface around the boundary is no longer guaranteed to be G2 continuous. Since the unconstrained algorithm would have delivered this property, and the vertices have been in effect snapped-back to the boundary away from their desired location, the smoothness is clearly affected. However this lack of smoothness is limited to the boundaries, since subsequent subdivisions will re-smooth the vertices and introduce new un-smooth vertices even nearer to the boundary. When proposing the limit surface as a design surface therefore, this lack of G2 continuity is not seen as a major issue for the building designer, especially given the added flexibility and practical applicability it allows.

4. Formfinding

The first question this research project seeks to address is that of how subdivision surface modeling techniques can be used in the building design process. This has been addressed by splitting the task of building design into two distinct phases. Firstly an idealized surface needs to be generated, which could represent the cladding surface for a roof, façade, shell or gridshell. Secondly, some sort of structure will be required, either within or slightly offset from this surface, which defines the structural members which would actually be constructed. The first task is more usually carried out by an architect, whereas the latter is typically more in the realm of a structural engineer. By delivering tools for both tasks in one single integrated environment, it is hoped that the different members of the design team might be able to collaborate more closely and develop a more holistic solution to the design brief.

4.1. Surface

In the spirit of including many disparate tools within the same software framework, a number of formfinding tools have been included which operate on the mesh vertices. These tools modify the coordinates, and sometimes velocities, of each vertex, and use system of time-stepping until a converged state is reached or until the user stops the tool.

A very basic *finite difference* based system of formfinding (Richtmyer [11]) has been implemented, which uses a simple coordinate-averaging scheme to converge to a possible design surface. This is a very basic approach that can be used to quickly validate the correct application of boundary constraints without the need to define material properties for elements.

In order to simulate cable-net type structures, a *force density* solver (Scheck [12]) has been included. This allows minimal surface geometries to be found from a model defined simply by edges connecting vertices. A post-processor which identifies closed loops of three or four edges can then be used to generate face- or quad-elements to actually represent the surface as a mesh. A *dynamic relaxation* solver (Barnes [3]) is also included, to allow the formfinding of triangular meshes. These two solvers require the incorporation of material properties to define the physical behavior of the geometric elements. This is done by specifying materials by a piecewise linear stress-strain curve and then associating each element (edge, face or quad) with a material. In the case of two-dimensional elements (faces and quads) two material properties are specified so that the elements can represent typical fabric structure materials, one for the warp (defined as the direction parallel to the element's first edge) and one for the weft (perpendicular to the warp by definition). This simplistic representation is viewed as sufficient to provide a formfinding "ideas" tool and is not intended to replace expert fabric structure detailed analysis and design.

4.2. Structure

A number of tools have been incorporated into the software to facilitate the generation of structural grid options once a design surface has been created.

Since the implementation of the formfinding solvers described above required the consideration of vertex velocities, fixity constraints and material properties, the addition of draping simulation algorithms was relatively straight forward. A grid of line elements can be defined as free to move and with a non-zero density (self-weight) above a surface mesh geometry that is specified to remain fixed in place. A dynamic relaxation style time-stepping process can then be initiated, with the grid constrained such that it cannot pass through the surface mesh. A draping simulation can thus be performed, whereby the grid drapes itself over the control surface. With careful consideration of the bending and axial stiffness of the grid, equal-length grid solutions can be found which take up the shape of the underlying surface. This underlying surface can either be chosen to be the control mesh itself, or more interestingly, its limit surface.

A further tool has been created which will automatically generate the graph-theoretic dual (Diaz et al[5]) of any given mesh. This dual can be offset from the original mesh by a given amount in the direction of the surface normal, and its vertices optionally linked back to the original vertices with new edge elements. In this way, from any given mesh, a space-frame structure of any given depth can be automatically generated.

A similar structure generating tool is available which uses every edge in a mesh to represent the centre-line of a box-section of specified dimensions. The intersection of many edges at a vertex is dealt with by joining the incoming box sections with a polygonal surface top-and-bottom, as shown on the left hand side of Figure 6. This structural profiling tool can quickly generate solid structural sections for rendering from simple line models.

5. Optimization

The second research question addressed by the project is that of how concurrent engineering principles (Kusiak [7]) can be adopted for use in building design. In this regard, some methods of assessing the performance of a building surface have been included into the software framework to provide up-front indications of the likely effects of design decisions. The premise is that by implementing simple, fast evaluations of both environmental and structural performance within a powerful geometric modeling package, the user will be able to explore different design solutions with near instant feedback on the effects those decisions are likely to have on the downstream design processes. In this way, the user is expected to find it easier to develop more "optimal" designs, assuming they have an idea of what they are aiming to optimize.

5.1. Environmental Optimization

Ray-tracing has been added to the software, such that the intersection of any vector with the surface mesh, or its limit surface, can be calculated. The main application for this functionality is in the assessment of *solar gain* of a proposed building. It is relatively trivial [2] to calculate the position of the sun relative to a given location on the globe for a specific time of day and day of the year. This is therefore used to calculate how much solar energy would be incident on a particular surface geometry, and is further modified to consider the angle the sun's rays make with the surface and how this would be refracted through, or reflected off, any given material such as glazing or foil cushions. Multiple intersections of the rays with the surface are taken into account, so that a surface can cast shadows on itself and the solar gain is reduced accordingly. By assessing the incident radiation for different times throughout the day and different times throughout the year, an overall picture of the solar gain of any proposed building geometry is displayed by using a color scale. Although qualitative and not quantitative, this tool is capable of comparing different designs, and the effects of dragging a vertex from one position to another can easily be seen.

Another implementation of the ray-tracing system is in the *particle tracking* functionality. In this case a set of unconnected vertices are modeled in position and velocity and the time-stepping system continually updates their coordinates. The ray-tracing functions are used to test whether any given vertex has passed through the model surface, and if so, the reflected position of the vertex is allocated instead. In this way, the vertices behave like Brownian-motion particles, and can be made to bounce-off either the control mesh or its limit surface. A possible application of this tool is in acoustic analysis, where the particles represent sound wave-fronts and distortion can be simulated by counting the distance travelled by each particle and the number of reflections. Other potential uses range from the assessment of wind loads or blast analysis to people-flow modeling.

5.3. Structural Optimization

The structural performance of a proposed building can be assessed in a number of ways. Since the framework can already hold information about a model's physical properties as well as its geometry, up-front information about its likely structural performance can easily be assessed. One approach would be to implement simple analysis tools such as pinned- or rigid-joined frame analysis or a finite element solver. In this case, advantage has been taken of the COM interface with the commercially available Robot software described in section 2.2. An iterative loop has been defined, which can automatically open up a copy of Robot, populate it with geometry and physical properties from the model, analyze it under an idealized load case (for example self-weight) and read back the results. The software can then either display the results to the user, or take decisions based on the results and repeat the process. For example, studies have been performed whereby all elements with stress below a certain threshold are deleted and then a further cycle performed until either there are no elements left or all elements are carrying a non-trivial load.

6. Example of Combined Tools

The true value of this integrated approach to research can be demonstrated through a case-study. By having many disparate tools available within one single software application, the tools can interact and build on each other's output to produce complex results with ease.

The abstract image below (Figure 4) was generated as a marketing logo to publicize the MPhil in Digital Architectonics, a Masters by Research course in the Department of Architecture and Civil Engineering of the University of Bath. It was generated using a combination of the research software's draping, profiling and subdivision functions.

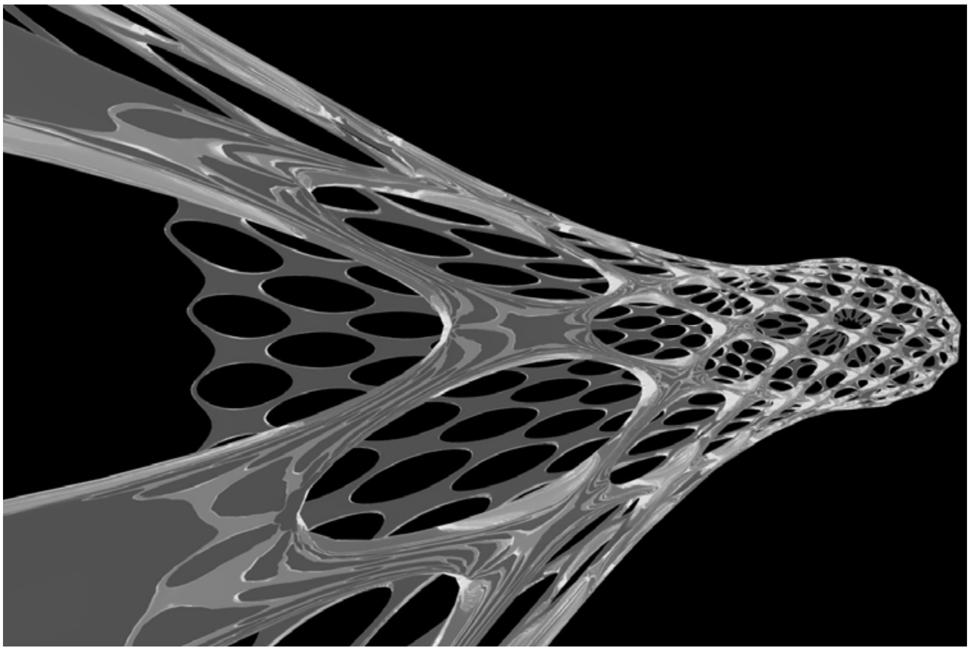

Figure 4: Logo for Digital Architectonics MPhil course.

The basic procedure for generating the form of the logo was to drape a net over a sphere. A number of starting net shapes were tested, including square and hexagonal grids. However, since the final form was desired to have a high degree of radial symmetry, starting from, for example a square grid, always produced large distortions in the corners where the squares had to undergo large shear deformation in order to take up the spherical shape. To eliminate this progblem, a spiral-based grid was generated parametrically inside the software and then draped over a sphere as shown in Figure 5. An additional pressure gradient was introduced into the system to cause the net to neck-in underneath the sphere and create a more flowing, aerodynamic type appearance.

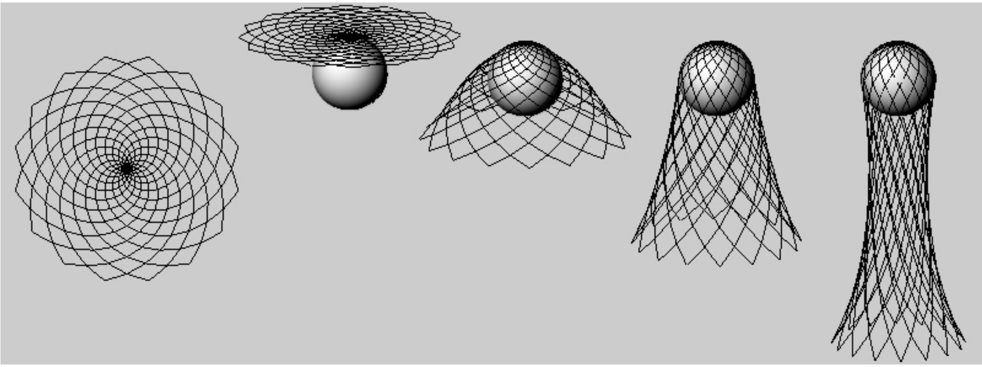

Figure 5: Generate a spiral net and then drape over a sphere

Every line of the draped net was then converted into a square box-section representation using the structural profiling tool, as shown in the two left-most pictures of Figure 6. This surface representation of the geometry was then subdivided using the Loop scheme as a smoothing tool, to turn the faceted box-box junctions into a more flowing shape, as though the net had been dipped in a viscous fluid. The three pictures on the right-hand side of Figure 6 show first, second and third subdivision generations. The final form for the logo was taken as the limit surface of the model, sampled at the third generation of subdivision.

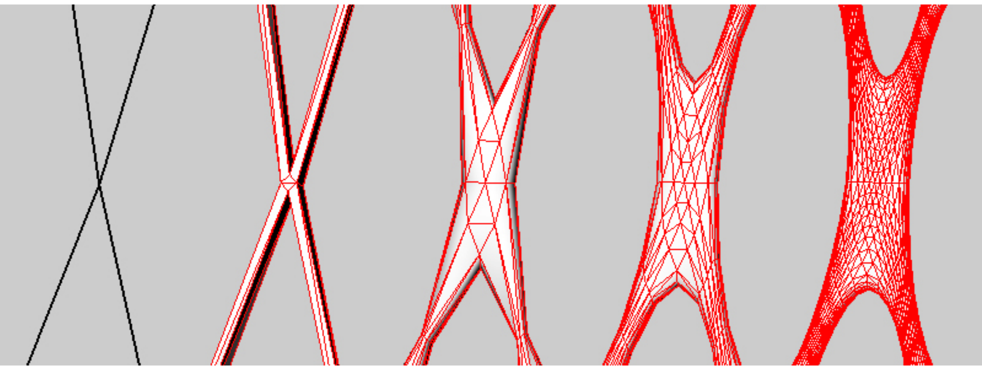

Figure 6: Add box-section profiles to lines and then subdivide surface

The resulting geometry was then exported directly to Rhino using the programmatic COM interface and rendered using V-Ray for Rhino with a wide field of view and refractive / reflective material.

Although very different from the building design process, the fast and simple work-flow required to create this complex geometrical model shows how powerful the research software framework can be through its integrated approach and the combination of physical modelling simulators with subdivision geometry processors.

7. Future Research Direction

This paper has outlined the research approach being adopted by the Digital Architectonics Research Group in the Department of Architecture and Civil Engineering of the University of Bath. A basic framework of software tools has been created, and extended to include subdivision surface modeling and formfinding. Assessment of the performance of resulting models for both environmental and structural considerations has also been included. The benefits of linking these tools together have been demonstrated through the easy creation of an elegant and complex geometric form, which has been rendered for use as a marketing logo.

The software has also been used with great success to help the win the architectural competition to design a new hothouse for the Aarhus Botanical Garden project in Denmark (Shepherd [13]). In this real-world building project, the advantages of adopting a concurrent engineering approach to building design led to an aesthetically pleasing dome structure which was also optimized for environmental performance. The practical application of using such tools within a design team of practicing architects and engineers has therefore been demonstrated. However, a broader range of case-studies will be needed to fully explore the relative merits of the different tools and their interaction.

Future research will develop in two directions. Firstly the geometric modeling capabilities of the software will be extended by including such tools as Voronoi meshing, conformal grid mapping and parametric constraint engines. Subdivision surface modeling will also be extended through non-stationary schemes and the open problem of dealing with the intersection of subdivision surfaces will be addressed.

The second branch of research will look at optimization and analysis algorithms and the assessment of performance against broader criteria. Topology optimization and genetic algorithms are likely to be an initial focus to help drive the building shape exploration. Structural buckling analysis and dynamic response calculations will also fit into the framework particularly well.

Acknowledgement

The authors would like to thank Informatix Inc. for their financial support of this research.

References

[1] http://docs.taoframework.com/Tao.OpenGl/, accessed 14th May 2009.

[2] http://ecotect.org/wiki/Solar_Position_Calculator, accessed 14th May 2009.

[3] Barnes, M., “Form Finding and Analysis of Tension Structures by Dynamic Relaxation”, *International Journal of Space Structures*, 14(2)pp89-104, 1999.

[4] Catmull, E. & Clark, J., “Recursively generated B-spline surfaces on arbitrary topological surfaces”, *Computer-Aided Design*, 10(6):350-355, 1978.

[5] Diaz, J. A., Togores, R., & Otero, C., “Geometry Applied to Designing Spatial Structures: Joining Two Worlds”, in *ICCSA 2004 Proceedings Part III*, Lagana A. et al (Eds), pp158-167, Italy, 2004.

[6] Dyn, N., Levine, D., & Gregory, J. A., “A butterfly subdivision scheme for surface interpolation with tension control”, *ACM Transactions on Graphics*, 9(2) pp160-169, 1990.

[7] Kusiak, A. (Ed), “*Concurrent Engineering: Automation, Tools and Techniques*”, Wiley, 1992.

[8] Levin, A., “Combined subdivision schemes for the design of surfaces satisfying boundary conditions”, *Computer Aided Geometric Design*, 16(5) pp345-354, 1999.

[9] Loop, C., “Smooth Subdivision Surfaces Based on Triangles”, M.S. Mathematics Thesis, University of Utah, 1987.

[10] Orenstein, J.A., “Multidimensional tries used for associative searching”, Information Processing Letters, 14(4) pp150-157, June 1982.

[11] Richtmyer, R. D. & Morton, K. W., “Difference Methods for Initial-Value Problems”, Interscience Publishers, New York, 1967.

[12] Scheck, H. J., “The Force density method for form-finding and computation of general networks”, Computer Methods in Applied Mechanics and Engineering, Vol 3 pp115-134, 1974.

[13] Shepherd, P., “Digital Architectonics in Practice”, in *Computation: The new realm of Architectural Design, Proceedings of the 27th eCAADe Conference*, 2009.

[14] Zorin, D., Schroder, P., DeRose, A., Kobbelt, L., Levin, A. & Sweldens, W., “Subdivision for Modeling and Animation”, SIGGRAPH 2000 Course Notes, New York University, 2000.

[15] Zorin, D., “C^k Continuity of Subdivision Surfaces”, Technical Report CaltechCSTR:1996.cs-tr-96-23, California Institute of Technology, 1996.