As a possible solution to the quandary of seeing two natural parts of the architectural process—free design and successive rationalization—in the hands of two separate professions when it comes to freeform architecture, this paper proposes the incorporation of respective geometric information into architectural design tools. An exemplary prototypical software is introduced, empowering an architect to interactively design and edit architectural freeform shapes represented as regular quad meshes with planar faces. The sustained planarity of faces is an integral part of the design process, thus considerably decreasing the need for elaborate post processing towards feasibility.
1 Introduction

Throughout the past few years, growing interest in the design and actual production of extensively curved architectural building hulls could be noticed. A state of the art, common solution for creating these is a two-step-process with freeform-design via NURBS or subdivision modeling packages as its first step. However, due to the absence of information on feasibility, the output of the latter in no way complies to geometric conditions and constraints relevant to actual needs of production.

Thus, if not the architects intend to have their freeform designs built without geometric rationalization (and by that necessitating parts of unaltered curved geometries to be produced individually and hence inefficiently in terms of cost and sustainability), step two employs vast successive post-processing, i.e. the approximation of these freeform surfaces by conveniently producible segments.

Hence, breaking down freeform geometries into meaningful parts has to be done by highly specialized third party companies, depriving the architect of two main challenges—the design of buildings based on structural necessities as well as the control over the final look.

This paper targets the design and production of glass/steel hulls, such that the term “feasibility” here translates to “planarity of panels”. Within this context, an (economically) feasible geometric representation of a freeform design is a number of flat segments appropriately approximating the intended complex geometry. These panels can be cut from flat sheets of glass in an easy and inexpensive way.

Presently, the approximation by triangles is the most common way to render a complex shape feasible. This method is relatively easy on one hand, but has its issues when it comes to structural complexity and extensive use of material. Therefore, the author addresses more promising ways of achieving feasible architectural freeform designs, especially planar quadrilateral panels.

Previous work in this area addresses the adaption of translational surfaces and other simple constructions to freeform design (Glymph et al., 2002), (Figure 2a). For a survey on further work on the design and applications of meshes with planar faces, especially in connection with support structures exhibiting torsion-free nodes, see (Pottmann et al., 2006).

As a new solution to this quandary of seeing two natural parts of the architectural process—free design and successive rationalization—in the hands of two separate professions, the author proposes the incorporation of respective geometric information into architectural design tools.

Exploiting all available degrees of freedom inherent to the geometrical constraints induced by material and production method requirements makes it possible to create and edit an already rationalized freeform design while maintaining important conditions such as planarity at all times. By that, an architect is granted a good deal of freedom at design time with total control over the structural look, such that the design process results in a freeform-design already accommodated to the successive process of production (Figure 1).
This paper introduces a prototypical software implemented as a plug-in for Rhinoceros 4 (a CAD modeling package very common to architectural design), empowering an architect to interactively design and edit freeform shapes represented as regular quad meshes with planar faces (Figure 3). The sustained planarity of faces is an integral part of the design process, thus considerably decreasing the need for elaborate post processing towards feasibility.

2 Definitions and Terminology

2.1 Planar Quad

A planar quad is a quadrilateral whose cornerpoints are elements of the same plane. Note that, while coplanarity of cornerpoints is always true for triangles, the cornerpoints of a general quad need not automatically be co-planar.

2.2 PQ Strip

A PQ strip (Figure 4a) is a sequence of two or more planar quadrilaterals, where each two are connected via one shared edge. See (Liu et al., 2006).

2.3 Interior Edges

A PQ strip is bounded by four sides, where two consist of open polygons exhibiting two or more segments. Edges connecting dedicated vertices on these opposing polygons are called interior edges (Figure 4a).

2.4 Conducted PQ Strip, Initial and Resulting Polygon

In the context of the presented software, a conducted PQ strip is a general PQ strip derived from a given polygonal boundary, which will be referred to as initial polygon (Figure 4a). Consequently, each single face of a PQ strip has its initial segment being part of the initial polygon. The opposing polygon resulting from the strip creation is called resulting polygon (Figure 4a).

2.5 Generative PQ Patch

A sequence of two or more PQ strips connected via their polygonal boundaries is called a PQ patch throughout this paper (Figure 4b). Please note that a PQ patch features regular connectivity, which means that its inner vertices (i.e. points) exhibit a valence of four,

Figure 3. Interactively designing an architectural freeform surface consisting of planar quadrilaterals.

Figure 4. Definitions. (a) Conducted PQ strip. (b) Regular PQ patch, where each strip is derived from the resulting polygon of the preceding strip.

Figure 5. (a) Rotating a PQ strip around its initial polygon. (b) Editing the strip width. (c) Individual segments of the resulting polygon may be rotated against their dedicated segments on the initial polygon.

Figure 6. (a) Rotating interior edges. (b) Explicit unplanarity by moving vertices away from their containing planes.

Figure 7. Design workflow sequence. (a) Editing polygon edge angles. (b) Editing strip angle. (c) Editing initial polygon. (d) Final result.
whereas boundary vertices have three, corner vertices two incoming edges. Due to the nature of the presented algorithm, the basis for each PQ strip being part of a PQ patch depends on its predecessor, such that changes at one PQ strip reflect in all following strips. This fact is taken into account by the term generative.

2.6 Conducting Polygon

The polygon made up of all first interior edges is called conducting polygon (Figure 4b).

3 Editing Options for Conducted PQ Strips

While general quad meshes exhibiting non-planar faces may be edited arbitrarily, the constraint of maintaining planarity limits possible editing actions to certain operations. In the following, these operations, as implemented in the software, will be introduced. Note that some of them operate on an entire strip, while some are applicable to each face or interior edge of a PQ strip.

3.1 Strip Angle

Each PQ strip is allowed to freely rotate one face around its initial segment at a specific angle (Figure 5a). In the presented software, this face typically will be the first one constructed for a PQ strip. Subsequent faces are constructed accordingly, such that by applying one value, the entire strip is rotated.

3.2 Strip Width

For the first face of a PQ strip it is allowed to alter the distance between its initial and its resulting segment, by that altering the overall width of the strip (Figure 5b).

3.3 Polygon Edge Angles

For each resulting segment of a PQ strip (and by that for each face), the angle it encloses with its dedicated initial segment can be manipulated (Figure 5c).

3.4 Interior Edge Angles

Per default, an interior edge is constructed as an element of a symmetry plane bisecting two adjacent segments of the initial polygon. These edges can be assigned angles by which their respective carrying planes deviate from symmetry (Figure 6a). Note that, since no such planes exist for the first and last vertex of the initial polygon, it will be perpetuated by two virtual segments according to the curvature. Refer to the section on the construction of a conducted PQ strip for further details.

3.5 Explicit Unplanarity within Production-Tolerances

Since planarity is not an absolute condition, depending on the reliability of measurements as well as on specifications of production techniques and material constraints, it is permissible to intentionally render quads unplanar to a certain extent, thus achieving interesting varieties of shapes. In the specific architectural context, this is valid as long as a flat panel does not break when mounted. For glass/steel structures, a typical magnitude of deviation from a plane would be 10-20 millimeters for a glass panel of 1-2 meters length and width. Each face of a PQ strip can be assigned a value defining the normal distance by which one of its four cornerpoints deviates from a plane defined by the other three (Figure 6b).

4 Algorithm

4.1 Rebuild Procedure

In order to generate a PQ patch to start with, the presented software lays out an equally spaced, straight initial polygon, exhibiting a user-supplied number of segments. From this polygon a first PQ strip is derived, whose resulting polygon will be used as the initial polygon for the generation of a next PQ strip.

This procedure is repeated n times, where n is the number of strips the user wants to have generated. The number of segments of the first initial polygon multiplied by the number of strips the PQ patch consists of equals the total number of faces on the resulting generative PQ patch.

4.2 Editing Procedure

Construction Values. As shown in the previous chapter, the construction of individual strips depends on certain values. These may be interactively assigned throughout the entire design process (Figure 7a, 7b), thus affecting the geometry of either entire strips, individual faces, or interior edges. Detailed information on the construction of these strips will be given in the next chapter.
Initial Polygon / Conducting Polygon. Additionally, the initial polygon of the first PQ strip is editable (Figure 7c, 7d). In a special mode, it is possible to also edit the conducting polygon. Refer to the chapter on design workflow.

As soon as construction values have changed or polygons have been edited, the entire PQ patch is rebuilt instantly based on these changes.

5 Construction of a Conducted PQ Strip

Because the first PQ strip of a generative PQ patch naturally possesses no predecessor, the construction for the general case of an interior PQ strip is covered.

5.1 First Quad

As a first step, two planes of symmetry $\varepsilon_1$ and $\varepsilon_2$ are created, bisecting the initial segment of the new quad and its predecessor/successor. These carry the new quad vertices $v_1$ and $v_2$ as well as its interior edges and thus are rotated according to the interior edge angles defined for these (Figure 8a). Note that, for the first and last segment of an initial polygon, an additional virtual segment is constructed, perpetuating the polygon curvature, such that this algorithm can be applied to any open polygon.

In order to receive vertices $v_3$ and $v_4$, which will be elements of $\varepsilon_1$ and $\varepsilon_2$ respectively, a line to intersect these planes has to be constructed. Therefore, a circle of radius strip width is placed, with line $L_{12}$ as its axis and the midpoint of that line as its center. Let point A be an intersection of that circle and the plane containing the preceding quad on the previously built strip. Point $A\sim$ is now received by moving A along the circle according to the strip angle $\alpha$ set for the new strip. Next, a straight line $l$, parallel to $L_{12}$, is placed through $A\sim$ (Figure 8b).

The intersection of $l$ with $\varepsilon_1$ and $\varepsilon_2$ yields points M and N (Figure 8c). Finally, $l$ is rotated around $A\sim$ within the plane spanned by $v_1$, $v_2$, M, and N, according to the polygon edge angle $\beta$ set for the new quad, yielding the final vertices $v_3$ and $v_4$ (Figure 8d).
5.2 Remaining Quads

Since vertex \( v_3 \) is received from the previously constructed quad, thus determining strip width as well as strip angle, the only thing left to do is the calculation of vertex \( v_4 \) (Figure 9a, 9b): A straight line starting at vertex \( v_3 \) that is parallel to \( L_{12} \) is rotated in-plane according to the polygon edge angle set for the new face and finally intersected with the rotated symmetry plane bisecting the current and the next initial segment.

6 Design Workflow

As mentioned before, each PQ strip exhibits a set of values, associated with certain editing actions taken when (re-)building it. Editing the PQ patch means interactively adjusting these values while the software immediately rebuilds the generative PQ patch on detected changes.

6.1 Selecting PQ Strips

In order to edit construction values of a PQ strip according to the aforementioned editing options, the strip must be selected first. If more than one strip is selected, changes affect all of the selected strips in the same manner.

6.2 Application Of Values

When editing a selected strip, either the strip itself, its faces, or its interior edges can be assigned construction values, depending on the desired editing action to be taken (Figure 10).

Values currently assigned to the respective elements of the selected strip are edited via increment/decrement: Strip-wise adjustment alters one value per strip, while for face-wise and edge-wise adjustment, values are changed by increasing/decreasing the current value of the first and last respective element separately. Values in between are interpolated.

6.3 Editing PQ Patch Dimensions

If more detail is wanted at some place in the design process, users may dynamically edit the PQ patch dimensions (Figure 11). Changing the number of rows alters the number of segments of the initial polygon accordingly, while changing the number of strips operates on the segmentation of the conducting polygon. Construction values for new strips, faces, or edges are interpolated.

6.4 Working with Strip Angles

Each PQ strip may be assigned a value influencing its rotation around the initial polygon it is derived from (Figure 12). Internally, this value represents the dihedral angle enclosed by the first face of the strip and its neighbor on the preceding strip.
6.5 Adjusting Polygon Edge Angles

For polygon edge angles, values apply to each face of a selected PQ strip. Face-wise adjustment sets the values of the first and last face respectively, while the others are interpolated, thus yielding a smooth distribution of angles along the strip (Figure 13). Optionally, it is possible to interpolate angles for the first and last selected strips against their unselected neighbors. This “fade-out” option results in smoother transitions between edited and unedited strips.

6.6 Adjusting Interior Edge Angles

Interior edges of a PQ strip are normally constructed to be elements of symmetry planes bisecting their respective initial segments. However, it is possible to rotate these planes, such that they deviate from symmetry at a specific angle, which can be set in an edge-wise manner (Figure 14). As an option, the software allows the adaptation of dedicated interior edges from previous strips to values of the selected strip.

6.7 Applying Explicit Unplanarity

To explore the variations of shapes achieved when purposely rendering faces unplanar within architectural tolerances, it is possible to assign values on a face-wise basis, stating the normal distance of one cornerpoint to the plane defined by the other three (Figure 15).

6.8 Editing the Initial Polygon

Since a generative PQ patch is derived from an initial polygon, it is also possible to edit its shape at design time, thus triggering a patch rebuild (Figure 16a).

6.9 “Edit Two Polygons” – Mode

In addition to editing the first initial polygon of a PQ patch via its control points, a special mode has been added which allows for also editing the conducting polygon (Figure 16b). In order to fine-tune these curves appropriately, the number of desired control-points can be set at design time. Because strip angles and strip widths are derived from that curve directly, editing these via selecting strips as described before is not available in this mode.
6.10 Developable Strips

A PQ strip can be unrolled into a plane without distortion and converges to a torsal ruled surface (developable surface) when continuously refining its interior edges. See (Liu et al. 2006; Pottmann et al. 2008). Since this kind of refinement can be done by increasing the PQ patch dimension in one direction, the option of deriving developable strips from sufficiently refined PQ strips has been implemented (Figure 17).

6.11 Visual Modeling Feedback

In order to gain visual feedback over the distribution of respective construction values, several color analysis modes have been implemented. For the purpose of visualization, an optional pseudo-architectural structure along with cylindrical nodes and node-axes can be displayed at design time, where the dimensions of a rectangular beam profile can be changed interactively. Node axes are achieved by averaging face normals of quads surrounding the respective vertices, beam orientations are approximated between adjacent axes (Figure 18). This feature has been used in the creation of (Figure 1, 21, 22, and 23).

7 Conclusions / Future Research

As an attempt of bringing cost-efficient design of complex surfaces closer to the architect, the described approach of using the planar quadrilateral as a bottom-up entity for creating architectural PQ meshes for sure holds a lot of potential. (Figure 1, 21, 22, and 23) show some design-studies that could easily be achieved with the presented software.

However, planar quads represent only one of many production-oriented constraints that in the authors opinion should find their way into modern CAD design tools for architects.

With this in mind, future implementations will target the following features:
7.1 Introduction of Triangles

Because it is valid to collapse a planar quad edge to a single point, one could think of this way introducing triangles in order to properly deal with narrow regions of a PQ patch (Figure 19a). See also (Glymph et al., 2002). Parts of this have already been implemented (Figure 19b).

7.2 Irregular PQ Meshes

By connecting a number of PQ patches other than 4 alongside their editable polygons as shown in (Figure 20), it is possible to achieve irregular PQ meshes, exhibiting inner vertices connecting other than 4 incoming edges, so-called irregular vertices.

Acknowledgements

Thanks to Benjamin Schneider for helping with the design and creation of some of the figures. The author gratefully acknowledges the support of the Austrian research promotion agency (FFG) under grant No. 813391.

References