Amigurumi Crochet Patterns from Geodesic Distances

Mirela Ben Chen[1] and Michal Edelstein[2]

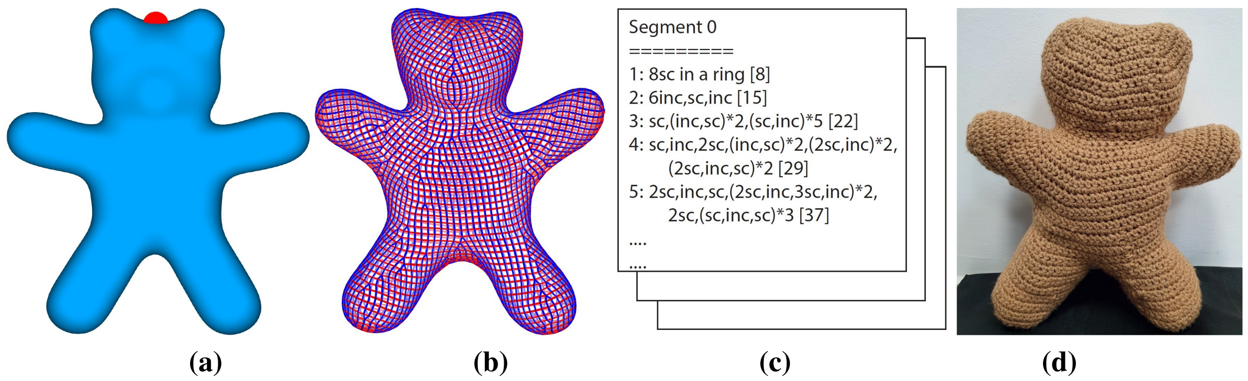
Computer Science Department, Technion - Israel Institute of Technology, Haifa
[1]mirela@cs.technion.ac.il, [2]smichale@cs.technion.ac.il

Abstract

We provide a brief overview of a recently published approach of generating an Amigurumi crochet pattern given an input 3D model. The main idea is to compute the *geodesic distance* on the surface from a given *seed point*, and then sample the equidistant curves non-uniformly, guided by the surface curvature. In addition, the mean curvature of the surface at crease lines guides the type of stitches used, to better adhere to the input shape. We believe that this approach would be of interest to both fiber artists and mathematicians, either as a starting point for crochet pattern generation, or as an interesting application of geodesic distances. This paper provides a brief introduction to our recent publication on computational crochet "AmiGo: Computational Design of Amigurumi Crochet Patterns". The figures are partly reproduced from that paper with permission.

Why Computational Amigurumi?

The interest in crochet has grown quickly in recent years, perhaps due in part to the pandemic. While many types of objects can be made with crochet, from garments to household items, many novice crocheters are drawn to the art of *Amigurumi*, or knitted stuffed toys. Designing amigurumi patterns involves non-trivial shaping to match some sought after 3D geometry. Hence, designing such patterns from scratch requires skill and experience, and is mostly inaccessible to crochet novices. Existing approaches to computational knitting target mostly garments and knitting *machines*, whereas crochet is done almost exclusively *manually* and thus poses different challenges. Hence, we designed a system that gets as input a 3D model, and generates a human-readable crochet pattern. One can directly crochet this pattern, or use it as a basis for a more intricate design. Figure 1 shows the pipeline of our approach, from an input 3D model + seed point (marked red) (a), to the *crochet graph* (b), crochet pattern (c) and the final crocheted object (d).



```
Segment 0
=========
1: 8sc in a ring [8]
2: 6inc,sc,inc [15]
3: sc,(inc,sc)*2,(sc,inc)*5 [22]
4: sc,inc,2sc,(inc,sc)*2,(2sc,inc)*2,
      (2sc,inc,sc)*2 [29]
5: 2sc,inc,sc,(2sc,inc,3sc,inc)*2,
      2sc,(sc,inc,sc)*3 [37]
....
....
```

Figure 1: *Generating an Amigurumi crochet pattern (c) from an input 3D mesh and seed point (red) (a). We also show the intermediate crochet graph (b), and the final crocheted result (d).*

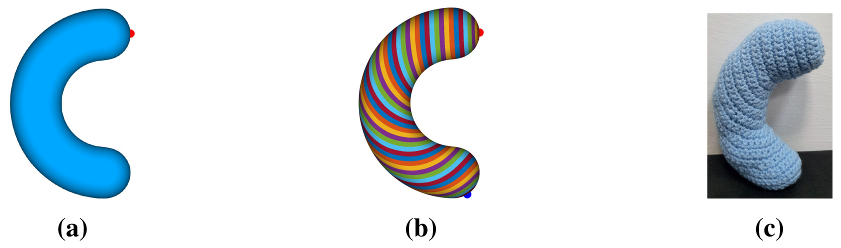


Figure 2: *(a) Model and the seed point (red). (b) Equidistant curves from the seed. (c) The crocheted model.*

Why Geodesic Distances for Crochet?

Crochet has a large variety of stitches of different heights. However, Amigurumi are almost always crocheted using the densest stitch, i.e., single crochet or SC, to prevent the stuffing from protruding. Furthermore, Amigurumi are crocheted *in the round*, starting with a small circle of SC stitches. Hence, usually the height of the crochet rows is *constant* and approximately equal to the width of the SC stitch. Thus, each stitch in a given row is *equidistant* to the starting point - the center of the starting circle. Of course, Amigurumi are not flat, and the way to measure distances on the surface is using *geodesic distances*. These are defined as the *length of the shortest curve connecting two points on the surface*. Therefore, geodesic distances are the best fit for modeling crochet rows. Figure 2 shows a model of the letter C and the curves equidistant from the seed point (red dot). Note how the crocheted rows follow the equidistant curves in the final crocheted model.

How?

The Amigo algorithm [2] constructs a *crochet graph* (Figure 1 (b)), where each vertical edge line (blue) represents an SC stitch. Two vertical lines connecting to the same point indicate an increase SC or decrease SC stitch. An increase SC turns one stitch into two and decrease SC does the opposite.

Rows, stitches and connecting the dots

Figure 3 demonstrates the process. Given a seed point (red), and the stitch height, we compute the geodesic distances of all the points on the model from the seed, and create curves that are equidistant to the seed, spaced by the stitch height (a). These curves represent the *rows* of the pattern (e.g., the marked cyan and yellow curves). The stitches should be spread evenly along the rows. To robustly compute the correct stitch locations, we generate a *flattening* of the model, such that the vertical direction corresponds to the distance from the seed (and thus the row number), and the horizontal direction corresponds to the distance *along* the equidistant curve, and thus to the stitch number. For that, we cut the model along the green path so it can be

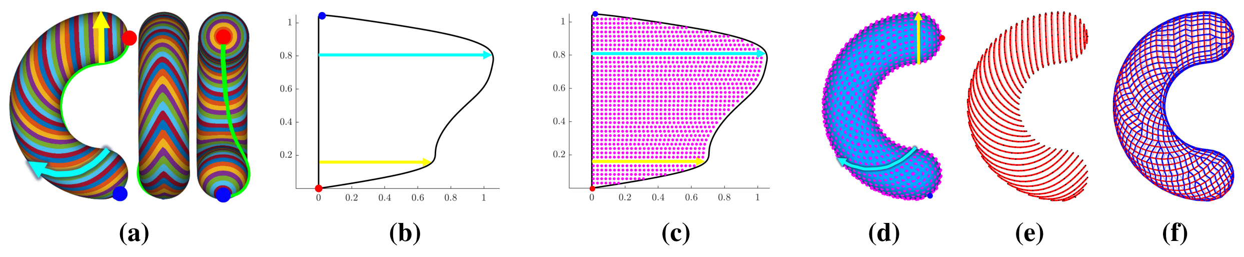


Figure 3: *(a) Equidistant curves (yellow, cyan) from seed point (red), and cut for the flattening (green). (b) Flattening. The seed goes to* $(0,0)$*, and the yellow and cyan rows to horizontal lines. (c) Sampled points on flattening. (d) Sampled points on input model. (e) Row edges. (f) Crochet graph.*

"laid out" on the plane (b). Once the model is laid flat, we evenly sample the row and stitch directions (c). These points are mapped back to the input model, and are the vertices of the crochet graph (d).

Consecutive sampled points on the same row are connected by a horizontal edge in the crochet graph (e). To find the vertical edges, we solve an optimization problem over all valid connections that minimizes the total length of the vertical edges, which provides the final crochet graph (f).

Curvature based inflating, sampling and creasing

Can any closed 3D model be realized as a stuffed crocheted object? We assume the crocheted "skin" will be maximally stuffed, and therefore will attain maximal volume. Under this assumption models with "craters", namely areas on the surface with negative mean curvature and positive Gaussian curvature, are *not* realizable. We therefore pre-process our input model, "inflating" it, to remove the craters (see Figure 4 (a,b)).

When the curvature is "saddle-like", specifically, when both Gaussian and mean curvatures are negative, we *can* realize the model, but need to adapt the row sampling to use *fewer* stitches in such areas. Figure 4 (c-e) demonstrates this. We show the input model (c), and the resulting crocheted models with and without the modification (d-e). The model with the modified sampling (d) is indeed more similar to the input.

Creases are also hard to reproduce, since the model tends to be smooth due to the stuffing. A standard way to introduce creases is to use BLO (Back Loop Only) and FLO (Front Loop Only) stitches. We modify the pattern accordingly in areas with high *maximal absolute* curvature, where additionally the curvature direction is *orthogonal* to the crocheting direction. Figure 4 demonstrates this on a model of a mushroom. We show the equidistant curve lines (f) from the seed point at the top of the mushroom (marked red), the resulting crochet graph with BLO edges (green) and FLO edges (cyan) (g), and the crocheted model (h).

Both the curvature adapted sampling and the BLO/FLO modifications are done automatically.

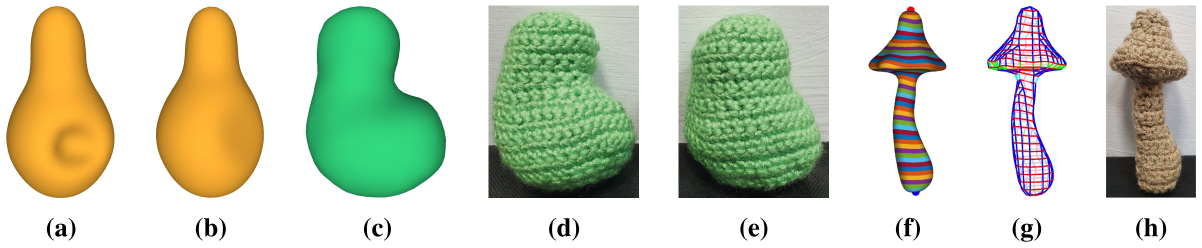


Figure 4: *Curvature mods and creases. (a) A "crater"ed model, and (b) its "inflation". (c) A "saddle"d model crocheted (d) with and (e) without adapted sampling. (f) A creased model with its distance curves, (g) crochet graph with BLO (green) / FLO (cyan) stitches, and (h) the crocheted model.*

Segmentation and visualization

When cutting the mesh to flatten it, we may need more than one cut, depending on the topology of the equidistant curves. For example, a human-like model with two legs will need more than one cut for a seed at the head. We identify this, partition the model into crochetable parts, and add the instructions for connecting them (noting which stitches to pick up). Figure 5 shows (a) the equidistant curves, (b) the partitioning and (c) the crocheted model. We crocheted each part using a different color to visualize the relation to the partitioning in (b). The assembly requires no sewing, and the seams between the parts are not visible (see Figure 6).

It is important for designers to get a feel for how the crocheted model will look, before attempting to crochet it. We provide a visualization by solving an optimization problem embedding the crochet graph in 3D [1]. We constrain the lengths of the SC edges and the row edges to a fixed width, fix the location of the seed point and require smoothness. The result, interestingly, is very similar to the crocheted output, even though we do not incorporate physical simulation. Figure 5 shows a crochet graph (d), the corresponding optimized

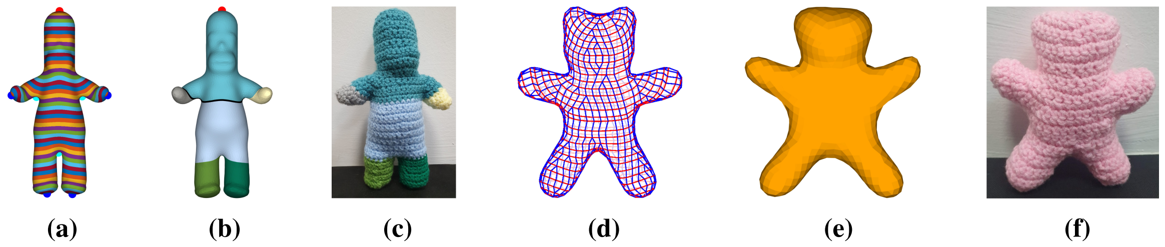


Figure 5: *Segmentation and visualization. Partitioning when the equidistant curves (a) have multiple components, (b) the segments, (c) the crocheted model. Visualizing the crochet graph (d) using a 3D embedding (e). The crocheted model (f) is very similar to the visualization.*

3D embedding (e) and the crocheted model (f). Note the similarity of the embedding to the crocheted shape.

Figure 6 shows a gallery of models that we and the testers that we recruited have crocheted. We also show the input 3D model, with the input seed point and the partition to segments.

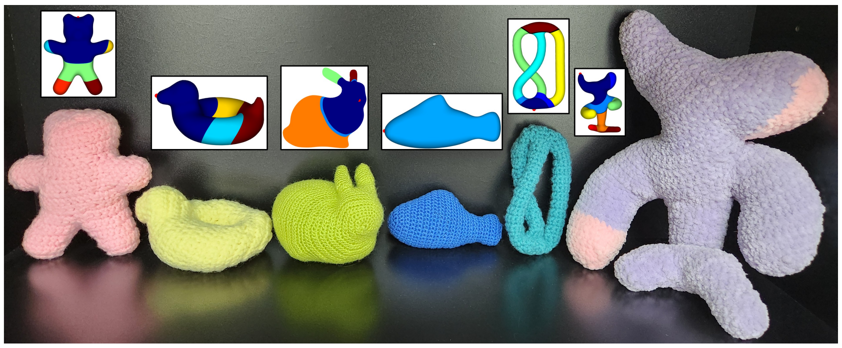


Figure 6: *Models crocheted using our instructions and the 3D models with marked seed point and segments.*

Caveats and What's Next?

Experienced crocheters may notice that our results do not look like "classic" amigurumi. This is because we join at the beginning of the round and turn, instead of crocheting in a spiral. We do this because spiral crocheting leads to slanting, where the stitches of consecutive rows are not aligned one on top of the other. However, our optimization problem for the vertical edges aims for the shortest edges, which is not compatible with this slanting. As spiral crocheting is sometimes preferable, we plan to modify our algorithm to be better compatible with this crocheting method. While we assumed for simplicity that the SC stitch is square, we can also handle rectangular SC stitches. Additional improvements, such as higher stitches, incorporating color, reporting on the required amount of yarn, etc., are planned for a future version. We believe that our visualization tool could be useful for amigurumi designers, even if they start from given instructions and not a 3D model. To that end, standardization of the crochet instructions would be a very important step that would allow inter-operability between computational crochet tools.

References

[1] S. Bouaziz, M. Deuss, Y. Schwartzburg, T. Weise, and M. Pauly. "Shape-up: Shaping discrete geometry with projections." *Computer Graphics Forum*, vol. 31, pp. 1657-1667.

[2] M. Edelstein, H. Peleg, S. Itzhaki, and M. Ben-Chen. "Amigo: Computational Design of Amigurumi Crochet Patterns." *Symp. on Computational Fabrication Proceedings*, Seattle, USA, Oct. 26–28