

# Out-Tree Generation Using Controllable Field Space Colonization and Application to Modeling

Toshiki Tanaka<sup>1,\*</sup>, Tomoya Ito<sup>2</sup>, Tsukasa Kikuchi<sup>3</sup>

<sup>1</sup>Graduate School of Media Science, Tokyo University of Technology, Hachioji, Japan

<sup>2</sup>Department of System and Information Engineering, Hachinohe Institute of Technology, Hachinohe, Japan

<sup>3</sup>School of Media Science, Tokyo University of Technology, Hachioji, Japan

## Email address:

g311900842@edu.teu.ac.jp (T. Tanaka), tomoya@hi-tech.ac.jp (T. Ito), kikuchitks@stf.teu.ac.jp (T. Kikuchi)

\*Corresponding author

## To cite this article:

Toshiki Tanaka, Tomoya Ito, Tsukasa Kikuchi. Out-Tree Generation Using Controllable Field Space Colonization and Application to Modeling. *American Journal of Computer Science and Technology*. Vol. 4, No. 2, 2021, pp. 28-37. doi: 10.11648/j.ajcst.20210402.11

**Received:** March 17, 2021; **Accepted:** April 6, 2021; **Published:** April 13, 2021

---

**Abstract:** The reproduction and modeling of natural phenomena using computer graphics is used in a wide range of fields. This requires a great deal of work on the part of the producer. Procedural techniques are an effective means of supporting this process, and this paper focuses on the generation and modeling of branching structures. First, this paper extends an existing branching structure generation algorithm, space colonization, by changing the positions of the points that make up the segments of the Out-Tree generated in 3D space. By using an induced vector field rotated by quaternions around the normal vector of the starting point of the Out-Tree generation, the algorithm can change the coordinate values of newly generated points, thereby allowing the shape of the branching structure to be manipulated by numerical parameters. The next step is to apply the Controllable Field Space Colonization as a framework for modeling. The application domain of the algorithm is determined by the simple model input by the user. The input model is converted to sparse volume data, and the attraction points are placed on the surface or inside the input model. This method can not only generate tree models similar to the L-System and existing space colonization, but also represent frost, lightning, blood vessels, rivers and mountains. As the future work, includes optimization to enable automatic input of numerical values for each generated model, and correction of polygon flipping due to normal vector errors.

**Keywords:** Controllable Field Space Colonization, Out-Tree, Branch Structure, Procedural Modeling

---

## 1. Introduction

There is a growing need for computer graphics technology to reproduce, model, and design natural phenomena, not only in the entertainment field, but also in architectural design and art. In the entertainment field, it is more realistic to use computer graphics to visualize natural phenomena (large-scale ground cracks, lightning) and other natural objects that are difficult to actually film. In the case of design or art using computer graphics, it is expected that a great deal of effort will be required to create a form with a branch-like ornamental motif, such as a Hollow Branch Sculpture.

The modeling of these tree-structured objects has been studied for a long time, and various methods have been developed until now. Basically, methods for reproducing tree

structures and their associated shapes can be categorized into geometry-based methods, image-based methods, and rule-based methods. As for rule-based methods, modeling methods using the L-System are well known, and the study of mathematical models of cell-cell interaction by Lindenmayer is famous (1968) [1]. And, it has been used for Web 3D modeling using the L-System (2011) [2]. However, these methods are mostly used to generate trees and other plants. Another known problem that can occur in L-Systems is the difficulty of finding control points and substitution rules by tracing back the structure. Also, known problem that can occur in L-Systems is the difficulty of finding control points and substitution rules by tracing back the structure.

In this paper, as an improvement of the problems in the conventional L-System based tree structure generation, we propose a method to control the shape more easily by using

the space colonization algorithm to avoid the problems related to the parameterization to make the model manipulable by users. Furthermore, conventional space colonization is limited to the creating simple branching structures, as it is only an iterative process of making connections to points scattered throughout the envelope. On the other hand, the proposed method can generate graphs with branching structures that cannot be reproduced by conventional space colonization by using vector fields. Using a polygon model, the user can arbitrarily decide the region to apply the algorithm. In addition, we consider the generated shape as a kind of outward-facing tree, and show that modeling using this method and concept can roughly reproduce the appearance of not only plants such as trees, but also objects with similar rules, and can be used for general purposes (Figure 1).



Figure 1. Images of the results of our works.

## 2. Previous Works

### 2.1. Tree Modeling Methods

Runions et al. proposed a biologically motivated algorithm for generating leaf vein patterns [3]. It is an algorithm that formulates the geometric patterns that exist between the distribution of hormones (auxins) in plant leaves and leaf veins. An extension of the algorithm to reproduce the same algorithm in 3D space was named space colonization [4]. This allows us to propose an approach that can generate a branching structure that focuses on the competition of branches for space. Because the branching structure takes into account the competition of branches in space, the shape of the branch is controlled only by the density of points placed in space to form the branch and the search range of the branch. Therefore, in addition to controlling the direction of branch propagation by parameters, intentionally generating branching structures with deformations such as spiral shapes cannot be reproduced in the original space colonization. Extending the traditional space colonization and focusing on tree modeling is the work of Patrick et al [5]. The algorithm has been extended to reflect environmental factors and to pursue more interactivity.

### 2.2. Road Generation Method

In their research, Fernandes et al. proposed an intuitive method for parameterizing road layouts by using an approach

to generate road layouts in and between urban environments based on the concept of attraction points introduced in space colonization [6]. The method in this paper also allows for fine-tuning of the layout in more detail by using flow fields, demographics, and other geographic factors. The first stage of the approach by Fernandes et al. generates a primarily tree-like branching structure, and the second stage performs processing on the connectivity between roads, effectively converting the initial tree structure into a road layout graph. It was found that there was a relationship between the parameter values and the geometry of the road layout. Future work includes parameterizing the attraction points to take into account points of interest such as shopping malls, train stations, and bus stops, as well as giving the primitives in the flow field a range of influence to enable retrieval of different primitives and obtain interesting road layout patterns.

### 2.3. Control Method of Vector Field

In many graphics applications, vector fields are often used as input data. In non-photorealistic rendering, vector fields are used to guide the direction of brush strokes [7]. Also, when considered in fluid simulation, external forces are vector fields, which guide the movement of particles [8]. Zhang et al. proposed a system for generating vector fields on a 3D mesh [9]. It is possible to create a wide variety of vector fields (curl-free, divergence-free, generic) using various types of basis fields. By utilizing this system, the singularity can be controlled arbitrarily, and the application to designs using vector fields and higher-dimensional assets, such as fluid simulation and hair design, is an issue for the future. As an extension of the aforementioned work, Paracios et al. have implemented an interactive 3D tensor field design system [10]. The system can generate a tensor field in a user-specified volume or on a 3D mesh.

### 2.4. Branching Ornaments

A research example focusing on decorative patterns with branching structures is the work by Zhang et al [11]. They proposed a method to automatically generate a sculpture called Shrub Branch-style Shell (Hollow Branch Sculpture) along the surface curvature of an arbitrary 3D model. Control options such as merging multiple branch clusters with different root locations, biasing the generation of the main branch, etc. are also implemented.

## 3. Background

In this paper, we use space colonization [4] as a base algorithm, which is an algorithm for generating branching structures by competing for space proposed by Runions et al. First, points are placed in 3D space, and the range of points to be used is set arbitrarily for the envelope, and whether they exist inside or outside the envelope determines the region used by the algorithm. In addition, it starts with an initial configuration of  $N$  target attractor points (usually hundreds or thousands) to be reached when extending the branch, and one

or more connection points that line up between the starting point and the target points. The required variables and sets are defined as follows:

The points used by the algorithm will be denoted by  $P_n$  for the connection points and  $P_a$  for the attraction points.

If  $A$  is the set of all attraction points, then the following relation holds:

$$\forall P_a, P_a \in A \# \quad (1)$$

If  $N$  is the set of all connection points, then the following relation holds:

$$\forall P_n, P_n \in N \# \quad (2)$$

In order to make it grow over time, it is necessary to replace some of the reached attractor points with connection points. Let  $S(P_n)$  denote the set of connection points affected by this process. If we denote the Euclidean distance between points  $a$  and  $b$  as  $D(a, b)$ , then the following equation holds:

$$D(a, b) = \sqrt{(a.x - b.x)^2 + (a.y - b.y)^2 + (a.z - b.z)^2} \# \quad (3)$$

The maximum radius of the search range between  $P_a$  and  $P_n$  is denoted by  $R_m$ . In order to prevent the same attractor point from being involved in the same process over and over again, it is necessary to delete the attractor points within a certain distance to thin them out. The radius of the fixed range to be deleted is denoted by  $D_k$ . Deletion is performed when the following relationship holds:

$$\exists P_n \in N | D(P_a, P_n) < D_k \# \quad (4)$$

$A$ ,  $N$ ,  $R_m$ , and  $D_k$  defined above are set as parameters that allow arbitrary numerical values to be entered. When each attraction point is an element in a whole set of attraction points, a connection point can influence the attraction point closest to its own position. The conditions for influencing can be expressed by the following relationship:

$$\exists P_n \in N | D(P_a, P_n) < D_k \# \quad (5)$$

When searching for surrounding attractor points, there is not always a single attractor point within range, which may place multiple attractor points under the influence of the connection point. In such a case, a set is created with multiple discovered connection points as  $S(P_n)$ . When each connection point is an element in the whole set of connection points, a new connection point  $P'_n$  is created if  $S(P_n) \neq \emptyset$ . This newly created connection point will be created at a distance  $D$  in the normalized vector direction from the old connection point. The new connection point  $P'_n$  can be expressed by the following equation:

$$P'_n = P_n + D(P_a, P_n) \frac{\vec{n}}{\|\vec{n}\|} \# \quad (6)$$

$$\vec{n} = \sum_{P_a \in S(P_n)} \frac{P_a - P_n}{\|P_a - P_n\|} \# \quad (7)$$

When an attraction point is intentionally deleted to prevent duplication when creating a new connection point, it is deleted if the following conditions are met:

$$(P_a, P_n) < D_k \# \quad (8)$$

The process up to this point is repeated until all the attractor points distributed in the space disappear. This produces an outgoing tree as shown in Figure 2.

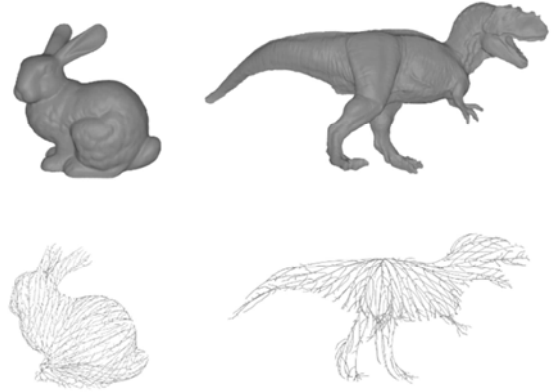


Figure 2. Example of Out-Tree by Space Colonization Algorithm.

## 4. Our Proposed Method

### 4.1. Detecting Contour Surfaces using Narrow-Band Level Sets and Point Cloud Placement

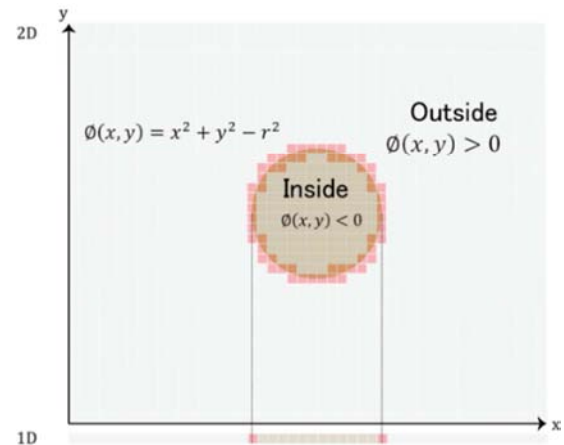


Figure 3. Image of a Narrow-Band level set.

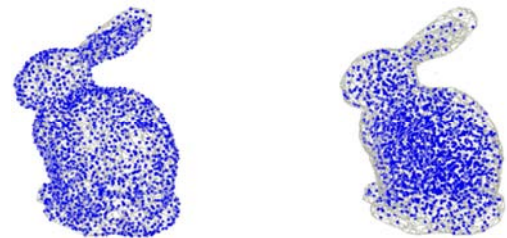


Figure 4. Placement of nodes in attractor type. (Left: Surface position of the model, Right: Using volume data).

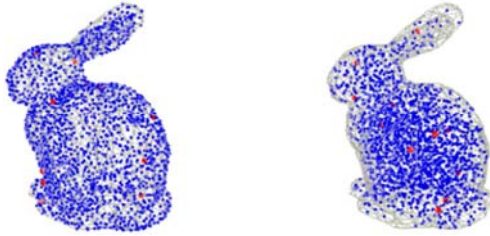
In the first step of this method, it is necessary to place a set of nodes in 3D space to apply the underlying space colonization. The placement of nodal points is based on the polygon model created in 3D space. When creating a point



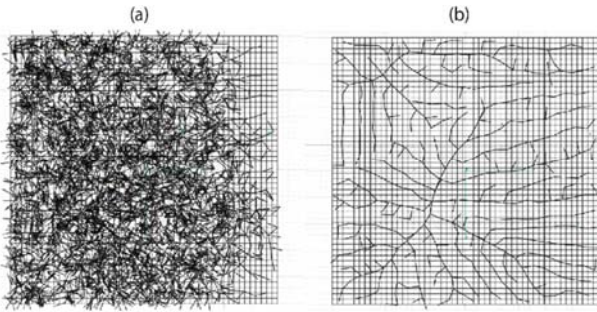
cloud to be used as a nodal point for the surface of an object, the point cloud is created at the object surface position and does not reflect the internal situation. The creation of the point cloud is done randomly for the unit area of the primitive.

On the other hand, in order to create a point cloud for the interior of an object as well, OpenVDB, a C++ library for manipulating sparse structured volume data discretized into a 3D grid, was used as the data format in this method. Each voxel, which is the smallest volume element in the sampling index space, is assigned a single numerical value (Figure 3). When some implicit function  $\phi(x,y)$  is defined, the Narrow-Band level set makes all voxels in the Narrow-Band active and all others inactive [14]. This can be used to create volume primitives, which can be used to place point clouds inside (Figure 4).

#### 4.2. Setting the Point Cloud Type



**Figure 5.** Placement of nodes in attractor type. (Red Points: Nodes type, Blue Points: Attractors type).



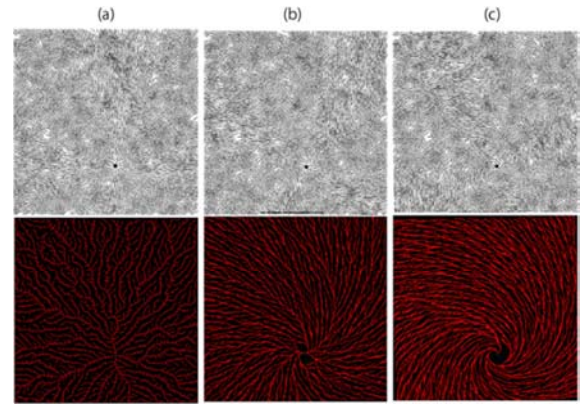
**Figure 6.** Example of shape when nodes are used multiple times.

Space colonization Algorithm searches for nodes that exist within a certain range from the starting node, and stores the unique numbers of the nodes found by the search as array data, making it possible to connect nodes together. Here, there are two types of nodes to be placed in the 3D space. In our method, we set up types using the values 0 and 1 to allow more than two types of nodes to exist in the space and to easily switch between the two attributes of node type and attractor type (Figure 5). In this method, the nodes that are the starting points of the Out-Tree is called node types, and the nodes that are the connection points are called attractor types. The nodes in the node-type and the nodes in the attractor-type are colored red and blue, respectively. The nodes in the node type and the nodes in the attractor type are used as vertices to create primitives, and the lines created by these primitives are treated as edges in the Out-Tree.

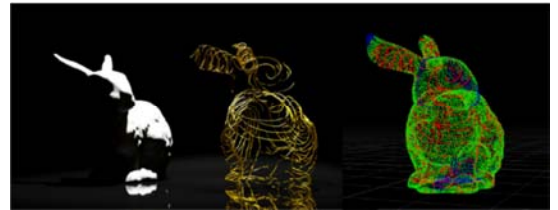
Space colonization is repeated infinitely as long as the

nodes exist in the 3D space. Therefore, as shown in Figure 6, the same node may be the starting point multiple times. In order to prevent this, we set up the system so that once a node is used, it is switched from the attractor type to the node type. However, if unused nodes belonging to the attractor type remain in the vicinity of nodes belonging to the node type that have already been used as vertices of the formed the Out-Tree, a new outgoing tree will be generated using the vertices of the already created Out-Tree as starting points, which is likely to result in undesirable shapes. To solve this problem, for nodes belonging to a node type that has been used as a vertex of the Out-Tree at least once, we set a search range as a parameter to delete nodes belonging to an attractor type that exist within a certain range. That makes it possible to thin out the nodes that belong to the attractor type, thus reducing the possibility of the same node being used as a vertex over and over again.

#### 4.3. Guide Vector Field



**Figure 7.** Out-Tree Generation Using Guide Vector Field. (a):  $\theta = 0^\circ$ , (b):  $\theta = 120^\circ$ , (c):  $\theta = 360^\circ$ .



**Figure 8.** Out-Tree Generation Using Guide Vector Field.

An extension in this paper is the use of Guide vector fields to control the shape of the Out-Tree. For the extension elements, we refer to the work by Zhang et al. [11]. And, the graph-based modeling of trees using guidance vectors as weights by Xu et al. [15]. In particular, the work by Xu et al. uses the Dijkstra's algorithm as the tree generation method. This is an algorithm that generates the shortest path between two points, and if there is no obstacle, a straight line is the shortest path. Therefore, there is a need to weight the generated branches with guidance vectors to give them branches and directions of progress.

On the other hand, this paper uses space colonization, in which the bifurcation structure changes depending on the density of points in the space, so it is necessary to influence

the point cloud to control the direction of progress. Therefore, considering the guide vector information as a field, it is possible to deliberately control the position where new nodes are created, and consequently, it is easy to control the shape of the entire Out-Tree. In this method, the normal vectors of the point in the node type are used as the axis for rotation by quaternions, and the vectors of the points in the 3D space are used to generate a field. The shape generated Out-Tree is controlled by considering the direction of the vectors when new points in the node type are created by space colonization. In this method, the normal vectors of the points in the node type are used as the axis for rotation by quaternions, and the vectors of the points in the 3D space are used to generate a field. The shape generated Out-Tree is controlled by considering the direction of the vectors when new points in the node type are created by space colonization. The first step is to generate the guide vector field to be used for control. Find the closest point belonging to the node type from the point cloud belonging to the attractor type, and obtain its coordinate value. When the point of the closest node type is  $P_c$ , the direction vector  $X_d$  used as a field can be expressed by the following equation:

$$X_d = \frac{P_a - P_c}{|P_a - P_c|} \# \quad (9)$$

The unit vector  $\hat{n}$  in the direction of rotation, and based on the created quaternions  $X_r$ , the final guide vector field  $X_f$  is expressed as follows:

$$q_o = \cos \frac{\theta}{2} + (\hat{n}x) \sin \frac{\theta}{2} + (\hat{n}y) \sin \frac{\theta}{2} + (\hat{n}z) \sin \frac{\theta}{2} \# \quad (10)$$

$$X_r = q_o x \overline{q_o} \# \quad (11)$$

$$X_f = \frac{X_r}{|X_r|} \# \quad (12)$$

If the position of the generated point  $P'_n$  belonging to the newly created node type can be adjusted by an arbitrary number  $P_d$ ,  $P'_n$  can be expressed by the following equation:

$$P'_n = \left( P_n + \left( D(P_a, P_n) \frac{X_d}{|X_d|} \right) \right) + X_f \# \quad (13)$$

Figure 7 shows the flow of growing Out-Tree reflecting the guide vector field, divided by step. Figure 8 shows an example of Out-Tree generated on the surface of the model.

#### 4.4. 3Dimensional Guide Vector Field

In Section 4.3, we showed an example of generating a whirling branching structure that cannot be generated by conventional space colonization by applying the guide vector field when nodes exist on the surface of the object. For example, objects that require point clouds to be prepared for the interior of the base object, such as trees, cannot be subjected to the guide vector field using the method in Section 4.3. By placing the base object inside the vector field, we can apply the guide vector field that acts as a guide. The calculation of the guide vector field is the same as in Section 4.3. After converting the data to point cloud data using the

sparse structure volume, the bounding box is placed to enclose the area where space colonization is applied. The bounding box in this paper is a grid composed of voxels, and density information is assigned to each voxel. In the same way, convert to voxel data of polygonal models, point clouds are placed by using a Narrow-Band level set to generate a 3Dimensional guide vector field. The vector information is obtained by using the node type in the point cloud as the base point and finding the direction vectors from the base point to the surrounding point clouds, forming a field. Figure 9 shows the results of generating Out-Tree under the influence of a 3Dimensional guide vector field.



Figure 9. An example of generating Out-Tree using a 3Dimensional guide vector field.

## 5. Applications to Modeling

In this section, we show some examples of objects that can be generated by using the Out-Tree generated by our method. When modeling natural objects or natural phenomena, graph theory can be applied when considering modeling for objects with branching structure [12]. If we define the concept of a graph as an abstract representation of things by means of points or vertices and lines connecting those points, there are many events in the real world that can be represented as graphs. For example, taking transportation networks as an example, nodes (vertices) can be replaced by cities and stations, and edges by roads and lines [6]. In the case of a molecular structure, nodes (vertices) can be considered as atoms and edges as bonds. Similarly, if we take a simplified view of the lightning shape, the nodes can be viewed as point charges in the potential space [13], which is set up when simulating lightning, and the edges can be applied to a staircase discharge path. We extracted objects that we thought could be represented by applying these concepts, and experimented with modeling using Out-Tree as a framework using Controllable Field space colonization.

### 5.1. Tree Model Generation

For the placement of the nodes that make up the tree, a random point cloud arrangement using the volume data in Section 4.1 is used. In order to add thickness to the shape, a sweeping process [16] is used to extrude the contour, which is the cross-section of the solid, using the generated Out-Tree as a trajectory. To adjust the thickness of the surface model, we use the variable value that stores the elapsed time of each node on the Out-Tree to be treated as a framework. The point number of the nodes on the Out-Tree is referenced when placing branches on the trunk. The detailed specification of

the placement position is determined by referring to the y-coordinate value of each node and sorting in ascending order of the y-coordinate value. This makes it possible to place a branch only at the node closer to the tip. The normal vectors of the nodes are used as references when placing the branches of the Out-Tree against the trunk Out-Tree. Figure 10 shows the flow of the above process. Node closer to the tip. The normal vectors of the nodes are used as references when placing the branches of the Out-Tree against the trunk Out-Tree. Figure 10 shows the flow of the above process.

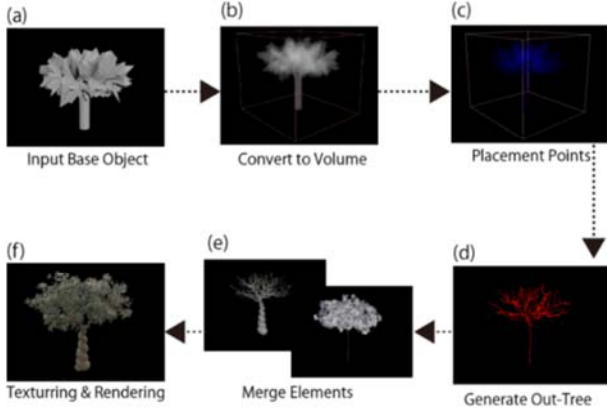


Figure 10. Tree model generating process.

## 5.2. Lightning Shape Generation

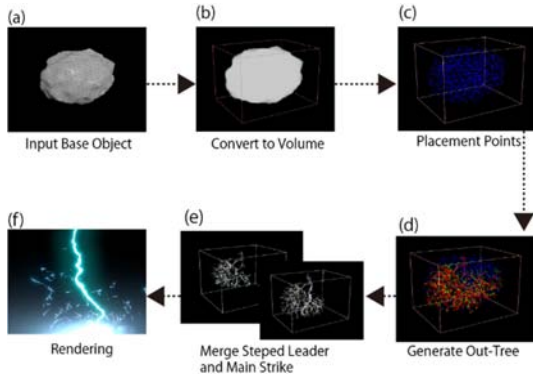


Figure 11. Lightning Shape generating process.

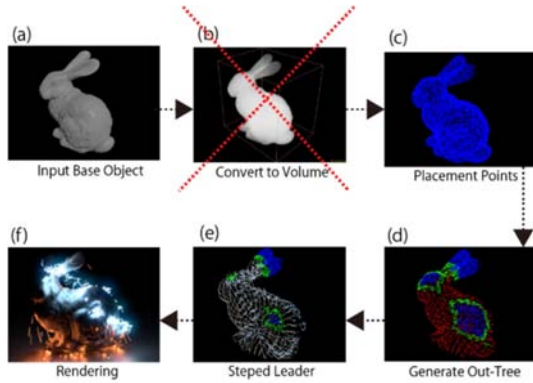


Figure 12. Lightning Shape generating process on the surface.

In order to reproduce the lightning bolt shape, it is

necessary to change the arrangement of the point clouds according to the final image. To represent lightning as a large-scale thunderstorm, it is necessary to use volume data (Figure 11). This makes it possible to reproduce the behavior of lightning in the atmosphere. However, it does not show the physically correct behavior by calculation. On the other hand, if we want to reproduce the behavior of a lightning bolt moving on the surface of an object, we need to place the point cloud with respect to the surface position of the object used (Figure 12).

When using Out-Tree with a space colonization algorithm as the lightning bolt, it is necessary to provide noise to the coordinate values of the nodes forming the Out-Tree. This makes it possible to reproduce the step-like path characteristic of lightning. In this method, noise due to fractional Brownian motion (fBm) is used. Based on the hypothesis that the discharge path of a lightning bolt is one of the shapes that have self-similarity in nature, this paper uses noise that shows self-similarity to reproduce the step-like path of a lightning bolt. If the normal Brownian motion is  $B(t)$ , the non-integer Brownian motion  $B_H(t)$  can be expressed by the following equation, where  $\Gamma(x)$  is the gamma function and  $H$  is the Hurst exponent:

$$B_H(t) = \frac{1}{\Gamma(H+1/2)} \int_{-\infty}^t (t-t')^{H-1/2} dB(t') \quad (14)$$

## 5.3. Frost Pattern Generation

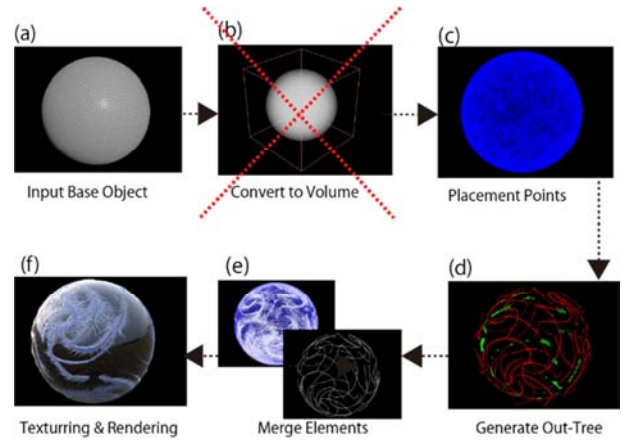


Figure 13. Frost Pattern generating process.

In the generation of frost, we attempt to control the shape of the Out-Tree using a space colonization algorithm with vector fields, which is the extended element in this paper. However, at this stage, we are only controlling the growth of the part corresponding to the frost trunk, and the arrangement of the branches extending from the trunk is the same as the arrangement of the branches in Section 5.1. Here, the point that belongs to the node type at the beginning of the process is used as the starting point of the frost, and the vector field that extends radially around this point is generated. It was confirmed that the rotation angle of the vector field affects the magnitude of the normal vector, and the larger the normal vector, the stronger the vortex strength in the vector field. Also, Figure 13 shows the process of frost generation using



the space colonization algorithm.

#### 5.4. Blood Vessels Shape Generation

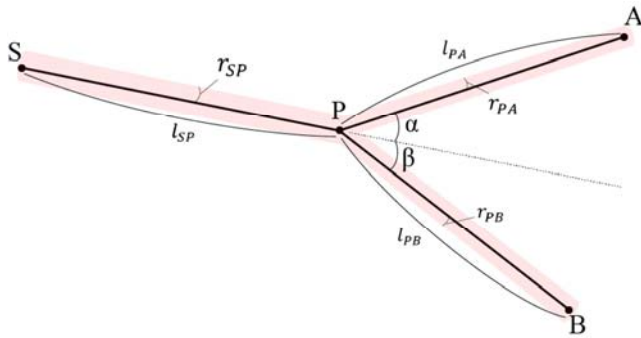


Figure 14. Relationship between branching angle and branch radius.

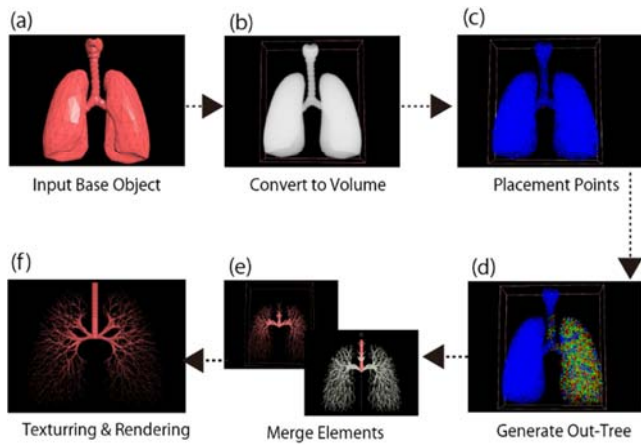


Figure 15. Blood Vessels Shape generating process.

Murray et al. reported that the laws of tree bifurcation are also present in the bifurcation of blood vessels. Here, the length at line segment SP is  $l_{SP}$  and the radius is  $r_{SP}$ . Similarly, when the same definition is applied to line segments PA and PB, the following equation is used to describe the branching angle [17-19].

$$\cos(\alpha + \beta) = \frac{r_{SP}^4 - r_{PA}^4 - r_{PB}^4}{2r_{PA}^2 r_{PB}^2} \# \quad (15)$$

By incorporating this relationship and rotating the point positions of the generated node types, a more faithful reproduction of the branching angle can be achieved (Figure 14). Strictly speaking, the bifurcation angle of a blood vessel is related to the volume in the vessel and the diameter of the vessel after bifurcation, and it has been reported that this law is also observed in the bifurcation of tree branches. The process in this method is shown in Figure 15.

#### 5.5. Terrain Pattern Generation

The modeling of mountain landscapes using computer graphics is often divided into two main objectives. The first is to use contour data or actual mountain elevation data as input for the purpose of simulating a structure close to the real one. The second is aimed at automatic generation for fictitious terrain [20]. The method in this section corresponds

to the latter. In particular, when we observe mountain ridge lines, valley lines, and the shapes of rivers along mountains, they are often tree structures without closed roads. Since space colonization Algorithm and the Controllable Field space colonization Algorithm proposed in this paper are algorithms for generating Out-Tree, if it is possible to map the numerical data of Out-Tree in the same way as sketch-based reproduction of undulations, it will be effective for generating the shapes of imaginary mountains and rivers. In this paper, we propose a method to generate the shape of an imaginary mountain or river. In this section, a polygon mesh is created for the Out-Tree by a sweeping process, and the position where undulations are given is specified by giving it RGB values and performing color mapping. For fine erosion of the terrain, sparse convolutional noise is mainly used [21]. Sparse convolution noise generates noise as a sum of randomly placed weighted kernels. If we denote an arbitrary kernel as  $k$  and the Poisson process as  $Y$ , the noise returned based on the weights can be denoted as  $N$ . Figure 16 is an image showing the change with the number of iterations. Figure 17 shows the flow of the process.

$$N(x, y) = \iint Y(u, v) k(x - u, y - v) du dv \# \quad (16)$$

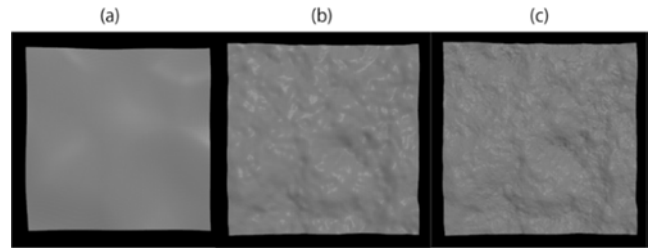


Figure 16. Comparison of different number of iterations.

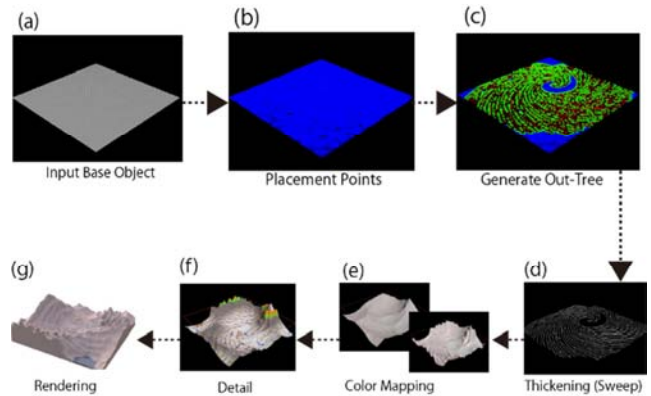
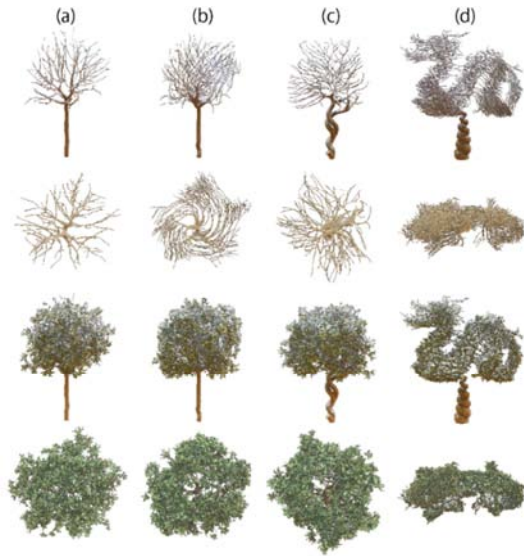


Figure 17. Terrain generation process using Out-Tree mapping.

## 6. Results and Evaluation

Examples of models that can be generated by using the method extended in this paper are shown. Algorithms that can produce branching structures similar to those produced by the Controllable Field space colonization algorithm or the conventional space colonization algorithm include the L-System and the Diffusion Rule Aggregation (DLA). It should be noted that this method is not based on accurate

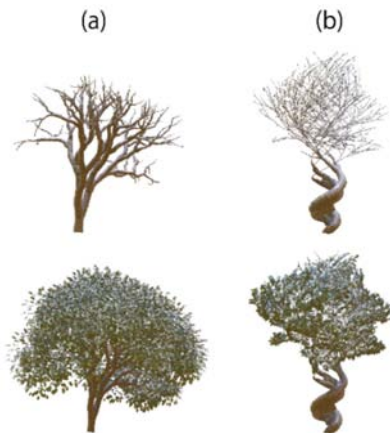
physical simulations or rules, as it is intended for use in the entertainment field. The base algorithm of this paper, space colonization, was proposed as a method for generating leaf veins and trees.



**Figure 18.** Comparison of tree models generated by our method with those generated by conventional space colonization algorithm.

- (a): Tree model using conventional space colonization algorithm.
- (b): Example of an induced vector field for a tree crown.
- (c): Example of an intensively guide vector field for the tree trunk part.
- (d): Example of changed the shape of the tree canopy to affect the guide vector field.

The model of a tree generated by the conventional space colonization algorithm is shown in Figure 18 (a). Figure 18 (b) shows the case where the crown of the tree is affected by the guide vector field. Figure 18 (c) shows the case where the trunk is affected by the rotation of the guide vector field. Figure 18 (d) shows the effect of the guide vector field on the entire model, and the Stanford Dragon is used as the user's input for the tree crown.

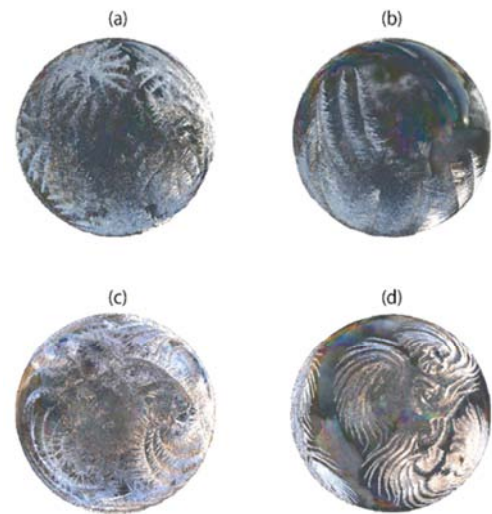


**Figure 19.** Comparison between the model using L-System (a) and the model using our method (b).

Figure 19 (a) shows the tree model generated by the

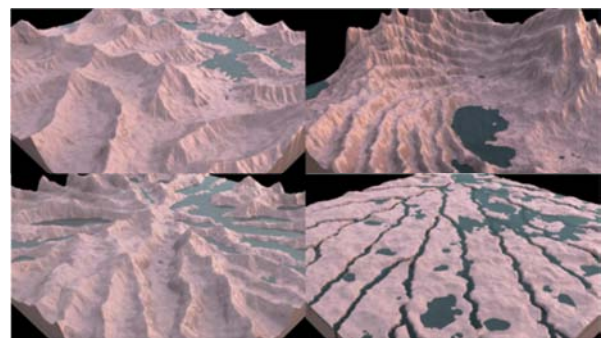
L-System, and Figure 19 (b) shows the tree model generated by the space colonization method extended in this paper. There are trees that show twisting, such as Pachira, Benjamin, and old cherry and plum trees. Our process requires a separate model or texture for the leaves, but the branches that anchor the leaves can be generated in a single step. While the L-System requires arbitrary rules to be written, the method in this paper guarantees that the branching structure is generated by the density of points in the space, and can show results comparable to existing methods in a simpler way.

Out-Tree generated by the Controllable Field space colonization algorithm can be used to generate other models, and as an example, Figure 20 shows an example of Frost Bubble pattern generation by this method.



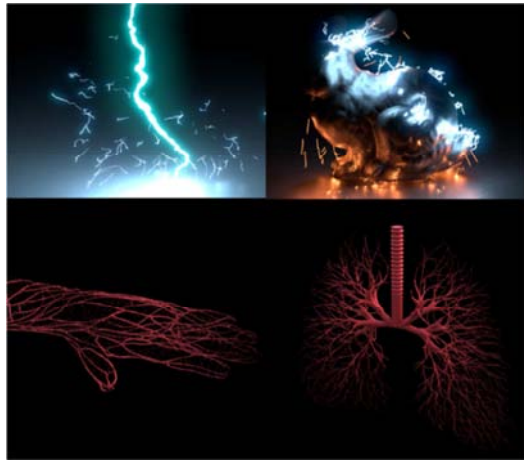
**Figure 20.** Example of Frost Bubble generation by Controllable Field space colonization Algorithm.

Figure 20 (a) shows the case with a rotation angle of 0 degrees, which is equivalent to the case with the conventional space colonization framework. In (b), the starting point is prepared on the bottom surface and grown by setting the rotation angle to 120 degrees while adding rotation; in (c), the rotation angle is set to 360 degrees; and in (d), the rotation angle is set to 720 degrees. Figures 21 and 22 shows the other possible objects that can be generated.



**Figure 21.** Examples of terrain applied Controllable Field Space Colonization Algorithm.





**Figure 22.** Other objects generated using the Controllable Field Space Colonization Algorithm.

## 7. Conclusions and Future Work

In this paper, we proposed a method to create Out-Tree, which is one of the graph structures, by using a Controllable Field space colonization Algorithm. Out-Tree generated by the Controllable Field space colonization algorithm is used as a framework for modeling, and a method for procedural and general modeling is proposed.

In this paper, it is assumed that a simple polygon model is used as the initial input data in the process of model generation. This is defined as the region for applying the algorithm, and the region is converted into voxel data according to the purpose, and the sparse structure volume is used to determine the inside and outside of the region, which is then referred to when placing the point cloud. The model with the branching structure targeted for generation cannot be shaped by the exact same process. Therefore, if a single digital set is to be controlled by the user, multiple processing networks need to be embedded in the asset and switched according to the purpose.

In addition, there are still some issues regarding the control parameters. Depending on the object to be modeled, the scale and the total number of point clouds required will vary. When generating a largescale model or a model with animation such as a lightning bolt after generating a small scale model, the location of the attracting point, which is the target coordinate position to place the next connection point, may not be included in the search range. It is necessary to adjust the granularity of the region and the maximum length of the edge of the Out-Tree that can be extended by one connection, etc. At present, some numerical sense is required. In order to solve this problem, it is necessary to prepare appropriate input values for each model and optimize them. In addition, for the models that can be generated at this stage, there are many cases of incorrect rendering due to the backtracking of normal caused by insufficient division of polygons. Similarly, it is necessary to determine the scale and the number of divisions required for each model, and thus more samples

need to be obtained and the process adjusted. Reducing the computational load is also listed as an issue for the future.

## References

- [1] Lindenmayer. A: Mathematical models for cellular interaction in development. *Journal of Theoretical Biology* 18, pp. 280-315, 1968.
- [2] Hang. Q, Ruichao. Q, and Jinyuan. J: L-system based interactive and lightweight web3D tree modeling. *VRCAI'11: Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pp. 589-592, December 2011.
- [3] A. Runions, M. Fuhrer, and A. Gaelle Rolland Lagan: Modeling and visualization of leaf venation patterns, *ACM Transactions on Graphics*, No. 3, Issue 0730-0301, pp. 702-711, 2005.
- [4] A. Runions, B. Lane, and P. Prusinkiewicz: Modeling trees with a space colonization algorithm, *Eurographics Workshop on Natural Phenomena 2007*, pp. 63-70, 2007.
- [5] O. Patrick, M. Rege and R. Bailey: Extending space colonization tree modeling for artistic control and environmental interactions. *2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*, Lisbon, Portugal, pp. 1-8. 2014.
- [6] G. Dias Fernandes and A. R. Fernandes: Space Colonization for Procedural Road Generation. *2018 International Conference on Graphics and Interaction (ICGI)*, Lisbon, pp. 1-8, 2018.
- [7] A. Hertzmann: Painterly rendering with curved brush strokes of multiple sizes. *SIGGRAPH'98, Proceeding of the 25th annual conference on Computer graphics and interactive techniques*, pp. 453-460, July 1998.
- [8] J. Stam: Flows on surfaces of arbitrary topology. *ACM Transactions on Graphics*, volume 22, Issue 3, July 2003.
- [9] E. Zhang, K. Mischaikow, and G. Turk: Vector field design on surfaces. *ACM Transactions on Graphics*, volume 25, Issue 4, July 2006.
- [10] J. Palacios, L. Roy, P. Kumar, C. Hsu, W. Chen, C. Ma, L. Wei, and E. Zhang: Tensor field design in volumes. *ACM Transactions on Graphics* Volume 36, Issue 6, November 2017.
- [11] Q. Zhang, J. Im, M. Park, M. Choi, C. Kim, and Y. Shim: Shrubbery-shell inspired 3D model stylization. *Computers & Graphics*, Volume 82, pp. 13-21, August 2019.
- [12] Team, C. F. (2020). Severe Outcomes Among Patients with Coronavirus Disease 2019 (COVID-19) - United States, February 12-March 16, 2020. *MMWR. Morbidity and mortality weekly report*, 69 12, 343-346.
- [13] Bollobas, B. New York: Springer Verlag, 1979. *Graph theory: an introductory course*
- [14] B. Sosorabaram, T. Fujimoto, N. Chiba, K. Muraoka: CG Lightning Model Taking into Account an Electric Field, *Journal of the Institute of Image Electronics Engineers of Japan*, Vol. 32, Issue. 1, pp 64-70, January 2003.

- [15] Ken Museth, Dreamworks Animation: VDB: High-Resolution Sparse Volumes with Dynamic Topology, ACM Transactions on Graphics, Vol. 32, Issue 3, July 2013.
- [16] L. Xu, D. Mould: Procedural Tree Modeling with Guiding Vectors. Computer Graphics Forum, Volume 34, Issue 7, October 2015.
- [17] J. Bloomenthal: Modeling the Mighty Maple. SIGGRAPH '85 Proceedings of the 12th annual conference on Computer graphics and interactive techniques, pp. 305–311, July 1985.
- [18] Norio Ino, Tokyo: CORONA PUBLISHING CO., LTD. 2018. Seibutsu kikai kougaku (Biomechanics).
- [19] Murray, C D. “The Physiological Principle of Minimum Work: I. The Vascular System and the Cost of Blood Volume.” Proceedings of the National Academy of Sciences of the United States of America vol. 12, 3 (1926).
- [20] N. Chiba, K. Muraoka, K. Yaegashi and M. Miura: A Geometric Modeling Method for a Mountain Produced in Erosion Processes Based on the Recursive Refinement of Ridge-Lines, The journal of the Institute of Television Engineers of Japan, vol. 45, No. 10, pp 1204-1248, 1991.
- [21] Lagae, A., Lefebvre, S., Cook, R. L., DeRose, T., Drettakis, G., Ebert, D., Lewis, J. P., Perlin, K., & Zwicker, M.: State of the Art in Procedural Noise Functions. Eurographics. 2010.