Algorithmic analysis and application of structural tessellation in design and optimization

Junho Chun[1*] and *Zai* Shi[1]

[1]School of Architecture, Syracuse University, Syracuse, NY, 13244, USA

Abstract. This paper explores the application of structural tessellation in architectural and structural design, where surfaces or spaces are divided into smaller repeating shapes or units to create aesthetic and functional structures. Structural tessellation offers various advantages, including improved stability, increased load-bearing capacity, and enhanced aesthetic appeal. The growing use of digital tools and advanced numerical algorithms has facilitated the creation of complex and intricate tessellations that can be tailored to suit specific project requirements. This research focuses on algorithmic methods for generating tessellations and their utilization in structural engineering and design optimization. Diverse patterns and configurations of tessellation are investigated through mesh generation algorithms, parametric approaches, and pattern gradation and repetitions, with data visualization accomplished using computing scripts and Grasshopper. The tessellation elements are employed to discretize the design domain in structural optimization and create initial patterns. The paper demonstrates the feasibility of proposed frameworks for structural design and application through the examination of various numerical examples. In conclusion, the strategic use of structural tessellation proves to be effective in producing unique and functional structures that seamlessly combine visual appeal with material and resource efficiency.

1 Introduction

Tessellations, the artful arrangement of repeated shapes to form patterns, have captivated the imagination across diverse disciplines. In recent years, tessellations have gained considerable attention in engineering and design fields due to their potential to optimize structures, enhance stability, increase load-bearing capacity, and improve aesthetic appeal. This paper is dedicated to exploring the realm of tessellations and their wide-ranging applications in engineering design. Beyond their traditional use in art and mathematics, tessellations have found innovative applications in engineering, offering novel ways to discretize and optimize design domains. By dividing surfaces or spaces into smaller repeating shapes or units, tessellations enable the creation of aesthetic and functional structures. They have been employed in diverse fields such as fracture mechanics, topology optimization, micromechanical analysis, computer graphics, and image processing, with promising results. The growing use of digital tools and advanced numerical algorithms has facilitated the

* Corresponding author: jchun04@syr.edu

creation of complex and intricate tessellations that can be customized to meet the specific requirements of a given project. With the aid of computing scripts and parametric approaches, researchers can generate diverse patterns and configurations of tessellation, allowing for seamless integration into various engineering design processes. Notable pioneers in the field, such as graphic artist Maurits Cornelis Escher and mathematics professor Roger Penrose, have explored the mathematical beauty and intricacy of tessellations, inspiring the integration of these patterns into engineering and design. Escher's mathematically inspired lithographs and impossible structures, as well as Penrose's ground-breaking work on self-similar quasicrystals with reflective and rotational symmetry, have demonstrated the captivating potential of tessellations in diverse domains.

This paper aims to investigate the algorithmic creation of tessellations and their practical application in structural engineering and design optimization. By examining the works of influential figures and studying the unique characteristics of tessellations, we seek to uncover the profound impact of geometrical patterns in engineering design. Through a comprehensive analysis of tessellation techniques and their integration into different engineering disciplines, this research endeavors to contribute to the evolving understanding and utilization of tessellations in cutting-edge engineering research and design.

2 Voronoi tessellation

Voronoi tessellation [1] is a mathematical technique that partitions a space into cells based on their distance to a set of points called generators. The Voronoi diagram, which is the graphical representation of the tessellation, has found numerous applications in various fields, including physics, computer science, biology, and geography. In this paper, we present the theory and principle behind Voronoi tessellation. The theory of Voronoi tessellation is based on the concept of proximity. Given a set of points in a space, the Voronoi diagram divides space into cells around each point, ensuring that any point in a cell is closer to its associated point than to any other point. This tessellation results in regions that are closest to specific points. The Voronoi diagram is constructed by finding the cells for each point, achieved by creating perpendicular bisectors between neighbouring points. The final Voronoi diagram is obtained by combining all the cells together. The Voronoi diagram is then obtained by taking the union of all Voronoi cells.

2.1 Generation algorithm

Given a set of n distinct points $\mathbf{P}$, the Voronoi tessellation of the domain $\Omega_V \subset \mathbb{R}^2$ is defined as:

$$\mathcal{T}(\mathbf{P};\, \Omega_V) = \{ V_{\mathbf{y}} \cap \Omega_V : \mathbf{y} \in \mathbf{P} \},\ V_{\mathbf{y}} = \{ \mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}-\mathbf{y}\| < \|\mathbf{x}-\mathbf{k}\|, \forall \mathbf{k} \in \mathbf{P} \setminus \{\mathbf{y}\} \} \tag{1}$$

where V_y denotes the Voronoi cell associated with point $\mathbf{y}$ (Fig. 1). The regularity of Voronoi diagrams is determined entirely by the distribution of the generating point set. To achieve a certain degree of regularity of Voronoi diagrams, the Lloyd's algorithm [2] can be used to iteratively improve a Voronoi diagram, a partitioning of a plane into regions based on a set of points called seeds. Each region in the Voronoi diagram corresponds to the area closest to a specific seed point compared to all other seeds. Centroidal Voronoi Tessellation (CVT) is a variation of the Voronoi diagram that emphasizes the notion of centroids. In a standard Voronoi diagram, the seed points serve as the centres of the Voronoi cells.

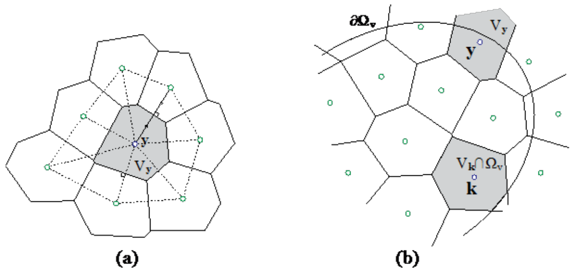

Fig. 1. (a) Voronoi diagram and its dual, the Delaunay triangulation, (b) illustration of V_y and $V_{\mathbf{k}}(\mathbf{P}) \bigcap \Omega_V$.

The centroids are points that minimize the average distance to all the points within their respective Voronoi cells. A Voronoi tessellation is centroidal if for every $\mathbf{y} \in \mathbf{P}$

$$\mathbf{y} = \mathbf{y}_c := \frac{\int_{V_{\mathbf{y}} \cap \Omega_V} \mathbf{x} \varphi(\mathbf{x}) d\mathbf{x}}{\int_{V_{\mathbf{y}} \cap \Omega_V} \varphi(\mathbf{x}) d\mathbf{x}} \tag{2}$$

and $\varphi(\mathbf{x})$ is a given density function defined over Ω_V. Lloyd's algorithm can be viewed as a fixed-point iteration for a specific mapping **L** that maps the point set **P** to the set of centroids of the Voronoi cells in $\mathcal{T}(\mathbf{P};\ \Omega_V)$ such as

$$\mathbf{L} = \mathbf{L}_{\mathbf{y}}(\mathbf{P}) = \frac{\int_{V_{\mathbf{y}}(\mathbf{P}) \cap \Omega_V} \mathbf{x} \varphi(\mathbf{x}) d\mathbf{x}}{\int_{V_{\mathbf{y}}(\mathbf{P}) \cap \Omega_V} \varphi(\mathbf{x}) d\mathbf{x}} \tag{3}$$

The fixed-point iteration nature of Lloyd's algorithm means that in each iteration, the seed points move closer to the true centroids of their respective Voronoi cells, and the algorithm converges to a set of centroids that minimize the average distance to the points within their cells (Fig. 2). This leads to an improved Voronoi diagram with better-balanced and more evenly distributed cells. Consecutive iterations of Lloyd's algorithm result in the decrease of an energy functional known as the Lloyd's Energy [3], [4].

$$\varepsilon(\mathbf{P};\ \Omega_V) = \sum_{\mathbf{y} \in \mathbf{P}} \int_{V_{\mathbf{y}}(P) \cap \Omega_V} \varphi(\mathbf{x}) \left\| \mathbf{x} - \mathbf{y} \right\|^2 d\mathbf{x} \tag{4}$$

This energy functional [5] is a measure of how well the Voronoi cells are balanced and how close the seed points are to the centroids of their respective cells. As Lloyd's algorithm iteratively updates the seed points to minimize the Lloyd's Energy (see Fig. 2 (d)), the Voronoi cells become better approximations of the regions closest to each seed point, and the algorithm converges to a more stable and optimal configuration of centroids.

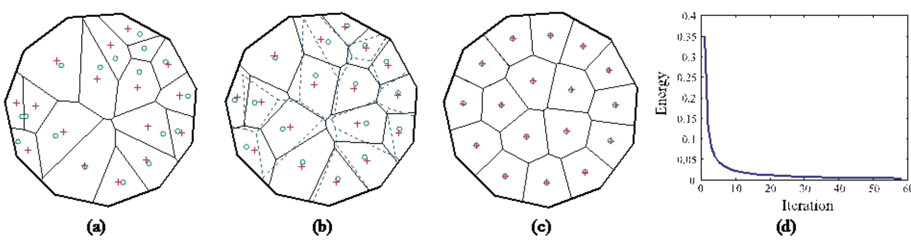

Fig. 2. (a) Random initial point set **P** (shown as green circles) and the corresponding Voronoi diagram, (b) first iteration of Lloyds' algorithm, (c) distributions of seeds and centroids (red cross marks) of the Voronoi diagram after 50 iterations, and (d) convergence of the energy functional.

2.2 Nonuniform Voronoi tessellation

The Lloyd's algorithm with a constant density function, $\varphi(\mathbf{x})$, produces a uniform distribution of seed points, resulting in a Voronoi discretization that is uniform in size across the entire domain. However, by choosing an appropriate density function, it becomes possible to generate non-uniform meshes with desired gradation. This density function biases the placement of points in specific regions, leading to a Voronoi diagram with varying cell sizes, providing a means to create meshes that adapt to the data distribution or prioritize specific areas of interest. One effective way to generate non-uniform meshes with desirable gradation is to use an initial distribution of seeds (points) such that regions requiring refinement have a higher density of seeds compared to other regions. Fig. 3 presents examples of graded meshes that can be produced using the algorithm presented in this study, along with the different density functions. The color bar in Fig. 3 represents relative or normalized mesh sizes across the domain. The color intensity indicates areas with finer or coarser mesh resolution, illustrating the adaptive nature of the meshing process based on the algorithm's refinement.

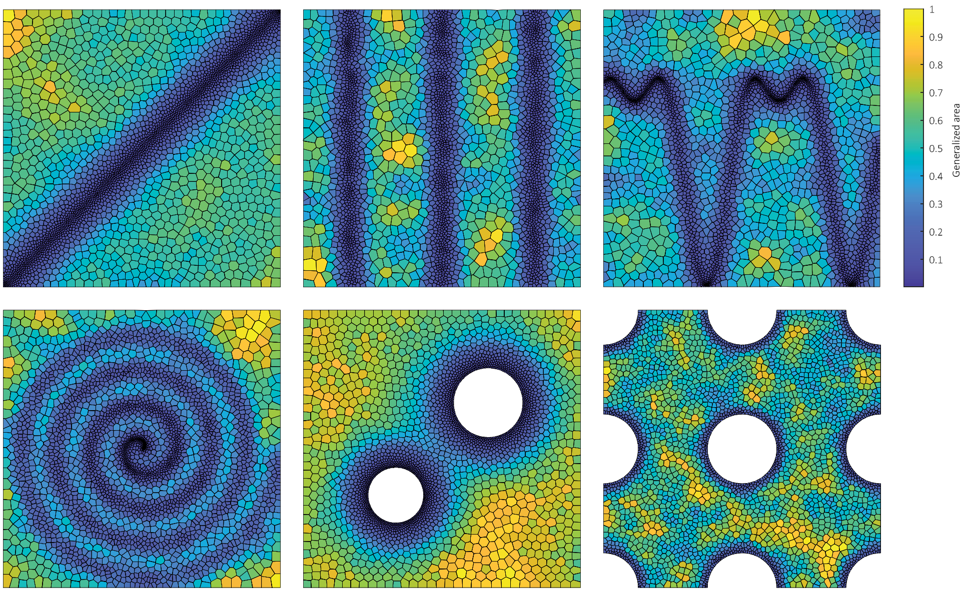

Fig. 3. Various sample graded meshes obtained through prescribed density functions, including linear, vertical lines, combined sine, and cosine functions, spiral, and circles around void areas.

3 General tessellation generation

In this section, the focus is on the generation of tessellations and their relevance in exploring the artistic aspects of engineering. The tessellations discussed here are based on basic two-dimensional shapes like triangles and quadrilaterals, serving as the underlying framework. These foundational shapes undergo modifications to derive motifs, which play a crucial role in creating the tessellations. The process involves utilizing fundamental concepts of translation and rotation to develop these captivating patterns and designs, showcasing the artistic potential within the realm of engineering. Using a triangle as the building block and using translation and rotation operations, diverse sets of tessellations can be constructed. For example, Fig. 4 shows the sequence of steps to create triangle-based tessellations. The red dashed line A-1 is transformed into the blue solid line. Afterward, the deformed line A-1 is rotated by 180 degrees around point 1, resulting in line A-B. Lines A-C and B-C are then obtained by rotating line A-B by 60 degrees each. By translating the patches horizontally and vertically, generated using the technique, the tessellations in Fig. 4 are created.

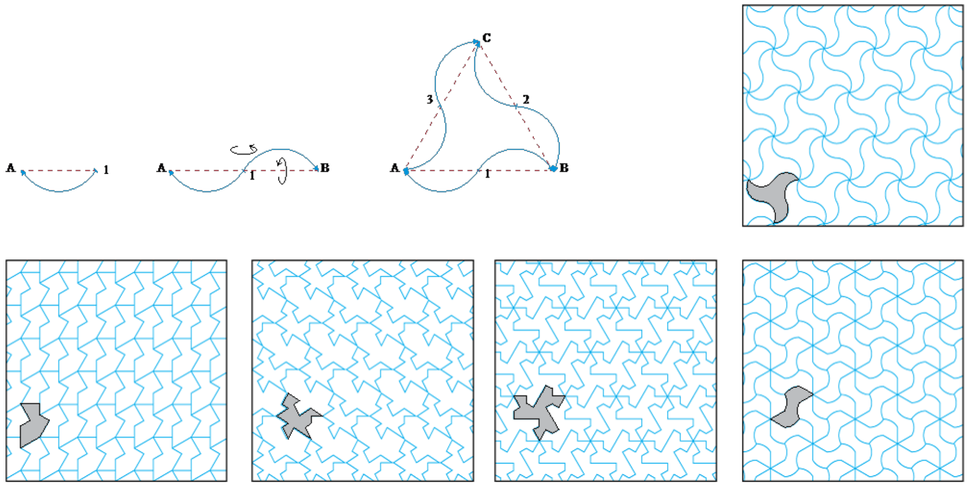

Fig. 4. The process for creating triangle-based tessellations involves the following sequence of steps, and the resulting tessellations generated using the patches shown in the corresponding insets.

3.1 Transformation of tessellation

In the realm of Euclidean geometry, an affine transformation, also known as an affinity, refers to a geometric transformation that preserves the essential property of lines and parallelism, though not necessarily Euclidean distances, and angles. It is a linear mapping method that conserves points, straight lines, and planes, ensuring that sets of parallel lines remain parallel even after the transformation. This characteristic makes affine transformations an enticing avenue for enriching the exploration of tessellations, which are patterns of repeated shapes without gaps or overlaps. Within this versatile canvas of tessellations, the application of affine transformations leads to captivating variations and distortions while upholding critical geometric attributes. Through the implementation of these transformations on individual domain within the tessellation, novel geometric arrangements emerge, retaining the distinctive features of the original pattern. For instance, scaling uniformly resizes the tiles, introducing varying sizes within the tessellation, creating an intriguing visual effect while maintaining the overall arrangement. Translation, another potent transformation, shifts the tiles in specific directions, giving rise to dynamic, offset versions of the original tessellation, evoking a sense of movement and liveliness. Rotation contributes to the introduction of symmetry, enabling the creation of rotationally symmetric patterns or intricate tessellations with diverse angles between neighboring tiles. Furthermore, the shearing transformation facilitates the skewing of tile shapes along one axis, leading to non-rectangular tiles and introducing asymmetry into the tessellation. By combining multiple affine transformations, such as scaling, translation, and rotation, one can embark on a journey of crafting complex variations and distortions of the original tessellation, resulting in a mesmerizing interplay of geometric forms. These transformative possibilities open up new avenues for creative expression and visual exploration within the captivating domain of tessellations.

An affine map can be expressed as the combination of two functions: a translation and a linear map. In the context of ordinary vector algebra, linear maps are represented using matrix multiplication with an invertible matrix A, while translations are represented using vector addition with a vector b. In the finite-dimensional case, for an affine map *f* acting on a vector **x**, it can be formally represented as the result of multiplying the vector **x** by the matrix **A** and then adding the vector **b**:

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{5}$$

Employing augmented matrices and vectors allows for the unified representation of both translation and linear maps through a single matrix multiplication. To implement this technique, all vectors need to be augmented with a "1" at the end, while all matrices require an additional row of zeros at the bottom, an extra column on the right to represent the translation vector, and a "1" in the lower right corner. The affine transformation matrices are defined as

$$\text{Translation: } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \quad \text{Scale: } \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Shear: } \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Rotation: } \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where t_x and t_y represent the displacement along the x-axis and y-axis, respectively; s_x and s_y specify the scale factor along the x-axis and y-axis, respectively; sh_x and sh_y specify the shear scale factor along the x-axis and y-axis, respectively; θ is the angle of rotation. Fig. 5 shows the affine transformation of the 2D mesh.

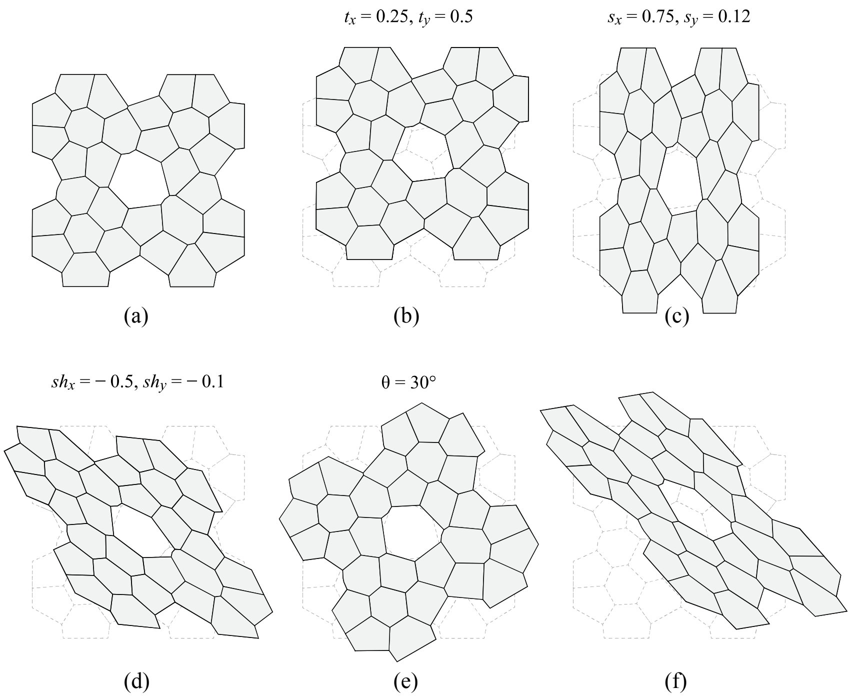

Fig. 5. (a) Original mesh, illustration of transformation of tessellations using (b) transformation, (c) scale, (d) shear, (e) rotation, and (f) combination of all given transformation matrices.

3.2 Conformal mapping, morphing and warping of tessellation

Conformal mapping [6], morphing and warping [7] of tessellations involve transforming the shape and arrangement of the tessellation elements to create variations or distortions while preserving certain geometric properties. These techniques provide a dynamic way to explore and manipulate tessellation patterns, enabling designers to achieve unique and visually captivating results. Grasshopper, a visual scripting plugin for Rhino (developed scripts are shown in Fig. 6), provides an efficient and intuitive platform to explore and implement these techniques. Using Grasshopper, one can apply warping and morphing to geometric elements, such as surfaces, curves, and meshes, by defining various transformation parameters. The process begins by creating the initial shape or geometry that serves as the base for the transformation. Then, a set of control points or vectors can be defined to influence the deformation. For warping, the control points can be moved, scaled, or rotated, causing the geometry to bend and distort while maintaining its overall structure. This allows for the creation of visually dynamic and organic forms that respond to changes in the control points.

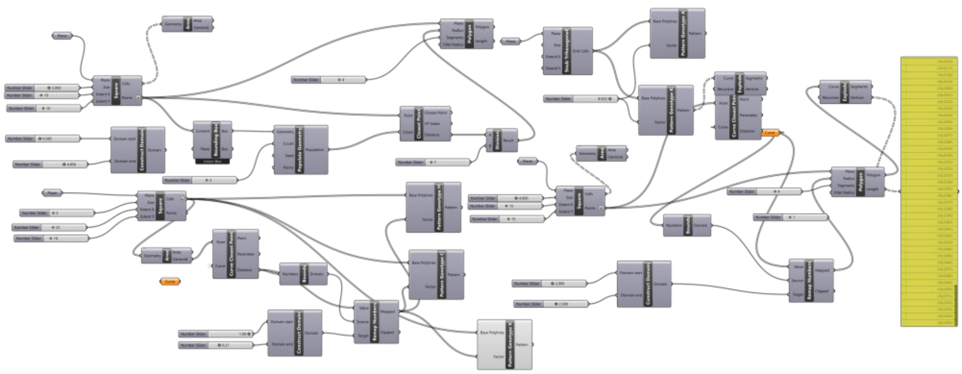

Fig. 6. Grasshopper (grasshopper.com/) scripts developed for conformal mapping, warping, and morphing of tessellations.

Morphing, on the other hand, involves interpolating between two or more shapes or geometries to create smooth transitions and animations. By defining a series of intermediate states and blending them together, the initial shape seamlessly transforms into the final form.

4 Structural design optimization

Structural design optimization [8] has become essential in engineering practice, as traditional prototyping methods suffer from limitations in terms of resources, cost, and time. Instead of relying solely on physical prototypes, engineers now turn to the power of mathematical models, simulations, and optimization to predict the behavior of structures under various conditions and find optimal designs. Classical structural optimization techniques, namely size [9], shape [10], and topology optimization [11], offer diverse approaches to achieving optimal designs. Size optimization focuses on determining the best thicknesses or cross-sectional areas of structural elements. In contrast, shape optimization explores the optimal boundaries of the design domain while keeping the topology fixed. Topology optimization, however, goes a step further by allowing for the modification of the overall layout, with design variables representing material densities or cross-sectional sizes. Continuum-based and discrete-based topology optimization methods play a crucial role in determining the optimal physical size, shape, and connectivity of the structure. These techniques generate material layouts that result in cost-effective and efficient designs, meeting the specified performance criteria. Overall, the integration of mathematical models, simulations, and optimization techniques in structural design has revolutionized the field, enabling engineers to make informed decisions, save valuable resources, reduce costs, and deliver high-performance structures that meet the demands of diverse applications. As computational capabilities advance further, structural design optimization will continue to evolve, unlocking even greater potential for innovation in engineering and construction. Deterministic optimization in both continuum and discrete design and topology optimization assumes that all design parameters, including material properties, loadings, and geometry, are deterministic during the optimization process. However, it is crucial to consider uncertainties in these parameters in structural engineering to prevent unexpected failures that could lead to catastrophic damage or loss of life. Therefore, incorporating optimization processes that address uncertainties becomes essential to achieve engineering solutions with a satisfactory level of reliability. This approach is commonly known as reliability-based design optimization [12]–[14]. This aims to achieve the optimal design and topology [15], [16] under given probabilistic constraints, arising from uncertainties in material properties or loads. A compliance minimization problem with a volume constraint in deterministic design optimization is formulated [8], [17] as

$$\begin{aligned} &\min_{\mathbf{A}} \quad \mathbf{f}^{\mathrm{T}}\mathbf{u}(\mathbf{A}) \\ &\text{s.t.} \quad \sum_{i=1}^{n_e} A_i L_i \le V_c, \;\; \mathbf{A}^{lower} \le \mathbf{A} \le \mathbf{A}^{upper} \\ &\text{with } \mathbf{K}(\mathbf{A})\mathbf{u}(\mathbf{A}) = \mathbf{f} \end{aligned} \tag{7}$$

where L_i represents the length of frame element i, n_e is the total number of truss elements, and $\mathbf{A}$ is the vector of cross-sectional areas of frame elements**,** V_c is the prescribed volume, $\mathbf{K}$, $\mathbf{u}$, $\mathbf{f}$ are the global stiffness matrix, the global displacement vector, and the global force vector, respectively. It is common practice to assign lower ($\mathbf{A}^{lower}$) and upper ($\mathbf{A}^{upper}$) bounds to design variables. The lower bound assignment is done to prevent the occurrence of a singular stiffness matrix during topology optimization. Equation (7) is commonly tackled using gradient-based optimization algorithms, which encompass sequential linear programming (SLP), sequential quadratic programming (SQP;[18]), convex linearization (COLIN; [19]), optimality criteria (OC; [20]), and method of moving asymptotes (MMA; [21]). These algorithms efficiently navigate the optimal solutions by iteratively updating design variables based on gradients of the objective function and constraints.

5 Numerical analysis and application

The proposed tessellation generation approach and optimization framework are tested on structural optimization problems. A gradient-based optimization algorithm, the method of moving asymptotes, is employed to solve all problems in this section.

5.1 Analysis of a cantilever beam discretized with tessellations

In this numerical example, a cantilever beam in Fig. 7 is studied under a point load to examine the influence of tessellation. The continuum domain of the beam is discretized with various types and shapes of tessellation, where each tessellation element comprises edges and nodes. Frame or bar elements are employed to replace each edge between adjacent nodes, depending on the structural type.

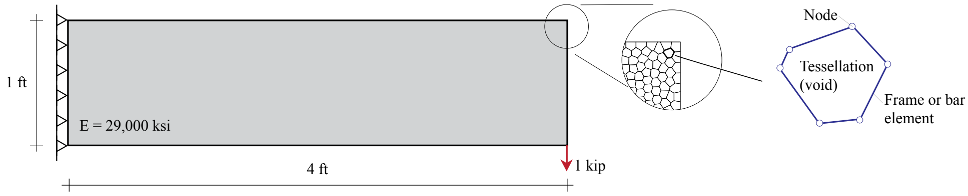

Fig. 7. Illustration of a cantilever beam that is subjected to a point load, accompanied by a representation of the material properties, loading, and boundary conditions. The construction approach for creating frame elements based on the outline of tessellation.

Fig. 8 (a) illustrates different tessellations generated through Voronoi, transformation, mapping, and warping techniques, as discussed in earlier sections. Upon constructing frame elements from these tessellations within the cantilever beam domain, the total frame lengths of cantilever beams with tessellations are measured. To ensure consistency, the cross-sectional areas of square steel (modulus of elasticity E = 29,000 ksi) frame elements are adjusted so that all cantilever beams have the same total volume, depicted in Fig. 8 (b).

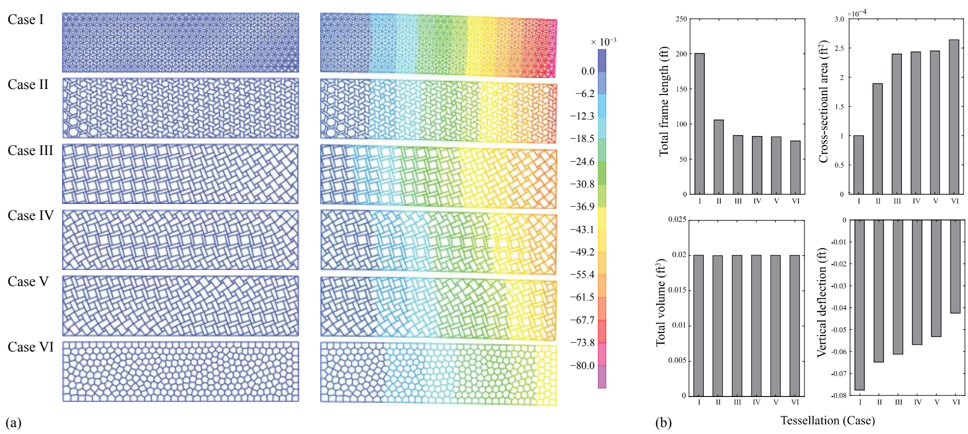

Fig. 8. (a) Illustration of the configurations of the cantilever beam with different shapes of tessellation, along with corresponding vertical deflections. (b) Depiction of the frame structure volume, cross-sectional areas, and vertical deflections of the cantilever beams.

Structural analysis is then conducted to assess vertical deflections under given loading and boundary conditions. The comparison results of the structural analysis are presented in Fig. 8. The analysis demonstrates the significant impact of discretization approaches and tessellations on structural behaviors. Notably, the cantilever beam with the highest number of frame elements in Fig. 8 (a) exhibits the largest deflection, while the one utilizing Voronoi elements displays the least vertical deflection under the same applied load magnitude. These observations underscore the critical role of frame element configurations and sizes in shaping structural behavior. The diverse tessellation configurations explored in this numerical example offer potential for creating aesthetically appealing structural elements while enhancing overall performance.

5.2 Structural optimization of a bridge structure

Fig. 9 depicts a 100 ft × 20 ft bridge structure subjected to point loads in this numerical application. The main objective is to investigate the optimal sizes of bridge frames created using various discretization techniques with tessellations. Distributed point loads are applied to the bridge structure on the bridge deck, and a roller support is placed at the right bottom location, while a pin support is assigned to the left top point. The approaches described in the previous section are employed for tessellation, and the objective function is to minimize the total volume (weight) of the bridge. Additionally, a constraint function (g) is imposed on the static compliance, represented as $g:\text{compliance}\,(=\mathbf{f}^T\mathbf{u}(\mathbf{A}))-0.25\leq 0$. Five different tessellations are utilized, incorporating transformation and density function approaches to discretize the continuum design domain. Frame elements are then placed for tessellation edges based on nodes, as depicted in Fig. 10 (a). Steel material properties are considered for frame elements having the same modulus of elasticity E. The specific parameter values employed for this optimization are summarized in Table 1. The optimization process is iteratively repeated until one of the termination criteria is met. These criteria include reaching the maximum number of iterations prescribed or satisfying the convergence criterion.

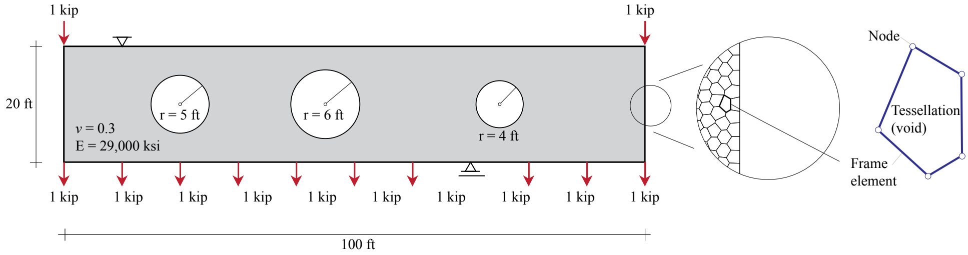

Fig. 9. Bridge design domain, loading and boundary condition.

Table 1. Parameters values used for the bride structure optimization problem.

Compliance limit	A^{lower}	A^{upper}	Modulus of Elasticity, *E*	Convergence criterion
0.25	0.05 ft^2	0.5 ft^2	29,000 ksi	10^{-4}

Fig. 10 shows the optimized bridge structures obtained through the tessellation-based optimization process. The corresponding cross-sectional areas of the optimized bridges are depicted in Fig. 10 (b), providing insight into the distribution of materials and load-bearing capacities. Furthermore, Fig. 10 (c) displays the internal forces acting on each optimized bridge structure, aiding in understanding their structural behavior under the applied loads. The convergence histories of both the objective and constraint functions are presented in Figures 10 (d) and 10 (e) respectively. These convergence plots track the changes in the objective function (e.g., total volume or weight) and constraint function (e.g., static compliance) over successive iterations, illustrating how the optimization process progresses towards an optimal solution. Additionally, Fig. 10 (f) offers a comparison of the volumes between the initial bridge structures and the optimized counterparts. This allows for an assessment of the efficiency gained through the tessellation-based optimization, showing potential reductions in material usage and resource consumption. The proposed method effectively identifies solutions that satisfy the specified constraints while minimizing the volume, demonstrating its efficiency and effectiveness for optimizing this bridge design scenario. Furthermore, the optimized bridge structures exhibit variations in material distributions and frame configurations, influenced by the initial tessellation layouts and compositions. Employing creative tessellations in engineering design opens up the possibility of discovering diverse and innovative solutions. These tessellations can result in architecturally and structurally appealing designs, offering a unique blend of aesthetics and functionality.

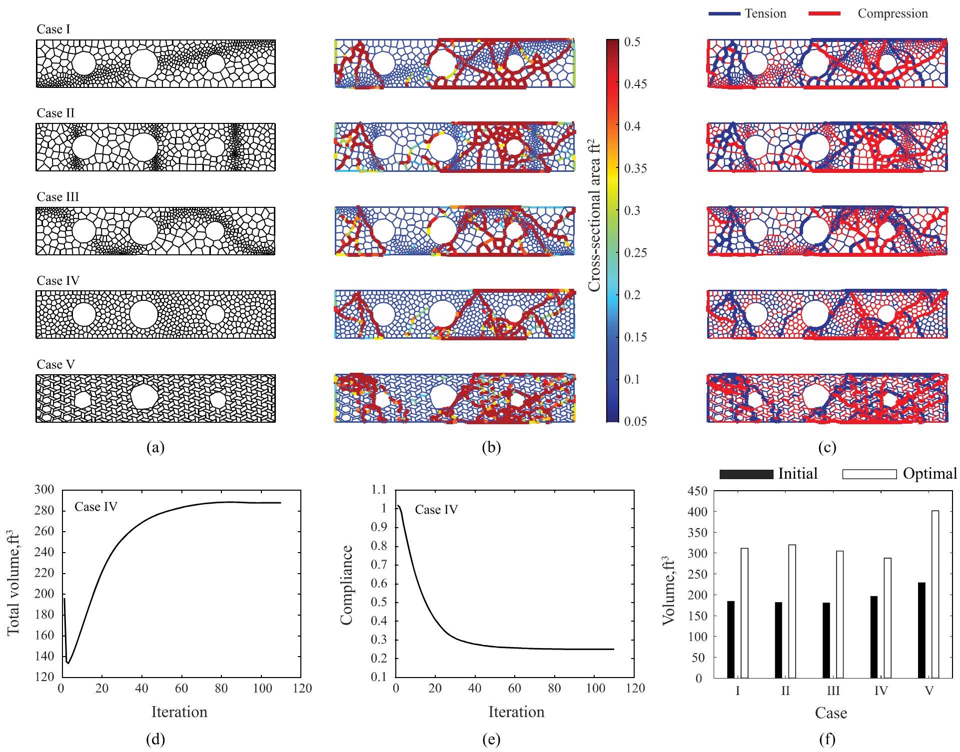

Fig. 10. (a) Bridge design domain discretization, (b) optimized frame member sizes, (c) member forces, (d) convergence history of Case IV – volume, (e) convergence history of Case IV – Compliance, and (f) volumes of initial and optimized bridge structures.

6 Conclusion

This paper delves into the fascinating realm of tessellations and their applications in engineering and design. Through an in-depth exploration of tessellation generation techniques, such as Voronoi, transformation, mapping, and warping, we have demonstrated their potential in creating captivating and diverse patterns. Lloyd's algorithm has been highlighted as a powerful tool for optimizing tessellation distributions, enabling uniform and non-uniform meshes with desirable gradations. Moreover, we have showcased the versatility of affine transformations in generating new geometries while preserving essential geometric properties. By combining translation, scaling, shearing, and rotation, engineers and designers can unlock a world of creative possibilities for aesthetic and functional designs. The numerical examples presented in the paper further exemplify the practicality of tessellations in structural optimization and architectural design. The optimized bridge structures and their convergence histories exemplify the efficacy of tessellations in identifying engineering solutions that meet constraints while minimizing volume. Employing creative tessellations in engineering design opens up the possibility of discovering diverse and innovative solutions. These tessellations can result in architecturally and structurally appealing designs, offering a unique blend of aesthetics and functionality. By exploring various tessellation patterns and configurations, engineers and designers can create visually captivating structures and components that not only meet functional requirements but also inspire and engage the senses. The use of tessellations in engineering design allows for a harmonious integration of form and function, transforming traditional approaches into artful expressions of creativity and ingenuity. Whether in architectural facades, structural elements, or product design, the incorporation of creative tessellations fosters an environment of endless possibilities, driving forward the boundaries of engineering aesthetics and innovation.

Acknowledgments

The authors acknowledge the support of the Innovative and Interdisciplinary Research Program of Syracuse University (Grant no. II-50-2021).

References

1. F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, 1991, doi: 10.1145/116873.116880.
2. J. Max, "Quantizing for Minimum Distortion," *IRE Trans. Inf. Theory*, vol. 6, no. 1, pp. 7–12, 1960, doi: 10.1109/TIT.1960.1057548.
3. Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi Tessellations: Applications and Algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, 1999, doi: 10.1137/S0036144599352836.
4. C. Talischi, G. H. Paulino, A. Pereira, I. F. M Menezes, A. Pereira, and I. F. M Menezes, "PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab," vol. 45, pp. 309–328, 2012, doi: 10.1007/s00158-011-0706-z.
5. Q. Du, M. Emelianenko, and L. Ju, "Convergence of the Lloyd Algorithm for Computing Centroidal Voronoi Tessellations," *SIAM J. Numer. Anal.*, vol. 44, no. 1, pp. 102–119, 2006, doi: 10.1137/040617364.
6. G. B. Arfken, *Mathematical Methods for Physicists*. Academic Press, 2013.
7. L. McMillan Jr., "An image-based approach to three-dimensional computer graphics," The University of North Carolina at Chapel Hill, 1997.
8. R. T. Haftka and Z. Gürdal, *Elements of Structural Optimization*. Springer Netherlands, 1992.
9. A. K. Peter W. Christensen, *An Introduction to Structural Optimization*. Springer Netherlands, 2008.
10. R. T. Haftka and R. V Grandhi, "Structural shape optimization — a survey," *Appl. Mech. Eng.*, vol. 57, pp. 91–106, 1986.
11. M. P. Bendsøe and O. Sigmund, *Topology Optimization: Theory, Methods, and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.
12. D. M. Frangopol, "Structural Optimization Using Reliability Concepts," *J. Struct. Eng.*, vol. 111, no. 11, pp. 2288–2301, Nov. 1985, doi: 10.1061/(ASCE)0733-9445(1985)111:11(2288).
13. J. Chun, "Reliability-Based Design Optimization of Structures Using Complex-Step Approximation with Sensitivity Analysis," *Appl. Sci.*, vol. 11, no. 10, p. 4708, May 2021, doi: 10.3390/app11104708.
14. J. Chun, "Reliability-Based Design Optimization of Structures Using the Second-Order Reliability Method and Complex-Step Derivative Approximation," *Appl. Sci.*, vol. 11, no. 11, p. 5312, Jun. 2021, doi: 10.3390/app11115312.

15. J. Chun, J. Song, and G. H. Paulino, "Structural topology optimization under constraints on instantaneous failure probability," *Struct. Multidiscip. Optim.*, vol. 53, no. 4, 2016, doi: 10.1007/s00158-015-1296-y.
16. J. Chun, J. Song, and G. H. G. H. Paulino, "System-reliability-based design and topology optimization of structures under constraints on first-passage probability," *Struct. Saf.*, vol. 76, pp. 81–94, Jan. 2019, doi: 10.1016/j.strusafe.2018.06.006.
17. J. Chun, G. H. Paulino, and J. Song, "Reliability-based topology optimization by ground structure method employing a discrete filtering technique," *Struct. Multidiscip. Optim.*, vol. 60, no. 3, pp. 1035–1058, Sep. 2019, doi: 10.1007/s00158-019-02255-1.
18. P. T. Boggs and J. W. Tolle, "Sequential Quadratic Programming," *Acta Numer.*, vol. 4, pp. 1–51, 1995, doi: 10.1017/S0962492900002518.
19. C. Fleury and V. Braibant, "Structural optimization: A new dual method using mixed variables," *Int. J. Numer. Methods Eng.*, vol. 23, no. 3, pp. 409–428, Mar. 1986, doi: 10.1002/nme.1620230307.
20. C. Fleury, "Structural weight optimization by dual methods of convex programming," *Int. J. Numer. Methods Eng.*, vol. 14, no. 12, pp. 1761–1783, Jan. 1979, doi: 10.1002/nme.1620141203.
21. K. Svanberg, "The method of moving asymptotes: a new method for structural optimization," *Int. J. Numer. Methods Eng.*, vol. 24, no. 2, pp. 359–373, Feb. 1987, doi: 10.1002/nme.1620240207.