ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
A- APPLIED SCIENCES AND ENGINEERING

16th Digital Design In Architecture Symposium
16th DDAS (MSTAS) - Special Issue 2022

2022, Vol.23, pp. 76-85, DOI: 10.18038/estubtda.11699938

# FROM STITCHES TO DIGITS AND BACK: COMPUTATIONAL CROCHETING OF BRANCHING GEOMETRIES

Özgüç Bertuğ ÇAPUNAMAN [1,*], Cemal Koray BİNGÖL [2], Benay GÜRSOY [1]

[1] Department of Architecture, College of Arts and Architecture, The Pennsylvania State University, University Park, USA.
[2] Management in the Built Environment, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Netherlands.

## ABSTRACT

Crocheting is a hands-on craft that involves repetitive manipulation of a single continuous thread with a hook-like tool to generate surfaces and 3D forms. In a previous study, we have presented a parametric model [1] that generates crochet patterns of NURBS surfaces using a 10-stitches-by-10-rows swatch to account for all the physical variables that affect the crocheted object (i.e., yarn thickness, hook size, crafter's grip). The dimensions of the previously crocheted tension swatches were used as the inputs of the crochet pattern generator algorithm, alongside the desired NURBS geometry, to generate individualized crochet patterns. These crochet patterns are text-based representations, similar to g-code in additive manufacturing, enabling the documentation and communication of the step-by-step hands-on crocheting process. Following these crochet patterns, the users can crochet physical objects with the same dimensions and form as their digitally modeled counterparts.

This paper presents the second stage of this research in which we expanded this computational framework to enable crocheting of parametric branching geometries with multiple components by multiple crafters. While the components of the branching geometries can be crocheted by a single user, it is also possible to have different users crochet the components since the tension swatch can capture crafter-specific variables. As a proof-of-concept, a branching structure made of 14 unique components is designed and crocheted by two students of architecture as part of the Advanced Digital Fabrication course at the Pennsylvania State University. The students each crocheted 7 components based on their individual inputs while maintaining the dimensions and form of the digitally designed branching geometry. The findings suggest the possibility of a collective and distributed crocheting platform which can be used to create crocheted artifacts in various scales. This can be considered an alternative way to transition from the digital to the physical without relying on digital fabrication tools.

**Keywords:** Digital craft, Computational making, Crocheting, Soft fabrication, Digital fabrication

## 1. INTRODUCTION

Crocheting is a hands-on-craft technique to produce 3D surfaces by stitching a single continuous thread with a hook-like tool based on instructional patterns. The procedural nature of the craft is similar to g-code in digital fabrication, especially additive manufacturing, due to the defined steps at each manipulation. [2-5] The text-based representation of the crochet patterns also allows documentation and communication of the step-by-step hands-on crocheting process.

In a previous study, we have explored the development of a computer algorithm that generates crochet patterns of single 3D objects modeled in CAD software [1]. In this computational framework, the users first crochet a 10-stitches-by-10-rows swatch. The dimensions (width and length) of the swatch are used as the main inputs of the algorithm, combining all the physical variables that have an effect on the crocheted object (i.e. yarn thickness, hook size, crafter's grip). Based on these dimensions, the algorithm subdivides the 3D modelled objects into nodes, and outputs a crochet pattern in the conventional text-based form. In other words, the algorithm generates unique crochet patterns based on each individual's

*Corresponding Author: okc5048@psu.edu
Received:04.09.2022 Published:23.12.2022

crocheted swatch. Following these crochet patterns, the users are able to crochet physical objects with the same dimensions and form as their digitally modelled counterparts.

This paper presents the second stage of this research in which we expanded this computational framework to enable crocheting of parametric branching geometries with multiple components. As a proof-of-concept, a branching structure made of 14 unique components is designed and crocheted. The findings suggest the possibility of a collective and distributed crocheting platform that can be used to create crocheted artifacts in various scales. This can also be considered an alternative way to transition from the digital to the physical, one that allows precise, circular, and sustainable materialization of digitally modeled objects.

## 2. CROCHETING: A SOFT FABRICATION METHOD

Crocheting is a textile craft that originated in the 19th century and is similar to knitting. While crocheting as a soft fabrication method in design has not been extensively explored beyond traditional crafts, there are research on delineating the algorithmic nature of crocheting and the potentials of this technique for making architectural artifacts [3–6]. Architectural potentials of knitting, specifically industrial knitting, on the other hand, have been widely explored through various large-scale projects [7–12]. Researchers have also recently explored the use of knitted textiles as formworks for concrete structures in various scales [13–15]. These studies can be considered a testimony for the architectural potentials of crocheting and the need to further explore the causal relations between form and pattern generation to design the crocheted forms in a more controlled way.

The fundamental difference between crocheting and knitting is in the way the stitches are constructed. While knitting stitches are "interlocking loops," which can make the knitted surfaces multidirectional and flexible, crochet stitches are "knots," which make the crocheted objects more solid and sturdy. Also, crocheting is done one stitch at a time as opposed to knitting, where all the stitches stay active on the needles until the rows are completed, making the knitted surfaces more susceptible to unravelling. With both techniques, it is possible to make planar surfaces and alter their shapes by increasing and decreasing the number of stitches. While with standard knitting, 3D geometries can be made by joining various planar knitted panels together or by introducing more needles in the process; by working in the round in crocheting, it is possible to make 3D geometries without the need for additional hooks or a panel construction [16]. As a matter of fact, in addition to Euclidean geometries, non-Euclidean geometries can be made with crocheting. Several mathematicians have used the crocheting technique to physically represent complex mathematical models and theories [17, 18].

Existing studies that explore computer-aided crochet pattern generation for 3D objects, such as the Crochet Lathe [19], and Knittink's Amigurumi Pattern Generator [20], enable the users to manipulate 2D profile curves to generate crochet patterns for revolved surfaces. In both crochet pattern generators the outcomes are limited with axially symmetric 3D objects. In a previous study, we have presented the computational framework to generate custom crochet patterns for various non-symmetric 3D objects [1]. This paper builds on this previous study and expands this framework to generate crochet patterns for branching structures with multiple components.

In addition to the formal complexity that can be achieved through crocheting, as a soft fabrication method, crocheting can enable more circular and sustainable fabrication scenarios. Yarns used in crocheted artifacts can be unraveled and reused multiple times to generate various artifacts. The potential to use crocheted textiles as lightweight structures or flexible formworks in architecture can as well open up sustainable construction possibilities.

## 3. CROCHET PATTERNS FOR PARAMETRIC BRANCHING GEOMETRIES

Branching structures are based on geometric systems that "expand through bifurcation without returning to form closed cells," and can be used as tension or compression systems in architecture. Various methods have been developed to generate branching geometries since the initial studies by Frei Otto from the early 1960's [21]. In the computational framework that we propose in this paper, branching geometries are generated in three stages. The digital workflow starts with the generation of point clouds in the 3D space (Figure 1a). These points are connected with single line segments to create branching line networks (Figure 1b), which are then transformed into continuous tubular surfaces (Figure 1c). These tubular surfaces are subdivided into branches (Figure 1d) and custom crochet patterns are generated for each branch based on the user inputs (Figure 1e-f).

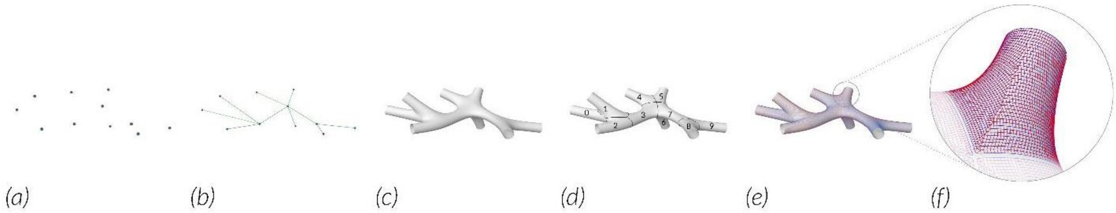


**Figure 1.** Process to generate branching geometries and crochet patterns simplified and illustrated in steps.

### 3.1. From Point Clouds to Line Networks

In the computational workflow, we defined three strategies to generate point clouds in the 3D space: a) random points within bounding solids, b) 3D grid-based ordered points, and c) points created on a surface by surface division (Figure 2). Each cluster of point clouds can be varied with parametric inputs that control the dimensions of the geometries and the number of points generated. Random point clouds have the potential to generate more irregular branching geometries. Whereas with grid-based ordered points, it is possible to obtain complexity through repetition. Points generated on surfaces, on the other hand, enable the creation of branching geometries that are constrained on surfaces.

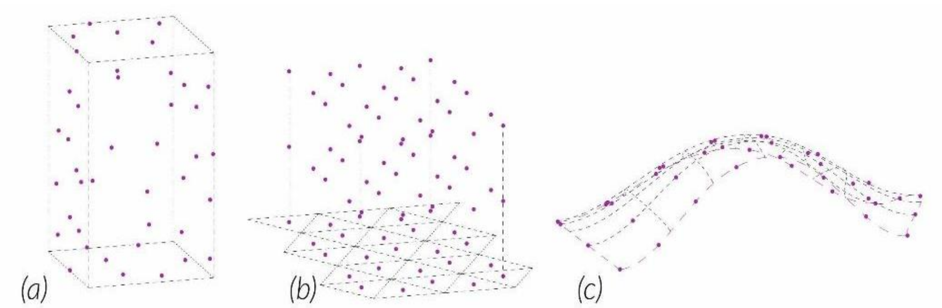


**Figure 2.** Point cloud generation approaches explored: a) random points in/on a bounding solid, b) 3D grid, and c) points on surface.

The second stage in the process is to connect the points with single-line segments. While this can be done manually, we employed two algorithmic strategies to facilitate the exploration of variations. The first method uses the distances between each node, the maximum number of nodes that can be connected, and the number of iterations for the spread of branches (Figure 3b).

The algorithm uses the closest points to the volumetric centroid as the seed point to spread out at each iteration of branching. It searches the point cloud to find the number of closest points to each point, then checks the number of possible points within a parameterized proximity. To further increase the control over branching and spread, a starting point for the algorithm is introduced. This is done to help the users see the volumetric centroid of the point cloud and select the closest point to this point within the point cloud for an initial branch. By defining the initial point, the users are able to see the branching at each step.

The second algorithm is developed and shared by Petras Vestartas at McNeel forums [22]. The algorithm connects point pairs in the point cloud regarding their proximity and generates points at the middle of each point pair (Figure 3a). The generated middle points are connected with line segments until there are no remaining points to connect. The connected point pairs then get relaxed by the physics engine of the Kangaroo, an add-on of Grasshopper. The relaxed branching structure generates line connection points where the angles between the lines are almost equal and close to 120 degrees. This relaxed geometry allows the construction of surfaces with more uniform curvatures in the subsequent stages, compared with the non-relaxed state and the former algorithm. Figure 3 illustrates how these two algorithms work to generate line networks using the three point cloud examples from Figure 2 as inputs.

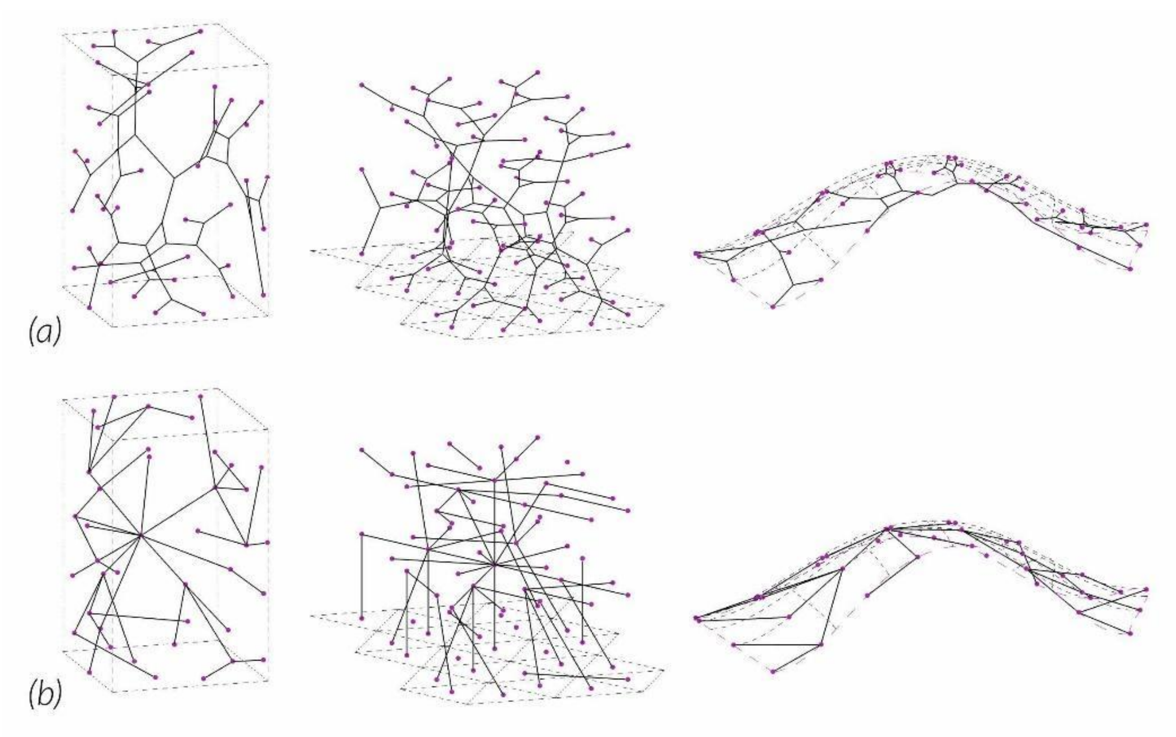


**Figure 3.** Line networks generated using two different algorithms using the point clouds in Figure 2.

### 3.2. From Line Networks to "Multipipes"

In the last stage, the "MultiPipe" component implemented within Rhinoceros 7 with the recent updates is used to create "SubD pipe frames with smooth conjunctions from intersected curves" [23]. With the Multipipe component, it is possible to vary the thicknesses of the branches and nodes, and generate smooth connections between the branches. Figure 4 exemplifies some Multipipes created using the line networks from Figure 3.

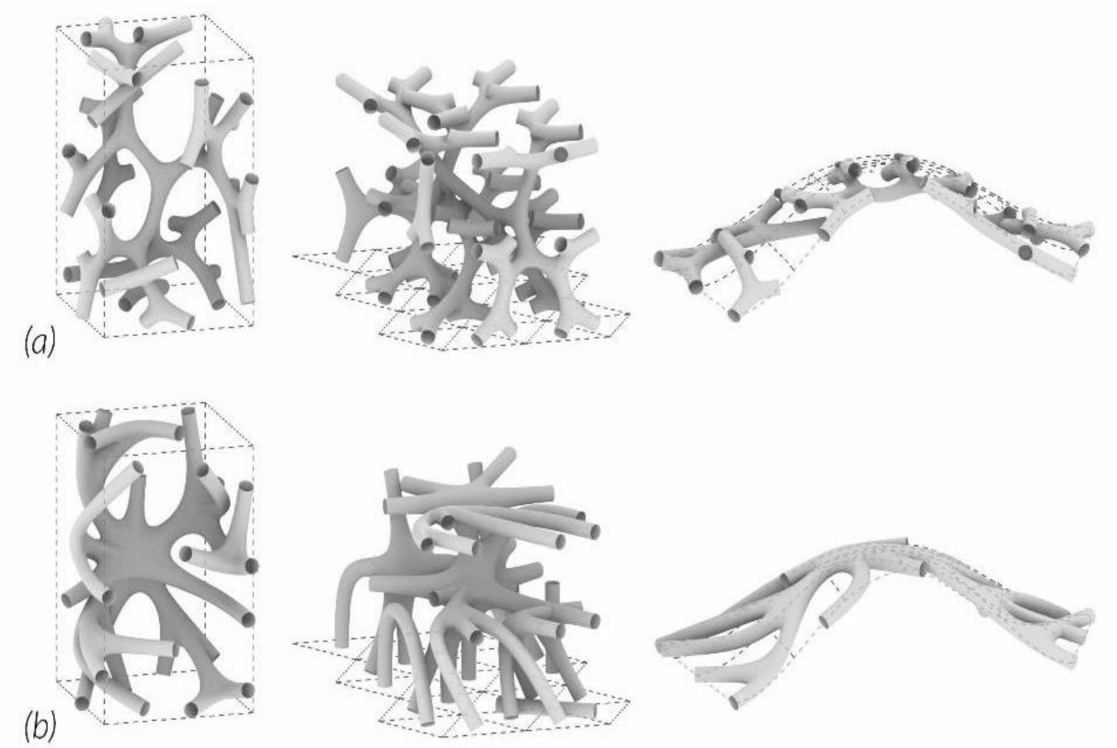


**Figure 4.** MultiPipe surfaces generated based on the line networks in Figure 3.

The crochet pattern generation algorithm that we have previously developed was based on the NURBS geometry class. SubD is a new geometry class in Rhinoceros 7 that "combines free-form accuracy while allowing quick editing" [24]. Since Multipipe is a SubD component, we needed to develop a method to convert SubD geometries to NURBS geometries. This conversion resulted in branches with multiple SubD surfaces that needed to be restructured as a single NURBS polysurface (Figure 5a) because the crochet pattern generation algorithm works best with single NURBS surfaces. This reconstruction is done in multiple steps. First, cylinders are generated around the branching curves to test whether the centroids of the SubD surfaces are within the cylinders or not (Figure 5b). This allowed the data structure of the branching structure to be mapped on each branching curve. SubD surfaces that are within the cylinders are grouped together. To reconstruct these as NURBS surfaces, planes are arrayed rotationally around branching curves and intersected with the grouped surfaces (Figure 5c). The emerging intersections are combined with the edge curves to create precise NURBS surfaces using the network surface command (Figure 5d). The reconstruction method also serves to check the crochetability of the geometries.

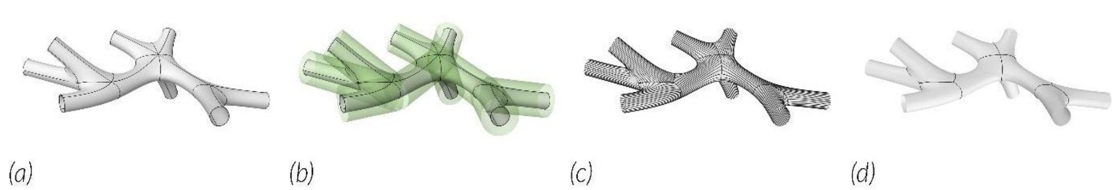


**Figure 5.** SubD to NURBS surface reconstruction: a) discrete SubD surfaces, b) cylinders around branching curves, c) intersection curves, (d) reconstructed NURBS surfaces.

### 3.3. From "Multipipes" To Crochet Patterns

Following successful reconstruction of individual branches within the larger structure, these NURBS surfaces are further processed to generate crochet patterns. As previously mentioned, the algorithm to generate crochet patterns from NURBS surfaces is based on the approach presented in our previous work. Together with the 10-stitches-by-10-rows swatch inputs by the users to approximate stitch width and height, this algorithm estimates a graph network of individual stitches which can later be used to export a series of crocheting instructions for each row, customized for the crafter. To achieve this graph network, the algorithm performs a series of geometric decomposition steps on the input NURBS surface. These steps are (b) generation of a spiral conformed to the NURBS surface, (c) division of the spiral into rows, (d) division of the rows into crochet nodes, and lastly (e) generation of the graph network between crochet nodes, as shown in Figure 6.

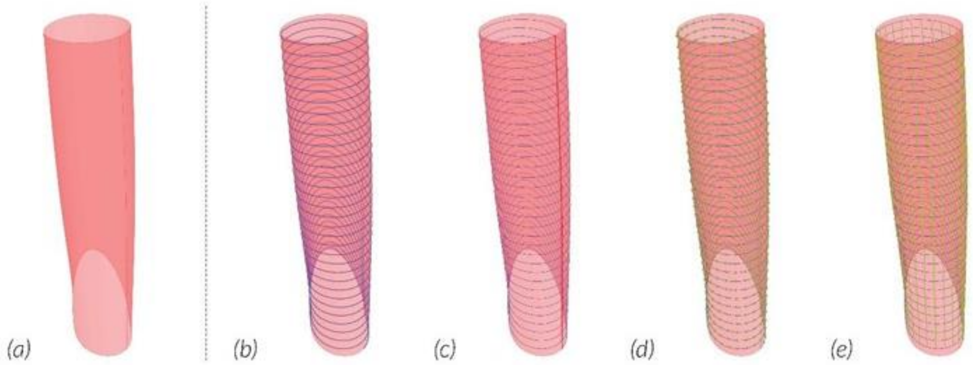


**Figure 6.** Computational stages of the pattern generation algorithm: (a) input surface, (b) generating the conformal spiral, (c) splitting the spiral into rows, (d) calculating crochet nodes, and (e) generating the graph network.

In the first step, a conformal spiral is drawn on the NURBS surface using UV isocurve intersections sampled based on the calculated stitch height. Following this step, the generated spiral is first split into rows using UV(0,0) isocurve and then divided into crochet nodes that are spaced out per the calculated stitch width. Once the nodes are generated, every node in a layer is compared to the nodes of the subsequent row. This is achieved by first establishing the node pairs between $row_n$ and $row_{n+1}$ by searching for closest nodes in $row_{n+1}$ for each node in $row_n$. In cases where this proximity pairing results in unmatched nodes, meaning that the surface geometry is bulging out locally or globally between layers in the direction of crocheting, another proximity pairing is carried out. In contrast to the first pairing, however, the order of rows is reversed, searching for closest nodes in $row_n$ for all unmatched nodes in $row_{n+1}$.

Using this approach, the graph network of node connectivity between layers results in one of the three different stitches: single stitch (st), increase stitch (inc), decrease stitch (dec). By drawing lines based on the connectivity between nodes in $row_n$ and $row_{n+1}$, this graph network is visualized for the user. Furthermore, by counting the number of nodes in $row_{n+1}$ that diverge from a node in $row_n$ and the number of nodes in $row_n$ that converge into a node in $row_{n+1}$ a text-based representation of the crochet pattern can also be exported by the user.

Although the computational approach to generating crochet patterns closely follows the approach detailed in our previous work, numerous changes were made to the script to address various issues we have identified. One major intervention to the script was to implement the algorithm in IronPython to enable more efficient computation of the crochet pattern. This was achieved mainly by switching to a dictionary-based connectivity graph which enables efficient storage and lookup of each individual node as opposed to geometrically calculating the number of lines connected to each node. In addition to computational efficiency, the other area of focus was to improve how the generated information is presented and communicated to the users. For this, a simple graphical user interface (GUI) was integrated into the script that allows users to preview and isolate the generated crochet pattern for (a) entire geometry, (b) a single branch and (c) single row (Figure 7). Additionally, exporting the text-based crochet pattern was also reworked allowing the users to select between a verbose or simplified version exemplified in Figure 8.

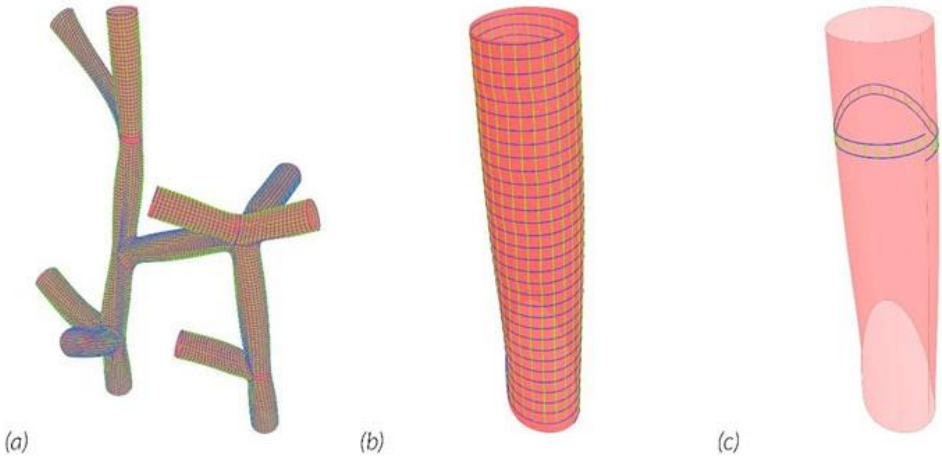


**Figure 7.** Computational stages of the pattern generation algorithm: (a) input surface, (b) generating the conformal spiral, (c) splitting the spiral into rows, (d) calculating crochet nodes, and (e) generating the graph network.

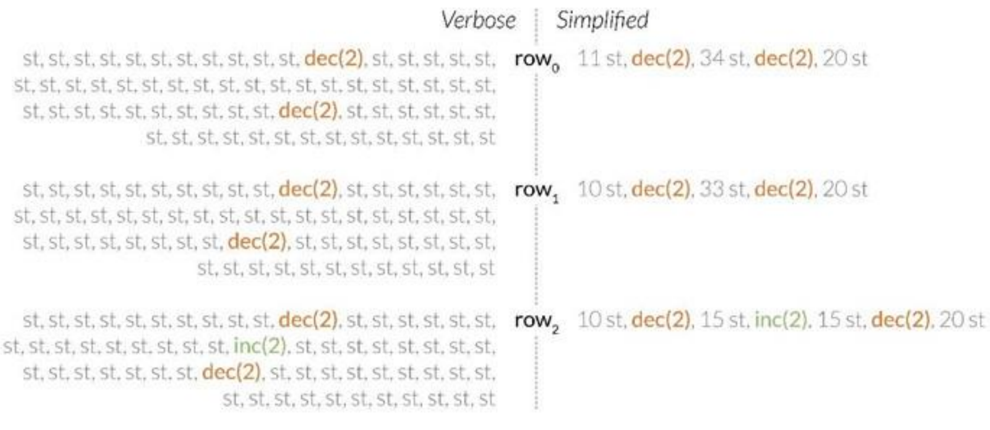


**Figure 8.** Computational stages of the pattern generation algorithm: (a) input surface, (b) generating the conformal spiral, (c) splitting the spiral into rows, (d) calculating crochet nodes, and (e) generating the graph network.

### 3.4. Proof-Of-Concept Crocheted Branching Structure

A proof-of-concept branching structure is crocheted by two students of architecture as part of the Advanced Digital Fabrication course at the Pennsylvania State University. The students generated the branching geometry shown in Figure 7a following the computational framework outlined above, for an exhibition to showcase various works from the course. Figure 9 shows the final crocheted structure as part of the course exhibition. The students first crocheted 10-stitches-by-10-rows swatches. The dimensions of these individual swatches were used as the inputs of the crochet pattern generator algorithm to generate individualized crochet patterns for each user. This way, the students were able to each crochet seven components based on their individual inputs while maintaining the dimensions and form of the digitally designed branching geometry. These components were crocheted together, filled with polyester fibers, and attached to / placed within acrylic boxes via crocheting. Both students had little experience in crocheting. They reported that they had spent around 1 hour crocheting each component, totaling around 7 hours of collective work to crochet the branching structure. The students also reported that the text-based crochet patterns were easy to follow, along with the digital interface that visualizes the stitches on each row.

**Figure 9.** Proof-of-concept crocheted prototype as part of an exhibition and a close-up view.

## 4. DISCUSSION AND CONCLUSION

This paper's focus is on the expanded computational framework that allows the users to generate individual crochet patterns for parametric branching geometries and the user interface developed to allow easy tracking of the stitches and rows on each component. While the components of the branching geometries can be crocheted by a single user, it is also possible to have different users crochet the components. In an upcoming publication, we will present the process and outcomes of an online design research workshop that we conducted with 20 participants from different locations around the world who collectively designed a branching structure and individually crocheted its components. This shows that the computational framework outlined in this paper can be used as a collective and distributed crocheting platform that allows the creation of large-scale crocheted artifacts.

Both crocheting and knitting are sustainable soft fabrication methods. Instead of ending up in landfills, the artifacts created through crocheting and knitting can be unraveled and yarns can be reused to generate new artifacts. Construction industry can also benefit from the circularity of these fabrication techniques. Both crocheting and knitting can be used to create lightweight tension structures and flexible formworks in architectural scales. While there are prominent examples of such applications in architecture with knitted textiles, crocheting as an architectural soft fabrication method is not explored yet. This is partly due to the possibility to automate knitting with industrial knitting machines. Industrial knitting machines that automate the knitting process can knit large and complex surfaces at once. However, there is no crochet machine developed to date to automate the crocheting process, and all crocheted artifacts are currently hand made. One reason behind this is the difficulty of simulating the complex and simultaneous hand movements necessary in crocheting stitches, so that it can be replicated by a machine. One possibility of automating the crocheting is through the integration of robotic arms in the process. We believe our research on computationally generating the crochet patterns can inform the robotic automation of the crocheting process.

Another future goal is to develop an open web-based user interface that can allow users who do not have access to (or are not proficient in) CAD software to easily generate 3D forms and obtain custom crochet patterns. Similar platforms exist to generate custom g-codes for FDM printing and paste extrusion (i.e. Potterware, SliceUp). This way, crocheting can become an alternative way to precisely transition from the digital to the physical without relying on digital fabrication tools.

## CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this article.

## ACKNOWLEDGEMENTS

The authors would like thank Puja Bhagat and Xi Jin and acknowledge their contribution in fabricating the generated patterns showcased in this paper.

## REFERENCES

[1] Çapunaman ÖB, Bingöl CK, Gürsoy B. Computing Stitches and Crocheting Geometry. Çağdaş G, Özkar M, Gül LF, Gürer E, editors. Computer-Aided Architectural Design. Future Trajectories Springer; 2017; 289–305. https://doi.org/10.1007/978-981-10-5197-5_16.

[2] Bernasconi A, Bodei C & Pagli L. (2007). Knitting for Fun: A Recursive Sweater. In P. Crescenzi, G. Prencipe, & G. Pucci (Eds.), Fun with Algorithms (pp. 53–65). Springer. https://doi.org/10.1007/978-3-540-72914-3_7

[3] Taimina D, Baurmann G. Crocheting Algorithm. Cornell Journal of Architecture 2013.

[4] Worden A. Emergent Explorations: Analog and Digital Scripting. Thesis. Virginia Tech, 2011.

[5] Kenning G. Crochet Lace as Expression of Digital Culture. Textile Society of America Symposium Proceedings. 2008.

[6] Kucukoglu JG, Colakoglu B. Algorithmic Form Generation for Crochet Technique. Proceedings of the 31st eCAADe Conference 2013; 2: 273–8.

[7] Ahlquist S, Menges A. Frameworks for Computational Design of Textile Micro-Architectures and Material Behavior in Forming Complex Force-Active Structures. Proceedings of the 33rd Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) 2013; 281–92.

[8] Baranovskaya Y. Knitflatable Architecture - Pneumatically Activated Preprogrammed Knitted Textiles. Proceedings of the 34th eCAADe Conference 2016; 1: 571–80.

[9] McKnelly CL. Knitting behavior : a material-centric design process. Thesis. Massachusetts Institute of Technology, 2015.

[10] Oghazian F, Vazquez E. A Multi-Scale Workflow for Designing with New Materials in Architecture: Case Studies across Materials and Scales - Case studies across materials and scales. Proceedings of the 26th CAADRIA Conference 2021; 1: 533–42.

[11] Sabin J, Pranger D, Binkley C, Strobel K, Liu J. Lumen. Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) 2018; 444–55.

[12] Thomsen MRT. Knit as bespoke material practice for architecture. ACADIA // 2016: POSTHUMAN FRONTIERS: Data, Designers, and Cognitive Machines [Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) 2016; 280-289

[13] Pal A, Chan WL, Tan YY, Chia PZ, Tracy KJ. Knit Concrete Formwork. Proceedings of the 25th CAADRIA Conference 2020; 1: 213–22.

[14] Popescu M, Reiter L, Liew A, Van Mele T, Flatt RJ, Block P. Building in Concrete with an Ultra-lightweight Knitted Stay-in-place Formwork: Prototype of a Concrete Shell Bridge. Structures 2018; 14: 322–32. https://doi.org/10.1016/j.istruc.2018.03.001.

[15] Popescu M, Rippmann M, Liew A, Reiter L, Flatt RJ, Van Mele T, et al. Structural design, digital fabrication and construction of the cable-net and knitted formwork of the KnitCandela concrete shell. Structures 2021; 31: 1287–99. https://doi.org/10.1016/j.istruc.2020.02.013.

[16] Spiers M. Knitting vs. Crochet: what's the difference? 2021. https://www.gathered.how/knitting-and-crochet/crochet/crochet-vs-knitting-difference/

[17] Henderson DW, Taimina D. Crocheting the hyperbolic plane. The Mathematical Intelligencer 2001; 23:17–28. https://doi.org/10.1007/BF03026623.

[18] Osinga H, Krauskopf B. Crocheting the Lorenz Manifold. The Mathematical Intelligencer 2004; 26. https://doi.org/10.1007/BF02985416.

[19] Avtanski A. Crochet Lathe2016. http://avtanski.net/projects/crochet/lathe/.

[20] Monroe A. Knittink's Amigurumi Pattern Maker Falls Short 2010. https://audreymonroe.wordpress.com/2010/10/05/knittinks-amigurumi-pattern-maker-falls-short/

[21] Von Buelow P. A Geometric Comparison of Branching Structures in Tension and in Compression versus Minimal Paths 2007.

[22] Vestartas P. Minimal Path Networks. McNeel Forum 2019. https://discourse.mcneel.com/t/minimal-path-networks/57158/32.

[23] MultiPipe 2021. https://docs.mcneel.com/rhino/7/help/en-us/commands/multipipe.htm.

[24] SubD 2021. https://www.rhino3d.com/features/subd/.