A METHOD FOR LOAD-RESPONSIVE INHOMOGENEITY AND ANISOTROPY IN 3D LATTICE GENERATION BASED ON ELLIPSOID PACKING

YAO LU[1], EDA BEGUM BIROL[2], COLBY JOHNSON[3],
CHRISTOPHER HERNANDEZ[4] and JENNY SABIN[5]
[1,2,3,4,5]*Cornell University*
[1,2,3,4,5]*{yl3229|ebb75|cmj73|cjh275|jes557}@cornell.edu*

Abstract. 3D lattice structures are gaining widespread application in multiple design fields. While the number of projects that utilize load-responsive inhomogeneous and anisotropic 3D lattices in design applications increase, accessible and effective algorithmic generation methodologies remain lacking. This paper addresses this gap by introducing a novel computational method for controlled load-responsive inhomogeneity and anisotropy in 3D lattice generation. The presented methods employ a responsive Ellipsoid Packing algorithm informed by the global tensor field of the packing geometry, followed by a Kissing Ellipsoids algorithm to generate the lattice. Load specific anisotropy and inhomogeneity in the ellipsoid packing process is achieved in response to the magnitude and directionality values of the global tensor field and specialized responsive lattices are easily generated. The proposed Ellipsoid Packing workflow is compared to various common lattice generation algorithms. Results show improvement in mechanical performance.

Keywords. 3D lattice; ellipsoid packing; bio-inspired; algorithmic design; ceramic brick.

1. Introduction

With increasing ease of fabrication through Additive Manufacturing Technologies (AMT), 3D lattice structures are gaining widespread application in architectural and design fields, as well as in fields such as biomedical and aerospace engineering. 3-Dimensional lattice structures offer advantages in these fields due to their ability to achieve high mechanical performance while maintaining a light weight (Niu et.al.2018).

Despite this increased ease of fabrication, "(...) there is relatively limited information available in the literature about designing large scale lattice structures (...) (Kantareddy, 2016)". This gap is especially apparent as it pertains to random, inhomogeneous lattice structures and the pertaining generation and evaluation strategies. This paper aims to address this gap by exploring the advantages of inhomogeneity and anisotropy in 3 Dimensional lattice structures within architectural load-bearing applications. Inhomogeneity and anisotropy are

RE: Anthropocene, Proceedings of the 25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2020, Volume 1, 395-404. © 2020 and published by the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

favored in natural load-bearing structures, such as the bone's trabecular lattice, and serve to achieve light weight and high mechanical performance. Within the context of bone's trabecular lattice, responsive inhomogeneity occurs as a result of thickening and thinning of trabecular struts in relation to high and low stress stimuli respectively. Anisotropy occurs through the adaptation of trabecular strut orientations parallel to the direction of pertinent compressive forces.

Establishing the anisotropic and inhomogeneous trabecular lattice structure as a natural precedent, the primary focus of this paper is to present detailed workflows of bio-inspired and responsive algorithmic lattice generation, whose manifestations can be realized at various scales through the employment of emerging AMT.

2. Background

Within the architectural field, various AMT including SLM (Selective Laser Melting), UV curing ceramic 3D printing, ceramic powder printing, and robotic extrusion have shown promise in realizing 3-dimensional lattice typologies and other load-bearing porous modules. In their paper Responsive Spatial Print, for example, Im and AlOthman present the potentials of real-time calibrated robotic extrusion as a means of fabricating "self-supporting spatial lattices" (Im, AlOthman, and Castillo, 2018). The PolyBrick series demonstrate the promise of ceramic powder printing (Sabin et.al 2014) and UV cured ceramic resin printing (Birol, Lu, Sekkin et. al. 2019) in the fabrication of load-bearing lattices. The realization of such projects consists of multiple phases including digital lattice generation, digital lattice thickening/meshing, and fabrication through the respective AMT. While the fabrication phase is making rapid progress in ease and accessibility due to recent developments in additive manufacturing technologies, preceding phases of digital lattice generation lack contained and accessible workflows for designer use. The outlined lattice generation algorithms aim to bridge this gap.

In introducing our lattice generation frameworks and possibilities, algorithmic generation processes, mechanical evaluations, and respective applications of various commonly used lattice typologies are outlined and compared to the proposed novel PolyBrick workflows and lattices. Hence, the paper not only compares the mechanical performance of generated lattice, but also tackles concerns about algorithmic efficiency and usability. Generated lattices are compared to commonly used periodic cellular lattices and various inhomogeneous anisotropic lattices from natural precedents. While uniform periodic lattice structures are advantageous in achieving high stiffness to mass ratio, it is difficult to introduce geometric specialization for heightened efficiency, allowing "unfavorable load cases like bending or shearing" (Reinhart and Teufelhart 2013) to occur. Inhomogeneous anisotropic lattice geometries break this constraint, differing from periodic cellular lattices in their specialized morphology. The bone's trabecular lattice is an important example of this, serving as a precedent for our outlined algorithms. As presented by several projects exploring applications of lattice structures in the architectural field (Reinhart and Teufelhart 2013) (Felder et al. 2016), the phenomenon of responsive morphology has immense potential in rethinking and expanding the capacities of load-bearing lattice structures. While there are projects integrating load responsiveness and responsive directionality within design applications; accessible, effective and user-friendly algorithmic streamlining of this type of lattice generation remains lacking.

This paper introduces a novel computational method for controlled load-responsive inhomogeneity and anisotropy in 3D lattice generation. The primary focus of the proposed framework is presenting streamlined algorithmic processes of informed and responsive lattice generation. We particularly contextualize the advantages of inhomogeneity and anisotropy in 3-Dimensional lattice structures in architectural load-bearing applications. The introduced algorithm allows for a user-friendly process of inhomogeneous anisotropic load-bearing lattice generation facilitated through a custom Grasshopper plug-in. The algorithmic workflows presented expand upon PolyBrick 2.0's (Birol, Lu, Sekkin et. al. 2019) algorithmic processes and present a detailed account thereof.

3. Method

3.1. LATTICE GENERATION ALGORITHM

3.1.1. Ellipsoids as an intermediate parameter to control lattice morphology

In our proposed processes for lattice generation, the trabecular bone is determined as a load responsive precedent of a 3D lattice geometry. The trabecular lattice is quantified by the morphological parameters of trabecular number, separation, length, and orientation (Bagi, Berryman, Moalli, 2011). In our workflow, these morphological quantifiers are algorithmically interpreted as controllable load response parameters. For this interpretation, ellipsoids are introduced as media. A number of ellipsoids are packed densely within a design geometry. Through a secondary algorithm, "Kissing Ellipsoids", the centroids of the touching ellipsoids are connected with a line segment. The morphological parameters of trabecular number, separation, and orientation are thus controlled by ellipsoid size, distribution, and orientation respectively, allowing the algorithm to interpret these major morphological features. The response condition is then introduced as a tensor field to inform ellipsoid packing parameters listed.

3.1.2. Tensor field as a user input to inform Ellipsoid Packing

The tensor field is the key input that establishes load-responsive inhomogeneity and anisotropy in the process of lattice generation. The tensor field is defined by a field of three orthogonal tensors at each point within the analyzed global design geometry. These points correspond to the "nodes" of the voxelized design geometry (Figure 2, left). For each ellipsoid packed in a design geometry with a corresponding tensor field, the three tensors interpolated at the ellipsoid centroid serve as a response condition in determining the length and orientation of semi-major, semi-median, and semi-minor axes (Figure 1).

The tensor field input can be generated in multiple ways. For processes outlined in this paper, ANSYS is used to apply Finite Element Analysis (FEA) to a user-determined design geometry under a set loading condition (Figure 2, right).

For each node, maximum, middle, and minimum principal stress directions are calculated as the three orthogonal tensors. In the responsive packing, areas of larger stress are packed with smaller ellipsoids. Ellipsoids' main axes are oriented to align with maximum principal stress direction.

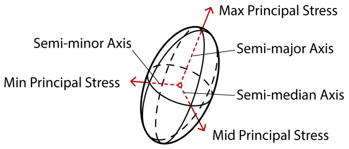

Figure 1. Semi-major, semi-median, semi-minor axis length aligned to and determined by tensor(stress) magnitudes.

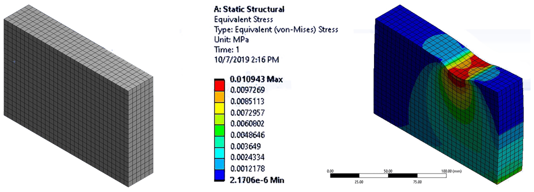

Figure 2. Left: Voxelized design geometry; Right: Finite Element Analysis by ANSYS.

3.1.3. Generation workflow

The Ellipsoid Packing and Kissing Ellipsoids algorithms are implemented using C#. The Ellipsoid Packing algorithm provides multiple adjustable inputs that serve as response factors in determining the lattice geometry. These inputs include boundary geometry, tensor field, maximum ellipsoid principal semi-axis length, minimum ellipsoid principal semi-axis length, pre-existing ellipsoids, and maximum iteration.

The lattice generation workflow starts with producing a tensor field for the boundary geometry with the assistance of ANSYS. The boundary geometry is first analyzed as a solid in ANSYS using a user-defined loading condition. Then, a CSV file containing the X, Y, Z coordinates of each node as well as key parameters evaluated at each node can be exported. As a convention of this workflow, the CSV file is exported in a format that contains maximum principal stress, middle principal stress, minimum principal stress, and Euler angles indicating the necessary angle of rotation to transform from the global coordinate system to each node's principal stress axis. Finally, the CSV file is translated to a tensor field.

Ellipsoid packing is initiated following the outlined process of tensor field generation. This consists of packing the design geometry with the associated tensor field with as many ellipsoids as possible. Simultaneously, the ellipsoids are scaled and rotated in accordance with the tensor field input.

The dense packing is realized by iteratively adding new ellipsoids and relaxing the existing ellipsoids. A user-defined input of an initial number of ellipsoids is needed to begin the packing into the design geometry. To continue ellipsoid packing, collisions are checked between pairs of ellipsoids. This process is particularly complex and computationally expensive when applied to ellipsoid geometries because the ellipsoid collision point doesn't necessarily correspond to the line that connects the centroids as it does in spherical colliding. To simplify this process, the orientation difference of two colliding ellipsoids is assumed to be small, hence the collision point is still assumed to be in close proximity to the connection line. For each pair of ellipsoids, the summation of the two rim distances along the center connection line is calculated (Formula 1-3) and then compared to the centroid distance. When calculating the rim distance, a rotation transformation is applied to align the ellipsoid axes to the global 3D Cartesian coordinate system so that standard ellipsoid function can be used to simplify the calculation (Figure 3). If the sum of rim distances is greater than the centroid distance, they are determined to be colliding, otherwise not. If no collision is detected, indicating all the ellipsoids are fully relaxed within the boundary geometry, one more ellipsoid is added into the boundary geometry. If collisions are detected, a relaxation process is initiated and the colliding ellipsoids are moved away from each other by half of the overlap distance.

$$v_1 = \text{Normalize}(R_1(C_2 - C_1)) = (x_1, y_1, z_1) \tag{1}$$

$$v_2 = \text{Normalize}(R_2(C_2 - C_1)) = (x_2, y_2, z_2) \tag{2}$$

$$d_{rim_i} = \frac{1}{\sqrt{\left(\left(\frac{x_i}{a_i}\right)^2 + \left(\frac{y_i}{b_i}\right)^2 + \left(\frac{z_i}{c_i}\right)^2\right)}} \tag{3}$$

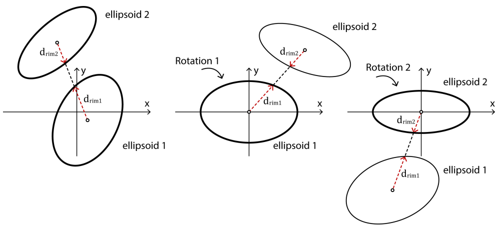

Figure 3. Left: A pair of ellipsoids in the global coordinate system; Mid: Transform the ellipsoids to the local coordinate system of ellipsoid 1; Right: Transform the ellipsoids to the local coordinate system of ellipsoid 2.

After the collision detection process is completed, the ellipsoid sizes and orientations are updated according to the tensors interpolated at the new centroid locations as a response to the tensor field. The three principal semiaxes of each ellipsoid are respectively aligned to the directions of the three tensors, and semi-axis lengths are determined by both the tensor magnitudes and user-decided maximum and minimum ellipsoid semi-axis lengths. Generally, an ellipsoid is expected to be smaller when the tensor magnitudes are higher, and is to be oriented towards the largest of the three tensors (Figure 4). In order to realize this, a mathematical mechanism is designed. Each tensor is mapped to a semi-axis length inversely proportional to its magnitude, with the maximum and minimum magnitudes of the tensor field as a source range and the user-input maximum and minimum semi-axis lengths as a target range. Each mapped length is then assigned to the two ellipsoid axes orthogonal to the tensor (Formula 4-6). Of the two lengths that an ellipsoid axis receives this way, the smaller of these two, corresponding to the larger orthogonal stress, is chosen as the length of that semi-axis.

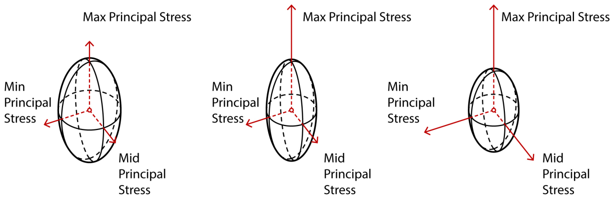

Figure 4. Larger tensor magnitude leads to a smaller ellipsoid.

$$l_x = \min\left(\frac{(t_y - t_{\text{min}})(l_{\text{max}} - l_{\text{min}})}{t_{\text{max}} - t_{\text{min}}}, \frac{(t_z - t_{\text{min}})(l_{\text{max}} - l_{\text{min}})}{t_{\text{max}} - t_{\text{min}}}\right) + l_{\text{min}} \tag{4}$$

$$l_y = \min\left(\frac{(t_x - t_{\text{min}})(l_{\text{max}} - l_{\text{min}})}{t_{\text{max}} - t_{\text{min}}}, \frac{(t_z - t_{\text{min}})(l_{\text{max}} - l_{\text{min}})}{t_{\text{max}} - t_{\text{min}}}\right) + l_{\text{min}} \tag{5}$$

$$l_z = \min\left(\frac{(t_x - t_{\text{min}})(l_{\text{max}} - l_{\text{min}})}{t_{\text{max}} - t_{\text{min}}}, \frac{(t_y - t_{\text{min}})(l_{\text{max}} - l_{\text{min}})}{t_{\text{max}} - t_{\text{min}}}\right) + l_{\text{min}} \tag{6}$$

Ellipsoids are added and relaxed iteratively until boundary geometry is filled (Figure 5). When no more collisions are detected within the number of iterations, the packing process is finished and the last pack of ellipsoids is output. The 3D lattice is then generated through the Kissing Ellipsoids algorithm, which connects the centroids of the kissing ellipsoids(Figure 6). This algorithm allows for the adjustment of the number of struts by changing kissing tolerance (Formula 7).

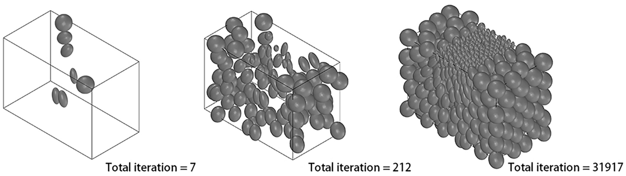

Figure 5. The pack of ellipsoids at different iterations.

Several optimization techniques are devised to improve the performance of the algorithm. When it comes to large-scale ellipsoid packing, looping over all pairs of ellipsoids for collision detection can be time-consuming and computationally expensive. To reduce the amount of required collision detection necessary during the packing iterations, the spatial partitioning technique is applied. The double of the maximum ellipsoid semi-axis length is used as the partition size, and the boundary is divided into voxels. Collision detection is only applied to a pair of ellipsoids if one of them was moved during the last iteration. If no movement occurred in the ellipsoid pair in the previous iteration, no new collision detection is applied.

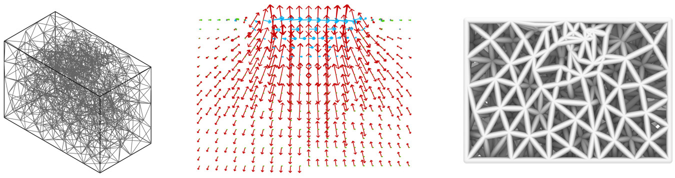

Figure 6. The lattice created by Kissing Ellipsoid algorithm and the comparison of the tensor field with the thickened lattice.

$$t_{\text{min}}(d_{rim1} + d_{rim2}) < ||C_2 - C_1|| < t_{\text{max}}(d_{rim1} + d_{rim2}) \quad (7)$$

3.2. COMPARATIVE STUDIES

3.2.1. Qualitative comparison of algorithmic design

Control lattices for comparison include both cellular periodic lattices generated by Crystallon for grasshopper, and other inhomogeneous anisotropic lattice precedents such as Sphere Packing followed by 3D Voronoi by Alvin Huang (2016), Ellipsoid Packing with 3D Delaunay by Felder et. al (2016).

Crystallon works by applying adaptive voxelization to a set design volume. A 3D unit linework pattern is created and mapped into each voxel of the design volume. A 3D periodic cellular lattice is generated with each voxel containing the same unit. The main input parameters include design volume, voxel size, and unit pattern. Users can morph the voxels in some areas using point attractors or curve attractors to locally densify or loosen the voxel division and potentially respond to external factors like loading. However, attractors are not explicit and are difficult for users to manipulate. The inhomogeneity and anisotropy of the periodic cellular lattice highly depend on the topology of voxelization and the anisotropy of the unit.

Huang's method of Sphere Packing with 3D Voronoi is employed within the context of a chair design. The furniture geometry is analyzed with Karamba, a Grasshopper FEA plug-in. The calculated principal stresses are used to inform the sphere packing process, where areas of larger stress are packed more densely with smaller spheres. This is similar to the PolyBrick ellipsoid packing logics. The two algorithms differ, however, in the process of lattice generation. Huang uses sphere centroids as an input for a 3D Voronoi algorithm and therefore loses control of strut orientations. While this method is able to achieve load-responsive inhomogeneity, it doesn't have enough control over lattice anisotropy due to a lack of strut orientation control.

The method of Ellipsoid Packing with 3D Delaunay employed by Felder et al. and our proposed packing method are similar in their use of FEA informed Ellipsoid Packing to control lattice directionality. However, there are three important differences in the algorithmic designs. Firstly, in Felder's method, Ellipsoid Packing is used to distribute lattice nodes and control inhomogeneity. The anisotropy is later achieved through a post-processing step, where struts not aligned with the stress field are removed. In our proposed Ellipsoid Packing algorithm, both anisotropy and inhomogeneity are controlled and enforced by dense packing. A second difference is the ways ellipsoids update in response to the stress field. In addition to aligning the ellipsoids to the stresses, Felder's method also tries to align the connection of the neighboring ellipsoids to the stresses. The lattice generation technique also shows a difference. In Felder's method, the lattice is generated by the 3D Delaunay triangulation algorithm, while our proposed method uses Kissing Ellipsoids.

3.2.2. *Quantitative comparison of lattice mechanical performances*

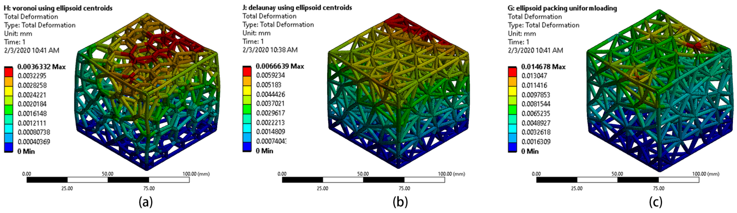

Figure 7. Part of the test lattices (a)3D Voronoi (b)3D Delaunay (c) Kissing Ellipsoids. Blue indicates zero deformation, red indicates the maximum deformation within each lattice.

Periodic cellular lattices, Sphere Packing with 3D Voronoi lattice, and Ellipsoid Packing with 3D Delaunay lattice are generated and tested with ANSYS (Figure 7). To begin the quantitative comparison, we generated lattices that had identical boundary sizes. When these line bodies were moved into ANSYS, they were all piped by the same radius, chosen to be 2mm. The total amount of force applied to each lattice is kept the same by dividing an arbitrary load of 1000N across the top face vertices of each lattice respectively. Controlling these parameters simplifies the post-processing of the axial force and directional displacement data because only lattice length is uncorrected for. Because some lattices have an overall longer length, meaning there was more material contained in the bounding box, they inherently perform better. To adjust for this, stiffness per volume is calculated using the maximum Y-direction displacement that occurs at the top face, the original height of the lattice, the total area of the top face piped vertices, and the 1000N uniform load. The maximum axial force and directional displacement data without this length correction are reviewed to get a general sense of the performance of the lattice.

4. Result

4.1. QUANTITATIVE COMPARISON OF LATTICE MECHANICAL PERFORMANCES

As the interest group, two ellipsoid packing lattices are tested, one generated with a uniform loading condition input and the other with a point loading input.

Figure 8 illustrates how our lattices perform in stress, deformation, and stiffness against the control lattices generated with different algorithms. An optimal lattice for load-bearing applications would minimize the highest stress found in a beam element, minimize the maximum deformation of the lattice, and maximize the stiffness because the forces would be most evenly distributed and protect the lattice well from breakage. Stiffness is essentially a combination of stress and deformation and by correcting for volume, we may consider the lattices independent of density. The ellipsoid packing lattices formed with uniform loading and point loading both show higher stiffness per volume ratios than a majority of the lattices tested, and outperform 11 and 14 lattices, out of the total 18 tested, respectively. The lattices each have a stiffness per volume of 172.7 kPa/m2 and 197.9 kPa/m2 respectively.

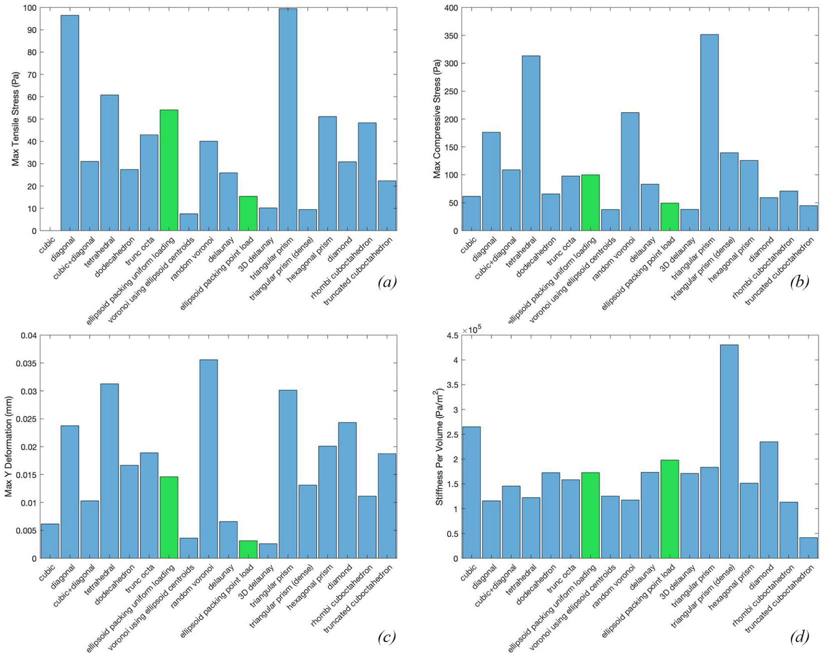

Figure 8. (a) Maximum Tensile Stress in one beam element. (b) Maximum Compressive Stress in one beam element. (c) Maximum deformation in the direction of the force application. (d) Stiffness per volume for each lattice. For a, b and c, lower value is more advantageous. For d, higher value is more advantageous.

5. Limitations and Conclusion

5.1. LIMITATIONS

In the quantitative comparison, we chose the piping radius of 2mm based on the expected physical dimensions of our physical prototypes. However, each individual lattice does have an optimal piping diameter to maximize performance.

Hence, we may not have chosen the ideal radius for the ellipsoid packing algorithm to demonstrate its full potential or we may have chosen a radius that causes other lattices to perform poorly. Therefore, our current results should only be considered on the scale of those prototypes. If we wanted to change the scale or extrapolate the results, more testing would be needed to explore the dependence between performance and piping radius. Additionally, the time and memory usage of different algorithms need to be compared for a better understanding of our method's computational performance.

5.2. CONCLUSION

In this paper, a new computational method based on ellipsoid packing is proposed for the generation of anisotropic and inhomogeneous lattice structures. In contrast with existing lattice generation algorithms, this method provides comprehensive control over both anisotropy and inhomogeneity and increases ease of generating specialized responsive geometries. For the next phase, a load-responsive strut thickening algorithm will be developed to create a complete lattice structure. Furthermore, the proposed method can be leveraged in architectural-scale projects with additional research of material and fabrication methods. Robotic clay extrusion is foreseen as a promising future trajectory to fabricate generated load-responsive lattice typologies at an architectural scale, contextualized as part of a wall design. Thus architectural advantages and potentials of proposed responsive 3D lattice algorithms will materialize.

References

Bagi, C.M., Berryman, E. and Moalli, M.R.: 2011, Comparative bone anatomy of commonly used laboratory animals: implications for drug discovery, *Comparative medicine*, **61**(1), 76-85.

Birol, E.B., Lu, Y., Sekkin, E., Johnson, C., Moy, D., Islam, Y. and Sabin, J.: 2019, PolyBrick 2.0: Bio-integrative load-bearing structure, *Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture*, Austin, 222-233.

Felder, A.A., Lewis, H., Piker, D., Pereira, A. and De Kestelier, X.M.V.: 2016, Mechano-adaptive space frame generation based on ellipsoid packing, *Proceedings of IASS 2016 Symposium: Spatial Structures in the 21st Century*, Tokyo, Japan.

Huang, A.: 2016, From Bones to Bricks: Design the 3D Printed Durotaxis Chair and La Burbuja Lamp, *Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture*, Ann Arbor, 318-325.

Im, H.C., AlOthman, S. and del Castillo, J.L.G.: 2018, Responsive Spatial Print: Clay 3D printing of spatial lattices using real-time model recalibration, *38th Annual Conference of Association of Computer Aided Design in Architecture*, 286-293.

Niu, J., Choo, H.L., Sun, W. and Mok, S.H.: 2018, Numerical study on load-bearing capabilities of beam-like lattice structures with three different unit cells, *International Journal of Mechanics and Materials in Design*, **14**(3), 443-460.

Sabin, J., Miller, M., Cassab, N. and Lucia, A.: 2014, PolyBrick: Variegated Additive Ceramic Component Manufacturing (ACCM), *3D PRINTING*, **1**(2), 78-84.