Journal for Geometry and Graphics
Volume 27 (2023), No. 2, 207–226

Computational Design Framework For Self-Forming Geometry: The Case Of Natural Tree Forks*

Anton D. Kerezov, Mikio Koshihara, Tomohiro Tachi

The University of Tokyo, Tokyo, Japan
ask@ankere.co, kos@iis.u-tokyo.ac.jp, tachi@idea.c.u-tokyo.ac.jp

Abstract. Utilizing material availability in architecture has become popular. With it, the problem of combining parts with variable geometry became apparent. Existing research defines approaches for regular elements, however, the use of highly irregular material poses new challenges. Therefore, inspired by nature, we propose a framework for material-first design, that creates equilibrated geometry from available raw material. Furthermore, the pieces are not fit to target geometry, but rather the geometry is a function of the pieces. We achieve that through a two-way optimization workflow. First, a dynamic relaxation combines the ordered members in triangular patterns. Next, to guide the solution, designers can specify indirect spatial requirements to be used in the genetic optimization of the first step. For this research, we utilized the currently neglected material of tree forks. While research on tree fork structures exists, none of it has dealt with the topic of reciprocally connecting forked branches into gridshell structures. Moreover, our focus is on generating digital models of the structures and not their fabrication.

Key Words: roundwood, rigid gridshell, dynamic relaxation, genetic optimization, reciprocal joints, tree forks, availability, architecture, material-informed

MSC 2020: 68U05 (primary), 49Q10, 51M15, 57R05

1 Introduction

Irregular roundwood forks have been neglected by the industry because of their complex shape and properties that are difficult to analyze. However, when compared to sawn rectangular timber, roundwood forks use less or no energy to process [22], reduce wood waste [2], and have structural benefits such as greater bending [4, 8, 19] and joint [20] strength. The use of such

*Extended version of the paper "From natural tree forks to grid shells: towards a self-forming geometry", published in ICGG 2022, Brasil – Proceedings of the 20th International Conference on Geometry and Graphics, Springer, 2022.

ISSN 1433-8157/

materials in construction is not fully researched, and one of the main problems is the lack of an adequate method for joining different parts despite their unpredictable sizes and sections. To fully utilize the roundwood forks for construction, we need to provide a computational framework based on the available material.

The informed use of the material availability for structural and architectural design is a relatively new approach. Workflows for man-made materials with various dimensions such as steel bars or metal plates have been partly established by Brütting [3] and Moussavi [18]. The autonomous assembly of dry stone walls by Johns [9] or fabrication of fiber-reinforced granular structures by Rusenova [23] show the possibility of deriving form from collections of particles. Structures of forked and curved wood were built in Hooke Park [17] by the utilization of digital scanning and computational optimization techniques to place the branches along target curves, forming arches and trusses. In the same way, we saw how Macdonald [13] fits straight branches to a predefined surface as Desai and Amtsberg [1, 6] also do for heavily trimmed tree crotches. Furthermore, Hamada [7] explored a branch formation of six elements with great size and weight which limited the construction by hand.

Existing workflows making arches, trusses, or gridshells using roundwood aim to realize a given surface not related to the material or its quantity and try to fit the natural forms of the branches onto perfect predefined geometries. However, this approach would require a large amount of processing of the material to create joints or trimming to match the given structural system; leading to inefficiency and potential structural weakening. In contrast, we consider the assembly of natural roundwood branches simply by pinning or tying them together by offsetting them by the radius of the members. This type of joint can minimize material waste and accommodate various wood sections thus providing a solution to one of the main problems stated earlier.

Therefore, the objective of this paper is to establish a new form-finding framework for efficient availability-driven geometric design of gridshell structures made with bifurcated roundwood. We focus on singly forked branches of medium size. However, the consideration of the individual shapes of members and their joining requires a novel approach of availability-informed design that lets the raw materials of varying sizes create the geometry instead. We refer to this approach to geometric design as "self-forming" because the material (the "self") is at the beginning of the process, rather than the form (see Figure 1).

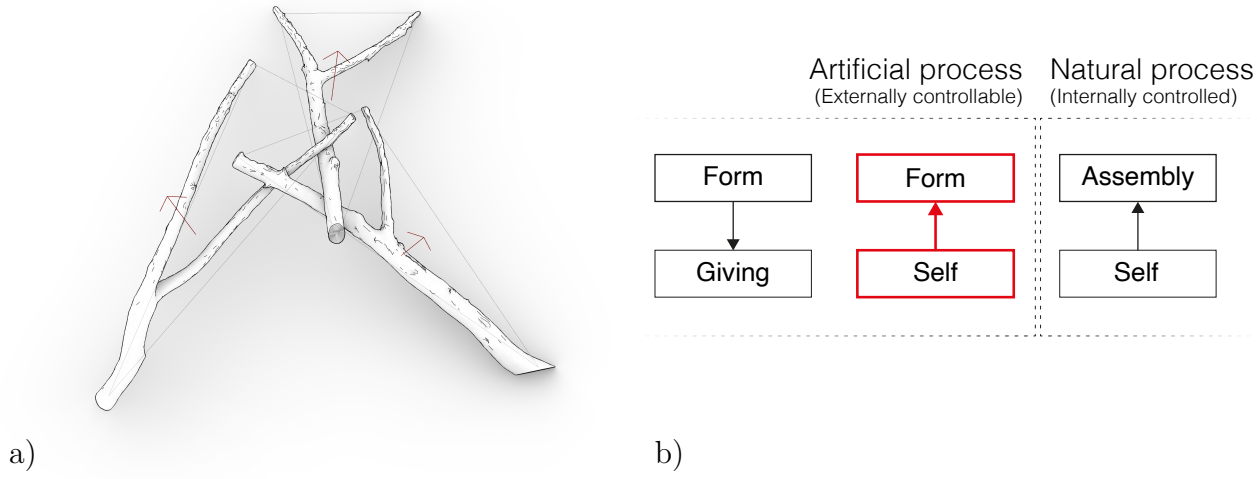

Figure 1: a) An assembly of three forked branches illustrates the challenges of using roundwood as a structural material. b) Comparison of natural and artificial form generation methods.

Self-forming was inspired by the natural "self-assembly" process, which is a spontaneous arrangement of individual components into static equilibrated structures that cannot be externally controled [15, 16]. In the self-forming process, we computationally emulate the process of self-assembly for finding the form informed by the materiality. Unlike self-assembly, however, self-forming does not use biological or chemical reactions, nor does it extend to the physical fabrication of the structures.

The main contribution of the work is the definition of the method by which irregular unprocessed forked branches are joined spatially with reciprocal connections in support-free gridshell surfaces; resulting in entirely new spatial experiences and structural formations. Although the natural process of self-assembly does not have a mechanism of external control, in the design process we need an external control to satisfy different criteria of the structures, e.g. minimum height, area, etc. Hence, we also provide such external control indirectly through the idea of "space effectors", without directly providing the form to be designed. This method of control is unlike the "form-giving" approaches devised by humans that modify the material until the predefined form is achieved. The rest of the paper can be summarized as follows:

- Section 2 introduces the control flow of the self-forming framework with its two-way optimization logic and the parameter groups.
- Section 3 shows the geometric patterns used to describe two types of triangle assemblies and the number of connections necessary to create a branch structure from irregular materials. That, however, leads to some open questions about the way branches are inserted into these skeletons.
- Section 4 defines a novel parametric reciprocal jointing method for forked elements that is unlike joints for timber laths [11] or round logs [24].
- Section 5 and Section 6 introduce an indirect method of formal control we call "space effector" and the possible surface variations, even if the same branches are used.

2 Workflow

The assembly workflow we propose in the paper builds upon our previous research methodology [10] and enhances it (see Figure 2) to be more modular, expandable, and deliver one definitive result. To start the computation we need to provide the branch 3D models and their characteristic metadata in the system. Altering any of the input parameters or scan data will generate different solutions. The *Ap* rule controls how the kits of parts are spatially arranged in patterns, *Js* – defines the joint logic (preserve, cut, extend the tree fork members), and *Ec* – representing the user spatial requirements such as area or structure height. The selection order of input elements is defined by sorting the branches by volume, size, etc. – *So*.

The self-forming approach is represented by two-way optimization. The first step is the dynamic relaxation with the Kangaroo solver [21]. For the simulations we used the preserve joint strategy, implementing a reciprocal joint, that can use the tree forks in nexorades without any secondary trimming. The joint behavior is defined by the *Js* input and its subparameters: u, a, f (see Section 4). Along with that a spatial distribution pattern (*Ap*) is provided.

The second step of the process is the evolutionary optimization of the first step. The objective function for this step can be composed of one or several space effector conditions (*Ec*) that can have a "weigh" assigned to them, giving different priorities to each one. They alter the "pool" of joint parameters u, a, f to obtain different spatial characteristics of the gridshell without force-fitting it to a target surface. The secondary optimization is optional.

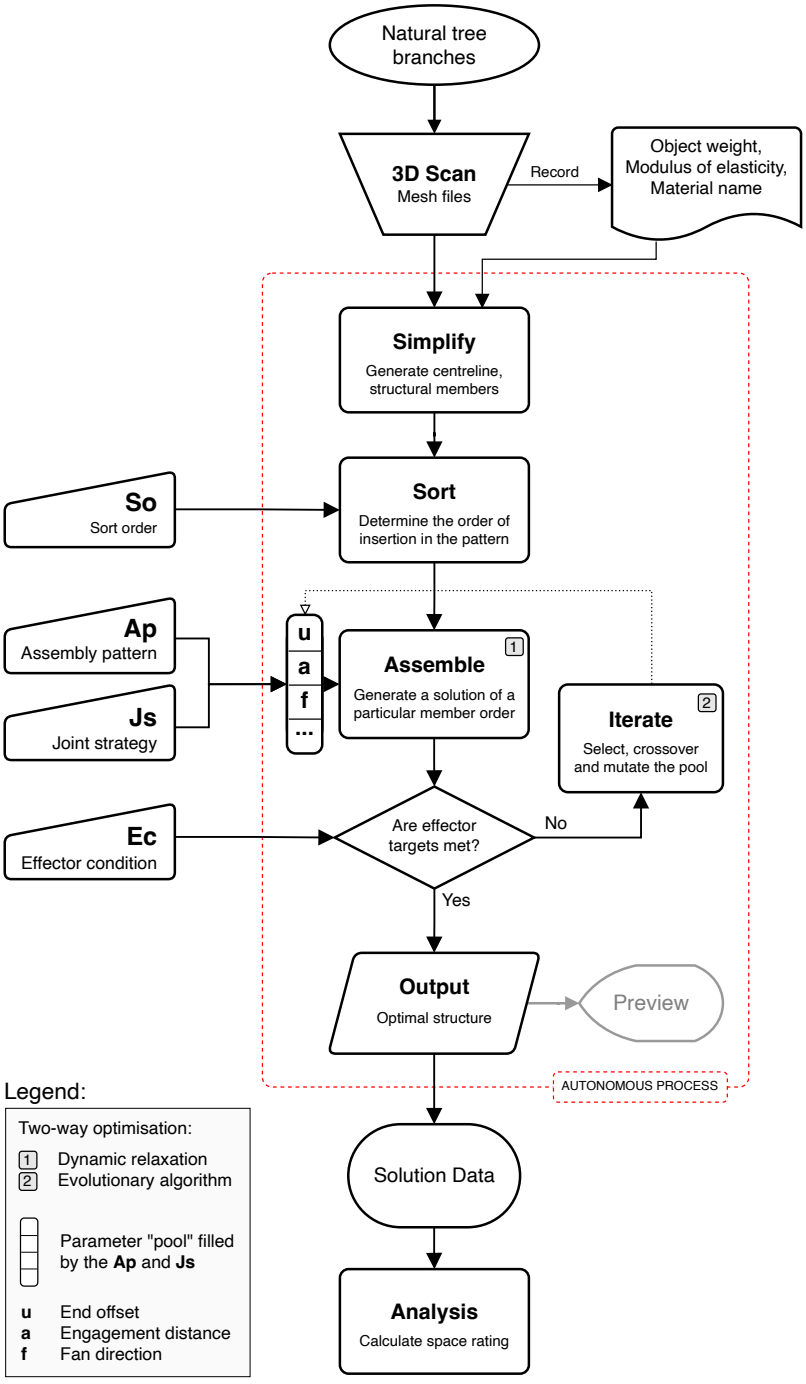

Figure 2: Data flow from real branches to gridshell solution. The user of this system can control it indirectly through the choice of the input parameter and the provided 3D data that are fed in the two-way optimization workflow.

3 Assembly Patterns and Surface Variations

Each branch fork has three ends – A, B and C; therefore, it is a triangle when simplified [10] (see Figure 3). A set of triangles can form surfaces with a wide design space that can have synclastic, anticlastic, or both curvatures at the same time. To assemble surfaces from bifurcated branches

we use these triangle simplifications in adaptable patterns – linear and radial (Figure 3). They can accommodate a different number of irregular branches because of the empty buffer triangles to compensate for the differences in the size of the triangles representing a branch [10]. Adding branch members is a function of the known triangle number series (1).

$$1, 3, 6, 10, 15, 21, 28, 36, 45, \ldots \tag{1}$$

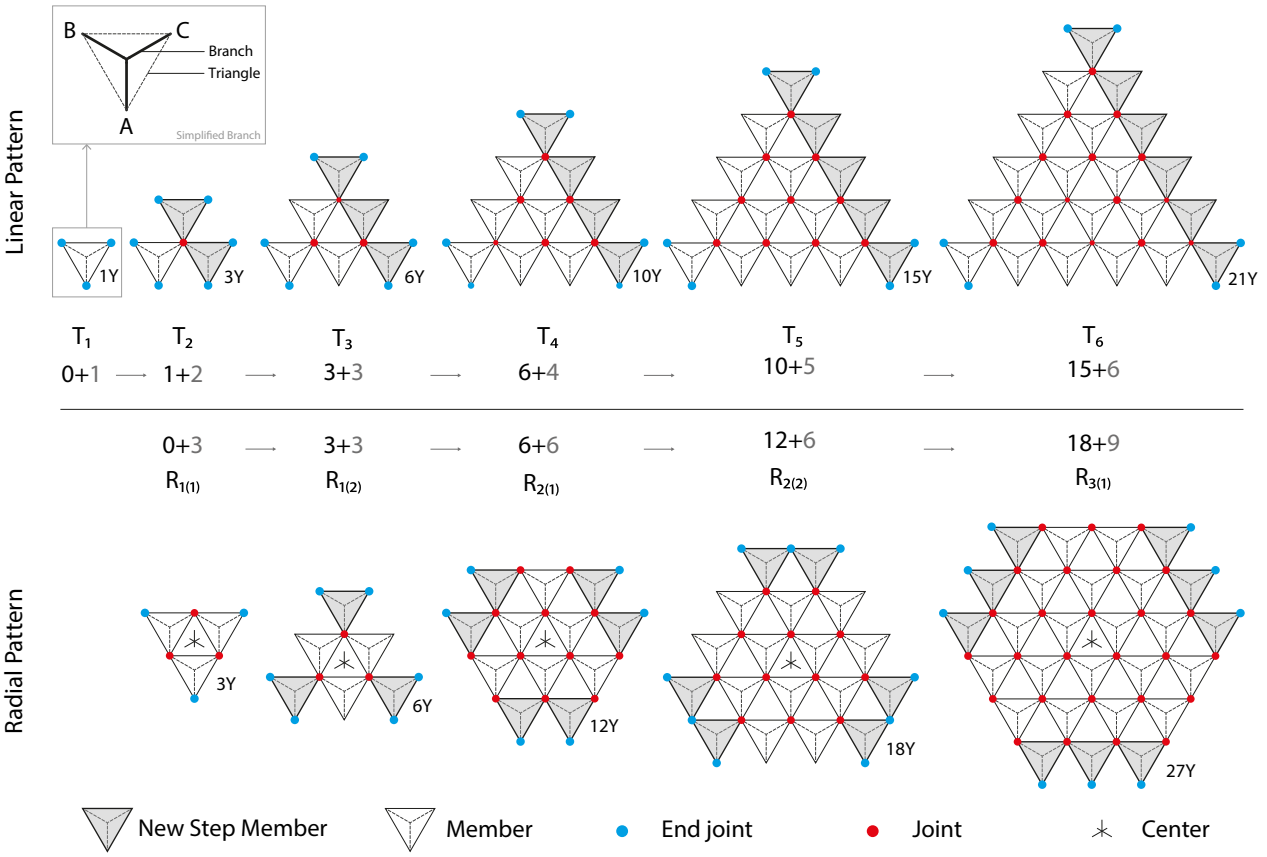

Figure 3: Branch placeholder (ABC triangle) plans of the linear triangle sequence pattern on the top and its modification into a pattern for symmetrical radial expansion on the bottom.

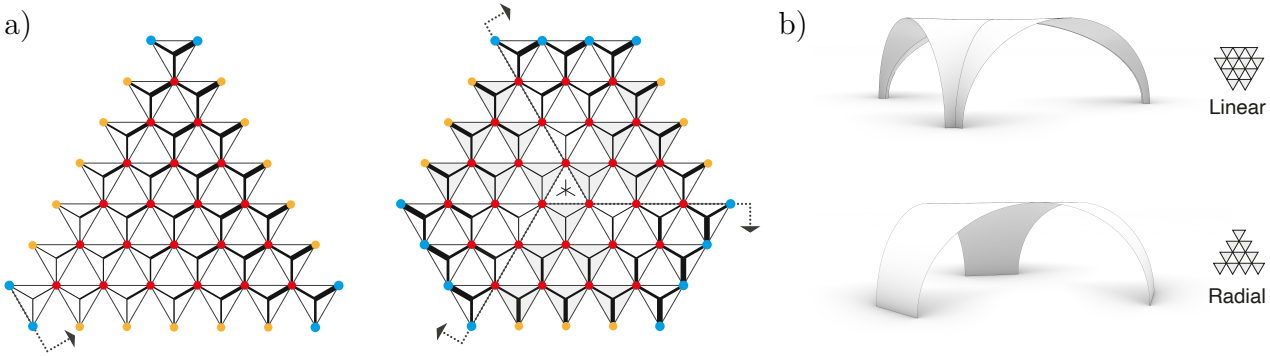

Figure 4: a) Distribution of members by size on the linear and radial pattern, provided that the input members are sorted correspondingly. b) Hypothetical concave geometry results when the appropriate joint parameters are used ($f = 1$, see Section 4.3).

$$T_n = \sum_{k=1}^{n} k = \frac{n(n+1)}{2} \tag{2}$$

$$JT_n = T_n - n \tag{3}$$

For the linear pattern, the number of triangles for each step can be found from Tn (2) and the number of joints from JTn (3), while the endpoints are always two for each of the three corners. The radial pattern is a derivative of the linear pattern. It spreads in spiral fashion in two steps $R_{n(1)}$ and $R_{n(2)}$. The number of triangles at each step can be found with the formulas (4) and (5) respectively. Each of these steps can be subdivided into three more sub-steps, therefore, providing granular insertion and better utilization of the available branches. The number of joints (6) is the same for both steps. Unlike the linear pattern, the radial has an alternating number of endpoints.

$$R_{n(1)} = T_n \times 3 + T_{n-1} \times 3 = 3n^2 \tag{4}$$

$$R_{n(2)} = T_n \times 3 + T_n \times 3 = 3n(n+1) \tag{5}$$

$$R_{n(1),n(2)} = R_{n(1)} \tag{6}$$

The structures can also have varied surface curvature but it is predominantly synclastic. The radial pattern has a structural advantage because it can effectively position stronger members near the ground (see Figure 4) and also creates wider footing. Excess input of members for both patterns is not used and is separated in a leftover data set.

It is possible to generate more patterns if the linear triangle pattern (Figure 3 top side) is mirrored, rotated, extended, etc. This could potentially create vaulted, arched, or domed surfaces. There is also the possibility of combining triangles and straight segments (such as non-forked branches) into freeform surfaces that have their members connected by joints with a valency of 3,4 or 5. This proves that the use of a triangles and lines as basic building elements of the patterns is very versatile (see Figure 5).

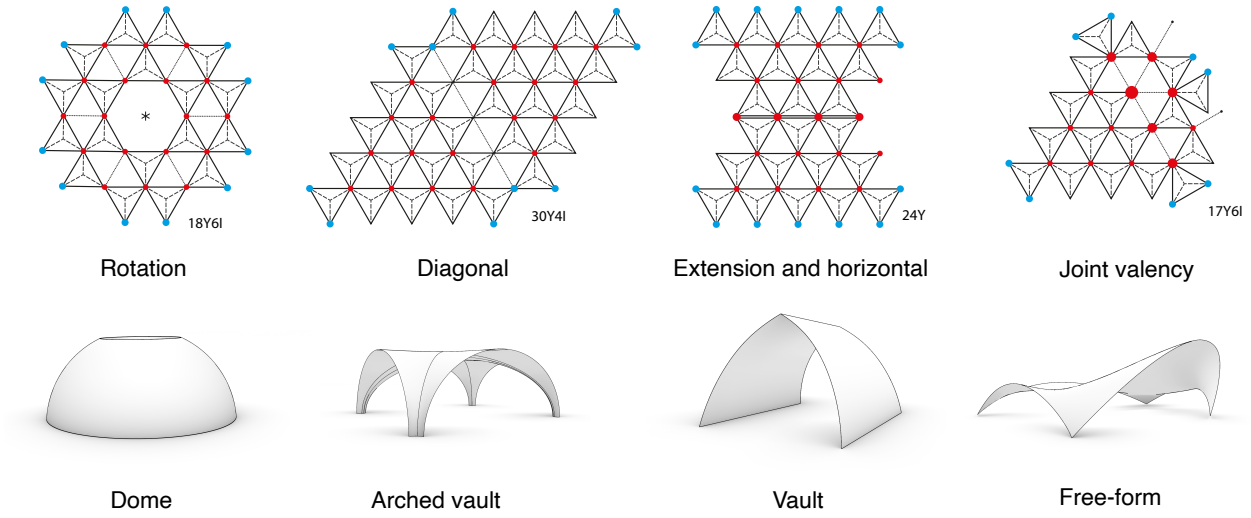

Figure 5: Other possible geometrical patterns and shapes are possible - domes, vaults, and freeform surfaces depending on the distribution pattern and the members connected together.

4 Joint Parametrization

After evaluation of different options, we selected the reciprocal joint as the means for connecting the forks in the triangular patterns, because it allows the use of irregularly shaped branches to their fullest potential without excessive cutting. Furthermore, it is easy to calculate its structural performance, because it is represented as three combined pin joints. There are some drawbacks associated with reciprocal frame structures, mainly the enormous complexity of the design process that requires non-linear solvers. Additionally, the non-redundant structural design of single-layer grid shells and the reduced node valency when compared to space frames could impact negatively the structural performance. However, those drawbacks are negligible for smaller structures. Furthermore, reciprocal structures have big column-free spans, use the material efficiently, and are easy to assemble on-site, making them preferable for use after natural disasters.

Reviewing the literature about reciprocal frame structures [5, 12, 14] revealed the important parameters that influence the behavior and geometry of one joint to be:

- Distance from the end of the branch (u)
- Engagement length (a)
- Fan direction (f) as influenced by the branch normal vector $\vec{n}$

However, we could not see examples of reciprocal joints utilizing forked roundwood; thus this paper presents for the first time an approach to connecting forked topologies into reciprocal frames (see Figure 6). Moreover, making this joint parametric allows for precise control of the self-forming process and the final surface geometry.

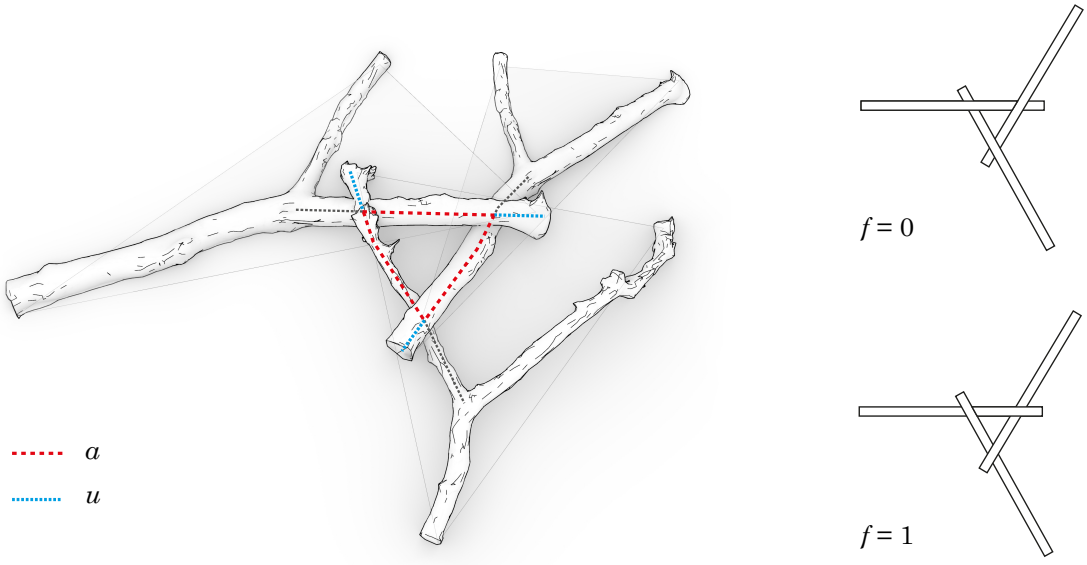

Figure 6: Illustration of the a, u and f parameters in one reciprocal joint.

4.1 Engagement Length and End Offset

Branch centerlines start at the fork's crotch, therefore, the ends are at the A, B, and C points (see Figure 3). The sizes of the engagement windows are set by the parameters u and a mentioned earlier. They measure, from the endpoint for u by a sphere with $R = u$ resulting in an intersection point C_1 and, for a from C_1 by a sphere with $R = a$ resulting in an intersection

point C_2 respectively. These two points define the way a branch connects to its neighbors. C_1 and C_2 can have different values for each one of the three centerlines (arms) of each branch. The values u and a can be any number $[0, \infty)$ but the tool will use values up to the maximum centerline length to the start (at the crotch), resulting in the six possible cases shown on Figure 7, however, case 4 must be avoided, and 5 is not optimal.

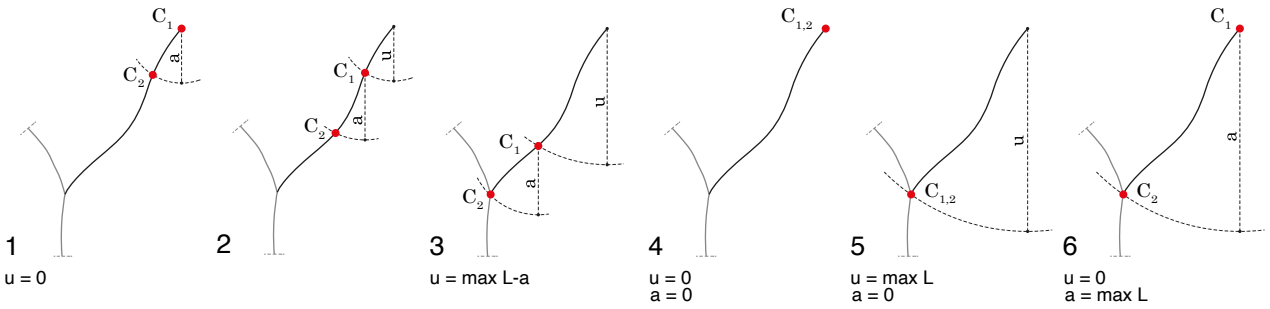

Figure 7: Engagement length and end offset have six possible point cases on one centerline for C_1 and C_2. Case number 4 is not desirable while 5 and 6 are corner cases.

4.2 Branch Orientation and Joint Links

Branch orientation is obtained from projecting the centerlines' points on a plane that contains the surface made of the endpoints A, B, and C (Figure 3 and Figure 8-c). The tallest of the three distances (from the highest H_n point to the surface H_{sn} point) is then selected as the magnitude of $\vec{n}$. A, B and C points are set in a way to make sure the branch's normal vector always points to its positive curvature; the one that creates a concave curve between the branch and the ABC surface. G is the center of gravity and O is the crotch point at which the centrelines connect. To make the reciprocal joints, all branches are linked in the respective order (determined by f) through line springs (Figure 8-a,b). During that process, the centreline points C_1 and C_2 are projected to the surface S_1 and S_2 points (Figure 9) and form the spring lines k_n. The springs are then shrunk to zero length using the dynamic relaxation physics software Kangaroo 2 [21], consequently bringing the members together into reciprocal joints.

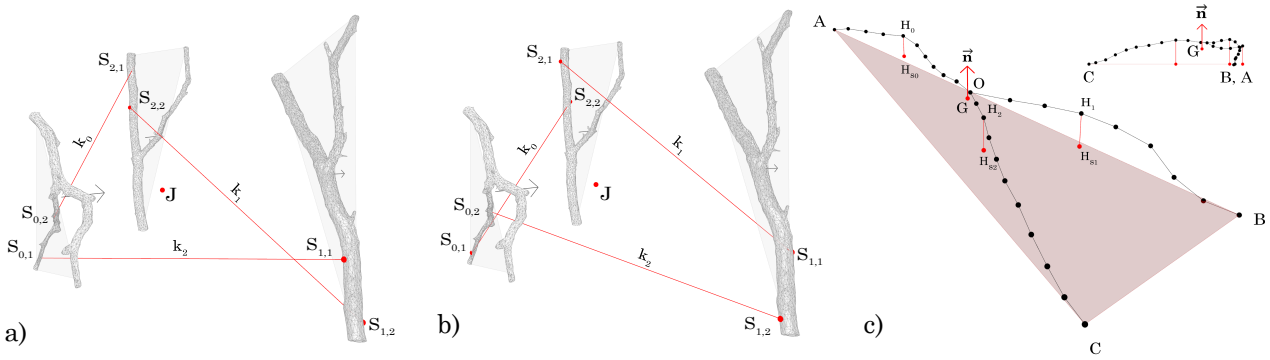

Figure 8: Linking for a) convex and b) concave joint fan. Fork normal orientation c) considered as the triangle surface normal vector $\vec{n}$ with a magnitude of the tallest height to the surface.

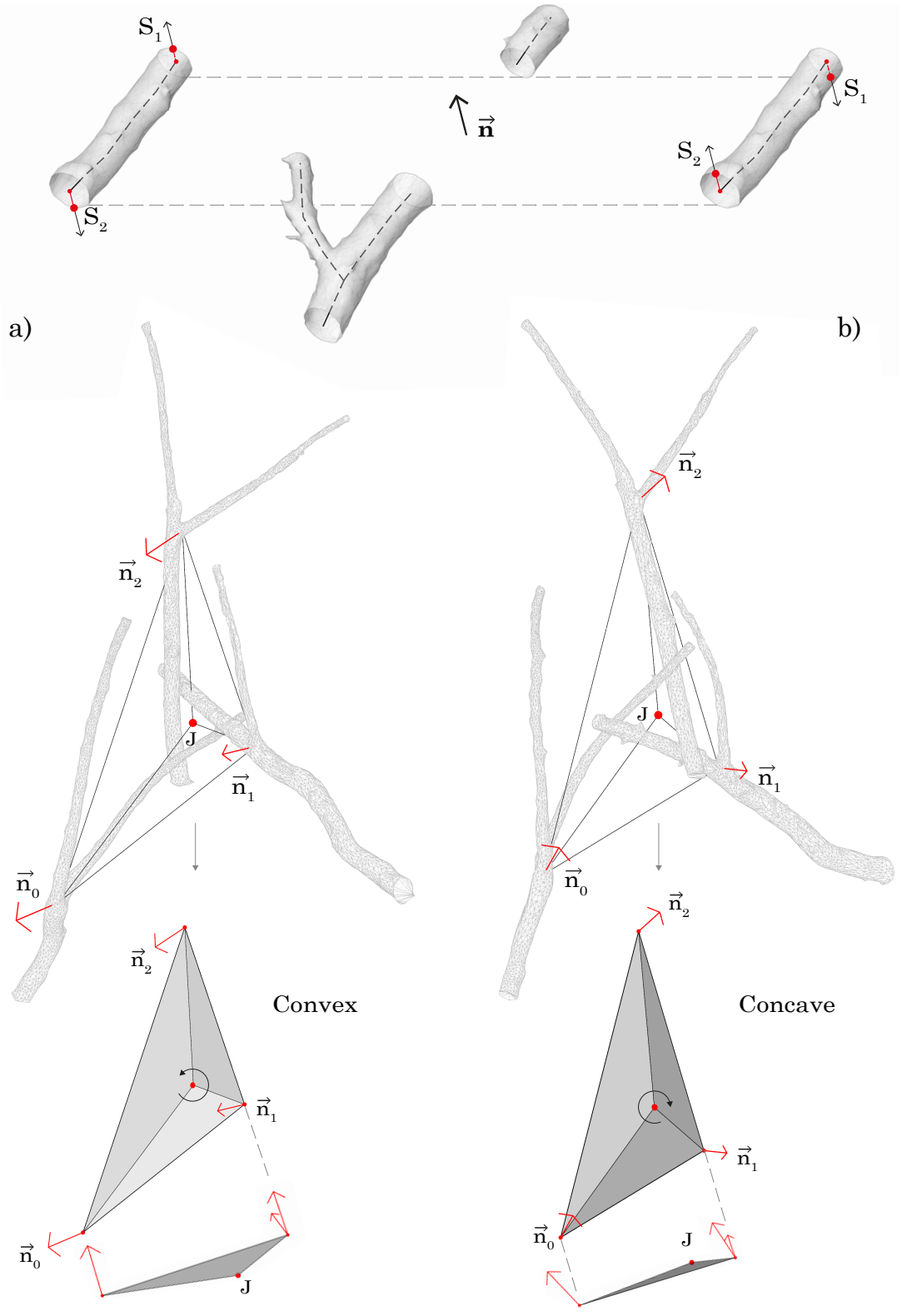

Figure 9: Setting fan direction can alter the pattern surface's shape just by switching the direction of the branch normal vector. In a) a convex resultant surface is formed and in b) a concave resultant surface. Results can vary based on the branch curvature.

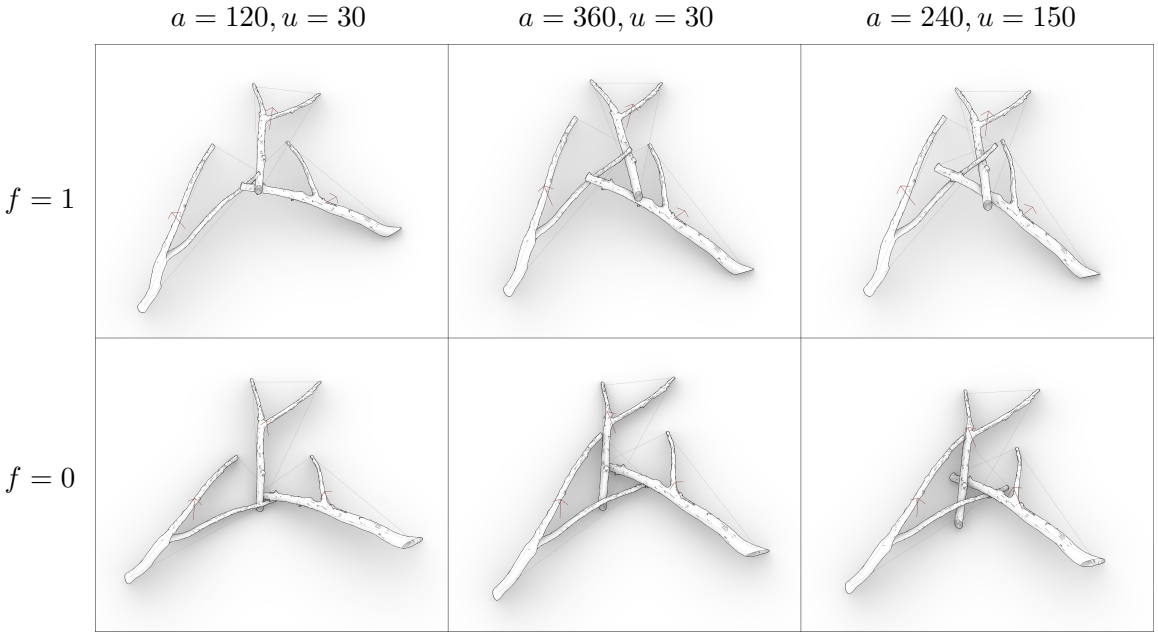

Figure 10: Examples of parameter variation and their influence on the geometry. We can see that changing the fan directions creates concave ($f = 1$) and convex ($f = 0$) shapes.

4.3 Surface Joint Points and Fan Direction

There are two points of interest S_1 and S_2 for each used arm of a branch. They are the intersection of the branch's mesh surface and vectors parallel to $\vec{n}$ with alternating directions and starting from the center points C_1 (defined by u) and C_2 (defined by a). Therefore, the branch's connection points S_i follow the branch's normal vector orientation that is related to the ABC surface (Figure 9). Furthermore, a point S_1 of one branch's arm always connects to the S_2 point of another branch's arm. Similarly easy, one can switch the fan rotation from clockwise to counterclockwise by negating the $\vec{n}$ vector of each branch found in one joint (Figure 10), which results in concave ($f = 1$) surface rather than a convex one (when $f = 0$).

5 Form Control and Optimization

The users of the self-forming implementation (a program we call Moose) can indirectly control the gridshell form by setting the So, Ap, Js, and Ec groups of parameters (see Section 3 and Section 4). Initial form finding with Kangaroo shrinks the link lines (spring goal with strength=5) to form reciprocal joints. Branch forks (centerlines, meshes, points, surface data, and structural and physical metadata) are defined as rigid body goal with strength=150 and are moved at once according to a clamp angle goal for θ with strength=1 (the default for all other goals). The curvature of the surface can be controlled by analyzing the reciprocal joints, the branch orientations and then modifying them accordingly by custom kangaroo goals.

5.1 Surface Conformity

There are cases when the Kangaroo solver would form reciprocal joints that have some of their members' normal vectors pointing in the opposite direction to the surface normal direction.

To ensure the surface is uniform, the branches' normal vectors must be relatively parallel and within specific angle tolerance $[0-\theta]$ where θ is less than 90° (Figure 11). We can enforce that by comparing each of the reciprocal joints' average normal vectors and orienting the branches to align with the needed direction.

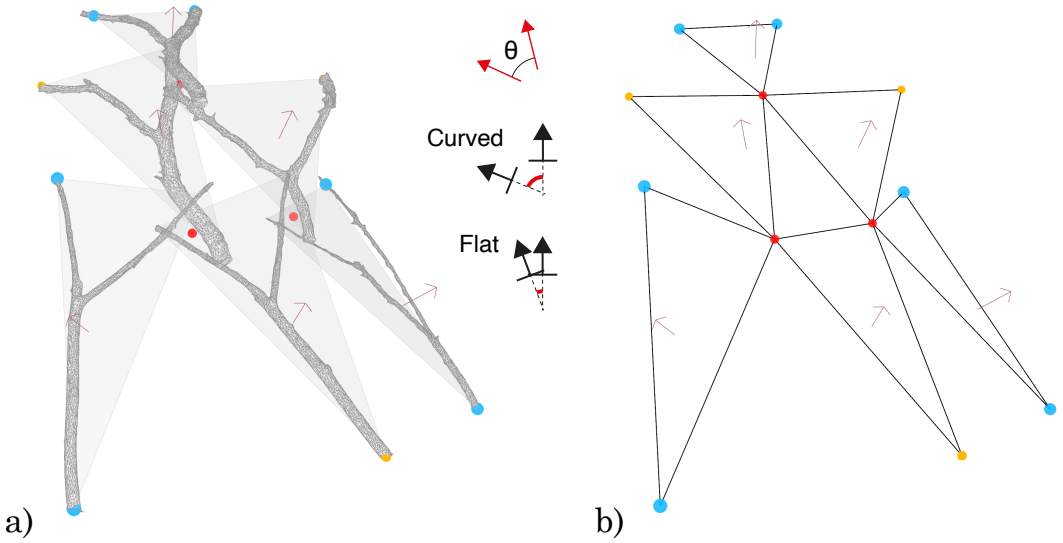

Figure 11: Aligning branch normal vectors to one direction and within tolerance θ, can control the smoothness of the structure: a) simple 6Y pattern, b) distorted line diagram connecting all joints to form the assembly pattern.

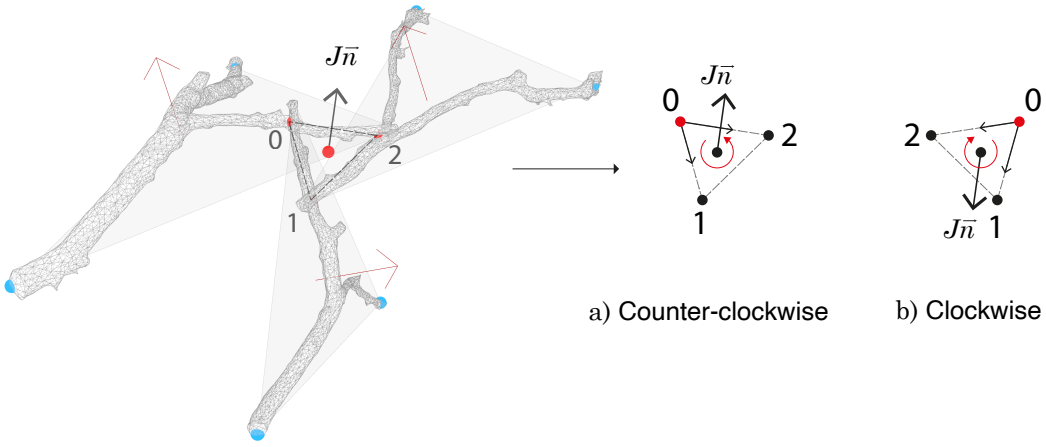

Figure 12: The joint normal vector $\overrightarrow{Jn}$ is defined from the position of the three links of the reciprocal joint. There are two possible cases: a) is the correct normal up direction and, b) is the reversed direction of the surface due to one or more reversed branches.

The average vector of each joint is calculated from the three link lines' (k_n) center points. Its direction is found by the right-hand rule for vector cross product: $a \times b = \overrightarrow{Jn}$, where $a = \overrightarrow{p_0p_1}$ and $b = \overrightarrow{p_0p_2}$ (Figure 12). If the points are in counter-clockwise order when viewed from the exterior side of the surface (Figure 12-a), it means all three branches of the joint have their normal vectors in the same direction as the average joint vector, and thus the surface vectors. On the other hand, if the points are in a clockwise formation, one or more

of the branches in the joint has their normal vector opposing the others. To make sure all normal vectors point in the same direction we programmed a custom Kangaroo goal in C# that conforms them by checking for dot product values greater than 0 and taking actions to position the members accordingly if that is not the case.

5.2 Ground Plane Orientation Methods

All solutions are generated without constraints that dictate their relationship to a ground plane. This is done for performance reasons, as the two-way dynamic relaxation and evolutionary algorithm optimization require the best possible performance from the dynamic relaxation part as it is repeated hundreds of times. To orient the structure the whole solution is analyzed and the correct position is detected based on points of interest on the existing members. Three interchangeable approaches were proposed:

1. Balance using an average plane made from all far endpoints (blue on Figure 3). Then by projecting the center of gravity of the solution on it, the zero reference origin point is found. This approach is very fast to calculate but might result in branch members under the ground plane because of the plane averaging.
2. Balance by finding the lowest point of the whole solution along the average normal vector of the structure. This routine requires slightly more computation time but ensures the members are not under the ground plane; however, parts of the structure can hang in the air. Additional support must be used in that case.
3. Balance using a plane defined from the lowest three endpoints (along the average normal vector) of the whole solution. The procedure starts as Method 2 to discover the lowest point. Then we perform a search for two other points on separate members that are further than a user-specified distance. This method is the slowest but most of the time results in the best orientation.

5.3 Space Effectors

The precise influence of the joint parameters a, u, f (see Section 4) on the whole form and its spatial characteristics is difficult to comprehend. Therefore, we introduced an easier-to-understand set of parameters that can define space – space effectors. The use of the word *effector* was inspired by the meanings it has related to orchestrating control in robotics, physiology, microbiology, etc. Space effectors are indirect boundary conditions that specify characteristics of space, letting designers and architects have intuitive control over the self-formation of geometry. There could be many space effectors active at the same time. Their priority is determined by a "weight" value that is then considered in the weighted average objective function for the optimization. Space effectors are applied in the secondary optimization of the control flow (Figure 2) and are optional.

The first and most basic space effector we created was the "Box effector". It specifies a target volume defined by box dimensions, weight, and an origin point. The box is positioned so that it is at the center of the structure. All reciprocal joints, that fall within the volume of the box are projected to the top side of the box (for example J_0 to PJ_0 and G_3 to PG_3 on Figure 13). The projection can be upwards or downwards but always the absolute distance values are taken and averaged, forming the objective value of the box effector. Consequently, the affected joints will be moved up or down, trying to minimize the objective value to zero. This indirect manipulation process affects the whole structure by trying to wrap it around the

specified box. Joints that are outside of the box volume are not measured and included in the objective value. We also developed the "Section area" effector – which tries to minimize the area of specific user-defined horizontal or vertical section of the structure (volume is defined by convex hull algorithm based on the joint points); and "Joint gap" effector – which sets minimum distance threshold after which the minimization is stopped, thus helping distribute computational resources between all joints. Many more effectors can be invented in future research as the system allows modular expansion.

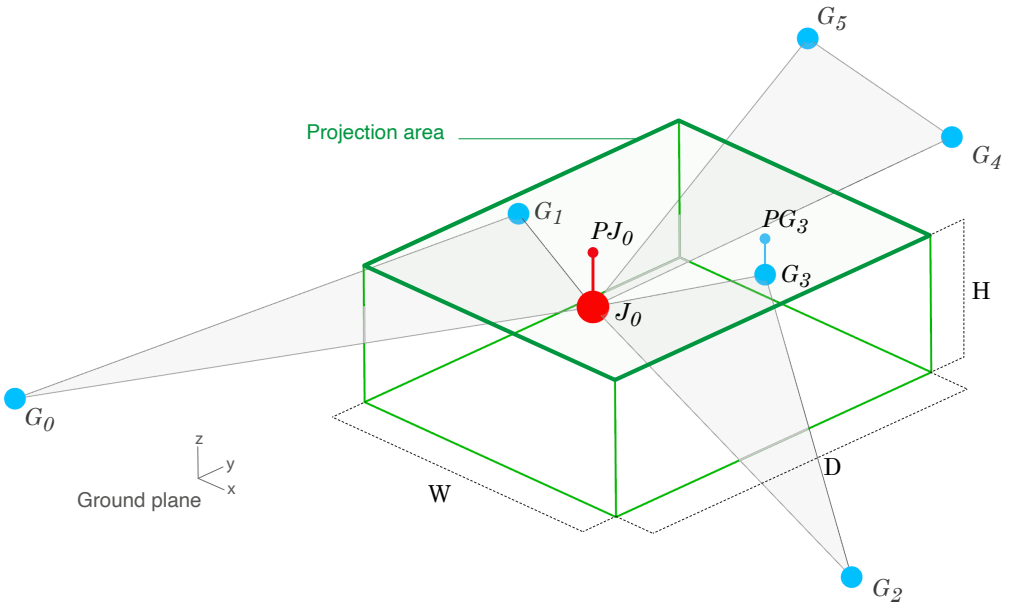

Figure 13: Box effector defined by width (W), depth (D), and height (H), exercising an influence on the two joint points that fall into its active volume: J_0 and G_3, while not affecting the other joints directly. The shown assembly is of three Y members in a linear pattern.

5.4 Effector Based Optimization

The self-forming geometry design is an approach implemented with two-way numerical optimization. The first step is the dynamic relaxation of the reciprocal frame. We do not optimize directly for a given target surface but provide effectors to indirectly guide the material-first formation. Thus, the second step is the application of these space effectors by an evolutionary multi-objective genetic algorithm optimization (GA). It changes iteratively the input parameters a, u, f for each joint by using gene mutations until an objective function value meets the optimization goals for minimum value or until the process is stopped (see Figure 14).

In the case of the linear expansion pattern, all joints that are not endpoints (in blue) are reciprocal ones and thus are made of three branch arms. For each arm are needed one u and one a value and for each joint, the fan direction can be set individually. That results in $JT_n \times 3$ times the end offsets and engagement lengths and JT_n fan directions (see Eq. 3).

For the radial expansion pattern, the joint types and counts vary: in step 1 ($R_{n(1)}$) there are pin joints at the circumference while in step 2 ($R_{n(2)}$), the joints are all rigid. The pin joint counts JR_{pin} can be deducted as $JR_{\text{pin}} = JR_{n(1)} - R_{n-1}$ and subtracted from JR_n (Eq. 6), but for the sake of simplicity, we considered $JR_n \times 3$ for a and u values and JR_n for the fan directions at the expense of computation time.

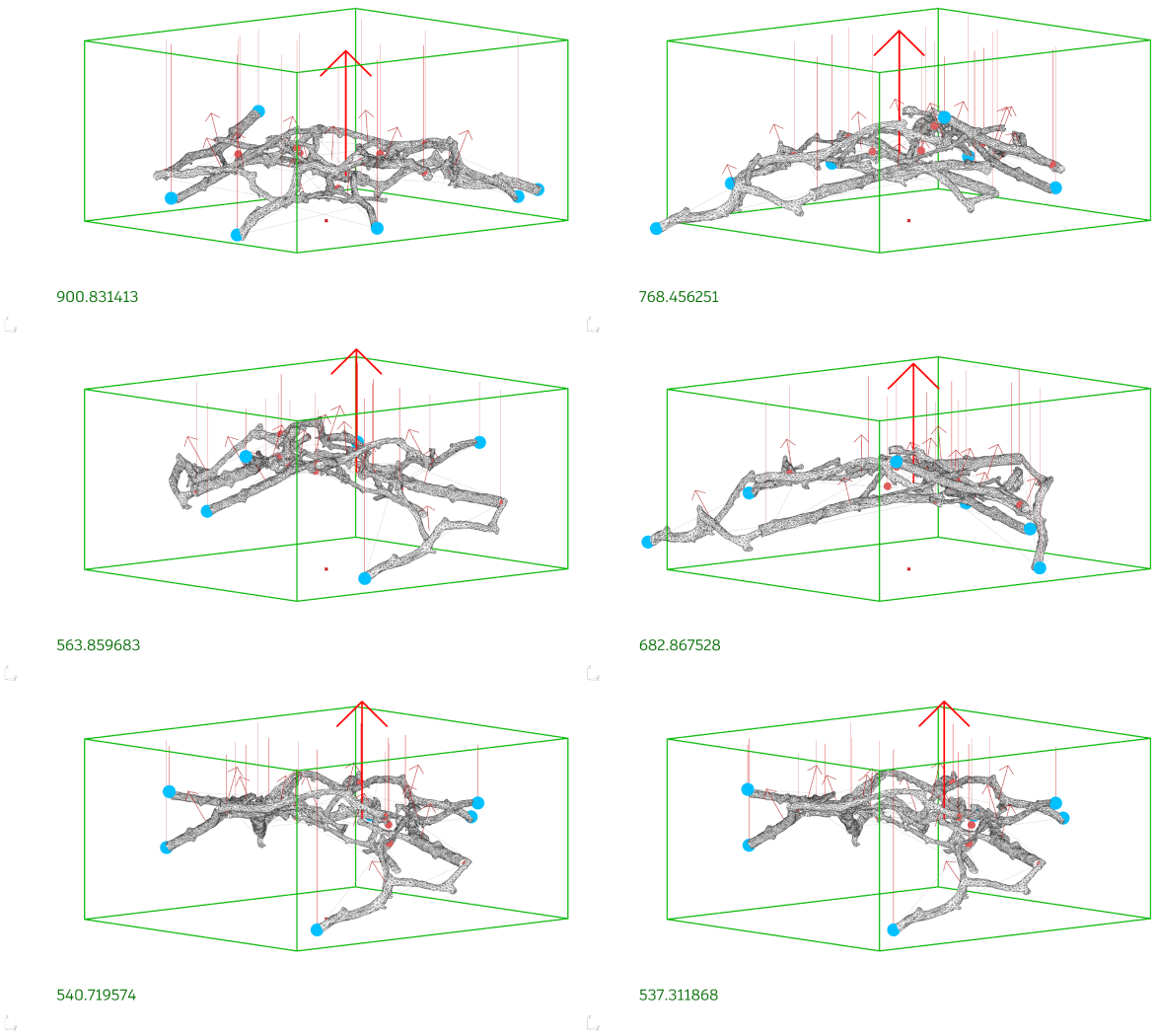

Figure 14: Different cycles of the effector-based optimization and a radial pattern of six branches: the initial state in the top left (objective: 900 mm) to the last run cycle at the bottom right (objective: 537 mm).

5.5 Space Analysis

To objectively compare different solutions we perform a spatial evaluation. It accounts for the volume, width, depth, and height of a specific form and its surface characteristics. To assess how smooth is the surface of the structure, as well as its curvature direction, a histogram of point data can be plotted. It compares the joint points' data in the Z axis to the actual Z centerline data of each branch member. The steepness of the histogram indicates how fast the structure elevates to its maximum height. We also use planograms – the distorted line diagram in plan view. Furthermore, we developed a space rating system, based on a cube, to categorize the self-formation results from the two-way optimization by a single numerical value. This evaluation is performed after the optimization as it is a calculation-heavy process.

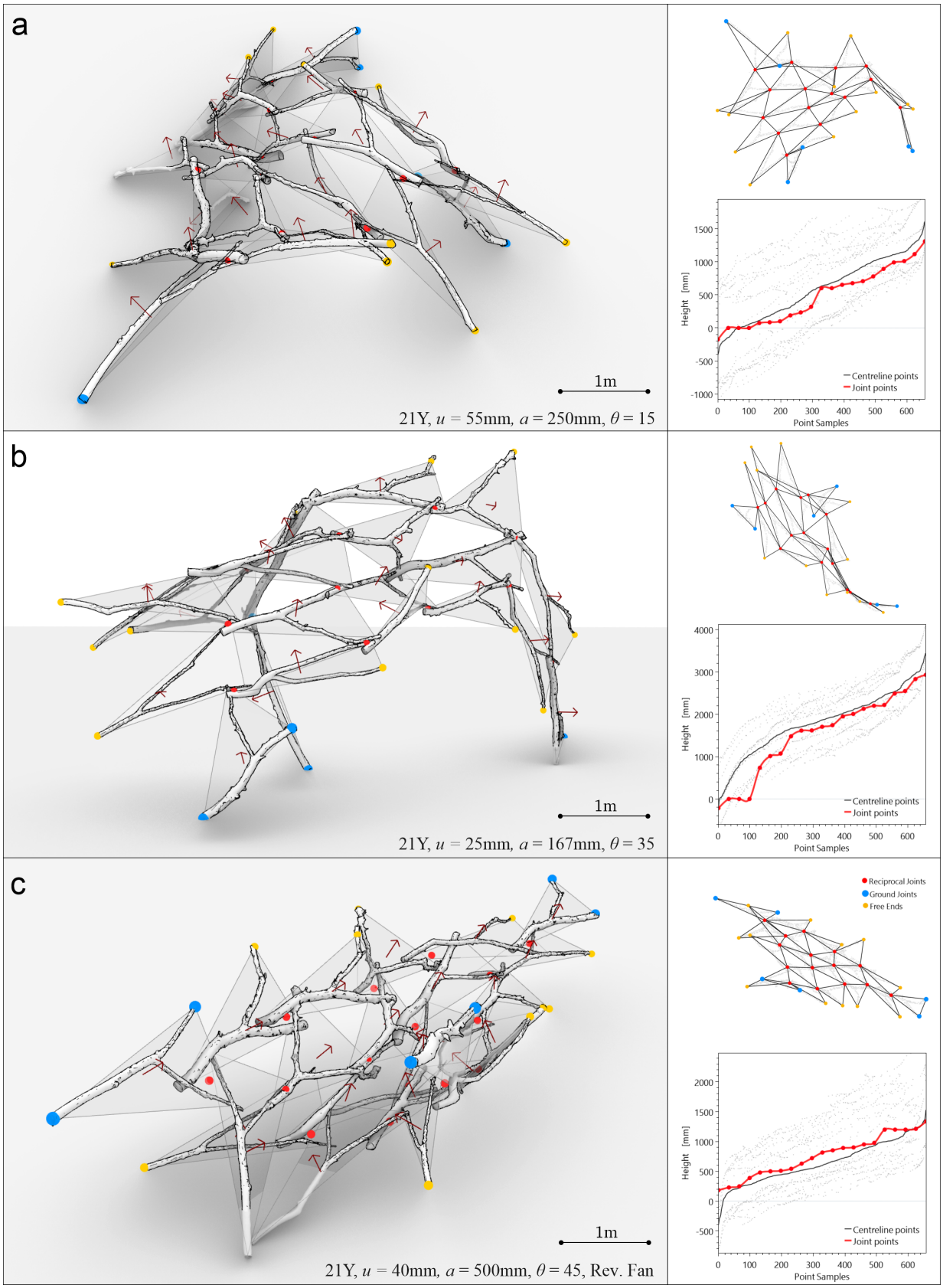

Figure 15: Comparative study of self-formed structures using linear expansion. The structures show branch normal vectors and triangle planes. On the right are the planogram and the space histogram, displaying in black – CPD, in red – JPD, and in grey – space trails as scaled 1 : 3 distance along the X and Y axis on top and bottom of CPD.

6 Results

We tested the presented self-formation hypothesis using different parameters on the same set of branches. The illustrated structures show how the parameters u, a, f (see Section 4) influence the final geometry. Furthermore, we tested the box effector on a small-scale structure thus performing the full cycle of self-formation.

Figure 15-a shows a structure with a maximum height of 1562 mm and a relatively smooth doubly curved surface. This result is expected with the selected short engagement length and smaller θ angle. The planogram covers a wide area and the space histogram displays a joint point surface (JPD) mostly below the centerline points data (CPD). The grey space trails that are related to each Z coordinate exhibit wider distribution on the X side (upper) than on the Y side (lower) and decrease towards the top of the histogram, indicating that the structure is longer along the Y axis in plan and has synclastic characteristics.

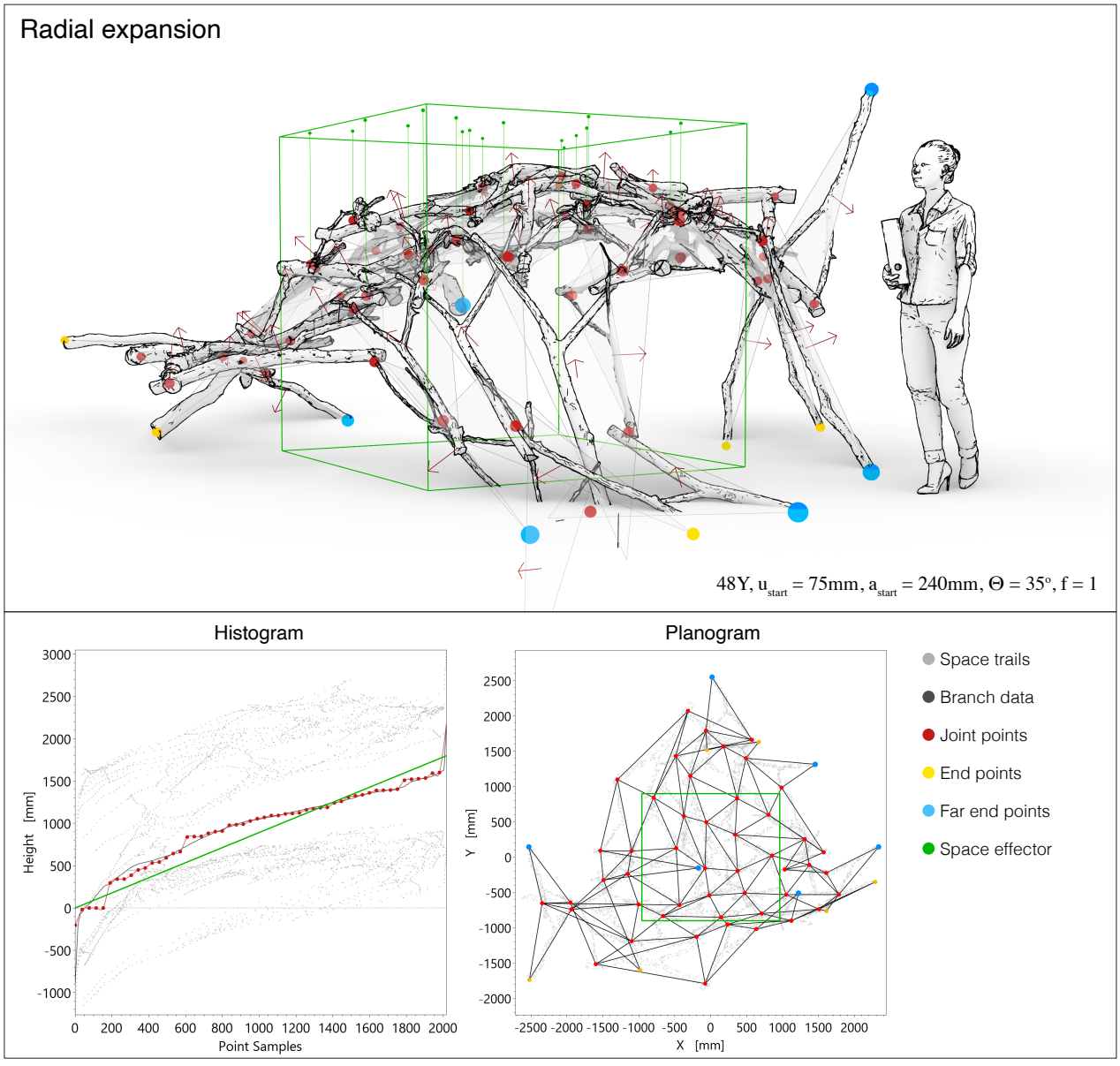

Figure 16: Self-formed assembly of 48 branches using the radial expansion pattern.

Figure 15-b shows a structure using the exact same 21 members but with much smaller $a = 167\,\text{mm}$ that is closer to the end of the centerlines, as well as θ of 35°, which altogether results in higher structure and JPD also below, but much closer to the CPD – an indication of a very smooth surface. This is the tallest and smoothest structure but as seen on the planogram and space trails it is very narrow.

In the last part – Figure 15-c, the fan direction was inverted for all joints, resulting in an anticlastic structure, with normal vectors pointing to each other, effectively creating a half-pipe shape. The JPD is firmly above the CPD indicating the catenary shape of the structure that is also indicated by the space trails expanding to the top of the CPD line.

Figure 16 shows a bigger self-formed structure that uses 48 forked branches arranged in the radial expansion pattern (see Section 3). For it, we used one box effector ($W = 1918\,\text{mm}$, $D = 1800\,\text{mm}$, $H = 1800\,\text{mm}$) and set a range of the search space of the genetic optimization for $u = [25 - 180]\text{mm}$, starting at $u_{\text{start}} = 75$ and $a = [180 - 400]\text{mm}$, starting at $a_{\text{start}} = 240$. For the other parameters we used $\theta = 35°$ and $f = 1$ (concave). Then after running the program for 30 min., we selected to our preference one of the solutions with small objective function results for the box effector.

Its maximum height is more than 2200 mm, however, that is due to a single branch standing higher than the others (an edge case that will need further constraints). The average height is therefore lower at around 1600 mm and not quite matching the expected 1800 mm from the box effector optimization. We concluded that we will need more branches to achieve that. The CPD and JPD are almost overlapping, thus signifying a smooth gridshell surface. The planogram also shows smooth distribution at the center and some irregularity at the edges.

7 Conclusion and Future Work

The original contribution of this research is the proposal of new material-first framework for self-forming the available material (natural tree branches) in geometrical gridshell formations. Their assembly is based on formation rules: parametrized reciprocal joints, space effectors, and adaptive spatial patterns united by a two-way optimization workflow. We created a computer program to implement and validate the self-forming framework. As a result, we could successfully generate, visualize and analyze the first in the world digital models of gridshell structures informed from bifurcated natural tree branches while using reciprocal connections and no target geometry fitting. We could also indirectly guide the geometry to a designer's intent for space through the new concept of space effectors.

The topic of availability-driven geometry and architecture is vast and yet unexplored. We have faced numerous challenges during our research. Firstly, the significant influence of the insert and sort order (So) of the branches was underestimated. To overcome this issue, the integration of advanced search heuristics is necessary. Secondly, the concept of orientation to the ground plane has some limitations and must be improved by including it as an effector condition of the self-forming generation. We also observed that the reciprocal joints are difficult to converge without any gaps, making the secondary genetic optimization obligatory, therefore revealing some shortcomings in using dynamic relaxation on arbitrarily positioned elements. Furthermore, the optimization is very time-consuming. The spatial and structural evaluation of the structures can also be further improved to more accurately take in consideration the irregular nature of the interior spaces.

In a future study, we must focus on the realization of the self-formed structures as real

architecture for shelters and huts. Developing the actual reciprocal joining mechanism in detail will be paramount to the safety of the structures. For that, the branches, the joint, and the gridshell structure must be evaluated by structural analysis. This routine can also become a part of the self-formation process. Using multi-layered grid shells can provide a solution to the redundancy concerns of single-layer ones. Confirming that the built geometry matches the digital one should be performed by 3D scanning. We will also explore the potential for expanding the system with a wider range of roundwood forks and straight branches, other materials, patterns, joints, and building elements such as envelopes, doors, windows, etc. As a long-term goal, we plan on researching more complicated geometries and how the material properties can be integrated further into the self-forming philosophy – even on chemical and interactional levels.

Funding

Open-access funding for the publication is provided by The University of Tokyo.

References

[1] F. Amtsberg, Y. Huang, D. Marshall, K. Gata, C. Mueller, F. Amtsberg, Y. Huang, D. J. M. Marshall, K. M. Gata, and C. Mueller: *Structural Upcycling: Matching Digital and Natural Geometry*. AAG 2020.

[2] O. Asamoah, S. Kuittinen, J. A. Danquah, E. T. Quartey, D. Bamwesigye, C. M. Boateng, and A. Pappinen: *Assessing wood waste by timber industry as a contributing factor to deforestation in Ghana*. Forests **11**, 2020. ISSN 19994907. doi: 10.3390/f11090939.

[3] J. Brütting, G. Senatore, A. Muresan, I. Mirtsopoulos, and C. Fivet: *Synthesis of Kit-of-parts Structures for Reuse*. In *Advances in Architectural Geometry*. 2020.

[4] A. M. Bukauskas: *New structural systems in small-diameter round timber*. Ph.D. thesis, Massachusetts Institute of Technology, 2015. https://dspace.mit.edu/handle/1721.1/99241.

[5] C. Castriotto, G. Giantini, and G. Celani: *Biomimetic Reciprocal Frames A design investigation on bird's nests and spatial structures*. In *Blucher Design Proceedings*, 613–620. Editora Edgard Blucher, Ltda., 2020. doi: 10.5151/PROCEEDINGS-ECAADESIGRADI2019_210.

[6] I. Desai: *Designing structures with tree forks: Mechanical characterization and generalized computational design approach*. Ph.D. thesis, Massachusetts Institute of Technology, 2020.

[7] A. Hamada and E. Sumitomo: *Using 3D Scanning And AR To Bring Out The Beauty Of Naturally Curved Logs*, 2021. https://fabcafe.com/magazine/using-3d-scanning-and-ar-to-bring-out-the-beauty-of-naturally-curved-logs/. Accessed: April 2023.

[8] M. Hartwig and E. K. Gamstedt: *On the composite design of wood branches leading to improved bending strength.* In *IOP Conference Series: Materials Science and Engineering Paper*, vol. 942. 2020. doi: 10.1088/1757-899X/942/1/012008.

[9] R. L. Johns, M. Wermelinger, R. Mascaro, D. Jud, F. Gramazio, M. Kohler, M. Chli, and M. Hutter: *Autonomous dry stone.* Construction Robotics **4**(3), 127–140, 2020. doi: 10.1007/s41693-020-00037-6.

[10] A. D. Kerezov and M. Koshihara: *A Study On Algorithm-Generated Assembly Of Curved I And Y Shaped Branches For Temporary Shelters.* Journal of the International Association for Shell and Spatial Structures **63**, 70–83, 2022. doi: 10.20898/J.IASS.2022.006.

[11] T. Kohlhammer, A. A. Apolinarska, F. Gramazio, and M. Kohler: *Design and structural analysis of complex timber structures with glued T-joint connections for robotic assembly.* International Journal of Space Structures **32**, 199–215, 2017. doi: 10.1177/0266351117746268.

[12] O. Larsen Popovic: *Reciprocal Frame (RF) Structures: Real and Exploratory.* Nexus Network Journal **16**, 119–134, 2014. doi: 10.1007/s00004-014-0181-0.

[13] K. Macdonald, K. Schumann, K. Saslawsky, and T. Sanford: *Branching Inventory: Democratized Fabrication of Available Stock.* In *26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia.* 2021. https://www.researchgate.net/publication/350740501.

[14] N. Mellado, P. Song, X. Yan, C.-W. Fu, and N. J. Mitra: *Computational Design and Construction of Notch-Free Reciprocal Frame Structures.* In *Advances in Architectural Geometry 2014*, 181–197. Springer International Publishing, 2015. doi: 10.1007/978-3-319-11418-7_12.

[15] T. Misteli: *The concept of self-organization in cellular architecture.* The Journal of Cell Biology **155**, 181, 2001. doi: 10.1083/JCB.200108110.

[16] S. Mohan, J. Jose, A. Kuijk, S. J. Veen, A. Van Blaaderen, and K. P. Velikov: *Revealing and Quantifying the Three-Dimensional Nano-and Microscale Structures in Self-Assembled Cellulose Microfibrils in Dispersions.* ACS Omega 2017. doi: 10.1021/acsomega.7b00536.

[17] Z. Mollica and M. Self: *Tree Fork Truss: Geometric Strategies for Exploiting Inherent Material Form.* In *Advances in Architectural Geometry 2016.* 2016.

[18] S. M. Moussavi, H. Svatoš-Ražnjević, A. Körner, Y. Tahouni, A. Menges, and J. Knippers: *Design based on availability: Generative design and robotic fabrication workflow for non-standardized sheet metal with variable properties.* International Journal of Space Structures **37**(2), 119–134, 2022. doi: 10.1177/09560599221081104.

[19] M. Nocetti, G. Aminti, C. B. Wessels, and M. Brunetti: *Applying machine strength grading system to round timber used in hydraulic engineering works.* Forests **12**, 2021. doi: 10.3390/f12030281.

[20] S. Özden, D. Slater, and R. Ennos: *Fracture properties of green wood formed within the forks of hazel (Corylus avellana L.).* Trees – Structure and Function **31**, 2017. doi: 10.1007/s00468-016-1516-0.

[21] D. Piker: *Kangaroo: Form finding with computational physics.* Architectural Design **83**, 2013. doi: 10.1002/ad.1569.

[22] M. H. Ramage, H. Burridge, M. Busse-Wicher, G. Fereday, T. Reynolds, D. U. Shah, G. Wu, L. Yu, P. Fleming, D. Densley-Tingley, J. Allwood, P. Dupree, P. F. Linden, and O. Scherman: *The wood from the trees: The use of timber in construction.* Renewable and Sustainable Energy Reviews **68**, 333–359, 2017. doi: 10.1016/J.RSER.2016.09.107.

[23] G. Rusenova, H. Mayer, F. Gramazio, and M. Kohler: *Jammed Architectural Structures.* In F. Melendez, N. Diniz, and M. Del Signore, eds., *Data Matter Design*, 146–159. Routledge, 2020. ISBN 9780367369156. doi: 10.4324/9780367369156-62.

[24] P. Vestartas and Y. Weinand: *Joinery Solver for Whole Timber Structures.* In *World Conference on Timber Engineering.* Santiago, Chile, 2020. https://infoscience.epfl.ch/record/281960.

Received April 21, 2023; final form July 31, 2023.