

SE390 Design Project Planning F23 Mini Project 3 - CrochetCraft

Simran Thind, *University of Waterloo, SE25*

Jasmine Tai, *University of Waterloo, SE25*

Dhananjay Patki, *University of Waterloo, SE25*

Allen Liu, *University of Waterloo, SE25*

Osman Wong, *University of Waterloo, SE25*

Abstract—We conducted a literature review which analyzes various fabric modelling research implementations and methodologies, particularly for knitting and some for crochet arts, in the hopes of learning processes that can be applied to our idea (making an AutoCAD application for 3D crochet designs) for the Software Engineering Final Year Design Project.

Crochet is an ancient textile craft that can be applied to a variety of objects, from making clothing, to [granny squares](#), to *amigurumi* (which is the Japanese practice of creating small, stuffed yarn characters) - it has even appeared in the practice of [yarn bombing](#)! At the core, crochet is an accessible art because it does not require expensive resources, can be easily taught, and allows for creativity with regard to shape, colour, and texture. While similar to knitting, crochet instead mainly uses one hook or needle in place of two needles, and that each stitch (at least in the common forms of crocheting) is closed before the next stitch is started. Given the aforementioned benefits, crochet is quite popular all around the world. However, there is a lack of tooling and support to help people involved in crochet design patterns and holistic products. While there is a glut of research related to computational knitting, there are not so many papers/talks/etc. about applying crochet to graphics, rendering, and algorithmic contexts. The team's goal for this literature review is to take stock of the current landscape of computational knitting and crocheting, and to learn from previous work such that we can understand where to begin and how to plan creating an application which would allow a user to input a crochet design, and would receive an image of the resulting textile in three dimensions (3D). Such an application would be useful to different people, like pattern/clothing designers so they can iterate faster

without using more physical material, and novice crocheters who would be able to amend patterns to suit their experience level.

3D MODELLING FOR KNITTING

In examining digital modelling and 3D rendering for crochet, it is also useful to examine a similar yet different yarn-based textile creation: knitting. Knitting, while mechanically different from crochet, does share the fundamental trait of creating fabric out of a single string of yarn. Thus, as we will observe later, some methods of modelling knit products may be adapted to crochet as well.

Stitch Meshes

Perhaps the most important method of modelling knit fabric is the stitch mesh. A stitch mesh, first introduced by Yuksel et. al in 2012 [1], is a method of representing stitch types and connections in knitted fabrics. These stitch meshes represent a single stitch as a polygonal tile, where each edge of the tile represents a connection to another stitch. These edges can be classified into wale edges, connections to the same row of stitches, or top edges, connections to other rows of stitches. This allows for specific modelling of various stitch types, while allowing the single stitches to be joined via their edges to model the whole structure. Stitch meshes constitute one of the most important ways to represent knit fabrics, and have been the basis for further exploration by other researchers, as we will examine shortly.

Knittable Stitch Meshes

In their 2019 paper *Knittable Stitch Meshes* [2], Wu et. al. (a team that included Cem Yuksel, from the original stitch mesh paper) proposed improvements to the original stitch mesh in order to ensure that the result would be fully hand-knitable. The original stitch mesh was mainly used for modelling purposes and often neglected knittability, resulting in unknitable models unsuited for practical use in knitting. To remedy this, Wu et. al. added representations of three knitting features. The first, shift-paths, is used for direct length-wise connections of adjacent rows. These are commonly seen in tubular knits, composed of a single row spirals in a helical fashion as opposed to several discrete parallel rows. The second is better support for mismatched directions, where the "directions" of adjacent stitches are not the same and the stitches do not naturally interlink. When this occurs in practice, the stitches are joined via one of several knitting techniques, which this paper proposed stitch mesh representations for. Finally, Wu et. al. propose a representation for short-rows, where a "short" partial row is inserted in between two complete rows, displacing the rows relative to each other. This is commonly seen when knitting and joining curved surfaces. Finally, the paper made one other notable proposition: an algorithm to turn a knittable object into step-by-step knitting instructions.

Wu et. al. acknowledge that their enhanced stitch mesh framework is unable to address some circumstances. Notable limitations include a reliance on an input mesh that may be difficult to create for unusual shapes, inability to account for variations originating from the forces applied by specific knitters, and a lack of support for limitations imposed by certain knitting machines. The first limitation has been addressed for the initial stitch mesh iteration, but not yet the updated framework, in a previous paper by a team also containing Wu and Yuksel.

Stitch Meshes from 3D Models

In the aforementioned paper, *Stitch Meshing*, Wu et. al. introduce a pipeline that takes an arbitrary 3D model and uses it to generate a stitch mesh, which can then be used to create a yarn-level rendering. The process begins by generating a mesh consisting of only triangles or quadrilaterals, ideally consisting mostly of quadrilaterals. Each edge's type (top or wale) is then determined, which in turn defines an optimization problem whose solution indicates the knitting direction. With this information available to be considered, a stitch mesh can now be generated via subdivision operations. This stitch mesh can then be used to

render yarn-level models.

Another paper was also published by Wu, Yuksel, and others that details generation of a knit model with a focus on two previously neglected aspects - feasibility for machine knitting and wearability. In this context, wearability is defined by the ability to put on and remove a knit covering a section of an object's surface without deforming the object. This paper, like the previous one, also uses stitch meshes to model the knit product as an intermediate step between the 3D object and the final product. The techniques introduced here are useful for dealing with unconventional objects and for applying 3D modelling to automated knitting.

Visual User Interfaces

Finally, in *Visual Knitting Machine Programming*, Narayanan et. al. devise a visual editor for machine knitting. Like in the other papers, the stitch mesh is used as a base, generated from an input mesh. A GUI is then provided for users to edit the stitch mesh as desired. The interface also provides some support for cosmetic features such as colours and textures. Also unique, the stitch mesh is enhanced via addition of machine-knitting related instructions, such that the stitch mesh can be easily be used by a knitting machine once modified to satisfaction by the user. While still in its early phases, this visualizer is a major development in the usage of 3D modelling and visualization for knitting in general and machine knitting in particular.

FROM KNITTING TO CROCHET

Representing Crochet with Stitch Meshes

In Guo et. al's work, *Representing Crochet with Stitch Meshes*, they use a novel technique to model 3D structures with stitch meshes, which are tiles that encapsulate the various geometric properties of the yarn. By using these meshes and providing a method by which several common crochet stitches can be converted to such tiles, they are able to model what some 3D structures would be like when converted to yarn and crochet. Additionally, it is possible to use their method to physically construct a 3D object, which are remarkably similar to their theoretical counterparts.

What is notable about the technique described in the paper is its ability to provide written text instructions for creating an object that has been modelled. Furthermore, this generation is possible automatically for some objects, with partial automation possible for a further class of 3D objects. The paper only uses a subset of well-known crochet stitches for constructing

objects and meshes. While not all stitches are easily accommodated into an automated system, adding additional stitches into the system would be a very useful avenue for future research and advancement in this technology.

AmiGo: Computational Design of Amigurumi Crochet Patterns

The work by Edelstein et. al. provides a conceptually similar framework to generate crochet patterns from an input model and input parameters. A key difference between this paper and the work by Guo et. al. is that this team focuses on models that are eventually stuffed to create a filled 3D volume, allowing them to use a different intermediate representation.

The paper largely focuses on using one type of stitch - what they refer to as single crochet (*sc*) - to create the basic shape for an object. To allow for rows with differing numbers of stitches, the number of stitches in a row can be increased or decreased using *inc(x)* and *dec(x)* stitches respectively. By varying the number of stitches in a row, curvature can be induced in the shape and by following a specific algorithm, this curvature can be made to conform to a desired shape.

By taking the target 3D model and converting it into a mesh, the authors are able to use a specified starting point on the surface to create a 'crochet graph', which maps out how the varying curvatures of fabric segments can be used to induce the proper curvature to create the desired target. This crochet graph can then be converted into instructions for a human to create their object via a process not dissimilar to compilation.

For geometries where crochet graph generation is immediately applicable - those with negative mean curvature, or with 'branching' rows - a few modifications to the graph generation algorithm are made. 'Craters' in the geometry are removed through preprocessing, and modifying the sampling rate (the density of stitch rows) for the other cases of negative mean curvature. Since the crochet graph cannot support branching rows, the geometry is also processed to produce separate segments which are crocheted sequentially and combined using a "stitch-as-you-go" method. This approach requires no sewing, as the crochet process for one segment simply begins using the last row of the previous one.

An interesting observation made by Edelstein et. al. is the similarity between the procedure of crochet and the following of instructions in a computer program. Both are sequential processes that require specific

orderings of steps in order to create a desired output. The fact that both often contain a series of similar steps repeated over and over only adds to this similarity. Additionally, the typically exponential-time method used in instruction generation is reduced to a linear-time problem in the number of stitches due to the properties of the crochet graph.

The authors aimed to make the end instructions produced as human-readable as possible - while this was not a major focus of the paper, the techniques used, such as loop folding, may be applicable to other methods as well.

With their focus on filled and stuffed 3D shapes, the applications of this research are somewhat limited - however, it does not seem entirely outlandish to extend the techniques described in the paper to objects that are not closed, or display different kinds of curvatures to the ones targeted at present.

Language and Tool Support for 3D Crochet Patterns

This document by Seitz et. al. provides 2 major contributions - a digital method to represent crochet designs, as well as a 3D editor to visualize the representations and modify them as required. Their editor allows crochet designs to be built stitch by stitch, much like they would be in real life, and then turned into a 3D visualization.

Much like the previously discussed work, this team found that representing crochet stitches as a graph was a useful construct. In particular, they are able to enforce the construction of valid stitches by checking properties of the graph - an invalid stitch connection would violate some property.

This research also documents the similarity between crochet and machine instructions, not unlike the work by Edelstein et. al. However, Seitz et. al. do not yet have a fully automatic method to convert their graphs to crochet instructions, though they do mention such work would be a good next step to build upon what they have accomplished. Without a method to automatically convert a pattern or graph representation to human crochet instructions, the utility of their work is somewhat limited, however, research from other teams that have partially automated this process would almost certainly assist in filling this gap.

Finally, it is important to note that the focus of this group was on crochet patterns that are being actively modified by a human - not converting 3D models or shapes to crochet. As such, their representation of crochet does not currently have the capability to automatically convert a model or 3D representation to

crochet, unlike the work discussed previously.

LIMITATIONS AND NEXT STEPS

In general, most approaches that combine elements of CAD with crochet require simplifying assumptions, whether it be limitations on the crochet techniques, or limitations on what models can be accurately represented. In Guo et. al., the stitch mesh modifications do not capture every way in which a stitch may involve the existing fabric, such as the front post double crochet. The stitch-mesh tile-set is also incomplete, although it allows for the formation of some complex stitches through the combination of simpler ones. In the work by Edelstein et. al., even fewer stitches are supported - only the `sc`, `inc`, and `dec` are used. While this simplification is necessary to make the crochet graph and related algorithm effective, the limited instruction set makes extending the work beyond Amigurumi an open area for research.

The limitations on what models can be accurately presented is a complex issue, as it is difficult to simulate what the output of a crochet pattern will be. Guo et. al. remark that the limitations of re-meshing algorithms can result in subpar crochet patterns, producing dense patches, gaps, or even failing to replicate sharper geometries. The issues associated with gaps in the crochet output, however, can be alleviated through the use of variable stitch aspect ratios - something that is applied in the work by Edelstein et. al.

Furthermore, not all models are compatible with all the techniques described previously, such as in Edelstein et. al. The focus on producing crochet patterns for stuffed toys means their approach cannot handle geometries with negative mean curvature and positive Gaussian curvature (informally referred to as a "crater"). While this is solved through preprocessing of the input mesh, it still serves as a limitation of what can be produced.

For projects that do not have the limitations discussed previously, namely the work of Seitz et. al., they lack the level of automation present in the work of other researchers. While the team were able to accommodate a significant number of crochet stitches into their framework, and are able to accurately model them, they do not have the tooling necessary to convert their graphical representation into a set of instructions that could be used to crochet the specified object. Furthermore, their work does not fully support converting arbitrary shapes into crochet, as their primary focus was accurate modelling with support for manual modification.

Ultimately, the simplifications used in previous re-

search are a necessity due to the complex nature of crochet, and the wide variety of techniques that it offers. While this has allowed for promising work in connecting 3D modelling to crochet, it means that current research has yet to capture the full potential of crochet.

Circling back to the general Final Year Design Project idea of creating an "AutoCAD for crochet", it is definitely feasible to build the groundwork for such an application - however, the team will have to try to determine the right balance of design inclusivity in our solution - making sure that enough processes in our application are automated so that using the application is effective for users, while making sure that we have enough crochet stitches/models involved so that users are able to build the items they want.

The team is also leaning towards hosting the application through the web as opposed to on the desktop, as is done in one of the aforementioned papers. Hosting the application on the web would mean that users would have to worry less about physical computing constraints on their own machines. There are also web technologies and frameworks for rendering and manipulating shapes in 3D. However, more thought will be needed as depending on how we decide to represent stitches in our application (i.e. graph-based nodes, basic shapes like triangles/quadrilaterals, etc.) some frameworks will be more amenable to certain methods than others. Hosting on the web also means that whatever third-party service we decide to use for hosting (i.e. Amazon AWS, Google Cloud, etc.) and managing servers should be able to appropriately handle a large amount of computation for rendering. Allocating large amounts of resources through these third-party services may incur non-trivial costs.

CONCLUSION

We critically examined research relating to computational knitting and crocheting, and discussed how this work may be applied to the idea we have for the Final Year Design Project. Overall, it will be feasible to continue down the path of making "autoCAD for crochet", but every design choice down to algorithm construction must be deliberate in order to most efficaciously build upon the work of previous and existing research and provide a good experience to end users. Discussing the results of the literature review with peers and faculty / graduate students at the University of Waterloo and potentially elsewhere will also be a subsequent step for the project.

REFERENCES

- [1] C. Yuksel, J. M. Kaldor, D. L. James, and S. Marschner, "Stitch meshes for modeling knitted clothing with yarn-level detail," *ACM Trans. Graph.*, vol. 31, no. 4, Jul. 2012, ISSN: 0730-0301. DOI: [10.1145/2185520.2185533](https://doi.org/10.1145/2185520.2185533). [Online]. Available: <https://doi.org/10.1145/2185520.2185533>.
- [2] K. Wu, H. Swan, and C. Yuksel, "Knittable stitch meshes," *ACM Trans. Graph.*, vol. 38, no. 1, Jan. 2019, ISSN: 0730-0301. DOI: [10.1145/3292481](https://doi.org/10.1145/3292481). [Online]. Available: <https://doi.org/10.1145/3292481>.
- [3] R. Guo, J. Lin, V. Narayanan, and J. McCann, "Representing crochet with stitch meshes," in *Proceedings of the 5th Annual ACM Symposium on Computational Fabrication*, ser. SCF '20, Virtual Event, USA: Association for Computing Machinery, 2020, ISBN: 9781450381703. DOI: [10.1145/3424630.3425409](https://doi.org/10.1145/3424630.3425409). [Online]. Available: <https://doi.org/10.1145/3424630.3425409>.
- [4] M. Edelstein, H. Peleg, S. Itzhaky, and M. Ben-Chen, "Amigo: Computational design of amigurumi crochet patterns," in *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication*, ser. SCF '22, Seattle, WA, USA: Association for Computing Machinery, 2022, ISBN: 9781450398725. DOI: [10.1145/3559400.3562005](https://doi.org/10.1145/3559400.3562005). [Online]. Available: <https://doi.org/10.1145/3559400.3562005>.
- [5] K. Seitz, J. Lincke, P. Rein, and R. Hirschfeld, *Language and tool support for 3D crochet patterns* (virtual crochet with a graph structure 137). 2021, ISBN: 978-3-86956-505-7. DOI: [10.25932/publishup-49253](https://doi.org/10.25932/publishup-49253).
- [6] K. Wu, M. Tarini, C. Yuksel, J. McCann, and X. Gao, "Wearable 3d machine knitting: Automatic generation of shaped knit sheets to cover real-world objects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 9, pp. 3180–3192, 2022. DOI: [10.1109/TVCG.2021.3056101](https://doi.org/10.1109/TVCG.2021.3056101).
- [7] V. Narayanan, K. Wu, C. Yuksel, and J. McCann, "Visual knitting machine programming," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019, ISSN: 0730-0301. DOI: [10.1145/3306346.3322995](https://doi.org/10.1145/3306346.3322995). [Online]. Available: <https://doi.org/10.1145/3306346.3322995>.
- [8] K. Wu, X. Gao, Z. Ferguson, D. Panozzo, and C. Yuksel, "Stitch meshing," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018, ISSN: 0730-0301. DOI: [10.1145/3197517.3201360](https://doi.org/10.1145/3197517.3201360). [Online]. Available: <https://doi.org/10.1145/3197517.3201360>.