Decorative Knots in 3D Artwork: Fabricating Models with Successive Knotting

Haruka Ikeda[1] Leo Miyashita Masahiro Hirano Masatoshi Ishikawa

The University of Tokyo, Japan
[1]Haruka_Ikeda@ipc.i.u-tokyo.ac.jp

Abstract

We propose a novel method for fabricating 3D shapes with decorative knots, which are unique patterns created by overlapping and tying thick cords and are an art form passed down since ancient times. Our proposed method automatically computes and displays a route for connecting decorative knots successively in the discretized input model. We introduce a new technique for converting the knotting process into a knitting-like procedure, which removes restrictions on the actual manufacturing steps for a number of knots. After the route is generated, appropriate types of decorative knots are selected that most closely imitate the detailed surface of the input model. Our rendering results show that a user can create large-scale and complex 3D artwork that makes good use of the rich geometric patterns of decorative knots, which will help and extend the fabrication possibilities in handmade contexts.

1 Introduction

Digital fabrication is the technology of shaping various structures of 2D or 3D objects based on digital data, such as polygon mesh models, and the application of algorithms to design and control the structures through computational methods. As a new application of digital design, this paper focuses on decorative knots [1], which are a type of artwork comprising a series of unique patterns and have been passed down since ancient times. Examples of decorative knots are shown in Figure 1. These knots are created by tying and crossing thick cords of about 5–10 mm diameter to express various geometric patterns including motifs of flowers as well as simple and complex patterns, mainly for display and ornamental use.

However, structures of decorative knots have thus far not been extended to construct 3D shapes. This is because knotting (i.e., the process of tying knots) includes procedures to pass the cord through loops made of other sections of the same cord such that a large structure with numerous knots requires a considerable amount of time in the manufacturing step, where the remainder of the cord must be passed through every loop in the knot. Because of this difficulty, there was no choice but to use very thin string, which is easier to maneuver through a loop. For example, in lace tatting, we wind the thread of about 0.3–1 mm diameter into a small lump. In knitting, on the other hand, string of about 1–5 mm diameter is used to create models by continuously connecting stitches, which are the basic or unit elements of the models and are made simply by hooking a part of the string (unlike a knot). Hence, the knitted fabric cannot express complex structures, which require more than just consecutive rows of the simple stitch.

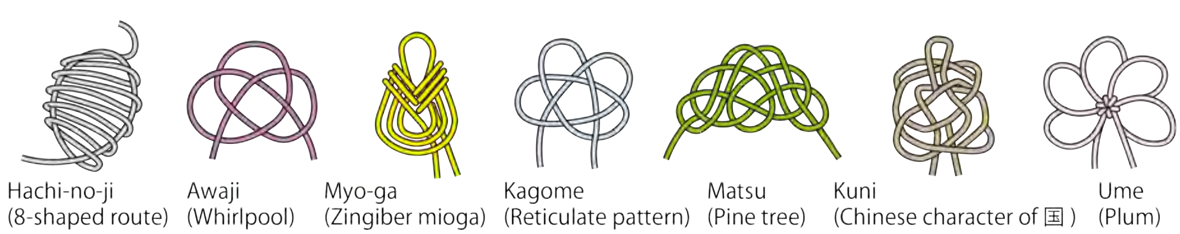

Figure 1 : *Examples of decorative knots with various shapes.*

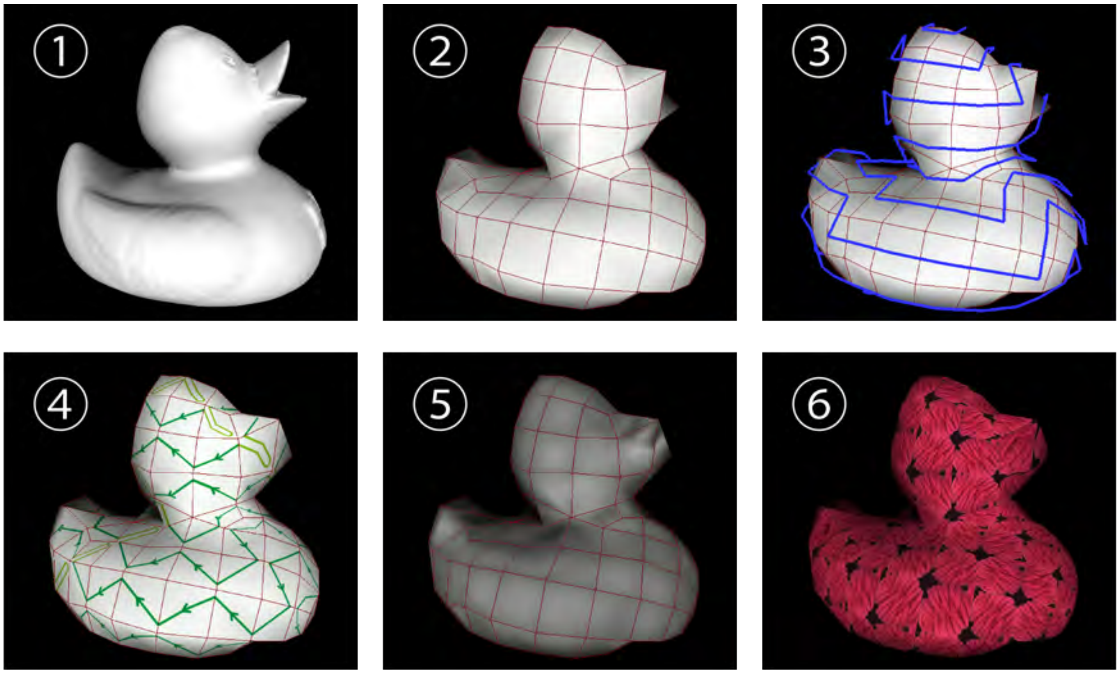

Figure 2 : *Pipeline of our method: ① inputting a model, ② generating a quad mesh, ③ finding a polygonal path, ④ finding a knot route, ⑤ calculating displacement maps, and ⑥ applying decorative knots.*

Therefore, we propose a fully automatic pipeline for constructing large-scale 3D objects, as shown in Figure 2. We first discretize the input model into mesh structures, then generate a path along which the knots can be successively connected, and finally apply the rich structures of decorative knots. This is an innovative approach to combining digital design and traditional handicraft to extend range of shapes possible.

2 Related Work

Various methods have been proposed in literature to support the fabrication of 3D objects in handmade contexts. Beady [5] is a system that assists the design and construction of 3D beadwork, which is the art of connecting beads together by wires, based on polygonal mesh structures that represent the whole model. The structure of the problem is similar to that of ours with knots in that the user connects the elements (i.e., a bead and knot, respectively) to create the whole model. However, they cannot be treated in the same manner because knots should be arranged along the cord that they are made from.

In particular, fabrication using strings is drawing increasing attention. An interactive design tool for authoring, simulating, and adjusting yarn-level patterns for knitted and woven cloths are proposed [9]. Stitch meshing [12] is a proposal to construct arbitrary 3D shapes by knitting. It converts input models into quad-dominant polygon mesh structures, each quad of which correspond to the smallest units of knitting. An extension called knittable stitch meshes [13] guarantees that the models designed with this method can actually be produced by manual knitting. Similarly, our problem deals with the given models as meshed structures to construct surfaces with decorative knots. However, these methods are inapplicable to our knotting problem because the decorative knots are not always connected in the same direction as the knitting. Further, decorative knots express their patterns as lumped units with definite sizes, whereas yarn-level knitting methods treat the whole structure as a set of small and simple stitches.

3 Problem Settings

Representation of 3D Models: Quad Mesh

In order to construct surface shapes with finite collections of decorative knots, we deal with them as structures made of discretized units of polygons [3]. Polygon meshes are suitable because we can use the topological information of the mesh structure to reflect the relationships between adjacent pairs of polygons. Of the available choices of polygons to subdivide the surface, we use quad faces so that we can easily connect them from one to the next by passing the cord along the diagonal of each quad. In this paper, we call the path for connecting quads as the *polygon path*. Further, we call the successive diagonal directions based on the polygon path as the *knot route* along which decorative knots are appropriately arranged.

Structure of the Problem on Mesh: Graph Theory

Once the meshes are generated on the model surface, we can define an undirected graph in which a node corresponds to each quad and an edge represents an adjacent relationship between the jointed nodes. In graph theory, a graph in which any two vertices are connected by exactly one path is called a tree. In this paper, if there are branching points in a tree and multiple routes extending from one node, we call each extended route as a *branch*. If a branch consists of only one polygon, we call it a *leaf*. In our problems, series of decorative knots must be connected by a tree that covers all of the nodes. The problem of finding a path that visits all of the nodes exactly once is equivalent to the Hamiltonian path problem [2]. However, determining whether such paths exist is known to be NP-complete [7]. Hence, we must define an approach to finding the appropriate path without relying on the conventional path planning problem.

Technique for Successive Knotting: Double Knotting

As shown in Figure 3, there are two types among structures of decorative knots: a closed path, where both the start and end point of the knot come at the same point, and structures whose end points extend in different directions. For the latter case, knots cannot be always created continuously along one-stroke paths because the direction of the cord depends on the type of the knot, which makes it difficult to connect knots according to a certain rule.

Therefore, we propose a preprocessing step that we call *double knotting*. First, we select twice the length of the cord that is required to make a knot with conventional techniques. Then, we fold the selected part at its midpoint and treat it as a single cord when making a knot. This generates a virtual double-tracked knot whose end point always comes back to the beginning point of the knot. In Figure 4, structures of the *Hachi-no-ji* knot made according to both (a) conventional method and (b) double knotting are shown. In the double-knotted structure, the exit side of the knot always comes to the entry side. Further, the processes of passing the remaining length of a cord through a loop every time to make a new knot can be removed, which used to be a problem when constructing large structures with thick cords.

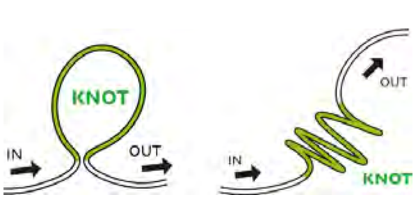

Figure 3: *End points of knots are sometimes at the same point and sometimes extend in different directions.*

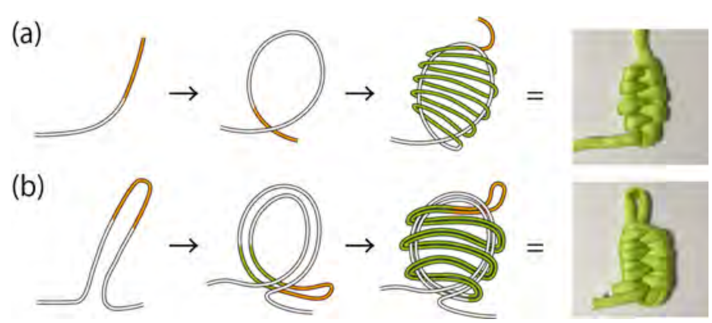

Figure 4: *Structures of the (a) conventional and (b) double knotted Hachi-no-ji knot.*

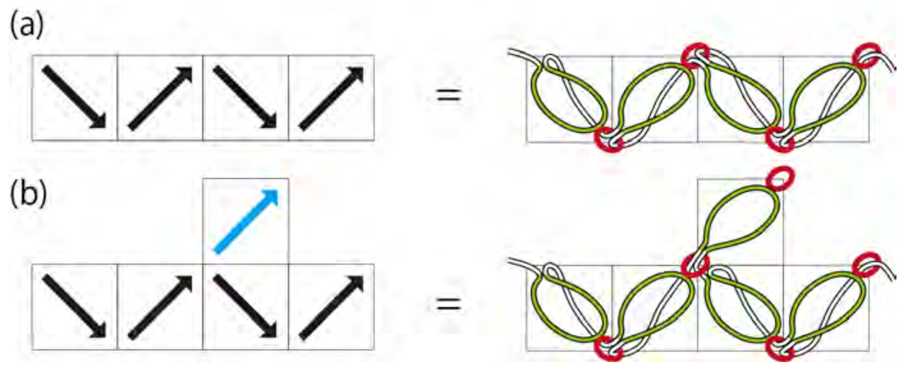

Figure 5: *Knot routes and corresponding decorative knots. The green areas are the bodies of the knots, and the red areas are the loops generated during the double knotting. (a) All knots are connected by the one-stroke path of the cord. (b) Even when a leaf (shown in blue) is inserted, the additional knot does not affect the original route.*

During the double-knotting process, a loop is generated at the points corresponding to the end of the knot with conventional methods, as highlighted in orange in Figure 4. In our method, we use these loops as the beginning points of the successive knot when connecting it to the already-generated one. As shown in Figure 5(a), by passing the cord to generate the next knot inside the loop, we can successively generate a new knot which is connected to the diagonal direction of each quad. At the same time, additional tension of pulling the cord toward the outside of the knot is caused, which prevents the generated loop from coming back towards the starting point of the knot and getting untied.

Constraints of a Route for Connecting Knots

Of the routes available to connect decorative knots, one-stroke paths with no branching points are the most desirable because we can generate knots without deviating from the route. However, such paths are not ensured to exist in arbitrary input models. Hence, we relaxed this restriction; as shown in Figure 5(b), even if a polygon path has branches constructed of one polygon (i.e., a leaf), other parts of the route are not affected by the existence of the additional suspended knot. Therefore, leaves are viable components. However, branches of two or more polygons are not acceptable because we cannot return to the starting point after generating the series of knots without influencing the connection to the next polygon.

4 Methods for Creating Models with Decorative Knots

Generating Quad Meshes

In order to create a model with decorative knots, our method converts the input model into quad meshes by applying instant meshes [6], which is a robust and efficient method that converts a surface into a naturally aligned mesh with high isotropy. The number of quads is determined based on the level of detail needed. After the mesh is generated, we regard it as an undirected graph in which each node corresponds to a quad in the mesh. We assign the structure of the decorative knot to each quad and solve the problem of appropriately connecting the knots where the corners of the quads are shared.

Finding Polygon Paths

On the generated graph, in order to find a path that passes through all of the nodes exactly once, we apply a type of greedy algorithm [4] instead of solving the Hamilton path problem. These algorithms choose a path based on minimizing the local cost at each stage without worrying about the effects that these decisions

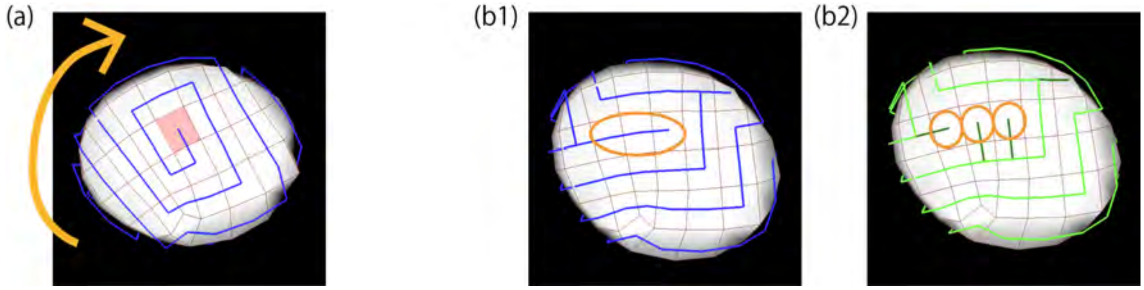

Figure 6: *(a) Spiral polygon path. (b1) Polygon path with a branch made of three polygons (circled in orange). (b2) Corrected path: the branch is disassembled into three individual leaves.*

may have in the future, which can be solved in polynomial time in many problems. Kruskal's algorithm [8] is one of the algorithms derived on the basis of this concept. It is a minimum spanning tree algorithm on a weighted graph, which regards all nodes as a tree and repeatedly combines the trees with the smallest-weight available edge up to a single spanning tree. In our problem, however, we cannot determine the order of the selected edges of the graph solely from the information of local points because there are no weights on them. Hence, we determine the order of connecting knots according to the following steps (i)–(iv).

(i) **Picking a Starting Point and Generating a Spiral Polygon Path**

To generate a polygon path, we pick one of the quads as the starting point of the search and then repeatedly select a polygon on the right from among the available polygons as the direction of extension of the path. This process connects the whole structure in a spiral path, as shown in Figure 6(a).

(ii) **Generating Branches to Deal with Dead Ends**

On drawing the spiral paths, if all the accessible polygons have already been passed, it is impossible for the path to move any further because of the rule that no polygon must be traversed more than once. In such cases, we leave the dead end as a branch or a leaf and search for the next available path according to the spiral rule (i) from the point where the branch begins.

(iii) **Disassembling Branches into Leaves**

If the generated branch contains more than one polygon, it disturbs the connecting knots continuously. Hence, we convert the branch into a structure made of multiple leaves. Figure 6(b1) is an example of a polygon path, where there is a branch made of three polygons (circled in orange). In such cases, we generate new leaves at the adjacent nodes and replace the branch with these leaves, as shown in Figure 6(b2). We continue disassembling all the branches on a path, as long as this is possible.

(iv) **Searching for the Best Path**

In case the generated path has too many leaves, we pick up another polygon as the starting point and repeat the processes of (i)–(iii). These processes can be calculated in a very short time because they are based on the fast greedy algorithm. After paths from all of the polygons are searched, we choose the best path with the fewest leaves.

Converting the Polygon Path into the Knot Route

Based on the generated polygon path, we make a knot route, which shows the order and direction for decorative knots to traverse in the fabrication process. We begin with the diagonal of the first polygon and continuously extend the route so that the knots are successively connected; diagonal directions are selected for each polygon, keeping the order of the polygon in the polygon path. Figure 2③ and ④ respectively show the polygon path and the corresponding knot route.

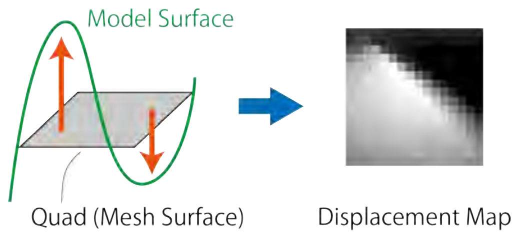

Figure 7 : *Process of calculating a displacement map. The brightness value is proportional to the distance from the mesh surface to the model surface.*

Figure 8 : *Displacement maps and corresponding knots. (L to R)* Hachi-no-ji *knot,* Kuni *knot, and* Awaji *knot. Standard deviations of the brightness are 12, 30, and 52.*

Finding Appropriate Decorative Knots

Finally, for each quad, we assign a decorative knot which reflects the patterns of the surface as much as possible. For this purpose, we compute a displacement map, which is an image acquired by calculating the brightness values at each point in proportion to the distances from the mesh surface to the surface of the input model, as shown in Figure 7. Then, we choose the best decorative knot whose surface has a texture closest to the input shape based on it. We select candidates for the assigned decorative knot from [11] and then choose one among them for each quad; we set thresholds of the standard deviations of brightness in the displacement map to classify images into three types according to the geometrical patterns on quads. We basically use the *Hachi-no-ji* knot because it is plain and can fit into a wide range of patterns. For quads whose standard deviations are a bit high, the *Kuni* knot, which has more uneven patterns, is selected. If the standard deviation is especially high, the *Awaji* knot is selected, whose shape is the most convex. Figure 8 shows examples of the displacement maps and decorative knots that are assigned to each type.

5 Results of Rendered Artwork Models

Representative results of artwork models where the quads are replaced by the handmade decorative knot images using computer renderings (Xeon Gold 5122 CPU @3.60GHz) are shown in Figure 9, as well as in Figure 2⑥. Various 3D shapes including even topologically different ones can be produced. Based on our method, a user can create a model both with high resolutions and with low polygon meshes. In

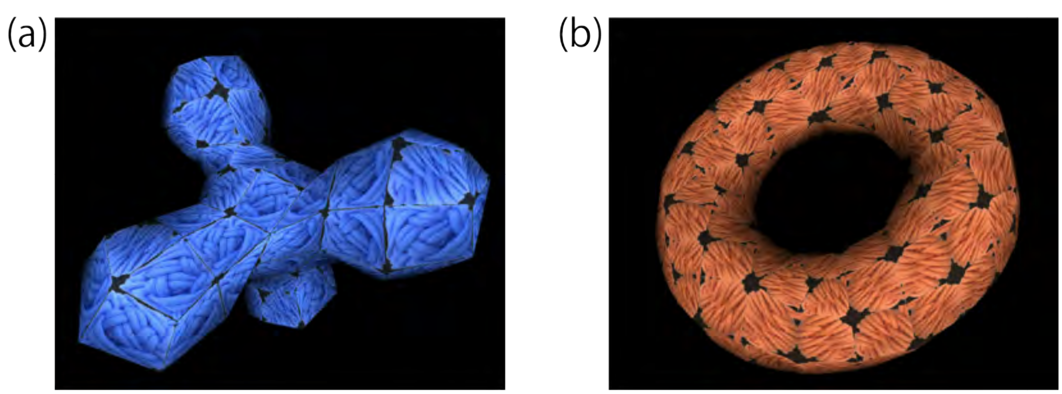

Figure 9 : *Rendered results of (a) a trahedral block, and (b) a torus. Appropriate decorative knots are assigned to each quad and detailed textures are expressed.*

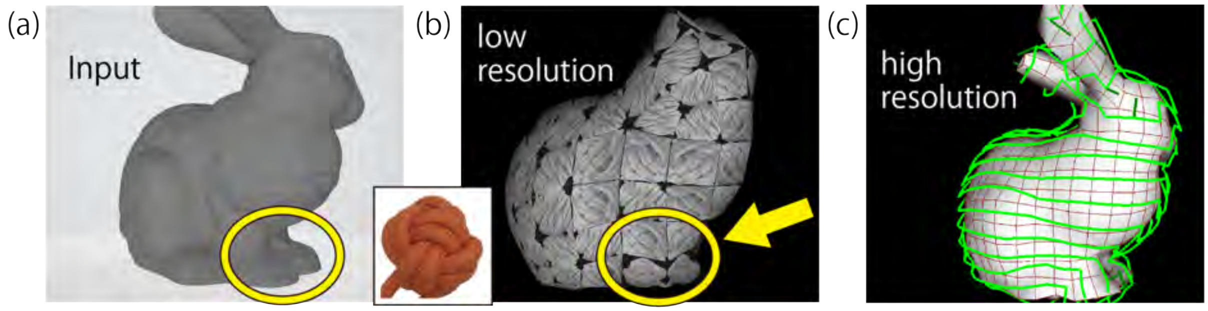

Figure 10: *(a) An input bunny model, (b) a rendered model with a low resolution, and (c) a polygon path with a high resolution.*

the latter case, the rich structures of decorative knots are especially important. For example, a model of a bunny in Figure 10(a) is discretized into a low resolution model with about 150 quads and is rendered into Figure 10(b), in which the front legs (circled in yellow) are appropriately expressed by the *Awaji* knot, whose spherical shape highly imitates the details of the corresponding parts in the input model. Further, our method can design the model with high resolution of more than 500 quads in a very short time (less than 10 seconds), which is very helpful for both skilled creators and beginners because previously creating such large models with complex structures was difficult to be considered.

6 Limitations and Future Work

There are some cases where our method cannot be smoothly applied. For example, in the process of discretizing input models into quad meshes, structures of narrow areas, like the ears of the bunny, are easily lost especially when the resolution is low. Also, even if such structures are expressed with high resolution models, it is not always ensured that branches can be disassembled into leaves on the generated polygon path. Actually, in the narrow region of the ears in Figure 10(c), no suitable paths were found without branches of more than two polygons. This is because once the path extends from the body to the top end of the ears of the bunny according to the spiral rule, there are no more available polygons remaining for the path to return to the body area. However, such problems are not unique to our method. For example, the traditional process of knitting a sweater is not all done continuously; the sleeves are usually knitted separately and then attached to the torso to construct the whole shape. Similarly, we can work around the problem by dividing the input model into parts in advance, applying our method to each divided part, and then assembling the parts. Hence, this limitation is not a serious disadvantage.

In the future, it may be possible to generate a polygon mesh that is optimized to produce a knot route. If the grid patterns of meshes are controlled interactively using the method like Boundary First Flattening [10], we will be able to freely design the surface and selectively assign polygons to the area we want to express with a specific decorative knot. Further, because we can easily pull the cord and stretch its shape to increase the variety of geometric patterns, if the deformability of knots are taken into consideration when an optimized route is generated, we will be able to fabricate models with more detailed expressions. In our classification algorithm, we currently use only three types of decorative knots. However, it is possible to use other types of knots to express more unique shapes of models. Such further extensions will lead to richer artwork based on the combination of both experimental techniques and computational strategies.

7 Conclusion

In this paper, we proposed a new pipeline to automatically design and construct 3D shapes with successive decorative knots. We first discretized input models into quad mesh structures and then determined routes of connecting knots. In order to construct model surfaces with a single-stroke cord, we proposed the double knotting technique, which converts the process of knotting into a knitting-like process to remove the difficulty of connecting a large number of knots in the actual manufacturing steps. Also, we proposed a practical algorithm of finding routes to smoothly tie knots, which makes it possible to connect the overall structure in a viable way in polynomial time. We showed rendered results where knots with appropriate geometric patterns were assigned to each quad, which demonstrate the possibilities of creating both a large-scale 3D artwork and a model with detailed expression. This type of artwork is first realized by our method of applying decorative knots to create 3D models with rich patterns on the surface and complex structures, which is important in the sense that we combined both computational methods and traditional techniques to expand the available range and possibilities of handmade artwork.

References

[1] K. Bien. *Hajimete-no kazari musubi*. Suiyosha Publishing, 2009.

[2] J. A. Bondy, U. S. R. Murty, et al. *Graph theory with applications*, volume 290. Citeseer, 1976.

[3] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon mesh processing*. AK Peters/CRC Press, 2010.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.

[5] Y. Igarashi, T. Igarashi, and J. Mitani. Beady: interactive beadwork design and construction. *ACM Transactions on Graphics (TOG)*, 31(4):49, 2012.

[6] W. Jakob, M. Tarini, D. Panozzo, and O. Sorkine-Hornung. Instant field-aligned meshes. *ACM Transactions on Graphics (TOG)*, 34(6):189:1–15, 2015.

[7] D. S. Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 6(3):434–451, 1985.

[8] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[9] J. Leaf, R. Wu, E. Schweickart, D. L. James, and S. Marschner. Interactive design of periodic yarn-level cloth patterns. In *SIGGRAPH Asia 2018 Technical Papers*, page 202. ACM, 2018.

[10] R. Sawhney and K. Crane. Boundary first flattening. *ACM Transactions on Graphics (TOG)*, 37(1):5, 2018.

[11] H. Takahashi. *Musubi dai-hyakka*. Boutique-sha, 2007.

[12] K. Wu, X. Gao, Z. Ferguson, D. Panozzo, and C. Yuksel. Stitch meshing. *ACM Transactions on Graphics (TOG)*, 37(4), 2018.

[13] K. Wu, H. Swan, and C. Yuksel. Knittable stitch meshes. *ACM Transactions on Graphics (TOG)*, 38(1):10, 2019.