

An Algorithm to Express the Weaving Process of Handwoven Bamboo Craft Using 3DCG

Ningping Sun and Mio Sassa

Abstract— We are interested in how to express the weaving process of Japanese bamboo baskets by 3DCG with reference to the existing handwoven methods. In this paper we provide our algorithm and approaches. Some developed patterns and discussions shall be presented also.

Keywords—Bamboo weave pattern, spline curve, modeling, rendering, 3DCG.

I. Introduction

Japanese style bamboo baskets are called “takezaiku” or “takekago” in Japanese, plays an important role in Japanese traditional craftsmanship because of its long history and excellent decorative art. Bamboo crafts have been popular in Japan since ancient times as handicrafts such as daily necessities, farm tools and tea props. Nowadays, the unique design and the texture of the bamboo become more popular, bamboo products and bamboo baskets are used in various life scenes. The hand-woven bamboo basket is woven from the bottom to the rim of the basket with bamboo strips made from bamboo trunks shown as Figure 1. The color and texture of the basket depends on the type of bamboo and the preparing process of strips. The longer you use, the more you will enjoy the evolution of the unique bamboo texture, the long-term use will change the appearance of the basket from the original color to the candy color.



Figure 1. Japanese household handwoven bamboo baskets

Braid of bamboo basket is called “henso” or “ami” in Japanese. There are various kinds of weave patterns such as “gozame-ami” or “su-ami” mainly used in the flat or cylindrical basket making, and flower-like weave such as “yotsume-ami”, “mutsume-ami”, “kikkou-ami”, “waguchi-ami”, “yatara-ami”, and so on [1] [2]. The new weave patterns based on these basic weave patterns have been born one after another, there are now hundreds of handwoven ways. Bamboo craft-worker uses and combines variations of weave methods on the bottom, the body and the rim of bamboo basket respectively to increase the originality. Even with the same length or number of bamboo strips, you can change the design of your work by different weave patterns. It can be said that it is one of the attractions of bamboo craftsmanship.

We aim to analyze the existing handwoven methods of bamboo basket and design an algorithm to express the handwoven process referring to the actual weave patterns and methods. In our work, the bamboo strip is represented by a 3DCG slender cylinder model, and the weaving motion is simulated using a cubic interpolation spline curve function [3]. We also want to control the mode by modifying parameters of the patterns.

We proposed and designed an approach to express the weave pattern called “gozame-ami” or “su-ami” and its derived patterns such as “uzu-ami” and “une-ami” using 3D spline curves. The shapes of model include circular plate and ones created by free hand-drawing.

In this paper we shall describe our algorithm and method how to model and render a bamboo basket using 3DCG. Some discussions about our method will be provided also.

II. Gozame Weave Pattern

In general, relatively shallow and flat baskets are called “goza” or “zaru” depending on the region, but deeper things are called “kago” in Japanese shown in Figure 2. The “gozame-ami” method also called “su-ami” produces one of the simplest and most basic weave patterns. In this study we focused on the weave method and its derived methods, and designed algorithms to reproduce them using 3DCG.

A. Gozame-ami Pattern

Gozame weave uses materials of different widths to weave, thicker bamboo strips are used in the longitudinal direction, and narrower and thinner bamboo sheets or strips are used in the transverse direction [1] [2].

Ningping Sun Ph.D
National Institute of Technology, Kumamoto College
Japan

Mio Sassa
National Institute of Technology, Kumamoto College
Japan

As shown in the illustration of Figure 3, the vertical bamboo strips are placed at regular intervals, and the horizontal strips are repeatedly woven around the vertically spaced strips to fill the space. Because this method has good strength and flexibility, it has been used to making “zaru”s and “kago”s for a long time, also is called “zaru-ami”.



Figure 2. “Zaru” (the up) and “kago” (the bottom)

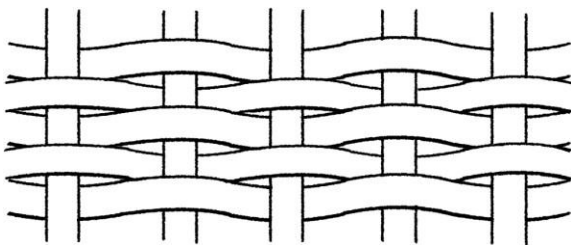


Figure 3. Gozame weave pattern

The braid passes over the vertical strips one above the other, up and down. By setting an odd number of vertical strips, the second row of the weaving strip is opposite to its first row, that is, the strip that was originally in the upper portion of the vertical strip will go to the lower portion of the vertical strip, and vice versa.

In order to express the weave pattern with computer graphics, the following process is employed.

- 1) Make a group of strips that becomes a set of vertical strips.
- 2) Thread the woven yarn through the vertical strip according to the "gozame-ami" pattern.

In the first stage, we form a "juji-kumi" organized like a cross, as shown in Figure 4.

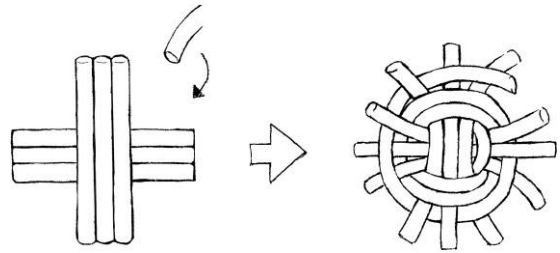


Figure 4. The illustration of "juji-kumi"

In the second stage, according to the "gozame-ami" pattern, we pass the yarn through the warp assembled in Figure 4 and weave the yarn into a wound shape as the purple line shown in Figure 5.

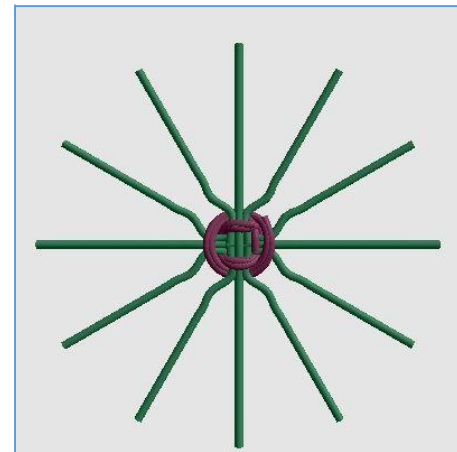
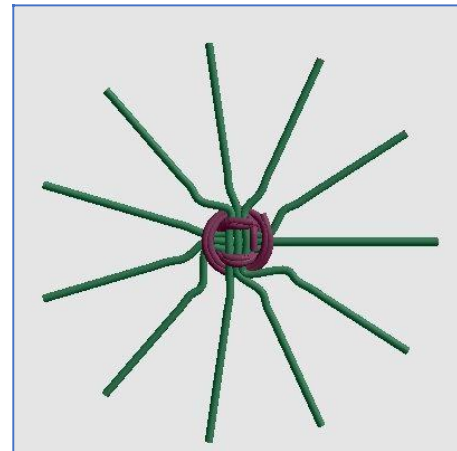


Figure 5. Warp assembled as Figure 4

The shape of pattern depends on whether the number of warp strips is odd or even. In our work, we use 11 as an odd number and 12 as an even number, the same as Figure 5.

After obtaining the coordinates of the warp yarn, the coordinate of points in the intersection position are obtained in the follows two ways denoted as WAY 1 and WAY 2.

We believe that the weft yarn draws a parallel spiral on the vertical warp strips, which has the x and y values of the vertical warp. Therefore, in WAY 1, we use the (x, y) coordinates of the vertical warp strip as the (x, y) coordinates of the knitting yarn, and change the z value of the knitting yarn to realize the “gozame-ami” pattern shown in Figure 6.

In the WAY 1, the spiral pattern is obtained by (1), where q is a variable indicating the distance from the center of the circle. If it is a normal spiral, it will increase constantly.

$$\begin{cases} x = q \cos \theta \\ y = q \sin \theta \end{cases} \quad (1)$$

In addition, by controlling the z value in (2), the circular model can be transformed into a “zaru” model, where h indicated the height of model and r is the radius of the “zaru”.

$$\begin{cases} z = q^2 k \\ k = l/r^2 \end{cases} \quad (2)$$

Figure 7 gives a “zaru” with “gozame-ami”.



Figure 6. Developed “gozame -ami” pattern with WAY 1

WAY 2, consider two vectors perpendicular to the vertical warp strip (inward, outward) and orient the vertex at a distance from the vertical warp strip in the vector direction, then choose the vertices and draw a spiral. WAY 2

is more freely and easily than WAY 1 in selecting the shape of the model. However, in WAY 2, it is impossible to adopt vertices of the same width, and thus it is difficult to express the gap.

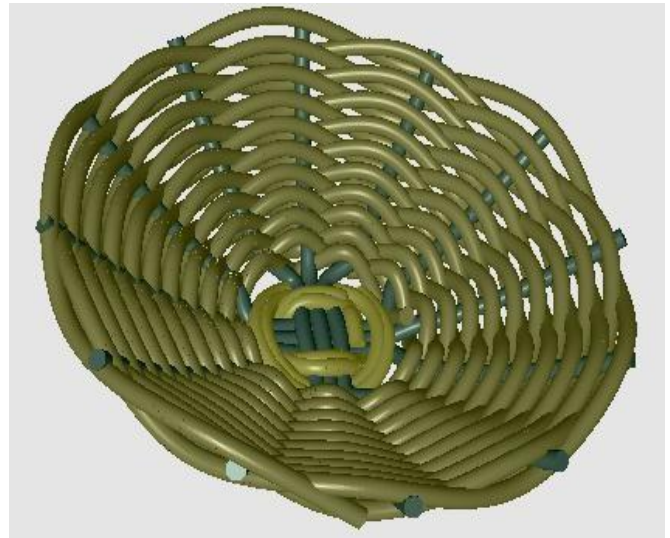


Figure 7. Developed “zaru” with “gozame-ami” pattern and WAY 1

B. Uzu-ami Pattern

“Uzu-ami” weave is a variant of “gozame-ami” weave, in which “uzu” means swirl in Japanese. Produce an odd number of warp yarns in the longitudinal direction, and jump the knitting strip on two warp at a time as shown in Figure 8. When weaving circular basketry by this pattern, the stitches look like “whirlpools”, therefore the pattern is called “uzu-ami”, it often is used to weave round baskets [1].

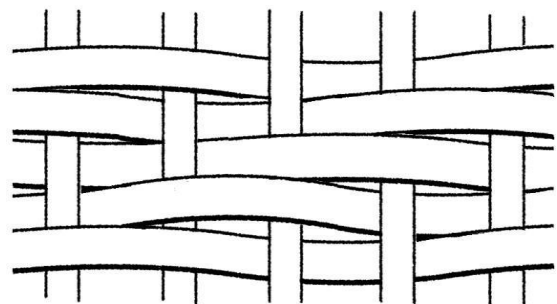
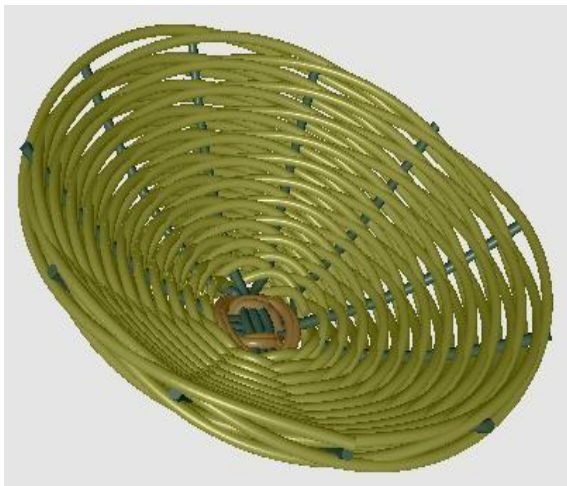


Figure 8. “Uzu-ami” weave pattern

This pattern is also implemented using WAY 1 and WAY 2 as described in above section. Figure 9 gives the developed “uzu-ami” pattern with WAY 1 and WAY 2. Using three-dimensional concave rendering and convex rendering, we can create left-handed whirlpool.

In the “uzu-ami” pattern, the warp strip is the important base thread. Depending on whether the number of warp is an odd number or an even number, the pattern to be drawn changes. Using WAY 2 we can change design of pattern during weaving easily.



WAY 1



WAY 2

Figure 9. Developed “uzu-ami” pattern with WAY 1 and WAY 2

C. Une-ami Pattern

The pattern also is a variant of "gozame-ami", and is named because it looks like ridges in the field. As shown in Figure 10, the three weft strips are grouped together to increase the twist and strength of the weft. It is usually used to make baskets for heavy loads [1].

This weaving method is significantly different from the first two patterns. First, we will weave with an even number of warp yarns in the longitudinal direction instead of an odd number. In addition, we use multiple yarns instead of one weaving yarn.

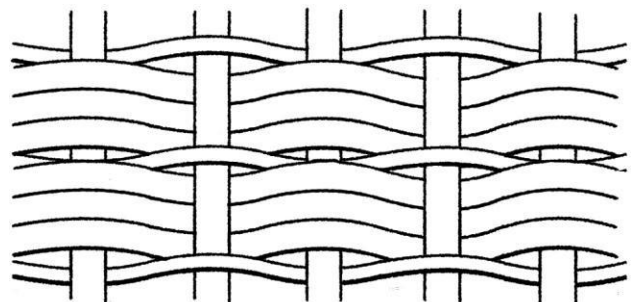


Figure 10. “Une-ami” weave pattern

In Figure 11, the “zaru” is woven with four lines, three thick parallel weaves, and one thin weave intersects with the other three bamboo strips. The pattern obtains a more ornate texture than the other two patterns mentioned above. Due to its special nature, it is difficult to implement the pattern with WAY 2. It only can be implemented by WAY 1.

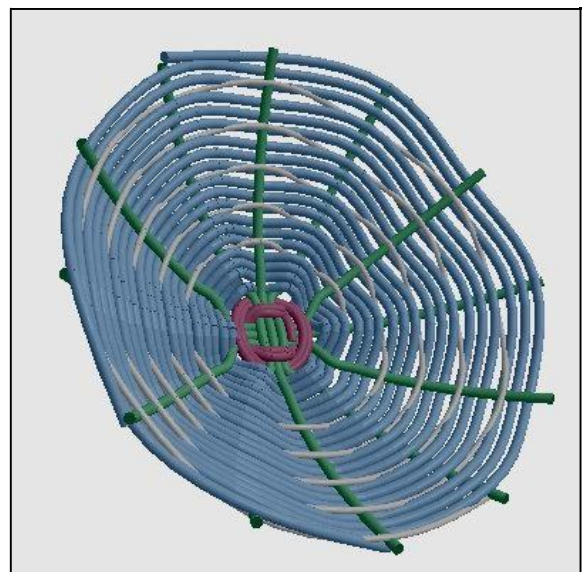


Figure 11. Developed “une-ami” pattern with WAY 1

III. Implement 3D Baskets with Spline Curve

In this study, we used a cubic spline curve to achieve the woven strip. Here we first explain how to create a hand-drawn curve using a cubic spline interpolation function polynomial, and then describe how to apply the spline curve to the weaving process of WAY 1 and WAY 2.

A. Cubic Spline Interpolation Curve

Cubic spline curves, the curves generated by the cubic spline interpolation function polynomial have good continuity, closest to the handwritten curve shown in Figure 12 [3][4].

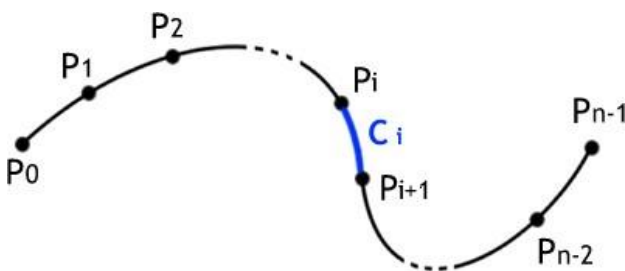


Figure 12. The cubic spline interpolation curve

Consider how to compute the uniform cubic spline curve that passes through a sequence of points in Figure 12. For convenience we denote the given points sequence as P_0, P_1, \dots, P_{n-1} ($n \geq 3$) and suggest that the generated curve will pass through the entire given points. $C_i(t)$, the curve segment between two consecutive points P_i and P_{i+1} is expressed as (3),

$$C_i(t) = p_i (2t^3 - 3t^2 + 1) + p_{i+1} (-2t^3 + 3t^2) + p'_i (t^3 - 2t^2 + t) + p'_{i+1} (t^3 - t^2) \quad (3)$$

where $0 \leq i \leq n-1, 0 \leq t \leq 1, C_i(0) = P_i, C_i(1) = P_{i+1}$.

$C(t)$ can be expressed as $C(x(t), y(t), z(t))$ in vector expression, where $x(t), y(t)$ and $z(t)$ are polynomials in t of the third-degree interpolation function, which is called the cubic spline interpolation function. In practice we interpolate and render curve within the n given points sequence by (4).

$$C_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2(P_i - P_{i+1}) + (P'_i + P'_{i+1}) \\ 3(P_{i+1} - P_i) - (2P'_i + P'_{i+1}) \\ P'_i \\ P_i \end{bmatrix} \quad (4)$$

where $0 \leq i \leq n-1$ and $0 \leq t \leq 1$. P_0, P_1, \dots, P_{n-1} , are the derivative of points P_0, P_1, \dots, P_{n-1} that can be obtained by (5) with natural bounded condition such as $C'_0(0) = 0$ and $C'_{n-2}(1) = 0$.

$$\begin{bmatrix} 2 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ & & \dots & \dots & \dots & \\ 0 & \dots & & 1 & 4 & 1 & 0 \\ & & & \dots & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} P'_0 \\ P'_1 \\ \vdots \\ P'_{n-2} \\ P'_{n-1} \end{bmatrix} = \begin{bmatrix} 3(P_1 - P_0) \\ 3(P_3 - P_1) \\ \vdots \\ 3(P_{n-1} - P_{n-2}) \end{bmatrix} \quad (5)$$

Using matrix LU decomposition to find the solution of (5) should reduce the quantity of the matrix calculations. The first-degree derivative of curve $C_i(t)$ is

$$C'_i(t) = \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 3\{2(P_i - P_{i+1}) + (P'_i + P'_{i+1})\} \\ 2\{3(P_{i+1} - P_i) - (2P'_i + P'_{i+1})\} \\ P'_i \end{bmatrix} \quad (6)$$

and we have the second-degree derivative

$$C''_i(t) = 6t\{2(P_i - P_{i+1}) + (P'_i + P'_{i+1})\} + 2\{3(P_{i+1} - P_i) - (2P'_i + P'_{i+1})\} \quad (7)$$

We can also find the curvature and the normal to $C_i(t)$ by (8) for shading or texture mapping application.

$$N_i(t) = \frac{C'_i(t) \times C''_i(t) \times C'_i(t)}{|C'_i(t)|^4} \quad (8)$$

By means of the above cubic spline polynomials we are able to model any curves based on the given points sequence P_0, P_1, \dots, P_{n-1} ($n \geq 3$). The smoothness of the curve depends on the number of vertices used in the segment. As the number of vertices increases, the curve becomes smoother, but processing costs also increase.

B. Present Weave Process with 3D Spline Curve

When rendering a spline curve in 3D, we replace the curve segment with a slender cylindrical model in order to set environmental information such as the light source and shading colors conveniently.

Now we consider how to use a cylinder to connect two points of p_1 and p_2 illustrated in Figure 13.

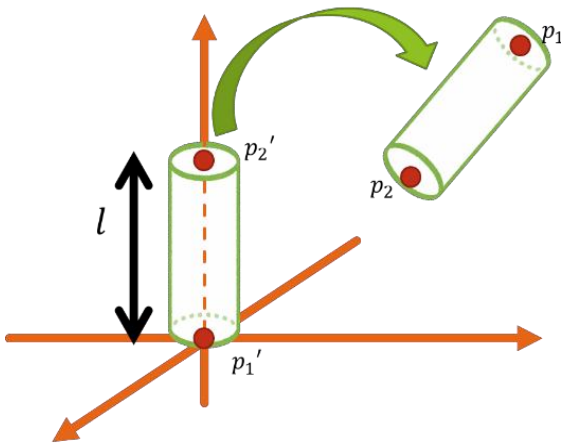


Figure 13. Connect two points of curve

The distance between p_1 and p_2 can be got by (9).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (9)$$

The coordinate of p_1' is $(0, 0, 0)$, p_2' is $(0, d, 0)$. Using translation and rotation we move the cylinder of p_1' and p_2' to the position of p_1 and p_2 .

Consider a new coordinate axis with the direction vector from p_1 to p_2 according to the positive direction of the y-axis. The new vector matrix is follows.

$$\begin{bmatrix} p_x & p_y & p_z \\ q_x & q_y & q_z \\ r_x & r_y & r_z \end{bmatrix}$$

Apply the rotation and translation functions to this matrix we can get the coordinates of the cylinder on the spline curve.

In order to achieve the up and down movement of the weaving yarn, it is necessary to obtain a normal vector at any coordinate point on the 3D spline curve. Because the initial normal vector is not to a 3-dimensional (x, y, z) space, but a normal vector to a spline curve on a 2-dimensional (x, z) plane. The vertex that the knitting yarn can pass through is obtained by the following steps.

- 1) By converting the coordinates of all the warp coordinates by z-axis rotation in order to convert the coordinate of point to one that can be expressed only once in the x-z coordinate system.
- 2) Find n vertex coordinates in one segment of spline curve. Then using (10) to find the approximation derivative values of the vertices, where h is a very small constant.

$$\begin{cases} f_x'(t) = \frac{f_x(t+h) - f_x(t-h)}{2h} \\ f_z'(t) = \frac{f_z(t+h) - f_z(t-h)}{2h} \end{cases} \quad (10)$$

- 3) Find the normal vector by the above derivative values. Let partial derivatives of coordinates $(f_x(t), f_z(t))$ be $f_x'(t), f_z'(t)$, the normal vector can be obtained by (11).

$$(\pm f_z'(t), \mp f_x'(t)) \quad (11)$$

- 4) Transform the two normal vectors into unit vectors. Find a normal vector with a length of margin from the unit vector.
- 5) Perform the operations of 2) to 4) for all segments of spline curve.
- 6) Using the obtained normal vector, create a list of vertices. Rotate the vertex in the vertices list to the position where the original warp strip was located and move the vertex.
- 7) Perform operations of 2) to 6) for all warps.

We can get two vertices lists by the above steps, for example $(vecList1, vecList2)$. Figure 14 illustrates our spiral drawing algorithm, in which a brown line is a warp strip and a red circle is a vertex position corresponding to $vecList1$ and $vecList2$, the blue line is the weave strip. We weave a basket by determining the number of times the warp is skipped according to weave pattern first, and then selecting an appropriate corresponding vertex from the two vertices list.

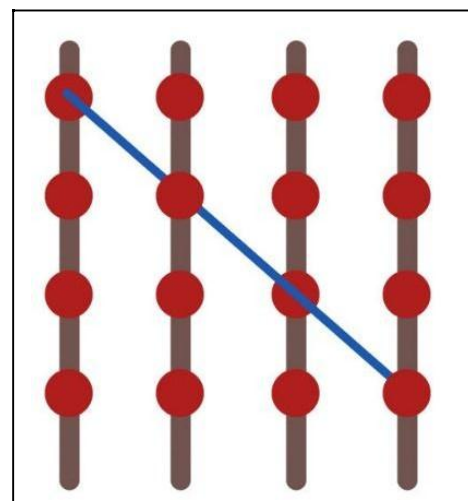


Figure 14. Spiral drawing algorithm

iv. Developed System

Figure 15 shows a graphics user interface of the system that we constructed. A spline curve is obtained from user's freehand-drawn input, the warp strips are limited to eleven, and a spline curve inputted by user starts from the bottom "juji-kumi" assembled as Figure 5. The system performs spiral knitting based on the warp strips obtained by rotating the curve on the z -axis. Also, the vertex coordinates of the knitting yarn are computed with acquiring the normal vector from the spline curve as we described in above section.

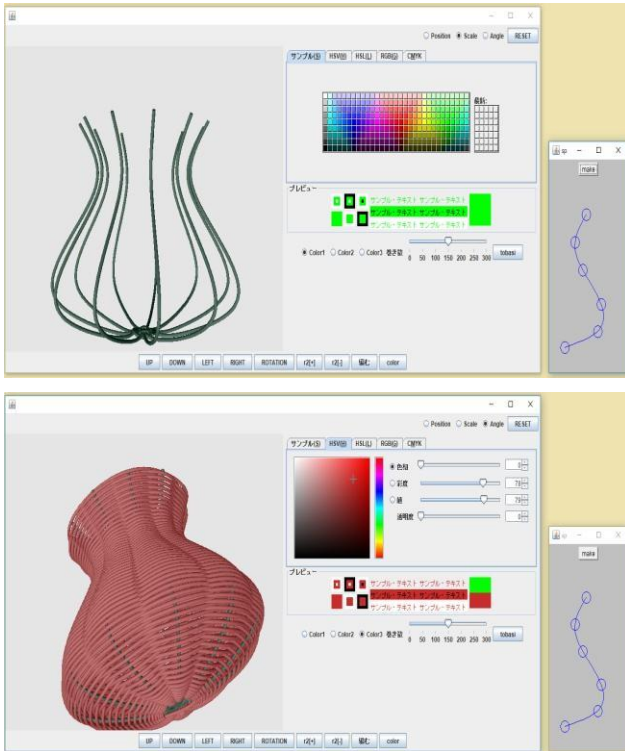


Figure 15. Graphics user interface of the system

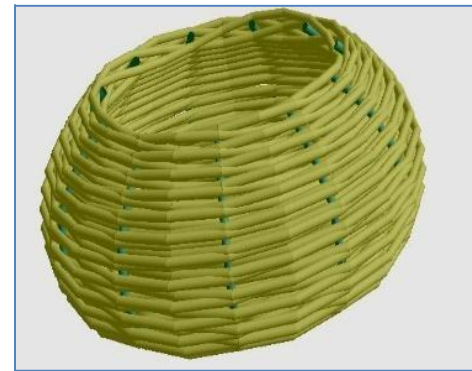
Since the coordinate information of the spline curve on the GUI is different from the coordinate system (OpenGL coordinate system) for rendering the three-dimensional model, it needs to be converted [5]. In this study, we convert the control points as follows and create new spline curves from the control points.

$$(OpenX, OpenY) = \left(\frac{mouseX}{T}, -\frac{mouseY}{T} \right) \quad (12)$$

where $(OpenX, OpenY)$ is the coordinates (x, y) of OpenGL, and $(mouseX, mouseY)$ is the coordinates of point on spline curve drawn on the GUI, constant $T = 100$.

In addition, the spline curve is drawn using the coordinates of the x - y plane. By rotating around the x -axis, converting it to the coordinates of the x - z plane, the starting point of the spline is moved to the position where the "juji-kumi" is performed. And rotating around the z -axis to draw

the warp strips in the three-dimensional vertical axis direction. Two developed baskets are shown in Figure 16.



"uzu-ami" pattern



Figure 16. Developed baskets

v. Conclusions

We designed and developed an algorithm to express the bamboo basketry weaving process. Our work on creating handwoven bamboo crafts using spline curve function and 3DCG methods demonstrated the possibility that use CG technology to express and preserve traditional handcrafting techniques. Dealing with more patterns and developing ways to better mimic the effects of handcrafted crafts is our future work.

References

- [1] Reiko Sasaki, The Complete Japanese Basket Making, Seibundo Shinkosha, 2013.
- [2] Takekagobu, The Bamboo Basket Handbook, Seibundo Shinkosha, 2015.
- [3] Yoshiyasu Minemura, Computer Graphics with Java - From Fundamentals to Visualization of Simulation (Japanese), pp.73-93, Mori Kita Publishing, 2003.
- [4] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, R. L. Phillips, Introduction to Computer Graphics, pp.321-366, ADDISON-WESLEY, 2000.
- [5] The Khronos OpenGL ARB Working Group, OpenGL Programming Guide, Addison-Wesley, 2013.