Computing Curved-Folded Tessellations through Straight-Folding Approximation

Suryansh Chandra[1], Axel Körner[2], Antiopi Koronaki[3], Rachelle Spiteri[3], Radhika Amin[3], Samidha Kowli[3], and Michael Weinstock[3]

[1]Zaha Hadid Architects
London, UK
suryansh.chandra@zaha-hadid.com

[2]University of Stuttgart
Stuttgart, Germany
a.koerner@itke.uni-stuttgart.de

[3]AA School of Architecture
London, UK

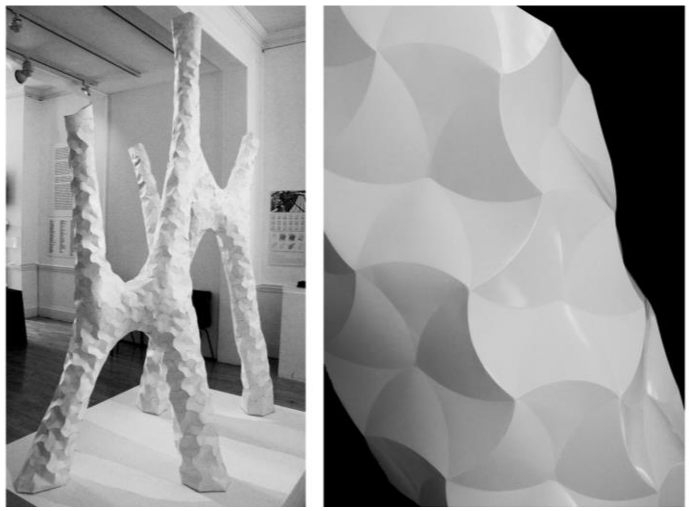

Figure 1: Prototype designed and produced using the method described in this paper.

ABSTRACT
The research presented in this paper explores curved-crease tessellations to manufacture freeform geometries for architectural and industrial design. The work draws inspiration from the ease of shaping paper into double-curved geometries through repeating fold patterns and the observed stiffening of curved surfaces.

Since production of large scale curved-folded geometries is challenging due to the lack of generalised methods, we propose an interactive design system for curved crease tessellation of freeform geometries. The methods include the development of curved folding patterns on the local scale as well as a novel computational method of applying those patterns to polysurfaces. Using discretized, straight-line fold approximations of curved folds in order to simplify computation and maintain interactivity, this approach guarantees developable surfaces on the local scale while keeping the double curved appearance of the global geometry.

Author Keywords
Curved-Folding, Edit-and-Observe Modelling, Design to Fabrication workflow.

SimAUD 2015, April 12 - 15, 2015, Alexandria, VA, USA

INTRODUCTION
Curved folding offers a very economical method of manufacturing curved surfaces from flat sheet material but a significant obstacle in working with curved-folded geometries is the lack of appropriate computational tools in commercial CAD software for describing such geometries. As architects and designers, we favour interactive methods that offer direct control over three-dimensional geometry. Contrastingly, the methods we have surveyed either require scanning of physical models and planar-quad meshing [7], or work on smaller scales of bending a singular surface [9], or require sequentially deriving surfaces through the method of reflection [8], but do not scale well to geometries with high number of creases. Further, curved-folded surfaces are simple to model physically with a limited number of creases, but substantially increase in complexity with several repeating creases as in a tessellation. Thus, the primary contribution of this paper is the setting out of a method for designing curved-crease tessellations and describing a simple and intuitive computational framework to simulate these curved-creases on straight-line tessellations. Critically, it should be pointed out that the geometry simulated by this method is a visual approximation and not precise curved folded geometry. However it does ensure that the unrolled two-dimensional shapes can be folded and fit together into a predefined form.

As case studies we investigated Ron Resch's straight folding patterns [10], as well as David Huffman's studies on curved folding [2]. In Ron Resch's pattern (Figure 2), polygonal faces (so-called inflated vertices) have straight fold-lines at each vertex, describing triangular faces. Aggregations of this folded pattern across a grid lead to global curvature. However, when all fold-lines converge into one vertex (so-called non-inflated vertex) instead of a polygonal face, it is not possible to fold the pattern with straight-line creases. David Huffman investigated patterns with even number of curved creases on inflated and non-inflated vertices.

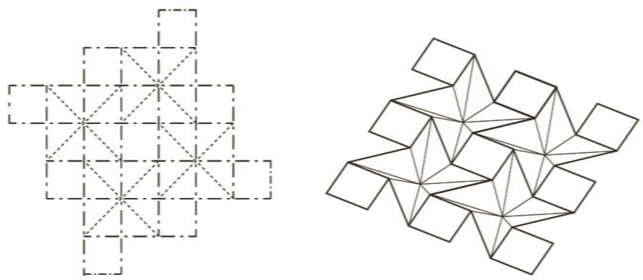

Figure 2: Ron Resch's folding pattern a. flat state; b. folded state.

For the purpose of our research we explored the possibility of achieving globally double curved geometries with aggregations of non-inflated vertices with curved creases.

METHOD

The method presented in this paper is broken down into two sections. The first part sets out a framework for designing curved-crease patterns at the local level (similar to Maekawa's and Kawasaki's theorems for straight-folding) [5] and the second part presents an interactive computational method for applying those crease patterns to freeform surfaces such that the resulting unrolled shapes are ensured to be developable.

Designing a Curved Crease Component

Non-inflated Vertex

We take a non-inflated degree-n vertex with an even number of alternating mountain and valley curved folds, similar to those developed by David Huffman [2]. In isolation, this configuration is versatile in terms of the number of mountain-valley folds at a vertex, and in terms of the curvature and type of two-dimensional curves that can be folded, which affect the three-dimensional depth of the folded geometry. For a given folding angle, a crease line with lower curvature will increase the three-dimensional depth of the folded component.

Boundary conditions

In order to develop a repeatable component that can be aggregated across a three-dimensional tessellation, the boundary of the non-inflated degree-n vertex in relation to its curved folds needs to be defined, so that each boundary edge can act as a developable seam between two folded components. In a curved folded tessellation, the surfaces adjacent to a crease are either cylindrical or conical, reflected by the plane on which the curved crease lies. This results in alternating concave-convex developable surfaces.

Hexagonal Boundary

Our initial attempt at aggregation is done with a folded non-inflated degree-3 vertex (Figure 3a). The aggregation is based on a regular hexagonal tessellation with the vertex of the folded module lying at the centre of each hexagon. No additional crease curves are added along the boundaries of the hexagon in this case. This results in the formation of secondary vertices of all-mountain or all-valley creases along the boundaries between components and in the formation of curved surfaces defined by four curved folds in the order *mountain-mountain-valley-valley*.

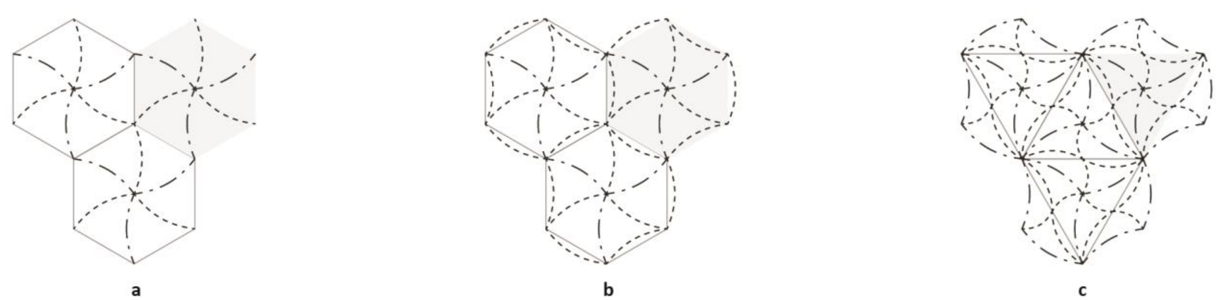

Figure 3: Aggregation of a folded non-inflated degree-3 vertex with hexagonal boundary (a, b) and triangular boundary adaptable to all n-gons (c).

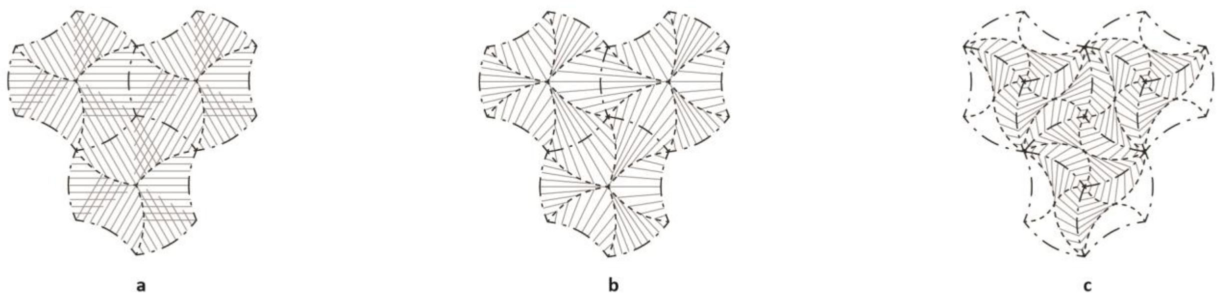

Figure 4: Surface rulings: cylindrical rulings showing conflict (a); conical rulings on hexagonal (b) and n-gon boundary (c).

We tested this aggregation on a number of material samples: 80gsm paper, 120gsm paper, thin cardboard and polypropylene. The paper folds easily into the pattern, however the thicker materials tend to unfold with increased aggregations of the pattern. This results from the fact that the four-sided surfaces which are formed as a result of two adjacent folded modules need to accommodate the transition from concave to convex, which adds bending into each face; this is investigated by analysing the surface rulings (Figure 4a).

This was resolved by introducing a valley crease with alternating curvature, splitting the 4-sided surface into a convex and a concave conical surface (Figure 3b, 5b). Tests in paper, cardboard and polypropylene (figure 5a) reflect the resolved rulings. Critically, this requires an alternating convex-concave curvature sequence within the n-gon, making it unsuitable for odd sided n-gons.

Tessellation of freeform surfaces

We used discrete meshes to represent our geometry due to their ease of use across platforms and the wide variety of available tools to control and manipulate them. However, meshes are usually made up of only tri or quad faces, while our folded components are better suited to polygons.

To achieve a mesh with convex polygonal faces we used the dual-mesh based on a triangulated mesh where the centres of adjacent faces are connected. Thus, new faces are created in which the initial vertices represent the centres of the new polygonal faces. An even number of edges and therefore an even and alternating arrangement of mountain and valley folds is guaranteed by dividing each edge of each polygonal face into two parts, as explained above.

For a smooth and homogenous appearance of the resultant polygon mesh, the triangular faces of the initial mesh should be as close as possible to equilateral. (Figure 6).

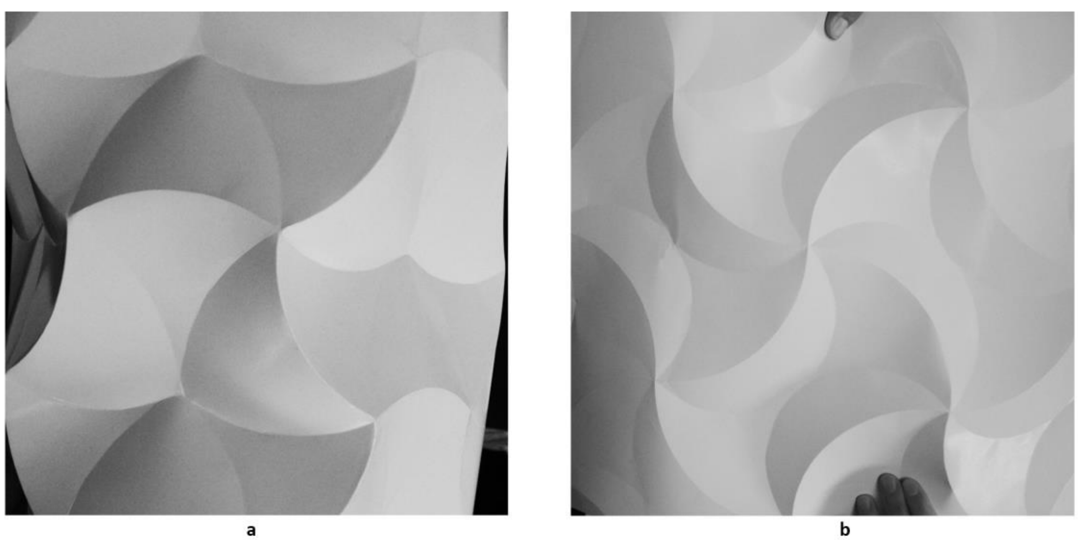

Figure 5: Hexagonal Boundary (a); N-gon Boundary (b).

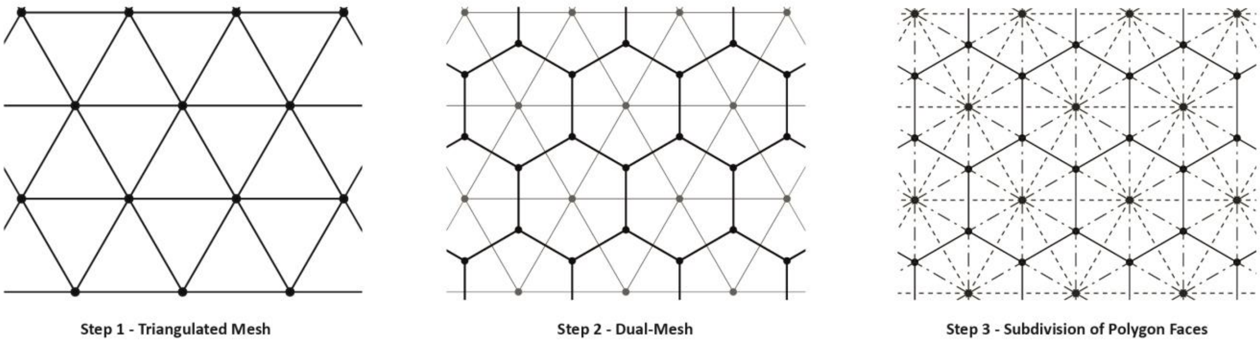

Figure 6: Steps of mesh tessellation: triangulation, dual-mesh and subdivision of polygon faces.

Although meshing algorithms implemented in commercial CAD and analysis software packages are able to generate meshes with regular faces (triangles close to equilateral or quads close to squares), they tend to create irregular faces adjacent to seams of surface patches that are not conducive to curved folding. (Figure 7)

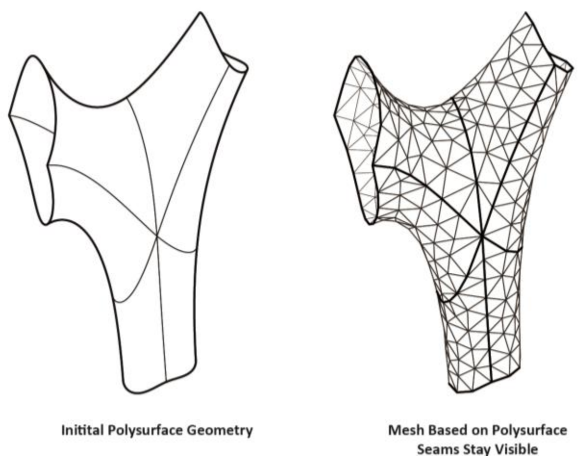

Figure 7: Mesh generated from nurbs patches.

To overcome this issue an alternative re-meshing approach is needed. The proposed strategy is based on a relaxation method using a particle spring system in combination with the half edge data structure [1]. This way of storing topological information of a mesh makes it easier to iterate through all adjacent edges, faces and vertices, and to locally modify the mesh topology, overcoming irregularities around seams without remeshing globally. Each edge of a given input triangulated mesh represents a spring with certain stiffness and adjustable predefined target length. The mesh therefore relaxes in a state of equilibrium where the distance between adjacent vertices is converging to an equal length and therefore the resultant mesh consists of close-to-equilateral triangles.

If the rest length is set to zero, the relaxation will lead to a minimal surface approximation between the boundary curves. To keep the information of the initial design geometry, the vertices are pulled towards the polysurface, acting as a secondary force during the relaxation process.

The edge length and aspect ratio of faces are evaluated after each relaxation iteration and edges are created or collapsed based on a given set of influencing rules and constraints, such as curvature of the base geometry (leading to smaller triangles in areas of high curvature while areas of low curvature will be populated with larger triangles) or refinement of boundary and edge conditions. The connectivity between vertices can be influenced by either emphasising consistent angles between adjacent edges (leading to more homogenous appearance) or equal valence of vertices (leading to a hexagon dominated dual-mesh) (Figure 8). After convergences are reached (approximately 100 - 200 iterations) the parameters can be modified and the system stays interactive during the relaxation process. We chose to weigh the relaxation towards maintaining consistent angles between edges as it is not possible to ensure a mesh in which all vertices have a valence of 6, meaning the dual mesh consist of only hexagonal faces. The used computational system is based on the particle spring system Kangaroo Physics 0.099, developed by Daniel Piker for Grasshopper 3D, a visual programming language for Rhinoceros 3D and the plankton half edge mesh library, developed by Daniel Piker and Will Pearson. Our algorithm was written in C# and python within the grasshopper environment.

Approximating Curved Creases

To convert a polygonal-faced mesh to the curved-folded geometry, we use a series of geometric relationships between the mesh polygons and the n-gon boundary curved folded crease pattern as described above, which enables an easy transition between the simplified triangular mesh and all its derivatives (Figure 9).

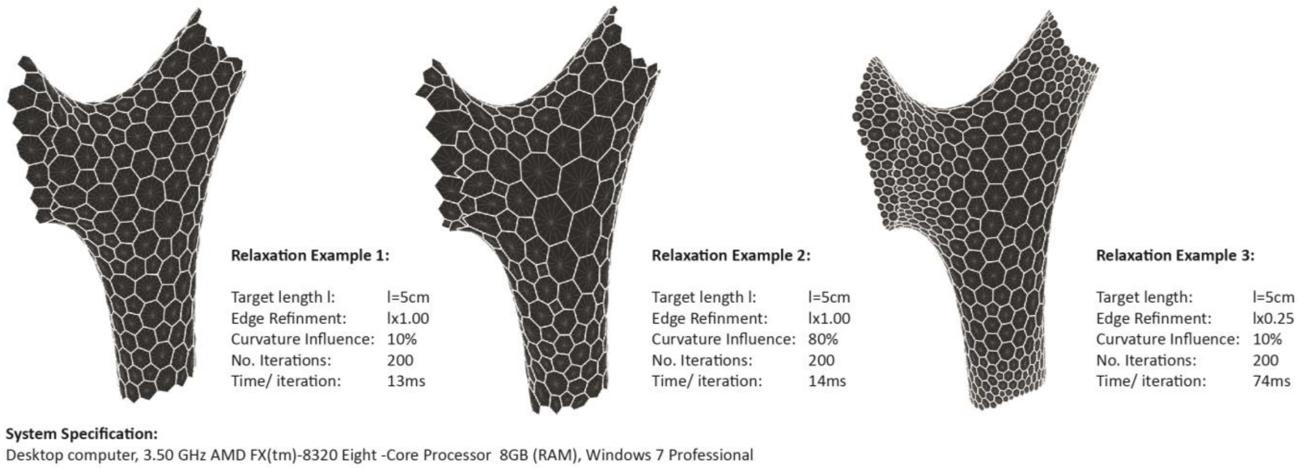

Figure 8: Mesh Relaxation under different influencing parameters.

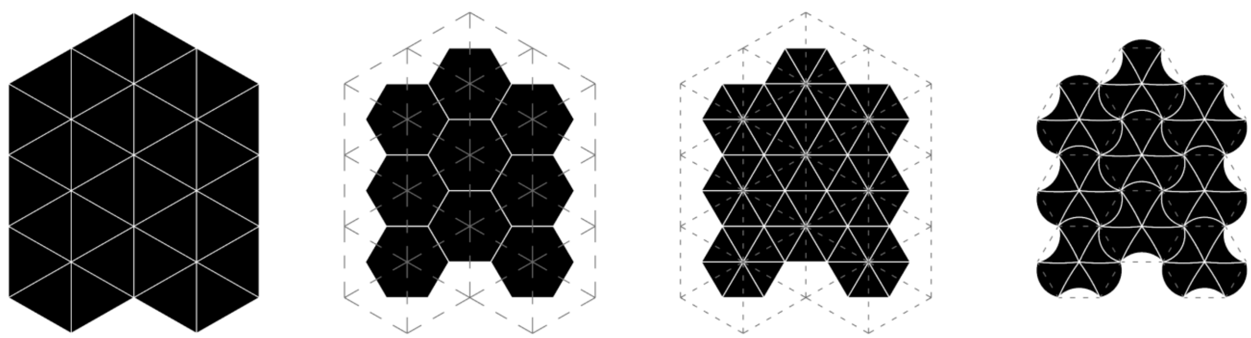

Figure 9: A triangular grid and its derivatives: it's polygonal dual, a triangular-subdivision mesh, and the curved-fold pattern.

This provides an initial estimation of the curved creases on the three-dimensional geometry, the curvature direction of creases and the mountain-valley assignment of folds. It is important to mention that because each curved-crease replaces a straight edge from the triangular-subdivision mesh, we assume the distance between its endpoints remain constant even when folded. Using this in combination with the method of reflection [8], we are able to compute the correct orientation of each curved crease by iteratively rotating them about the straight-line connecting their endpoints until the surfaces on either side of the crease are reflections of each other. Figure 10 illustrates this for a high and a low curvature crease.

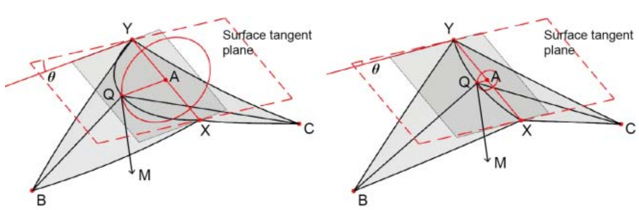

Figure 10: Rotating creases about the straight-line connecting their endpoints

The points X and Y represent the straight-line that connects the endpoints of the curved crease XQY, and points B and C are the centres of the polygons adjacent to the edge XY in the polygonal mesh. The curved crease XQY is iteratively rotated about the chord XY by an angle θ until QM bisects the angle BQC, QM being normal to the plane of XQY.

We implemented this method in Rhinoceros 5.8.4 using C# in Grasshopper 0.90076 without multi-threading or using any third-party plugins. To test its performance, we used a sample mesh of 25 n-gon faces with 95 creases on a laptop computer with 8GB of RAM and a 1.6GHz Intel Haswell processor with Windows 8.1. The solution took 17ms to compute on average with each crease being iteratively rotated 57 times on average. Some creases were rotated only once and some up to a 120 times, and the longest solving time for the method was 19ms. The computing time appears to increase linearly with 50 faces taking 34ms and 100 faces taking 68ms. To solve much larger meshes, the method could be multi-threaded with relative ease as it solves each crease independent of all others, thereby being very parallelizable.

Figure 11: Using a modified Prim's algorithm to produce developable strips of triangles.

Unfolding

A key difference between our method and the precedents we studied is that we unfold the underlying triangulated mesh instead of unfolding the curved folded dimensional geometry and convert the straight edges to curves as a two-dimensional operation. This eliminates the need to produce an accurate three-dimensional model of the folded geometry. Instead, the folded digital model only approximates the visual appearance of the object, which is critical to a designer.

As the base geometry is a freeform surface exhibiting Gaussian curvature, the triangulated mesh is required to be split into developable strips of contiguous mesh faces. We developed a custom version of Prim's algorithm (4) on the face-centres of the mesh which culls edges on the minimum spanning tree graph to ensure that all vertices have a maximum valence of two, effectively converting the minimum spanning tree to chains of faces (figure 11). In addition, our algorithm also takes into account if the unfolded faces in a chain overlap and modifies the graph accordingly. The strips of triangles are unfolded by a simple trigonometric method to maintain edge lengths. The flattened strips of triangles are converted to curved creases, applying the geometric relationships described in figure 9. The distance between the vertices of the triangles remain unchanged when converted to curves (figure 12), ensuring the strips fit together correctly upon folding.

It should be noted that in order to maintain interactivity of the design method, the unfolding is computed only once the tessellation and curved creases have been finalized. As the unfolding does not affect the appearance of the surface, excluding it from the interactive method results in speedier performance.

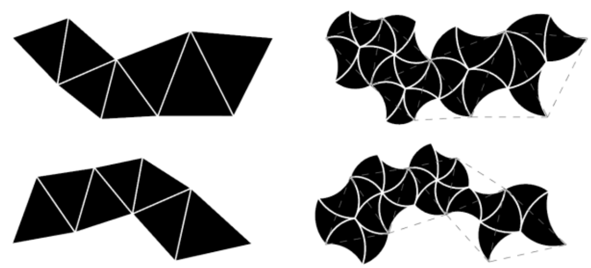

Figure 12: Converting strips of unrolled triangles to curved creases.

RESULTS AND DISCUSSION

The folding of a sheet material either causes or is caused by a change in its boundary conditions. In the case of curved folded tessellations, the depth of the folding is a direct consequence of the boundary condition. Irregular folds can often be caused by non-uniform forces applied at the boundaries, or in the case of cylindrical tessellations (figure 13), unidirectional forces applied on a multidirectional tessellation.

Another critical relationship is that of the curvature of base geometry and the size of the tessellation. Curvature causes each polygon in the tessellation to bend with respect to its neighbours, and this bending in turn causes the polygon to fold. Figure 13 shows two different tessellation sizes on the same boundary condition, one barely folds, and the other folds significantly.

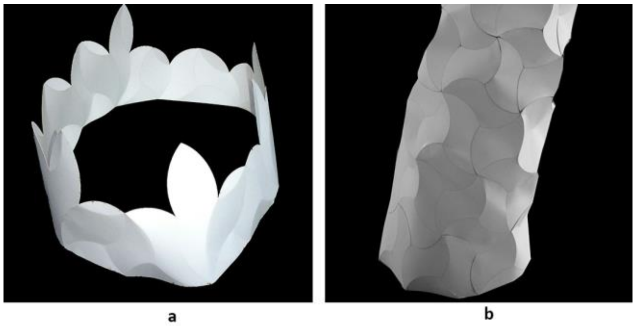

Figure 13: Boundary condition with small tessellation size (a), and the reworked boundary with large tessellations.

In our current process of approximating curved creases as mentioned in 2.3, we make a fundamental assumption that distances between vertices of the triangular mesh stays constant as the curved creases are folded. Although this is essential to how the process is setup, we have observed that this distance may reduce slightly during folding. To quantify this behaviour, we measured areas of unrolled triangular strips and compared them to the areas of their corresponding curved creased strips (figure 12), and observed a variation of 40-80% per strip. As is evident in figure 12, some of this can be attributed to each strip losing or gaining significant portions of triangles on being converted to curved creases. However, if this variation is summed up across all strips, the cumulative area change seemed to be less than 0.2%. Therefore, it appears that the final geometry produced will have a proportional shrinking, depending on the local arc curvatures of the tessellation pattern. A precise reconstruction of the three-dimensional geometry would improve our understanding of this phenomenon.

CONCLUSION

In this paper, we have demonstrated a method to design curved folded surfaces and a computational framework to apply them to freeform surfaces, producing fabrication data that ensures consistent assembly. Further, the method presented is applicable to any freeform surfaces that can be tessellated into convex polygons. Figure 14 summarises the method presented including future developments planned.

The domain of curved folding offers exciting potentials in architecture and manufacturing, but it remains largely unexplored. With the recent developments in computer hardware and the sophistication of tools available to architects and designers, it is becoming increasingly feasible to work with multi-constraint optimization algorithms such as those required for curved folding.

The method presented offers numerous opportunities to inform parameters such as folding depth, tessellation size, etc by external criteria such as required structural performance, material specific attributes, etc. We hope to further this work by enriching it with relevant contemporary research in related fields such as Gattas and You's recent analysis [3] of structural performance of fold-core panels demonstrated the benefits of panels with curved-folded cores over straight-fold ones highlights the potential of using curved folds to lend stiffness to surfaces.

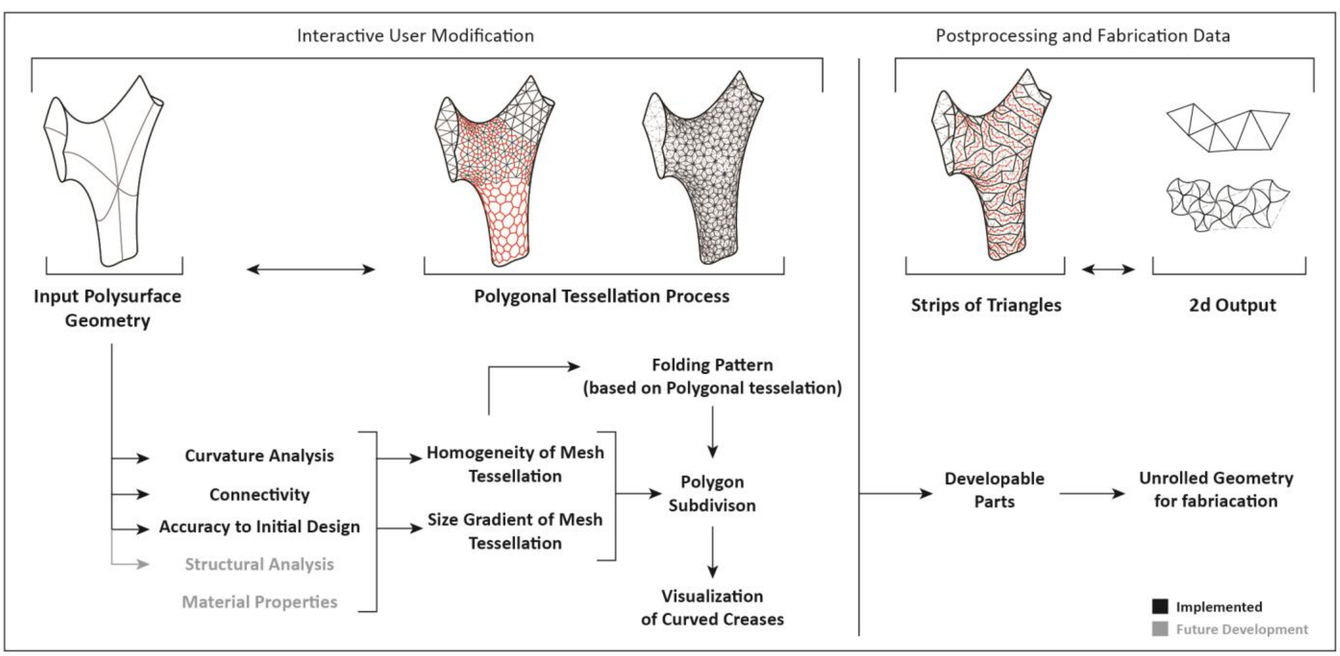

Figure 14. Logic Diagram of the proposed system including future development

ACKNOWLEDGEMENT

Portions of this research were undertaken in the Emergent Technologies and Design Programme at the Architectural Association School of Architecture during the 2013-2014 academic year.

Academic Supervisors:

Michael Weinstock, George Jeronimidis, Evan Greenberg and Mehran Gharleghi.

Workshop Coordinator:

Suryansh Chandra and Axel Körner

Design and Fabrication Team:

Radhika Amin, Amro AmerKabbara, Antiopi Koronaki, Samidha Kowli, Rachelle Spiteri, Arghavan Arjmand Nik, Quesada Arias, Rebecca Jane Bradley, Silvia Daurelio, Surbhi Bansal, Stanley Carroll, William John Haviland, Alessandra Lazzoni, Alessandro Guidetti, Anja Hein, Nikol Vergini.

The authors would also like to thank Zaha Hadid Architects Code Research Group for valuable inputs regarding the mathematical aspects of simulating curved folding.

REFERENCES

1. Campagna, S., Kobbelt, L. and Seidel, H.-P. Directed Edges - A Scalable Representation For Triangle Meshes. In Journal of Graphics Tools 3 (4), (1998).
2. Demaine, E. D., Demaine, M., & Koschitz, D. Reconstructing David Huffmans legacy in curved-crease folding. In Origami5, (2011), 39-51.
3. Gattas, J.M. and You, Z. The behaviour of curved-crease foldcores under low-velocity impact loads. In International Journal of Solids and Structures, (2014)
4. Greenberg, H.J., Greedy Algorithms for Minimum Spanning Tree, Univeristy of Colorado. (1998), online available at http://glossary.computing.society.informs.org/notes/spanningtree.pdf (Accessed November 22, 2014)
5. Hull, T. C. The Combinatorics of Flat Folds: a Survey. In arXiv:1307.1064 [math.MG], (2013).
6. Hull, T.C. Notes on Flat Folding. MA 323A Combinatorial Geometry! online available at http://mars.wne.edu/~thull/combgeom/flatfold/flat.html, (Accessed November 20, 2014).
7. Kilian, M., Flory, S., Chen Z., Mitra, N., Sheffer, A., Pottmann, H., Curved folding. In ACM SIGGRAPH 2008 papers, (2008)
8. Mitani, J., Igarashi, T., Interactive design of Planar Curved Folding by Reflection. In Pacific Graphics, (2011)
9. Solomon, J., Vouga, E., Wardetzky, M., Grinspun, E., Flexible developable Surfaces, In Eurographics Symposium on Geometry processing, (2012)
10. Tachi, T., Freeform Origami Tessellations by Generalising Resch's Patterns, In Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, (2013)