

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338584125>

Optimization and prefabrication of timber Voronoi shells

Article in *Structural and Multidisciplinary Optimization* · May 2020

DOI: 10.1007/s00158-019-02445-x

CITATIONS

2

READS

548

3 authors, including:



Hao Hua

Southeast University

16 PUBLICATIONS 51 CITATIONS

[SEE PROFILE](#)



Ludger Hovestadt

ETH Zurich

36 PUBLICATIONS 259 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



javakuka [View project](#)



Digital Fabrication [View project](#)



Optimization and prefabrication of timber Voronoi shells

Hao Hua^{1,2} · Ludger Hovestadt³ · Peng Tang^{1,2}

Received: 26 February 2019 / Revised: 16 October 2019 / Accepted: 23 October 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

We introduce a variant of the timber grid shell, the timber Voronoi shell, whose surface is reticulated by Voronoi tessellation and whose edges are made of discrete dimensional timber. This work explores form-finding methods of the Voronoi shell as a compressive funicular shell. Two closed-form solutions to shape initialization are proposed. We develop methods for minimizing the deviation from coplanarity between timber members and adjacent surface normal in order to facilitate manufacturing. A well-defined fabrication process is important for making the physical structure consistent with the structural model. A 6-axis robot with a motor spindle is employed to prefabricate the timber so the in situ manual assembly becomes easier. A parametric model describes the joint details. We formulate the robotic toolpath as a closed-form function of the resultant mesh from form finding. Thus, a general-purpose programming language can directly implement the mesh optimization and manufacturing processes without CAD or CAM software. The physical implementations, including an exhibition pavilion, validated the approach.

Keywords Grid shell · Form finding · Timber · Fabrication · Structural optimization · Toolpath

1 Introduction

The shell structure is a thin curved plate structure that transmits applied forces by normal forces and shear forces developed on the surface. The thickness of the shell is often much smaller than other dimensions of the entire structure. Thus, thin shells are efficient structural systems covering large spaces with a relatively small amount of materials. Much research has contributed to the theory of shell structures and the practical techniques of construction. Recently, the rapid

integration of digital fabrication technologies and computational design approaches has led to a new peak on this topic. Following this line, this work focuses on the form finding of shell structures and on the digital prefabrication of the components. The rationalization of the manufacturing process contributes to a true, precise implementation of the virtual geometry. Our approach drives a fabrication-oriented design in the early stages, especially concerning the joint details.

1.1 Timber grid shell

There are at least two types of timber-framed shells (Naicu et al. 2014):

1. The structure uses continuous grid members with flexible laths spanning across the structure. The laths intersect with each other at the nodes. This system is called a timber grid shell. One great idea behind such a system is the transformation from a 2D net to 3D grid shell. The structure is assembled flat on the ground to form a 2D articulated mat. The 3D shell structure is then achieved by pushing on the edges of this mat and, eventually, the structure takes its most adequate form (Paoli 2007). One celebrated case is the 1975 Mannheim Multihalle (Liddell 2015).
2. This work focuses on another type of grid shell, which is made of discrete grid members that connect at nodes.

Responsible Editor: Mehmet Polat Saka

✉ Hao Hua
whitegreen@163.com

Ludger Hovestadt
hovestadt@arch.ethz.ch

Peng Tang
tangpeng@seu.edu.cn

¹ School of Architecture, Southeast University, Sipailou 2, Nanjing, China

² Key Laboratory of Urban and Architectural Heritage Conservation (Southeast University), Ministry of Education, Nanjing, China

³ CAAD, ETH Hönggerberg, Zürich, Switzerland

Such structures often use rigid glued laminated timber (glulam) as members. Connections for glulam members are typically made with bolts and steel plates. The discrete grid shell often employs the triangular mesh for its stability, e.g., in the case of the Pods sports complex (2011), Scunthorpe, UK. Quadrilateral grids and N-gon grids are less popular probably for two reasons: (1) the problem of mobility of rectangle or n-gons should be resolved to ensure the stability of the whole structure (Dimčić 2011) and (2) Each rectangle or polygon of the grid does not typically form a plane, which may bring difficulty to the design of details.

1.2 Form finding of shells

An adequate shape is crucial to the load-bearing behavior and a consistent style of shells. Antoni Gaudí and Frei Otto were famous for using hanging chain models to design funicular structures. Based on computing technology and finite element methods (FEM), various numerical models have been developed for form finding of shell structures (Veenendaal and Block 2012). Not only have structural engineers but architects have also strived to develop new form-finding methods. The principle “form follows force” has a strong appeal in the discipline of architecture.

The force density method amounts to solving a system of linear algebraic equations, unlike other methods which are formulated as iterative algorithms (Schek 1974; Linkwitz and Veenendaal 2014). However, the external load and the force density should be predefined. Dynamic relaxation presents an iterative numerical integration of Newton's second law of motion until the entire system settles down in a static equilibrium via viscous or kinetic damping terms (Barnes 1999; D'Amico et al. 2014). The particle-spring method is intuitive: the loads and self-weight of particles are balanced with the forces generated by the elongation of the virtual ‘springs’ between the particles (Kilian and Ochsendorf 2005; Bhooshan et al. 2014). The thrust network analysis is appropriate for finding the shape of compressive funicular shells. It controls the force density by choosing a fixed horizontal projection of the solution and by manipulating the reciprocal force diagram (Block 2009; Van Mele et al. 2014).

Recent progress has been made on geometric and structural form finding with polyhedral meshes (Tang et al. 2015). Besides static equilibrium, other design criteria such as boundary interpolation, planarity of faces, panel size of shape, and costs have been integrated in the form finding. The topology of the mesh is fixed after initialization in most approaches, while Pietroni et al.'s work explored the anisotropic, density varying remeshing of grid shells (Pietroni et al. 2015).

1.3 Timber Voronoi shell

We developed a timber Voronoi shell as a prefabricated compressive shell. The shell surface is reticulated by a Voronoi tessellation. Every node connects three edges. Each edge is implemented by a dimensional timber. One can view such a network as a dual graph of the timber-framed geodesic domes (e.g., the PlayWorld Bristol, UK) whose surface is made of triangles. We have built two prototypes:

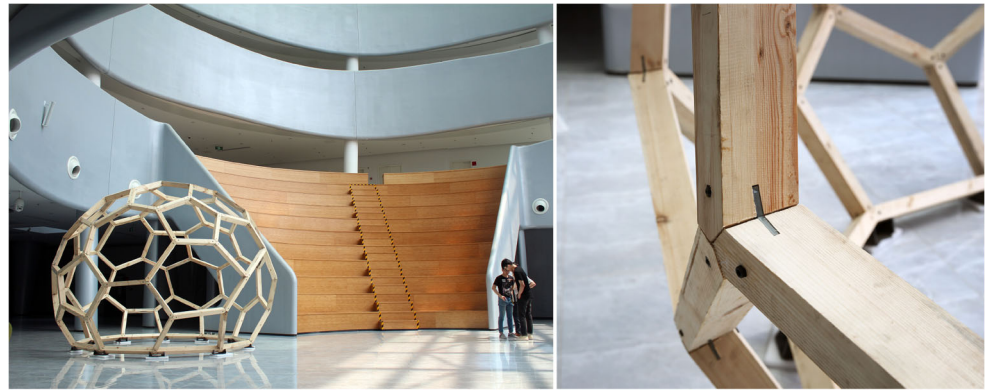
1. The mero sphere (at QINGXIAO Ltd, Nanjing, 2018) stands for the simplest prototype of the timber Voronoi shell (Fig. 1). The spherical shape (diameter 4 m) is not structurally optimized. It consists of 114 pieces of timber (cross section 68×95 mm) and 76 nodes (7.5 mm stainless steel plates). It uses 0.50 m^3 timber in total.
2. The Upsilon pavilion (at National Exhibition and Convention Center, Shanghai, 2018) is a typical timber Voronoi shell. The dimension is $6 \times 6 \times 3.7$ m. The form-finding process finds a compress-only ‘tree’ shape. It consists of 145 pieces of timber and 120 nodes. The timber weigh 243 kg (0.56 m^3) in total. Manual assembly takes six hours while de-assembly takes one-and-a-half hours. The project video can be found in Hua (2018b).

Contrary to the timber grid shell featuring flexibility (2D mat to 3D shell), the timber Voronoi shell has rigid wooden bars and, thus, is not intended to be deformable. Each piece of timber is attached to neighboring pieces of timber by metal connectors. For the connection (Fig. 2):

1. Compression is transmitted between two neighboring pieces of timber through the (flat) touching surfaces of the two.
2. The neighboring pieces of timber do not interlock with each other, in contrast to mortise-tenon connections.
3. The connectors do not bear major forces if the structure is optimized to be compression-only. However, they act on bending moments if (1) unexpected forces are applied or unexpected deformations occur or (2) the structure is not optimized (e.g., the mero sphere in Fig. 1).
4. The thickness (along the direction that is orthogonal to the mesh surface) of the timber, working together with the connectors, helps to prevent out-of-plane buckling. The width of the timber, working together with the connectors, contributes to the resistance of in-plane distortion (corresponding to diagonal stiffness in a typical timber grid shell).

An example of such a connection is illustrated in Fig. 2. The connection consists of timber (as edges), a laser-cut metal plate (as node), nuts, and bolts.

Fig. 1 Left: The mero pavilion as a timber Voronoi Shell. Right: Every metal plate (node) connects three pieces of timber (edges)



To reduce the risk of buckling, our strategy is modeling the timber pavilion as funicular structure with pin joined members, while using rigid connections (of dimensional timber) for physical construction. The rigid joints can resist unexpected loads and deformations. For example, the shape was not optimized in the mero pavilion (Fig. 1) where the joints are strong enough to bear bending moments.

1.4 Robotic fabrication

One essential appeal of shell structure is that it brings strong synergy between architects, structural engineers, industrial partners, and builders (Willmann et al. 2018). Bespoke digital fabrication has recently made great progress. Design, engineering, and construction have become three aspects of one development (Gramazio et al. 2014), rather than three intertwined processes in a traditional workflow. This is convincing in the contemporary cutting-edge study of shells (Block et al. 2018): designers do not only create the geometry

by formal logic through computer programming but also precisely inform fabrication processes by programming the CNC machines.

Robotic fabrication technologies allow the designers to directly manipulate (though not in real time) every detail in manufacturing, especially for timber structures (Willmann et al. 2016). Thus, designers are responsive for making the joints effective and assembling all the components correctly (Eversmann et al. 2017), according to the structural behavior of the shell. In our projects of timber shells, we pursued a closed-form expression of the toolpath for prefabrication, towards complete integration of design and materialization.

1.5 Contributions

We developed a timber-metal prefabricated system whose structural behavior is governed by a linear algebraic model. The structure stands as a compressive funicular shell in static equilibrium of member axial forces and gravitational pull. Compared to existing works, this paper features:

1. An integrated workflow from concept to building, including Voronoi diagram construction, mesh optimization, joint articulation, and robotic fabrication of timber structures.
2. A closed-form solution to shape initialization of shells (Section 2.1). Our geometric approach minimizes the sum of the squared lengths of all members. The resultant form typically approximates a doubly curved surface.
3. Methods for optimizing the nodal normals to satisfy coplanar conditions (Section 3).
4. In Section 4, the toolpath is formulated as a function of the resultant mesh from the form-finding process. Thus, the translation from the form finding to the executable machine instructions (for prefabrication) becomes straightforward.

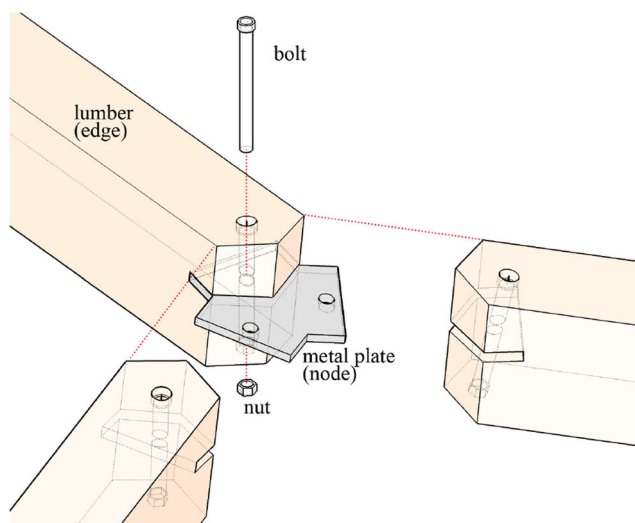


Fig. 2 The joint details of timber Voronoi Shell. There are gaps (ca. 0.25 mm) between the metal plate and the cut in the timber; thus, the compression is transmitted through each pair of neighboring pieces of timber rather than through the metal plate

The workflow of our system is illustrated in Fig. 3. The Voronoi diagram is initialized using the geometric method (LSL) or the static method (UFD). The resultant

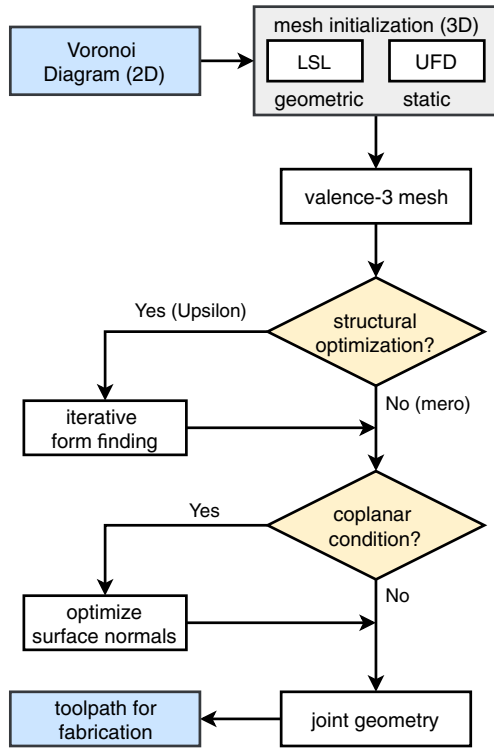


Fig. 3 Workflow from Voronoi diagram to toolpath planning

valence-3 mesh (each node links three edges) is ready for constructing geometric details for fabrication. But there are two important options. First, the shape of the mesh can be optimized to reach static equilibrium between axial forces and gravity. Second, one may impose coplanarity between timber members and adjacent surface normal in order to facilitate manufacturing. Finally, the toolpath for robotic fabrication is created from the joint geometry.

1.6 Notations

We employ a 3D Euclidean space for modeling the mesh structure. The z -axis is upwards. So the direction of the gravity can be represented by a row vector $\mathbf{t} = [0 \ 0 \ -1]$.

A given graph, as a starting point of the form-finding process, can be characterized as follows:

- K : the degree of the node. $K = 3$ in our case.
- N : number of graph nodes;
- M : number of graph edges (members);
- S : the set of indices of the variable nodes whose coordinates are to be computed.
- C : an $N \times M$ incidence matrix (C^T is the branch-node matrix):

$$C_{ij} = \begin{cases} 1 & \text{if edge } j \text{ ends in node } i, \\ -1 & \text{if edge } j \text{ begins in node } i, \\ 0 & \text{otherwise.} \end{cases}$$

$C_N \in \mathbb{Z}^{|S| \times M}$ and $C_F \in \mathbb{Z}^{(N-|S|) \times M}$ are the submatrices for the variable (or non-supported) nodes and the fixed nodes, respectively:

$$C = \begin{bmatrix} C_N \\ C_F \end{bmatrix}$$

The form-finding process is concerned with the geometry of the mesh:

$\mathbf{x} \in \mathbb{R}^{N \times 3}$: an $N \times 3$ matrix; each row represents the coordinates of a node.

\mathbf{u} : an $M \times 3$ matrix representing the directions of edges

$$\mathbf{u} = C^T \mathbf{x} \quad (1)$$

l : a vector consists of the lengths of all edges (bars):

$$l_i = \sqrt{\sum_{j=1}^3 \mathbf{u}_{ij}^2}, \quad \forall 1 \leq i \leq M \quad (2)$$

$\hat{\mathbf{u}}$: a normalized version of \mathbf{u} , i.e., $\hat{\mathbf{u}}_i = \mathbf{u}_i / \|\mathbf{u}_i\| = \mathbf{u}_i / l_i$.

2 Mesh construction and optimization

Our methods of shape initialization, form finding, and joint design require the nodes to be of degree 3, i.e., there are three edges incident to each node. The 2D Voronoi diagram is often more flexible than the hexagonal grid, for example, when the cells' sizes are differentiated. Two steps were employed to create a 2D Voronoi diagram:

1. Randomly place a set of points in a 2D polygon representing the construction site. Each point pushes its neighboring points (within a predefined radius) iteratively. Eventually, this self-organizing process makes all points evenly distributed within the polygon. Alternatively, the points are directly distributed on a 3D surface, e.g., on a sphere (Fig. 1).
2. Create the 2D Voronoi diagram (e.g., Fig. 4 top) from the distributed points. The diagram is represented as an undirected graph.

2.1 Shape initialization

Given a fixed network topology, most form-finding processes (Veenendaal and Block 2012) commence with a flat geometry. How to initialize the shape when the fixed points are not located at the same height is a rarely treated problem. There are both pragmatic and aesthetic concerns on initialization. The former requires that the geometry should not significantly differ from the one produced by structural calculation while the latter requires that the shape should be smooth and the polygons should not be visually

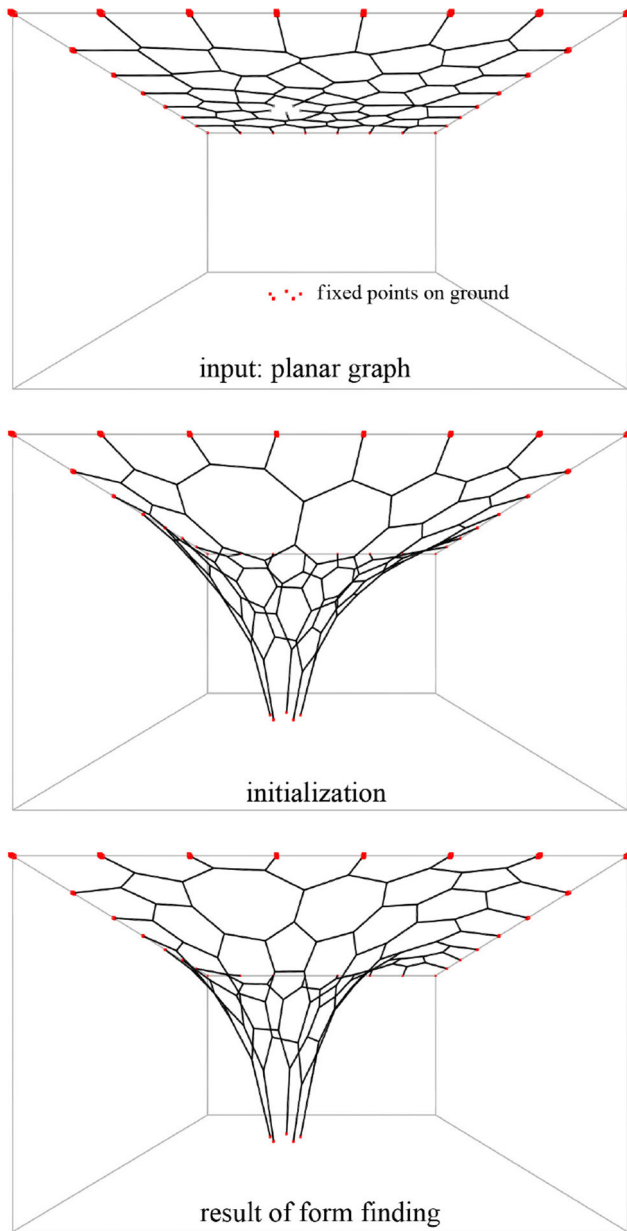


Fig. 4 The input planar graph, shape initialization (LSL), and form finding in the Upsilon project. Fixed points are marked in red

distorted (Dimčić 2011). In this subsection, two closed-form solutions for shape initialization are proposed. The first minimizes the sum of the squared lengths of the network edges while the second simplifies the force density method.

2.1.1 Least squared lengths (LSL)

The geometry of the predefined ground structure is often a planar graph assuming that the fixed points are located at the same height (on the ground). However, the shape initialization becomes non-trivial when the fixed points have different positions on the z -axis. We propose a geometric

solution to this problem by minimizing the sum of the (squared) lengths of all bars

$$\mathbf{x} \leftarrow \arg \min_{\mathbf{x}} \|\mathbf{C}^T \mathbf{x}\|_F^2 \quad (3)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The three columns of matrix \mathbf{x} correspond to x , y , and z -component of the 3D coordinates of all nodes. The matrix \mathbf{x} can be decomposed into $\mathbf{x}_N \in \mathbb{R}^{|S| \times 3}$ and $\mathbf{x}_F \in \mathbb{R}^{(N-|S|) \times 3}$ for the variable (or non-supported) nodes and the fixed nodes, respectively:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_N \\ \mathbf{x}_F \end{bmatrix}$$

Thus, submatrix \mathbf{x}_N consists of the actual variables of the objective function (3). Submatrix \mathbf{x}_F provides the boundary conditions for the structure. The quadratic function reaches the minimum value when its gradient is equal to 0. It leads to a system of $3|S|$ linear equations

$$-\mathbf{C}_N \mathbf{C}_N^T \mathbf{x}_N = \mathbf{C}_N \mathbf{C}_F^T \mathbf{x}_F \quad (4)$$

Solving (4) gives \mathbf{x}_N as the initial shape of the network, which often resembles a doubly curved surface connecting the fixed points. Compared to an initial shape of flat mesh (with fixed points with distinct z -components), such a ‘curved surface’ resembles the final result of form finding (Fig. 4).

Compared to Dimčić’s (2011) relaxation method, which tries to keep the member lengths as similar as possible, LSL minimizes the sum of the squared lengths of members instead. The advantage of LSL is that it results in a closed-form solution (4). Dimčić (2011)’s model is more sophisticated as it concerns other criteria (such as keeping the Voronoi diagram on the predefined surface) and subsequently relies on metaheuristic for obtaining solutions.

2.1.2 Uniform force density (UFD)

Although LSL is developed from a purely geometric perspective, its formulation is closely associated with the force density method, which is based on static analysis. The force density method has succeeded in deriving an analytical model by regarding the force-length ratio as constants for each member (Schek 1974). Linkwitz and Veenendaal (2014) recently generalized this model to nonlinear cases. Here, we take an opposite approach: simplifying the original method further for a convenient initialization of the shell shape.

The force density method features

$$-\mathbf{C}_N \mathbf{Q} \mathbf{C}_N^T \mathbf{x}_N = \mathbf{p}_N + \mathbf{C}_N \mathbf{Q} \mathbf{C}_F^T \mathbf{x}_F \quad (5)$$

The diagonal matrix \mathbf{Q} denotes the force density, or the force-length ratio of members. Matrix \mathbf{p}_N denotes gravitational pull on all non-supported nodes. By imposing a uniform density, i.e. $\mathbf{Q} = \mathbf{I}$, we have the uniform force

density (UFD) model for initialization. The only unfixed term (except for the variable \mathbf{x}_N) is the load $\mathbf{p}_N \in \mathbb{R}^{|S| \times 3}$, which can be uniformly specified by

$$\mathbf{p}_N = gJ \otimes \mathbf{t}$$

where \otimes denotes tensor product; g is a constant specifying the shell's weight; J is a $|S|$ -dimensional column vector of 1s.

Equation (5) is identical to the LSL model (4) when $Q = I$ and $\mathbf{p}_N = 0$. The UFD method is probably the simplest initialization method concerning the static equilibrium of member axial forces and external loads.

2.2 Determining geometry towards static equilibrium

This section introduces an iterative algorithm that finds the shell geometry in static equilibrium. The sum of vectorial forces from the three incident bars must equal the load lumped in the node. The goal is to determine the (non-fixed) nodal positions as variables, which determine the directions of members $\hat{\mathbf{u}}$. The static equilibrium is represented by the equation

$$H(CF\hat{\mathbf{u}} - \mathbf{p}) = 0 \quad (6)$$

where F is an $M \times M$ diagonal matrix belonging to the member forces f_j , $j \in [1, M]$. H is an $N \times N$ diagonal matrix

$$H_{ii} = \begin{cases} 0, & \text{if node } i \text{ is fixed,} \\ 1, & \text{otherwise.} \end{cases}$$

The external forces on the nodes are

$$\mathbf{p} = g\mathbf{w}\mathbf{t} + E \quad (7)$$

where \mathbf{w} (N -dimensional column vector) denotes the nodes' mass; g is a constant of gravity; E ($N \times 3$ matrix) denotes the external forces applied to the nodes.

A more realistic way is to distribute the weights of bars onto the nodes

$$\mathbf{p} = \frac{g}{2}\tilde{C}l\mathbf{t} + E \quad (8)$$

as l (M -dimensional column vector which represents the lengths of all bars) would change during the form-finding process. \tilde{C} is an absolute version of C , i.e., $\tilde{C}_{ij} = |C_{ij}|$, $\forall i, j$.

In general, people have not discovered analytic solutions (except for approaches imposing additional constraints) to the equilibrium problem (6) given load conditions and boundary constraints (Veenendaal and Block 2012). The axial forces F are associated with the mesh geometry and the nodal loads \mathbf{p} depend on the lengths of bars as (8), so the nonlinear (6) is difficult to be written in a closed

form for standard solvers. There could be many possible solutions instead of one unique solution. Thus, specific iterative numerical methods have been widely investigated. For example, the particle-spring system (Bhooshan et al. 2014) provides an intuitive heuristic mechanism to solve (6). Recent form-finding methods (Tang et al. 2015; Pietroni et al. 2015) have integrated static equilibrium constraints and other design criteria such as surface fairness, boundary-alignment, and surface remeshing.

The structure reaches static equilibrium between member axial forces (there are no shear forces or bending moments in the model) and external loads if the nodal positions satisfy (6). Thus, an iterative process towards (6) implicitly reduces shear forces or bending moments in the structure. This section introduces a simplified variant of the particle-spring system. The particle-spring method proposes a physical-based algorithm (Kilian and Ochsendorf 2005; Bhooshan et al. 2014) as a heuristic to satisfy the (6). To simulate the spring behavior, the axial force of each bar is proportional to its elongation (Hooke's law of elasticity). So the (signed) magnitude of the j th member axial force is

$$f_j = k(l_j - \varrho_j) \quad (9)$$

where k is a constant factor characteristic. l_j is the (variable) length of edge j during the form-finding process. The constant ϱ_j denotes the initial length of edge j , it is convenient in controlling the final shapes' 'curvature.' (Fig. 5).

The particle-spring model becomes sophisticated when dealing with motion and acceleration (Bhooshan et al. 2014), according to Newton's laws of motion. However, if one is only concerned with the final equilibrium of forces rather than the exact trajectories of the nodal motions, the process can be simplified to a non-linear Richardson iteration:

repeat

 compute \mathbf{u} , l , F , \mathbf{p} , and $\Delta\mathbf{x} = H(CF\hat{\mathbf{u}} - \mathbf{p})$

$\mathbf{x} \leftarrow \mathbf{x} + \mu\Delta\mathbf{x}$.

until $\|\Delta\mathbf{x}\|_F < \varepsilon$.

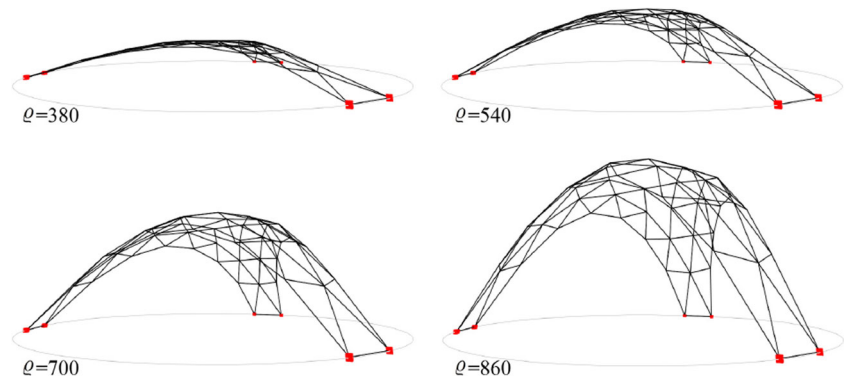
μ is a constant learning rate.

ε is a significantly small constant.

where the solution \mathbf{x} is updated by a fraction of the residual at every iteration.

The iterative algorithm makes heuristic local searches. The convergence behaviors are illustrated in the chart in Fig. 6. We observe that $\|\Delta\mathbf{x}\|$ is monotonically decreasing in four cases. The algorithm's termination condition $\|\Delta\mathbf{x}\|_F < \varepsilon$ appears as a threshold for approaching the condition (6).

Fig. 5 The results of the spring system method with different settings of initial length ϱ . The initial geometry of the four cases are identical. The fixed points are marked in red



3 Coplanar conditions for surface normals

The resulting mesh from the structural calculation resides on a virtual surface. One can estimate both the surface normals at nodes, and subsequently construct the local frames of bars and the joint details (as in the Upsilon project). However, some fabrication process may prefer that each member (bar) and the two normals at both ends lie in the same plane (Fig. 7, bottom). Such coplanar conditions are associated with the construction of parallel mesh (Pottmann et al. 2007). We aim to adjust the nodal normals to 1) fully or partially satisfy coplanar conditions, and 2) minimize the deviation of the adjusted normals from their original estimations. Following the two goals, this section introduces three different approaches:

1. The coplanar conditions are formulated as constraints and the deviation is presented as the objective function. The constrained optimization problem can be solved by using Lagrange multipliers.

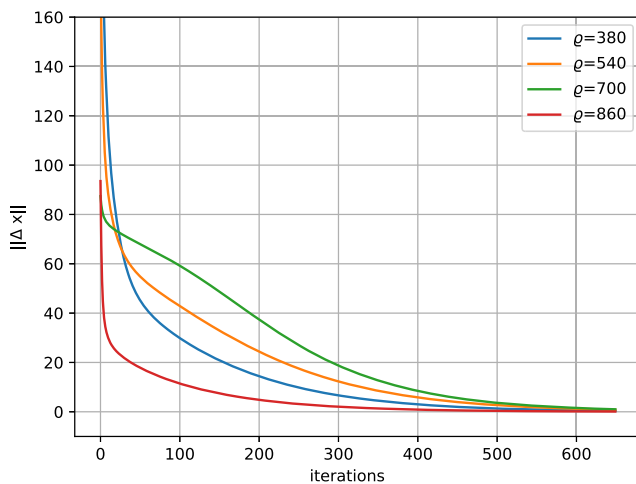


Fig. 6 The convergence behaviors of the heuristic algorithm (Section 2.2) in the four cases shown in Fig. 5. $\|\Delta \mathbf{x}\|$ is plotted against computational time

2. A condition of parallel mesh (stronger constraints than the coplanar conditions) is introduced as a constraint which results in a system of linear equations.
3. The objective function integrates both the coplanar conditions and the deviation so that the two design criteria can be negotiated. A heuristic iterative algorithm is introduced.

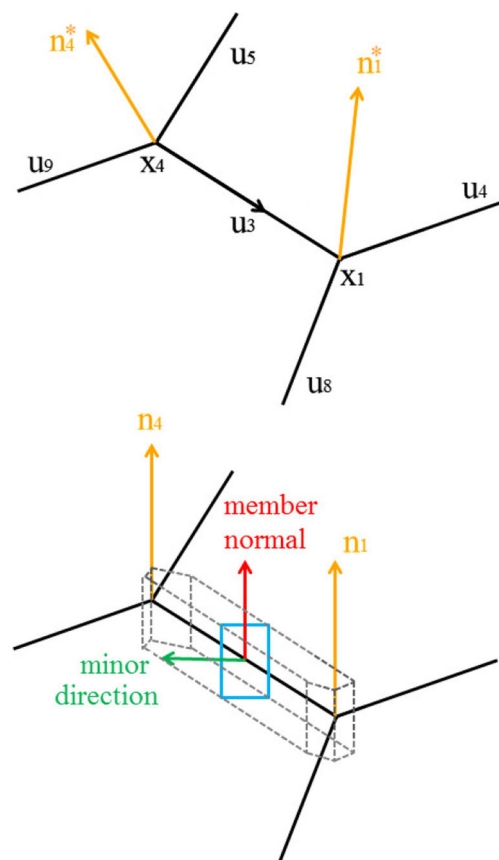


Fig. 7 Top: two nodal normals n_1^* , n_4^* and u_3 are not in one plane. Bottom: After modifying the nodal normals, n_1 , n_4 , and u_3 lie in one plane (meeting coplanar conditions). The member normal lies in this plane and the minor direction is orthogonal to the plane

The Upsilon project did not employ these methods, which provide more options in designing the geometry details for prefabrication.

First, we introduce two related vector definitions:

$$\begin{cases} \mathbf{y}_{i,k} = \widehat{\mathbf{x}_{D_{i,k}} - \mathbf{x}_i} \\ \mathbf{m}_{i,k} = \mathbf{y}_{i,k} \times \mathbf{y}_{i,(1+k)\%K} \end{cases} \quad \forall 1 \leq i \leq N, 0 \leq k \leq K-1$$

where the hat symbol denotes vector normalization; symbol % denotes integer modulo; D is an $N \times K$ matrix of integers; and the i th row contains the indices of the nodes that link the node i . The order is clockwise (following the right-hand rule, the surface normal as the axis of rotation).

The nodal normals are defined as the average of the normals of the adjacent triangles (defined by the two members attached to the node):

$$\mathbf{n}_i^* = \sum_{k=0}^{K-1} \mathbf{m}_{i,k}, \quad \forall i \in [1, N] \quad (10)$$

3.1 Direct formulation of coplanar conditions

The desirable normals \mathbf{n}_i ($1 \leq i \leq N$) are the variables. We wish that \mathbf{n}_i is aligned with its initial estimation \mathbf{n}_i^* , so the objective is to maximize

$$\sum_{i=1}^N \mathbf{n}_i \cdot \mathbf{n}_i^*$$

under constraints

$$\|\mathbf{n}_i\|^2 = 1, \quad \forall i \in [1, N] \quad (11)$$

For member j , the coplanar condition can be formulated as

$$\det[\mathbf{n}_{W_{j,1}} \quad \hat{\mathbf{u}}_j \quad \mathbf{n}_{W_{j,2}}] = 0, \quad \forall j \quad (12)$$

where W is an $M \times 2$ matrix of integers, the j th row contains the indices of the two ends of edge j . The three vectors $\mathbf{n}_{W_{j,1}}$, $\hat{\mathbf{u}}_j$, $\mathbf{n}_{W_{j,2}}$ in (12) are illustrated in Fig. 8 (top).

One can use Lagrange multipliers to obtain the solution as the partial derivatives of the objective and the constraints in (11,12) can be written in a closed form. A practical choice is using the first-order Lagrangian method (Bertsekas 2016).

3.2 Linear model with minimum norm solution

This approach formulates a stronger condition than coplanar conditions in Section 3.1 and, consequently, leads to a system of linear equations whose solution can be obtained by linear solver. This model defines the modifications \mathbf{n}_i of the nodal normals as variables, i.e.,

$$\mathbf{n}_i = \mathbf{n}_i^* + \mathbf{n}_i', \quad \forall i$$

The modifications should be minimized. The coplanar conditions in (12) leads to quadratic equations. To construct

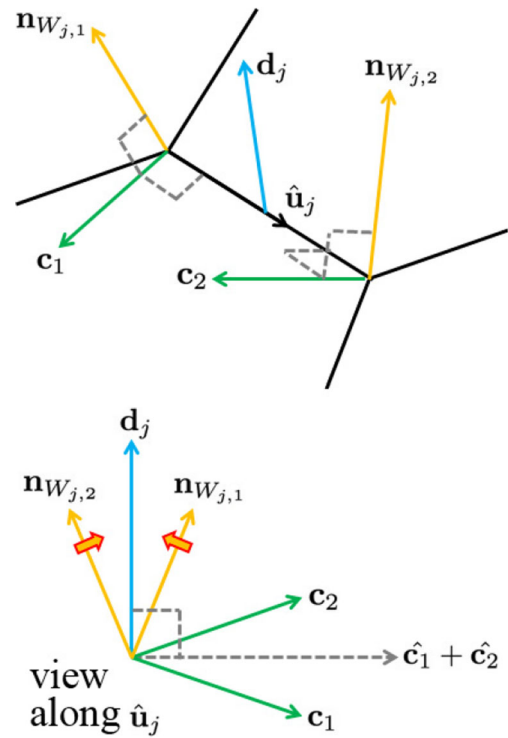


Fig. 8 Member-wise rotation of nodal normals. Top: edge j is associated with two nodal normals $\mathbf{n}_{W_{j,1}}$ and $\mathbf{n}_{W_{j,2}}$. Bottom: the view along the member direction $\hat{\mathbf{u}}_j$. The two nodal normals are rotated towards \mathbf{d}_j to meet coplanar condition

a system of linear equations, we employ the parallel conditions

$$\hat{\mathbf{u}}_j \times (\mathbf{n}_{W_{j,2}}^* + \mathbf{n}_{W_{j,2}}' - \mathbf{n}_{W_{j,1}}^* - \mathbf{n}_{W_{j,1}}') = 0, \quad \forall j \quad (13)$$

which are linear with respect to \mathbf{n}_i' . The parallel conditions (Pottmann et al. 2007) are stronger than coplanar conditions. Whenever (13) holds, so holds (12), but not vice versa.

The goal is to reduce all the modifications. One can implicitly implement this objective with the minimum norm solution of an underdetermined system. Let the $3N$ -dimensional vector \mathbf{z} contain all the elements in matrix \mathbf{n}' :

$$\mathbf{n}_i' = \begin{bmatrix} \mathbf{z}_{(-2+3i)} \\ \mathbf{z}_{(-1+3i)} \\ \mathbf{z}_{(3i)} \end{bmatrix}$$

The objective is to minimize $\|\mathbf{n}'\|_F$, or

$$\|\mathbf{z}\| \quad (14)$$

Borrowing the hat-map notation from the Lie algebra for $\mathbf{SO}(3)$ group, we define

$$\tilde{\mathbf{u}}_j \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -\hat{\mathbf{u}}_{j3} & \hat{\mathbf{u}}_{j2} \\ \hat{\mathbf{u}}_{j3} & 0 & -\hat{\mathbf{u}}_{j1} \\ -\hat{\mathbf{u}}_{j2} & \hat{\mathbf{u}}_{j1} & 0 \end{bmatrix}$$

The linear constraints in (13) are equivalent to

$$\begin{bmatrix} \mathbf{z}(-2+3W_{j,1}) \\ \mathbf{z}(-1+3W_{j,1}) \\ \mathbf{z}(3W_{j,1}) \\ \mathbf{z}(-2+3W_{j,2}) \\ \mathbf{z}(-1+3W_{j,2}) \\ \mathbf{z}(3W_{j,2}) \end{bmatrix} = \hat{\mathbf{u}}_j \times (\mathbf{n}_{W_{j,2}}^* - \mathbf{n}_{W_{j,1}}^*) \quad (15)$$

where \otimes denotes the tensor product. These edgewise constraints can be assembled into a global system of linear equations $A\mathbf{z} = \mathbf{b}$, where A is a $3M \times 3N$ matrix given by

$$\left[-C^T \otimes \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right] \circ \left[\underbrace{[1, \dots, 1]}_N \otimes \begin{bmatrix} \tilde{\mathbf{u}}_1 \\ \tilde{\mathbf{u}}_2 \\ \vdots \\ \tilde{\mathbf{u}}_M \end{bmatrix} \right]$$

where \circ denotes the Hadamard product.

A Voronoi mesh typically has more edges than nodes, i.e., there are more rows than columns in A . However, SVD gives the number of non-zero singular values, which indicates that the system is undetermined. Thus, the minimum norm solution of the system can be obtained from SVD's orthonormal bases corresponding to the non-zero singular values. As a result, the norm (14) is minimized and the coplanar conditions are satisfied.

3.3 Heuristic algorithm via Rodrigues' rotation

This method treats nodal normals \mathbf{n} as variables and formulates the coplanar conditions as an unconstrained optimization problem. We have two conflicting goals: (1) imposing coplanar conditions and (2) keeping the angle between each evolving nodal normal and its initial estimation \mathbf{n}_i^* below the threshold ϕ_{max} . Thus, we minimize the following objective function

$$\sum_{j=1}^M |\det[\mathbf{n}_{W_{j,1}}, \hat{\mathbf{u}}_j, \mathbf{n}_{W_{j,2}}]| + \eta \sum_{i \in S} \max(\phi_i - \phi_{max}, 0) \quad (16)$$

where

$$\cos \phi_i = \mathbf{n}_i \cdot \mathbf{n}_i^*$$

η is a constant weight.

Member-wise rotation is a simple method to make the member direction and the two nodal normals in one plane. It involves the Rodrigues' rotation

$$r(b, \theta, v) \stackrel{\text{def}}{=} v \cos \theta + (b \times v) \sin \theta + b(b \cdot v)(1 - \cos \theta) \quad (17)$$

where b is the axis of rotation about which v rotates by an angle θ .

For each member j , one can compute

$$\begin{aligned} \mathbf{c}_1 &= \hat{\mathbf{u}}_j \times \mathbf{n}_{W_{j,1}} \\ \mathbf{c}_2 &= \hat{\mathbf{u}}_j \times \mathbf{n}_{W_{j,2}} \\ \cos \theta_j &= \hat{\mathbf{c}}_1 \cdot \hat{\mathbf{c}}_2 \\ \mathbf{d}_j &= \hat{\mathbf{u}}_j \times (\hat{\mathbf{c}}_1 + \hat{\mathbf{c}}_2) \\ s_j &= \begin{cases} 1, & \text{if } \hat{\mathbf{c}}_1 \cdot \mathbf{d}_j < 0 \\ -1, & \text{otherwise} \end{cases} \end{aligned}$$

for rotating the nodal normals around member j (Fig. 8) to meet coplanar conditions. θ_j denotes the angle between $\hat{\mathbf{c}}_1$ and $\hat{\mathbf{c}}_2$.

We developed a heuristic iterative algorithm to minimize the objective (16). Each iteration consists of (1) member-wise rotation of nodal normals and (2) the nodal normals' alignment. One can terminate the iterations when the rotations are significantly small. Each iteration is as follows:

```

Initialize  $\Delta_i = 0, i \in S$ 
FOR member  $j \in [1, M]$ 
     $\Delta_{W_{j,1}} += (1 - \alpha)\mathbf{n}_{W_{j,1}} + \alpha r(\hat{\mathbf{u}}_j, -s_j/2, \mathbf{n}_{W_{j,1}})$ 
     $\Delta_{W_{j,2}} += (1 - \alpha)\mathbf{n}_{W_{j,2}} + \alpha r(\hat{\mathbf{u}}_j, s_j/2, \mathbf{n}_{W_{j,2}})$ 
ENDFOR
FOR node  $i \in S$ 
     $\mathbf{n}_i += \Delta_i$ 
     $\cos \phi_i = \mathbf{n}_i \cdot \mathbf{n}_i^*$ 
    IF  $\phi_i > \phi_{max}$ 
         $\mathbf{n}_i \leftarrow r(\mathbf{n}_i \times \mathbf{n}_i^*, \beta(\phi_i - \phi_{max}), \mathbf{n}_i)$ 
    ENDFOR
    
```

where α, β denote constant learning rates. The typical value of α is 0.1-0.15. When $\beta = 1$, the angle between a nodal normal \mathbf{n}_i and the original estimation \mathbf{n}_i^* is always below the threshold ϕ_{max} . When $0 < \beta < 1$, the algorithm balances between the coplanar conditions and the nodal normals' alignment.

The three charts in Fig. 9 show the convergence behaviors of the algorithm with different settings of the threshold ϕ_{max} . It indicates that the objective's first term (responsible for coplanar conditions) is always smoothly decreasing, while there are three stages for the behaviors of the second term (responsible for the deviation): (1) the value of the term is zero in the beginning, (2) the value increases erratically in the following stage, (3) the value becomes constant in the final stage. Eventually, the ratio between the value of the first term and the second becomes constant.

3.4 Finalize the local frames of bars

The timber prefabrication will be organized in the local frames of the bars. A comprehensive discussion on frame

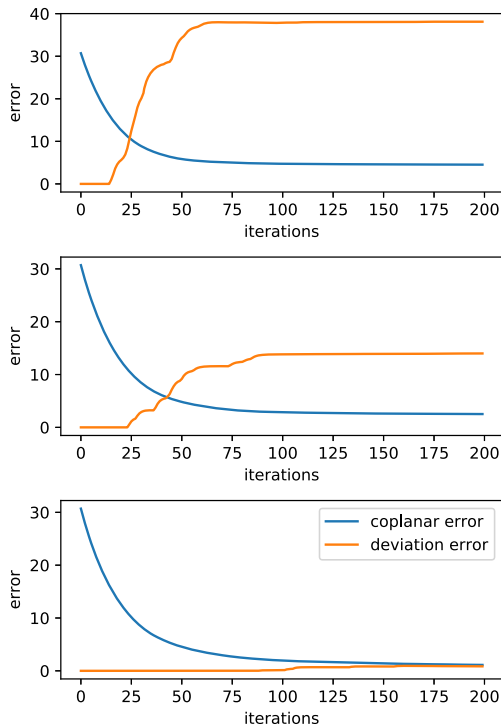


Fig. 9 The convergence behaviors of the algorithm in Section 3.3 with three different configurations (Top: $\phi_{max} = \pi/12$, middle: $\phi_{max} = \pi/9$, bottom: $\phi_{max} = \pi/6$). Both the coplanar error and the deviation, the two terms in the objective (16), are plotted against computational time. The deviation is scaled 100 times along the vertical axis for illustrative purpose

representations can be found in Craig (2005). A 3×3 unitary matrix describes the frame of the j th bar:

$$\Psi = \begin{bmatrix} e_{1,1} & e_{2,1} & e_{3,1} \\ e_{1,2} & e_{2,2} & e_{3,2} \\ e_{1,3} & e_{2,3} & e_{3,3} \end{bmatrix}$$

where $e_1 = \hat{\mathbf{u}}_j$, $e_2 = p(e_1, \mathbf{n}_{W_{j,1}} + \mathbf{n}_{W_{j,2}})$, and $e_3 = e_1 \times e_2$ stand for the three axes of the local frame. The function $p()$

creates a new vector orthogonal to u (normalized) from the resource vector v :

$$p(u, v) \stackrel{\text{def}}{=} \frac{v - (v \cdot u)u}{\|v - (v \cdot u)u\|} \quad (18)$$

The frame's the origin is at

$$O_j = \frac{\mathbf{x}_{W_{j,1}} + \mathbf{x}_{W_{j,2}}}{2}$$

Vectors e_2 , $\mathbf{n}_{W_{j,1}}$, and $\mathbf{n}_{W_{j,2}}$ lie on one plane if the coplanar conditions are met, otherwise they are generally not on one plane.

4 Joint geometry and robotic fabrication

Turning a shape-optimized mesh into a timber shell is a complex materialization process with many options on materials, joint details, and fabrication. Our bespoke joint is designed to facilitate the planning of manufacturing procedures. Computer numerical control (CNC) manufacturing in practice heavily relies on computer-aided design (CAD) and computer-aided manufacturing (CAM) software (Sarcar et al. 2008; Chang and Wysk 1997). A typical workflow is “3D solid model (CAD) - toolpath from model discretization (CAM) - fabrication”. The map from the 3D model to the toolpath is essentially a black box and often follows *ad hoc* assumptions. Here, we take another approach: representing the toolpath as a closed-form function of the resultant mesh of the form-finding process. Subsequently, the machine codes for fabrication can be directly derived from the closed-form functions.

The details of joint geometry (Figs. 2 and 10) include (1) two flat slopes, each slope touching an adjacent piece of timber; (2) a straight flat cut, which the metal plate will be inserted into; and (3) a hole, which the bolt will go through.

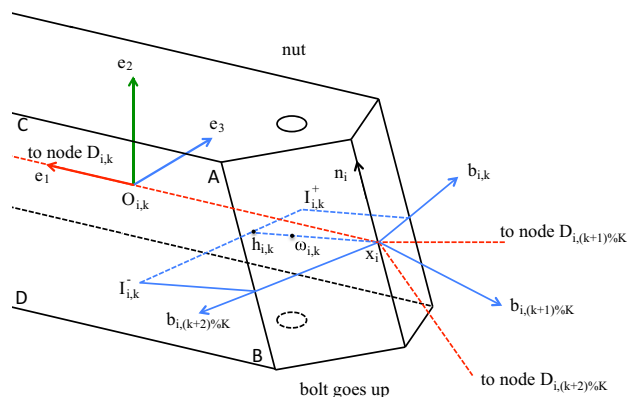


Fig. 10 Joint geometry. The metal plate is not orthogonal to the bar's cross section

The following characteristics guarantee the three connected pieces of timber will fit together:

1. The common line of the two slopes of one piece of timber coincides that of the other two adjacent pieces of timber. The common line is along the surface normal \mathbf{n}_i at node i .
2. The nodal normal \mathbf{n}_i is perpendicular to the plane that the flat cut (for accommodating the metal plate) is made on.
3. The hole for bolt and nut is made parallel with the bar's cross section (along e_2). It leads to a slight bias from the right angle between the metal plate and the bolt.

If the nodal normals meet the coplanar conditions (Section 3), \mathbf{n}_i lies in the e_1 - e_2 plane (Fig. 10 right). Otherwise, it is not the case. The coplanar conditions were not imposed in our projects as the robotic milling device is flexible enough to handle the irregular geometry. So the joint geometry and the toolpath are generally formulated where the coplanar condition is just a special case.

4.1 Joint geometry

At node i , two touching pieces of timber are supposed to share an in-between vector $\mathbf{b}_{i,k}$ (Fig. 10):

$$\begin{cases} \mathbf{b}_{i,k} = p(\mathbf{n}_i, r) \\ r = r(\mathbf{m}_{i,k}, \cos^{-1}(\mathbf{y}_{i,k} \cdot \mathbf{y}_{i,(k+1)\%K}), \mathbf{y}_{i,k}) \end{cases}$$

where $r()$ denotes the Roderigues' rotation (17). The other two in-between vectors are defined likewise.

Points $\omega_{i,k}$ and $\mathbf{h}_{i,k}$ are on the plane of the metal plate. The bolt goes through $\omega_{i,k}$ along the direction of e_2

$$\begin{cases} \omega_{i,k} = \delta_\omega p(\mathbf{n}_i, \mathbf{y}_{i,k}) + \mathbf{x}_i \\ \mathbf{h}_{i,k} = \delta_h p(\mathbf{n}_i, \mathbf{y}_{i,k}) + \mathbf{x}_i \end{cases}$$

where δ_ω, δ_h denote two predefined constants. Function $p()$ is defined in (18).

The metal plate intersects with the two surfaces of the timber (Fig. 10 right) at point $I_{i,k}^-$ and $I_{i,k}^+$. Let wid denote the width of timber (long the minor direction of the cross section, Fig. 7). The point $I_{i,k}^-$ is given by intersecting the plane ABCD ($O_{i,k} - \frac{\text{wid}}{2}e_3$ is a point on the plane; vector e_3 is the normal of the plane) and the line ($\mathbf{h}_{i,k}$ is a point on the line; vector $\mathbf{y}_{i,k} \times \mathbf{n}_i$ is the direction of the line), i.e.

$$I_{i,k}^- = q(O_{i,k} - \frac{\text{wid}}{2}e_3, e_3, \mathbf{h}_{i,k}, \mathbf{y}_{i,k} \times \mathbf{n}_i)$$

where $q()$ is the plane-line intersection function which gives an intersection point (if $\hat{n} \cdot v \neq 0$):

$$q(p, \hat{n}, b, v) \stackrel{\text{def}}{=} b - \frac{\hat{n} \cdot b - \hat{n} \cdot p}{\hat{n} \cdot v} v \quad (19)$$

where p and \hat{n} denote the point and the normal of the plane, respectively; b and v denote the point and the direction of the line, respectively.

Likewise, the other intersection point is

$$I_{i,k}^+ = q(O_{i,k} + \frac{\text{wid}}{2}e_3, e_3, \mathbf{h}_{i,k}, \mathbf{y}_{i,k} \times \mathbf{n}_i)$$

4.2 Joint geometry in local frames

Eventually, transforming all data (in the global coordinates system) into the bar's local frame would facilitate the toolpath planning for that individual bar. For example, one can transform the relevant vectors into the bar's local frame:

$$\begin{aligned} \mathbf{n}'_i &= \Psi^T \mathbf{n}_i \\ \mathbf{b}'_{i,k} &= \Psi^T \mathbf{b}_{i,k} \end{aligned}$$

and represent the relevant points in the bar's local frame:

$$\begin{aligned} \mathbf{x}'_i &= O_{i,k} + \Psi^T (\mathbf{x}_i - O_{i,k}) \\ I'_{i,k}^\pm &= O_{i,k} + \Psi^T (I_{i,k}^\pm - O_{i,k}) \\ \omega'_{i,k} &= O_{i,k} + \Psi^T (\omega_{i,k} - O_{i,k}) \end{aligned}$$

The $'$ symbol indicates the local version of a vector (or a point). The three axes are $e'_1 = [1, 0, 0]^T$, $e'_2 = [0, 1, 0]^T$, and $e'_3 = [0, 0, 1]^T$.

4.3 Robot milling

Both a 5-axis CNC machine and a robotic system (Chen and Dong 2013) are adequate for creating the details on the timber's ends. We employed a high payload 6-axis KUKA robot for prefabrication. An automatic tool change (ATC) spindle is mounted to the robot flange. Each piece of timber, as the workpiece, is fixed horizontally (Fig. 11). The video (Hua 2018b) briefs the robotic fabrication process.

Before working, the robot requires the 'BASE' frame calibration and the 'TOOL' frame calibration (Craig 2005). The BASE frame, including the origin, the x , y , and z -axes, is the coordinate system of the workpiece relative to the robot's pedestal (often called the WORLD coordinate). We calibrated the BASE origin to $O_{i,k}$, the three axes of the BASE along the bar's local frame (e'_1, e'_2, e'_3). The TOOL frame is the coordinate system of the tool (the milling bit on the spindle) relative to the robot flange (Craig 2005). We calibrated the tool center point (TCP) to the center of the head of a flat end-mill and the three axes of TOOL along the frame of the robot flange (in our case, the x -axis of TOOL happens to be along the bit (outward)). Then, the coordinates of the tool (milling bit) trajectory can be directly expressed in the BASE frame, as the robot will automatically use the TOOL and BASE frames when computing kinematics.

The geometry of each joint is uniquely parametrized. Thus, the data in the machine instructions for cutting each joint is unique. Conventional off-line programming could

language to feed the calculated data into corresponding places inside the machine instructions. The instructions for all pieces of timber are first created and saved on a computer. The instructions for a particular piece of timber are stored in one source file (.src file). Then the file is copied to the robot, which can execute the instructions to fabricate the corresponding timber (Fig. 11).

5 Results and discussions

The two built prototypes, the mero sphere and the Upsilon pavilions (Fig. 13), indicate multiple options in the shape design of shells (Flowchart in Fig. 3):

1. The simplest approach is using certain ‘a priori’ shape (e.g., the mero sphere, Fig. 1) or a manually modeled free surface, if the joint connections are strong enough to bear irregular bending moments.
2. A convenient option is to use the initialization shape without iterative form finding. It is efficient to use either the LSL or UFD model, as the shape can be obtained by solving a system of linear equations.



Fig. 13 The Upsilon pavilion as a timber Voronoi shell

3. A more reasonable approach is to employ the form-finding method to create a compressive funicular shape in static equilibrium.
4. If the detail design or the fabrication process prefer coplanar conditions on nodal normals, the optimization methods in Section 3 are helpful.

Timber Voronoi shells present a prefabrication system that unifies computational form finding and prefabrication-assembly solutions. The adjustment of the nodal positions in the iterative form-finding process finds the geometry in static equilibrium of member axial forces and external loads. On the other hand, the joint details allow the timber to mainly experience compression and the metal plates (with bolt and nut) to be idle until unexpected loads or deformation occur. Further, the geometry of the timber details and the metal connections are designed for systematic fabrication. Eventually, the toolpath can be expressed in a closed-form function of the shell geometry.

The detailing at the valence-3 node is a key problem in the design of the timber Voronoi shells. Regarding the whole shell as a compressive funicular structure, our main strategy is that the compression should be transmitted between the neighboring timber directly, while the “bolt-metal plate” mechanism provides safety under unexpected loads and deformations. The shell structure should be in static equilibrium of member axial forces and gravitational pull so that the out-of-plane bending moments and in-plane shear forces at the nodes are small in the physical installation. Employing the flat steel plate simplifies the prefabrication processes of timber and sheet metal; however, the angled cut in the timber weakens the timber bar. Thus, the angle between the nodal normal and the bar’s cross section should be limited. In future work, we are going to carry out the numerical and physical load tests of the node details.

Traditional timber structures opt for a small category of timber components. Every instance of the same category has the exact same geometry. This is widely-recognized wisdom for manual or manual-machine combined working processes. By contrast, all timber components have different shapes in our Voronoi shells, though they belong to one category (one class in the programming language). This could be a nightmare for conventional manufacturing systems, just considering the off-line programming for each irregular component respectively. Thus, exploiting symmetry and reducing the number of types of components became an important strategy. However, the direct translation from the closed-form expression of the toolpath to the executable robot instructions is suited for the individually customized pieces. We also developed an open-source library (Hua 2018a) to output robot instructions (KUKA .src files) from the Java language.

Our current fabrication plan still demands much manual labor in assembly. Future work concerning further automation includes (1) automating the processes of loading and unloading timber in the machine. An option is to let the robot arm grab, move (when fabricating), and unload the timber while the cutters are fixed; and (2) automating the assembly processes, e.g., with multiple robotic arms attached to gantry systems (Thoma et al. 2018). However, auto de-assembly of timber structures also poses a great challenge but there are hardly any experiments on this topic.

The efficiency of the computation is not our major concern as the problem's scale is relatively small. We ran our Java implementation of the algorithms on a MacBook with a 2.4GHz i5 processor. For the Upsilon pavilion (120 nodes and 145 members), the shape initialization, form finding (600 iterations), and coplanar optimization (linear model) take less than one second.

6 Conclusions

Fabrication-oriented geometry design has gained great interest from both the academy and practice. Based on the advanced computational methods and fabrication technology, people now try to unlock the relation between form, structure and material, which follow an established set of preconceived rules (Menges 2016; Willmann et al. 2018). For example, one will probably avoid connecting three pieces of timber at one point (because it seems tedious to describe and materialize the geometry) and prefer triangles to polygons in a timber-framed shell (because triangles are stable). However, our method makes the timber Voronoi shell feasible by minimizing undesirable forces and defining the fabrication process with closed-form expressions. The interplay between formal logic and materialization processes could be an active and fruitful participant in shell design.

Recent years have seen two trends in shell structures design. On the one hand, user-friendly software (or plug-ins) make shell design accessible to non-experts. For example, Kangaroo and the RhinoVAULT plug-in for Rhinoceros are easy-to-use form-finding tools for architects. People can also find convenient software that translates given geometry into robot toolpath. On the other hand, the cutting-edge development of shell design (e.g., the Armadillo vault Block et al. 2018) demands deep expertise on structural analysis, spatial design, material science, and CNC fabrication. Without intensive cooperation between different disciplines, very little progress can be made. In this work, we propose a well-defined link from the form finding to the toolpath for fabrication and attempt to establish an explicit, consistent formalism from form finding to CNC manufacturing. As a result, researchers from different fields

can work together and new user-friendly software may link experts and new users.

Acknowledgements The prefabrication and assembly of the structures were organized by Prof. Biao Li's Institute of Architectural Algorithms and Applications, Southeast University, Nanjing. The team members of the Upsilon project include H. Hua, J.C. Wang, B. Li, P. Tang, Y.Y. Chen, C.Y. Cai, J.N. Xu, J.Z. Chen, S.Y. Li, H.J. Li, N.L. Liu, H. Li, J.R. Zeng, H.C. Guo, X. Wang, Y.L. Chen, H.D. Wu, L.X. Wei, and J.S. Zhang.

We thank the reviewers and the editor for their constructive remarks towards improving the manuscript. We also thank Prof. Lei He from School of Civil Engineering at Southeast University for his advice on numerical optimization.

This work was financially supported by the National Natural Science Foundation of China [51778118] and the Ministry of Housing and Urban-Rural Development of China [UDC2017020212].

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results The Java codes at <https://github.com/whitegreen/Voronoi-Timber-Shell> can fully reproduce the results.

References

- Barnes MR (1999) Form finding and analysis of tension structures by dynamic relaxation. *Int J Space Struct* 14(2):89–104
- Bedi S, Mann S, Menzel C (2003) Flank milling with flat end milling cutters. *Comput Aided Des* 35(3):293–300
- Bertsekas D (2016) Nonlinear programming, 3rd. Athena Scientific
- Bhooshan S, Veenendaal D, Block P (2014) Particle-spring systems: design of a cantilevering concrete canopy. In: *Shell structures for architecture: form finding and optimization*. Routledge, pp 103–113
- Block PPCV (2009) Thrust network analysis: exploring three-dimensional equilibrium. PhD thesis, Massachusetts Institute of Technology
- Block P, Van Mele T, Liew A, DeJong M, Escobedo D, Ochsendorf J (2018) Structural design, fabrication and construction of the Armadillo vault. *Struct Eng* 96(5):10–20
- Chang TC, Wysk RA (1997) Computer-aided manufacturing. Prentice Hall PTR
- Chen Y, Dong F (2013) Robot machining: recent development and future research issues. *Int J Adv Manuf Technol* 66(9–12):1489–1497
- Craig JJ (2005) Introduction to robotics: mechanics and control, vol 3. Pearson/Prentice, Hall Upper Saddle River
- Dimčić M (2011) Structural optimization of grid shells based on genetic algorithms. PhD thesis, Universität Stuttgart
- D'Amico B, Kermani A, Zhang H (2014) Form finding and structural analysis of actively bent timber grid shells. *Eng Struct* 81:195–207
- Eversmann P, Gramazio F, Kohler M (2017) Robotic prefabrication of timber structures: towards automated large-scale spatial assembly. *Construc Robot* 1(1–4):49–60
- Gramazio F, Kohler M, Willmann J (2014) The robotic touch: how robots change architecture: Gramazio & Kohler Research ETH Zurich 2005–2013. Park Books
- Hua H (2018a) Javakuka open source library <http://javakuka.org>. accessed 12 09 2018

- Hua H (2018b) Upsilon pavilion <https://vimeo.com/287979548>, Accessed 18 11 2019
- Kilian A, Ochsendorf J (2005) Particle-spring systems for structural form finding. *J Int Assoc Shell Spatial Struct* 46(2):77–84
- Larue A, Altintas Y (2005) Simulation of flank milling processes. *Int J Mach Tools Manuf* 45(4-5):549–559
- Liddell I (2015) Frei Otto and the development of gridshells. *Case Studies Struc Eng* 4:39–49
- Linkwitz K, Veenendaal D (2014) Nonlinear force density method. *Shell Struc Archit–Form Finding Optim*, 143–155
- Menges A (2016) Computational material culture. *Archit Des* 86(2):76–83
- Naicu D, Harris R, Williams C (2014) Timber gridshells: design methods and their application to a temporary pavilion. In: *Proceedings of World conference on timber engineering (WCTE), WCTE 2014; Conference date: 10-08-2014 Through 14-08-2014*
- Paoli CCA (2007) Past and future of grid shell structures. PhD thesis, Massachusetts Institute of Technology
- Pietroni N, Tonelli D, Puppo E, Froli M, Scopigno R, Cignoni P (2015) Statics aware grid shells, vol 34, Wiley Online Library
- Pottmann H, Liu Y, Wallner J, Bobenko A, Wang W (2007) Geometry of multi-layer freeform structures for architecture. *ACM Trans Graph (TOG)* 26(3):65
- Sarcar M, Rao KM, Narayan KL (2008) Computer aided design and manufacturing. PHI Learning Pvt. Ltd
- Schek HJ (1974) The force density method for form finding and computation of general networks. *Comput Methods Appl Mech Eng* 3(1):115–134
- Tang C, Sun X, Gomes A, Wallner J, Pottmann H (2015) Form-finding with polyhedral meshes made simple. In: *ACM SIGGRAPH 2015 posters SIGGRAPH '15*. ACM, New York, pp 5:1–5:1, <https://doi.org/10.1145/2787626.2787631>. <http://doi.acm.org/10.1145/2787626.2787631>
- Thoma A, Adel A, Helmreich M, Wehrle T, Gramazio F, Kohler M (2018) Robotic fabrication of bespoke timber frame modules. In: *Robotic fabrication in architecture, art and design*. Springer, pp 447–458
- Van Mele T, Panozzo D, Sorkine-Hornung O, Block P (2014) Best-fit thrust network analysis. *Shell Structures for Architecture-Form Finding and Optimization*, 157–170
- Veenendaal D, Block P (2012) An overview and comparison of structural form finding methods for general networks. *Int J Solids Struct* 49(26):3741–3753
- Willmann J, Knauss M, Bonwetsch T, Apolinarska AA, Gramazio F, Kohler M (2016) Robotic timber construction-expanding additive fabrication to new dimensions. *Autom Construct* 61:16–23
- Willmann J, Block P, Hutter M, Byrne K, Schork T (2018) Robotic fabrication in architecture, art and design. Springer

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.