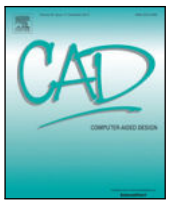
Computer-Aided Design 166 (2024) 103633

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad



Reducing the Number of Different Faces in Free-Form Surface Approximations Through Clustering and Optimization

Yuanpeng Liu [a], Ting-Uei Lee [a], Anooshe Rezaee Javan [a], Nico Pietroni [b], Yi Min Xie [a,*]

[a] Centre for Innovative Structures and Materials, School of Engineering, RMIT University, Melbourne, 3001, Australia
[b] School of Computer Science, University of Technology Sydney, Sydney, 2007, Australia

ARTICLE INFO

Keywords:
Architectural geometry
Rationalization
Clustering
Optimization
Façade design
Free-form surface

ABSTRACT

Free-form structures are highly valued for their aesthetic appeal in architecture, but they typically comprise panels of many different shapes, which can pose great challenges for building construction. In this study, we aim to address this issue by proposing a novel clustering–optimization method to reduce the number of different n-gonal faces in free-form surface approximations. The method partitions the faces into several groups of similar shapes through clustering and transforms the ones within each group toward congruent forms through optimization. By utilizing this approach, the number of geometrically different panels can be reduced while also satisfying a user-specified error threshold. The potential practical application of this method is demonstrated by redesigning the façade of a real architectural project to achieve cost-effective solutions.

1. Introduction

Free-form surfaces create 3D shapes that possess innovative and elegant appearances. In recent years, the adoption of free-form surfaces has been rapidly increasing in architectural designs due to the development of computer-aided design technology [1–3]. However, free-form surfaces are typically complex due to the inherent double curvature, which often poses great difficulty for real-world construction. Therefore, in the last decade, significant effort has been focused on the problem of architectural rationalization, i.e., making the construction of complex surfaces feasible and affordable [4–8].

One major rationalization strategy is to reduce the shape variety of building components, such as beams, nodes, and panels [9]. Typically, free-form structures consist of elements with many different shapes, which often results in expensive and time-consuming construction [10]. By reducing the number of different elements, mold-based manufacturing techniques like casting and vacuum forming can use the same mold for repeated elements, resulting in lower production costs [11]. For other no-mold manufacturing processes like CNC (computer numerical control) machining and 3D printing, reducing the shape variety may simplify programming and setup for the machinery, leading to increased efficiency and quality control [12]. Fewer types of components can also ease storage and transportation [13]. Moreover, for on-site assembly, using fewer types of elements may reduce the amount of backup components, hence further lowering overall fabrication costs [14].

Despite the above-mentioned benefits, reducing the number of different elements in complex free-form geometries is not an easy task. To facilitate numerical analysis, free-form structures are often simplified as polygonal meshes, with beams treated as edges, nodes as vertices characterized by the angles between adjacent edges, and panels as faces. To reduce the number of different edges, several methods have been proposed in different contexts, including architectural structures [15,16] and sphere tessellations [17,18]. Regarding nodes, methods include replacing low-valence nodes with fewer types of high-valence ones [15,19], or merging different nodes by similarizing their respective shapes [14,20]. In this study, however, we only focus on faces. Several studies have investigated using congruent regular triangles to create various three-dimensional shapes [21–23]; the results suggest that only limited target profiles can be achieved. For complex free-form surfaces, a related approach has been proposed to reduce the number of different molds for panel fabrication [24], where panel shapes can be different. Other studies have instead explored free-form approximations using congruent groups of triangles [25,26] or quadrilaterals [11]. However, these methods are not generic enough; general polygonal meshes composed of one or more types of different n-gonal faces remain underexplored. These meshes contain unique properties [27,28] and visually appealing patterns that hold a wide range of practical applications in architecture and artistic designs [29–31].

The motivation for this study comes from a real architectural project, the Soumaya Museum, as shown in Fig. 1. The initial façade design contains a large number of uniquely shaped hexagonal panels. The

* Corresponding author.
E-mail address: mike.xie@rmit.edu.au (Y.M. Xie).

https://doi.org/10.1016/j.cad.2023.103633
Received 19 April 2023; Received in revised form 11 October 2023; Accepted 16 October 2023
Available online 21 October 2023

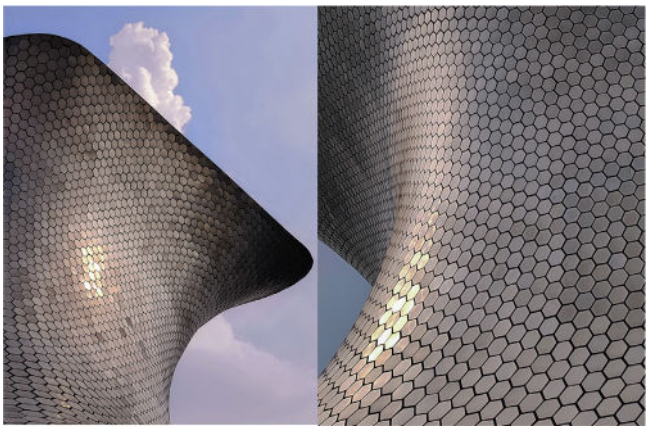


Fig. 1. The façade of the Soumaya Museum, which is mostly made up of hexagonal panels, with some trimmed ones near the boundaries.

Fig. 2. A redesign of the Soumaya Museum using the proposed method. Given an initial mesh with 1,924 faces, the final group number is significantly reduced from 500 (using clustering alone) to 285 (after incorporating the proposed optimization method).

original approach [29,32] for reducing the shape variety is to cluster all panels into a specified number of groups, with each being replaced by the representative shape of its corresponding group (i.e., the group centroid). However, since panels typically remain different from their respective centroids, such shape discrepancy would cause nonuniform gaps between neighboring panels. The required group number by using clustering alone could still be high in order to achieve a small shape discrepancy. In this study, we propose to add an optimization process after clustering to convert faces within the same group toward congruent shapes. This approach has great potential to further reduce the group number while meeting the specified level of shape discrepancy. Although similar concepts have been explored in the contexts of triangular [25] and quadrilateral [11] meshes, further research is still required for general polygonal meshes that contain more complex faces and even mixed different n-gonal faces.

This study presents a new method to reduce the number of different faces in free-form surface approximations based on polygonal meshes that comprise general n-gons. The main ingredients are clustering and optimization. In clustering, faces are adaptively partitioned into a specific number of groups of similar shapes. The geometric closeness between two different faces is evaluated based on their best-fit alignment while considering all possible vertex correspondences. In the subsequent optimization, faces within the same group are transformed toward congruent shapes by equalizing their corresponding edges, angles, and signed distances from vertices to their corresponding best-fit planes. This strategy is generic and can be applied to different n-gonal faces. Clustering and optimization are interleaved until the clustering results remain unchanged, which is found to be beneficial for improving the outcome. Along the clustering–optimization process, the topology of the input mesh remains unchanged, faces are kept connected, and the positions of vertices are adjusted based on the applied geometric goals. Based on the clustering–optimization process, a global framework is developed to determine the required group number that satisfies a specified error threshold. The error-driven property allows users to explore various results to accommodate projects with different error requirements.

The remainder of this paper is organized as follows: The details of clustering and optimization are explained in Section 2. Section 3 presents a variety of numerical examples to investigate the performance of the proposed method. In Section 4, the potential practical application of the method is demonstrated through a redesign of the Soumaya Museum, as shown in Fig. 2. Section 5 discusses the limitation and potential extension of the method. Section 6 draws conclusions.

2. Methodology

2.1. Overview

The core of our method is clustering and optimization. We first define a similarity metric to evaluate the geometric closeness between two different n-gonal faces in Section 2.2.1. Based on this metric, we apply an adaptive clustering method to partition faces into groups of similar shapes, as described in Section 2.2.2. After clustering, we seek to transform faces within the same group toward congruent shapes through optimization. An additional goal is considered to approximate the target surface for architectural applications. Here, each geometric goal is formulated as an objective subterm, as detailed in Sections 2.3.1 and 2.3.2. Summing up the weighted terms gives the overall objective function, as stated in Section 2.3.3.

Essentially, this is a multi-objective optimization problem with the positions of vertices treated as the design variables. The objective function can be minimized by manipulating the values of design variables based on the gradient information. Since the objective function is built based on the geometric goals, the movement of vertices drives the geometry to gradually approach the specific goals. However, in cases where the geometric goals are in conflict, the objective function usually cannot be reduced to zero; the final outcome is typically a compromising solution with each geometric goal only partially achieved.

2.2. Clustering

2.2.1. Similarity metric

To evaluate the geometric closeness between two n-gonal faces, P and Q, each represented by a set of n vertices, the key idea is to determine their best-fit alignment [11,25]: With Q fixed, we perform rigid body motions (i.e., translations and rotations) on P to minimize s, the sum of squared distances between corresponding vertex pairs, as shown in Fig. 3. To achieve this, different possible vertex correspondences between P and Q need to be first analyzed. Assuming that the vertex permutation of Q is fixed, any vertex in P can be chosen as the starting vertex, resulting in n different permutations. If flipping the face (i.e., reversing the direction of the vertex set) is permitted, there are n additional permutations possible. Since each permutation matches a unique vertex correspondence, all possible correspondences can be identified. For a given vertex correspondence, the best-fit mapping process T can be determined using the method described in [33], which yields the alignment that globally minimizes the summation s under the specific vertex correspondence. By examining all possible alignments,

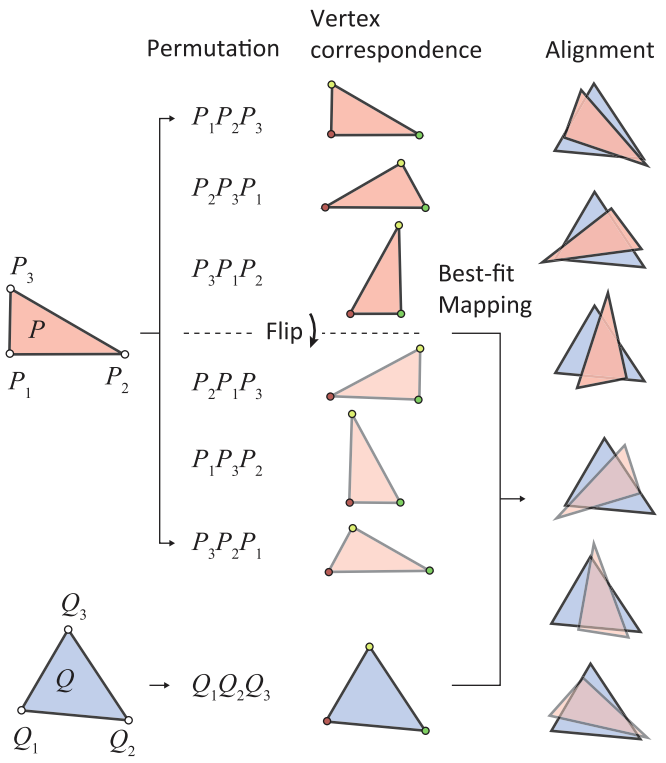


Fig. 3. To calculate the similarity metric between two different triangular faces, P and Q, three or six (if allowing flipping the face) different permutations need to be considered, each leading to a unique vertex correspondence. For a given correspondence, P is best-fit mapped to Q, resulting in an alignment with the sum of distances between corresponding vertex pairs minimized.

the one with the minimal s can be determined as the best alignment. Finally, the similarity metric d is defined as follows:

$$d(P,Q) = \min_{j=1}^{n \text{ or } 2n} \left(\sqrt{\frac{\min(\sum_{i=1}^{n} \|T_{P_jQ}(P_{ji}) - Q_i\|^2)}{n}} \right) \tag{1}$$

where i is the vertex index, j is the permutation index, T_{P_jQ} is the best-fit mapping process between the jth permutation of P and Q, and n and $2n$ refer to the cases without and with flipping, respectively. Flipping is allowed for the numerical examples in this study, unless otherwise specified. To eliminate the effect of the vertex number on the result, the summation s is divided by n in Eq. (1). The final similarity metric approximates the averaged distance between a vertex in P and its corresponding vertex in Q at the best-fit alignment. In general, the lower the similarity metric d, the more alike these two faces are; when d equals zero, these two faces are congruent.

The proposed similarity metric might not adequately capture the shapes of faces. This is because it is only based on the coordinates of vertices, whereas the intermediate face shape also depends on the curvature. For example, a planar face may share the same vertices as a cylindrical face, yet it is obvious that they are not congruent, even though the similarity metric between them is zero. To avoid such undesirable cases, we assume that vertices are connected with straight lines and faces are minimal surfaces in this study.

2.2.2. Clustering method

In clustering, we seek to partition mixed different n-gonal faces into a specified number of groups of similar shapes. This can be achieved through the farthest point sampling strategy [34] and k-means clustering [35]. Specifically, the group number is gradually increased by one at each iteration until reaching the specified value. During iteration, each face is treated as a point, and the distance between two different points is evaluated using the similarity metric defined in Eq. (1). The overall iteration process is summarized as follows:

1. Divide all faces into different groups according to their vertex numbers to separate different n-gons. In each group, choose one face at random as the initial group centroid.
2. In each group, best align all faces with the centroid to create a group of superimposed faces. Then, shift each vertex of the centroid to the average position of the corresponding vertices of the superimposed faces.
3. Calculate the distance from each face to the updated centroid.
4. Check if the current group number reaches the target value. If so, end the iteration; otherwise, continue the following steps to add a new group.
5. Choose the face that is currently the farthest away (i.e., contains the most dissimilar shape) from its corresponding centroid as the centroid of a new group.
6. For the centroid of the new group, calculate its distances to the faces that have the same number of vertices. Allocate the faces that are closer to the new centroid than their current ones to the new group.
7. Return to step 2.

After clustering, faces are divided into a specified number of groups of similar shapes. Each face is assigned to a group, and its vertex correspondence with the respective group centroid is determined. Increasing the group number tends to make faces within the same group more alike, as the most dissimilar shapes are gradually filtered into new groups. However, the extent of geometric closeness within groups is highly dependent on the input mesh. This is because throughout the clustering process, the vertices are not moved, and the faces retain their original shapes. Given a "poor" input mesh with faces mostly in different shapes (which is often the case for an arbitrary mesh generated from a free-form surface), clustering alone is typically not very useful for achieving groups of near congruent shapes (discussed further in Section 4). In this regard, based on the clustering result (i.e., the grouping information and the vertex correspondence), an optimization process is proposed to further transform faces within the same group toward congruent shapes by fine-tuning the positions of vertices, as detailed in the following subsection.

Note that other clustering methods [36] may also be adapted to partition mixed n-gonal faces into groups of similar shapes. However, most existing clustering methods are only based on the shapes of input faces without considering the subsequent optimization. Using these techniques, a better clustering result does not necessarily guarantee a better optimization outcome [20]. Pre-considering the optimization performance while clustering is a challenging task, which will be further discussed in Section 5.

2.3. Optimization

2.3.1. Achieving congruent shapes

The main goal of optimization is to transform geometrically different n-gonal faces into congruent shapes. Our key idea is to control the shape of a face by manipulating its edges and angles. For instance, the shape of a triangle can be determined by the lengths of its three edges; two triangles with identical edge lengths are thus congruent. Following this idea, to achieve congruent shapes for different n-gonal faces, we propose to equalize their corresponding edges and angles. To verify the effectiveness of this strategy, we must ensure that each single n-gonal face with specified edge lengths and angles has a fixed and unique shape. For the convenience of implementation and analysis, in the following discussion, each angle is replaced by an "angle edge", which is a virtual edge that connects the two neighboring points of this angle's vertex, as shown in Fig. 4.

Given a face with specified values of edges and angle edges, we must first ensure its shape is fixed. Here, we approach this problem through

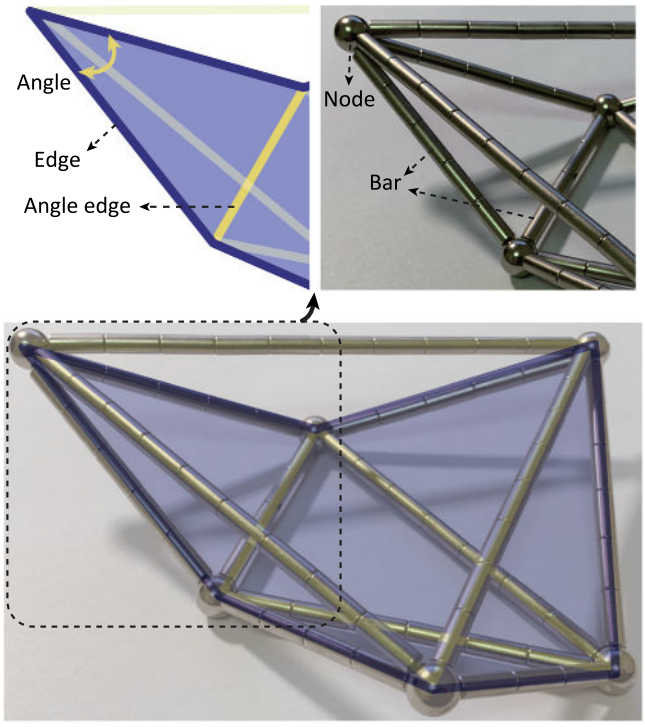


Fig. 4. A face (in light blue) can be treated as a node-bar structural system. Nodes are the face vertices, and bars are the edges (in dark blue) and angle edges (in yellow) with specific lengths. In this example, the nodes cannot move without elongating or shortening the lengths of bars, which indicates the face shape is fixed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

kinematic analysis. A face can be treated as a node-bar structural system, where nodes are the vertices and bars are the edges and angle edges with specific lengths, as shown in Fig. 4. To fix the face shape, we need to ensure that the nodes cannot move without elongating or shortening the bars. In the node-bar system, each node contains three degrees of freedom (DoF), and each bar poses one constraint. Six additional constraints are required to connect the system to a foundation to eliminate the rigid body motions. Thus, the number of remaining DoF for an n-gonal face with n vertices and m bars is $3n-m-6$. According to Maxwell's rule [37,38], when $3n-m-6 \leq 0$, no node movement is possible, so that the face shape can be verified to be fixed.

In our proposed strategy, $2n$ bars are added (n from edges and another n from angle edges) for an n-gonal face. However, for triangular faces, the angle edges are repeated with the edges; thus, only three bars are effective ($m = 3$). For quadrilateral faces, the angle edges are self-repeated, with only six bars being valid ($m = 6$, four from edges and two from angle edges). For cases with $n > 4$, no repetition exists between edges and angle edges ($m = 2n$). Together, the remaining DoF for any n-gonal face is summarized in Table 1. Specifically, when $n > 6$, the remaining DoF is positive, meaning that the face shape can still be deformed. For these cases, we add n additional constraints by limiting the distances between the ith and $(i+3)$-rd vertices within each face. This reduces the remaining DoF to $n-6-n=-6$, which is a negative constant, thus ensuring fixed face shapes regardless of the value of n.

Given a face whose shape is fixed (with no remaining DoF), we also need to ensure its shape is unique. Unfortunately, this is violated by the cases of mirror images, as shown in Fig. 5. The shape difference here is caused by the opposite directions in which the vertices deviate from the mirror plane. To exclude such undesired instances, we further require the signed distance (considering the direction) from each vertex to the best-fit plane to be identical for different faces. The best-fit plane is the

Table 1
The number of nodes n and effective bars m in an n-gonal face. The number of the remaining degrees of freedom equals $3n-m-6$; when it is non-positive, the face shape is fixed.

n-gon	node(n)	bar(m)	remaining DoF
3	3	3	0
4	4	6	0
$4 < n$	n	$2n$	$n-6$

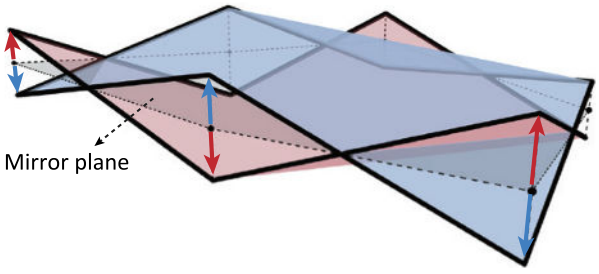


Fig. 5. A face shares equivalent values of edges and angle edges with its mirror image, but their shapes are different. This difference is caused by the opposite directions in which the vertices deviate from the mirror plane.

one that minimizes the sum of the squared distances to all its vertices. For faces within the same group, the normal directions of their best-fit planes must be unified with the group centroid.

Finally, based on the above analysis, faces will be congruent (for cases with $n \leq 6$) if they contain equivalent values of edges, angle edges, and signed distances from vertices to the best-fit planes. The corresponding similarity term F_s can be formulated as follows:

$$F_s = \sum_{i=1}^{F} \sum_{j=1}^{n_i} \left((l_{ij} - \bar{l}_{ij})^2 + (l'_{ij} - \bar{l}'_{ij})^2 + (h_{ij} - \bar{h}_{ij})^2 \right) \tag{2}$$

where F is the number of faces; for the ith face, n_i is the number of edges, l_{ij} and l'_{ij} are the jth edge length and angle edge length, respectively, and h_{ij} is the signed distance from the jth vertex to the best-fit plane; and $\bar{l}_{ij}$, $\bar{l}'_{ij}$ and $\bar{h}_{ij}$ are their respective mean values of the corresponding group. Specifically, to calculate h_{ij} for vertex V_{ij} in face F_i, one needs to find the best-fit plane P_i of F_i and determine the closest point $V^\star_{ij}$ of V_{ij} on P_i. Then, h_{ij} can be computed as the signed distance from V_{ij} to $V^\star_{ij}$. However, during this process, P_i and $V^\star_{ij}$ typically introduce strong nonlinearity into h_{ij}, making the optimization difficult to solve. Therefore, to simplify computation, P_i and $V^\star_{ij}$ are treated as constants during optimization, with their values updated at the beginning of each iteration based on the current geometry (using the Rhino built-in function *FitPlaneToPoints* [39]). In doing so, h_{ij} is simplified as the signed distance from V_{ij} to a predefined position. This converts F_s into a quadratic form, enabling it to be solved using standard gradient-based methods.

2.3.2. Approximating the target surface

In architectural applications, it is often required that the generated mesh closely approximates the given target surface. This can be achieved by requiring each vertex to be close to the target surface, vertices on the boundary (boundary vertex) to be close to the boundary curves, and corner vertices to remain in their original positions. These corner vertices can be identified by their corresponding turning angles θ, as illustrated in Fig. 6. In this study, we designate boundary vertices as corner vertices if their corresponding turning angles θ are greater than 1. The closeness term F_c can be formulated as:

$$F_c = \omega_{sur} \sum_{i=1}^{V} {d_{si}}^2 + \omega_{bou} \sum_{j=1}^{V_{bou}} {d_{bj}}^2 + \omega_{cor} \sum_{k=1}^{V_{cor}} {d_{ck}}^2 \tag{3}$$

where d_{si}, d_{bj}, and d_{ck} denote the distances from each vertex to the target surface, each boundary vertex to the boundary curves, and each

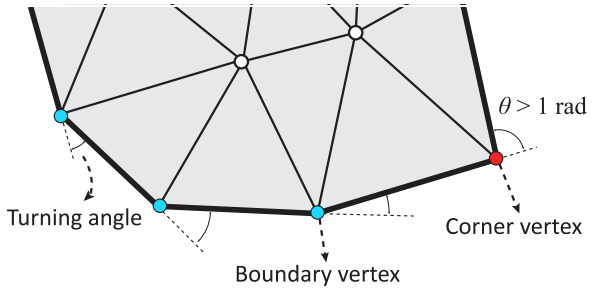


Fig. 6. Corner vertices are distinguished from boundary vertices based on the turning angle θ. If $\theta > 1$, the corresponding boundary vertex is designated as a corner vertex.

corner vertex to its initial position, respectively. ω_{sur}, ω_{bou}, and ω_{cor} are the corresponding weights for each term. The total number of vertices, boundary vertices, and corner vertices are denoted by V, V_{bou}, and V_{cor}, respectively. In this formulation, d_{si} and d_{bj} are highly nonlinear in relation to vertex positions. To simplify computation, in optimization, we first identify the closest point from each vertex to the target object. Specifically, the closest points from vertices to the target surface (given as a mesh) [40] and to the boundary curves (given as polylines) are computed using the Rhino built-in functions *Mesh.ClosestPoint* [41] and *Polyline.ClosestPoint* [42], respectively. These closest points are then treated as constants, updated at each iteration, and assigned as the target positions for their corresponding vertices. As a result, minimizing F_c becomes the minimization of distances from vertices to predefined target positions. This redefined objective involves a quadratic formulation that offers feasible solutions.

2.3.3. The global objective function

Combining all subterms (F_s and F_c) gives the global objective function F_g:

$$F_g = \omega_s F_s + \omega_c F_c \tag{4}$$

where ω_s and ω_c are the corresponding weights. Users can control the bias of optimization toward each subgoal by adjusting the values of the weights. In this study, we set $\omega_s = \omega_c = 1$ as the default values, unless otherwise specified. During optimization, adjacent faces are kept connected, with the geometry remaining a mesh throughout the whole process. The design variables are the positions of mesh vertices. Moving each vertex simultaneously changes the shapes of all its adjacent faces. The movement of each vertex is determined by the resultant effects of all its adjacent faces (related to F_s) and the vertex-wise closeness constraint (related to F_c). All optimization problems are solved using the nonlinear conjugate gradient method [43]. Minimizing F_g transforms the input mesh toward the user-defined geometric goals.

3. Results

In this section, we assemble the proposed clustering and optimization techniques to solve multiple problems. We begin in Section 3.1 by defining two error metrics to assess the method performance. Then, in Section 3.2, we examine the effectiveness of the optimization strategy for achieving congruent shapes via benchmark tests. Section 3.3 investigates the impact of input parameters on outcomes and explores an iterative strategy that may enhance results. The versatility of our method in handling general n-gons is demonstrated in Section 3.4 by testing diverse models with various mesh patterns. Finally, Section 3.5 offers a comparative analysis with a relevant approach focused on triangular meshes.

3.1. Implementation details

Two error metrics, δ_s and δ_c, are introduced to assess algorithm performance. The similarity error, δ_s, is evaluated by normalizing $d_{\max}$ with the average edge length of the input mesh, where $d_{\max}$ is the largest similarity metric (as defined in Section 2.2.1) between faces and their respective centroids. The closeness error, δ_c, represents the maximum distance from vertices to the target surface. The interpretation of distance varies for different vertex types: inner vertices are measured with respect to the surface, boundary vertices to boundary curves, and corner vertices to their initial positions. For comparison, all models are uniformly scaled, with the maximum side lengths of their respective bounding boxes set to one.

The algorithm has been scripted as a plugin using C# codes with double precision in the Rhino–Grasshopper CAD platform. All numerical results are obtained on an ordinary personal computer with an i7-7700HQ Intel core. It is worth noting that our method is a numerical approach, which cannot produce exact solutions but only results with numerical errors. Nevertheless, for convenience, faces that are highly similar with error δ_s below 1E-13 (close to the machine error) are referred to as "congruent faces" in the following discussions.

3.2. Method validation

A series of examples are first tested to examine the effectiveness of the proposed optimization strategy F_s (see Section 2.3.1), as shown in Fig. 7. For each test, the objective is to optimize faces of the same topology into congruent shapes ($\omega_s = 1$, $\omega_c = 0$). In this figure, the superimposed faces (obtained as described in the second step of the clustering iteration) are given at the bottom of each geometry.

In (a), we have two connected right triangles with the same edge lengths and orientations. If flipping is permitted, the input faces can be perfectly superimposed, indicating they are already congruent and thus requiring no optimization. However, if the face orientation needs to be preserved (i.e., not allowing flipping), which may occur in real projects, the superimposed input faces cannot match. Here, optimization is necessary; eventually, the input right triangles are successfully converted into congruent isosceles triangles, with their orientations remaining unchanged.

In (b), we discuss two hexagonal faces that are mirror images of one another. We first only equalize the lengths of edges and angle edges; after optimization, the shapes remain unchanged and incongruent, as expected (see Section 2.3.1). Then, we further equalize the signed distances from vertices to their best-fit planes. This successfully optimizes the faces into congruent shapes that are also planar. Here, planarity is caused by the average signed distance being equal to zero, as the distances of mirror images only differ in sign but have the same magnitude.

In (c), we test three Platonic solids and two Archimedean solids, which are closed meshes composed of different n-gons. Specifically, the two Archimedean solids contain faces with $n > 6$, which are solved by additionally constraining the distances between the ith and $(i + 3)$-rd vertices within each face based on Eq. (2) (as described in Section 2.3.1). In all cases, we start from random geometries that are topologically equivalent to the target solids but contain faces mostly in different shapes. After optimization, with the preserved mesh connectivity, faces of the same group seamlessly transform into congruent regular polygons, achieving the intended Platonic/Archimedean solids.

3.3. Numerical investigation

3.3.1. Effects of the input parameters

This section investigates the effects of input parameters on the final outcome, as shown in Fig. 8. The required input parameters for our algorithm are the weight values for achieving congruent shapes (ω_s) and approximating the target surface (ω_c), and a target group

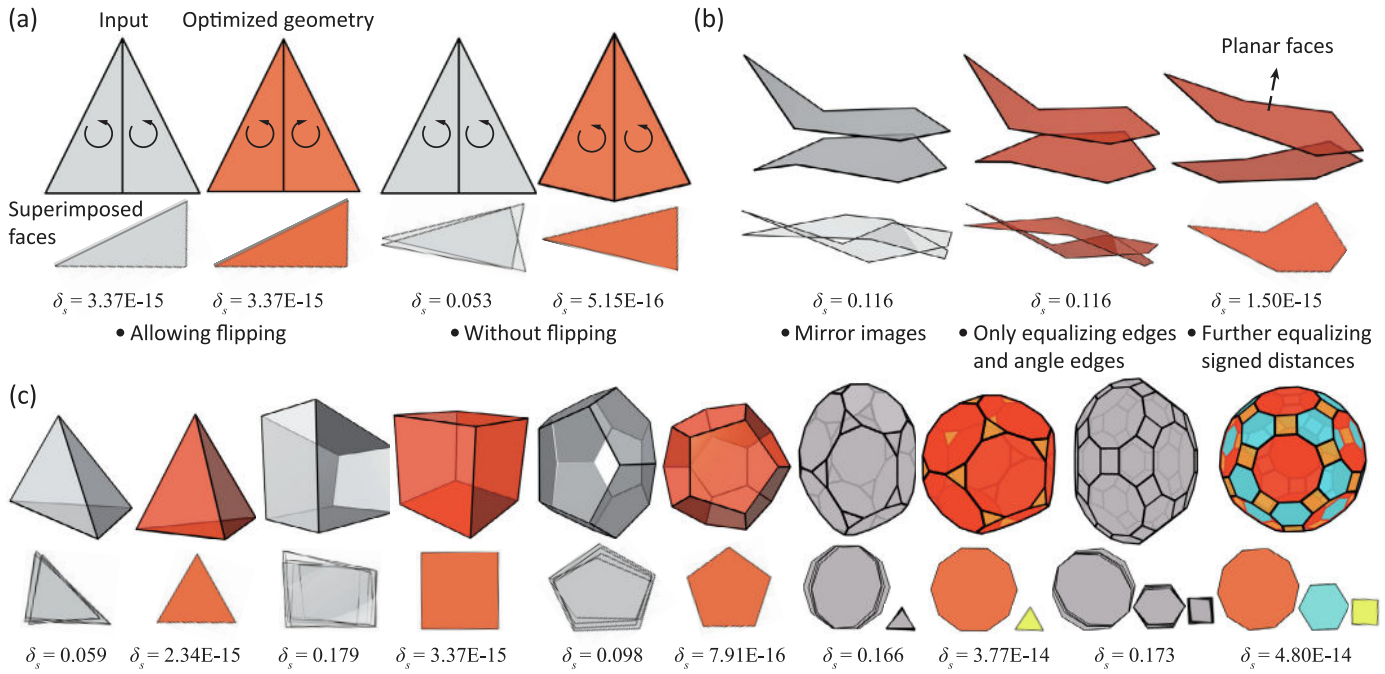


Fig. 7. Method validation: for each input geometry, we seek to optimize faces of the same topology into congruent shapes without other constraints ($\omega_s = 1$, $\omega_c = 0$). The input geometries include (a) two connected right triangles with identical edge lengths and orientations, (b) two separate hexagonal faces that are mirror images of one another, and (c) closed n-gonal meshes that are topologically equivalent to Platonic or Archimedean solids but with variously shaped faces. Here, all numerical tests are completed within 10 s.

number k. The investigation is based on a simple free-form geometry, as shown in Fig. 8(a). Three different scenarios are discussed, with the corresponding results detailed in Fig. 8(b). Faces within the same group are superimposed below the corresponding output mesh. In each scenario, the target group number k is set to 1, 3, and 8, respectively.

We start with $\omega_s = \omega_c = 0$ in Scenario 1, where the faces of the input mesh are only clustered into different groups with no shape optimization. In this case, the closeness error δ_c is always zero because the input mesh is built on the target surface. The similarity error δ_s decreases as the group number increases since the most different shapes are filtered into the new groups. However, here, the face shapes remain unchanged; the similarity error δ_s is thus highly dependent on the input mesh.

In Scenario 2, we include optimization to transform faces toward congruence ($\omega_s = 1$ and $\omega_c = 0$) based on the clustering result. In the obtained geometries, faces within the same group are successfully optimized into congruent shapes, but the overall geometry is largely modified. For instance, when $k = 1$, the input geometry is converted into a barrel-like vault, losing the features of double curvature, with the boundaries also distorted into a zigzag pattern. Although increasing the group number reduces shape distortion, the produced shape is difficult to anticipate, and users still lack control over the profile of the final geometry.

In Scenario 3, an additional optimization goal is added to approximate the target surface ($\omega_s = 1$ and $\omega_c = 1$). In the generated geometries, the overall shapes are better preserved, including the boundaries, with δ_c significantly decreased. However, due to the imposed extra constraints, faces within the same group cannot achieve congruent shapes. One can manipulate the weight values to control the errors, but decreasing one error typically increases the other. For real fabrication, it is often desired to maintain δ_c at a reasonably small value and decrease δ_s as much as possible; in this case, one generally needs to increase the group number.

3.3.2. Clustering–optimization iteration

This subsection discusses an iterative clustering–optimization strategy that may further reduce the similarity error without needing more groups. In Scenario 3, we optimize for congruent shapes based on the clustering result of the input faces. When optimization reaches convergence, faces have been partially transformed toward their respective centroids. At this point, faces are different from the initial shapes; certain faces may be closer to other centroids compared to their current ones. Thus, allowing faces to move across groups may further reduce the similarity error. To take advantage of this possibility, we propose an iterative clustering–optimization strategy: every time the optimization is converged, we recluster the obtained faces and run another optimization until the clustering result remains unchanged. The effectiveness of this strategy is examined based on the geometry obtained in Scenario 3 with $k = 8$. After the first optimization, we continue the clustering–optimization iteration, as shown in Fig. 8(c). It can be seen that the clustering result is changed after reclustering the configuration at the end of the first optimization, and reaches convergence after three additional iterations. The final similarity error ($\delta_s = 0.019$) is reduced by 27% compared to the result of the first optimization ($\delta_s = 0.026$). For subsequent numerical tests in this paper, to prevent complex cases requiring excessively long runtime to converge, the maximum iteration number is limited to 5.

3.4. Applicability to general n-gons

Our method has broad applicability to meshes composed of general n-gons, as shown in Fig. 9. Diverse models with various mesh patterns are presented, with their detailed data listed in Table 2. In the generated geometries, faces are replaced by their respective group centroids. This introduces gaps and overlaps between adjacent faces due to the remaining similarity errors. Yet, as illustrated in Fig. 9(a), the similarity error can be effectively reduced by increasing the group number. Gaps and overlaps are correspondingly mitigated, even to an unnoticeable level. In Figs. 9(b) and (c), we investigate geometries that contain inherent symmetries. Results suggest that our method can adeptly identify the symmetry (even in cases less perceptible to the human eye), hence allowing fair approximations of these geometries with just a few groups of distinct shapes. However, for general free-form geometries (see Figs. 9(d–f)), reducing the group number is more challenging; a moderate number of groups is typically needed to achieve relatively low similarity errors.

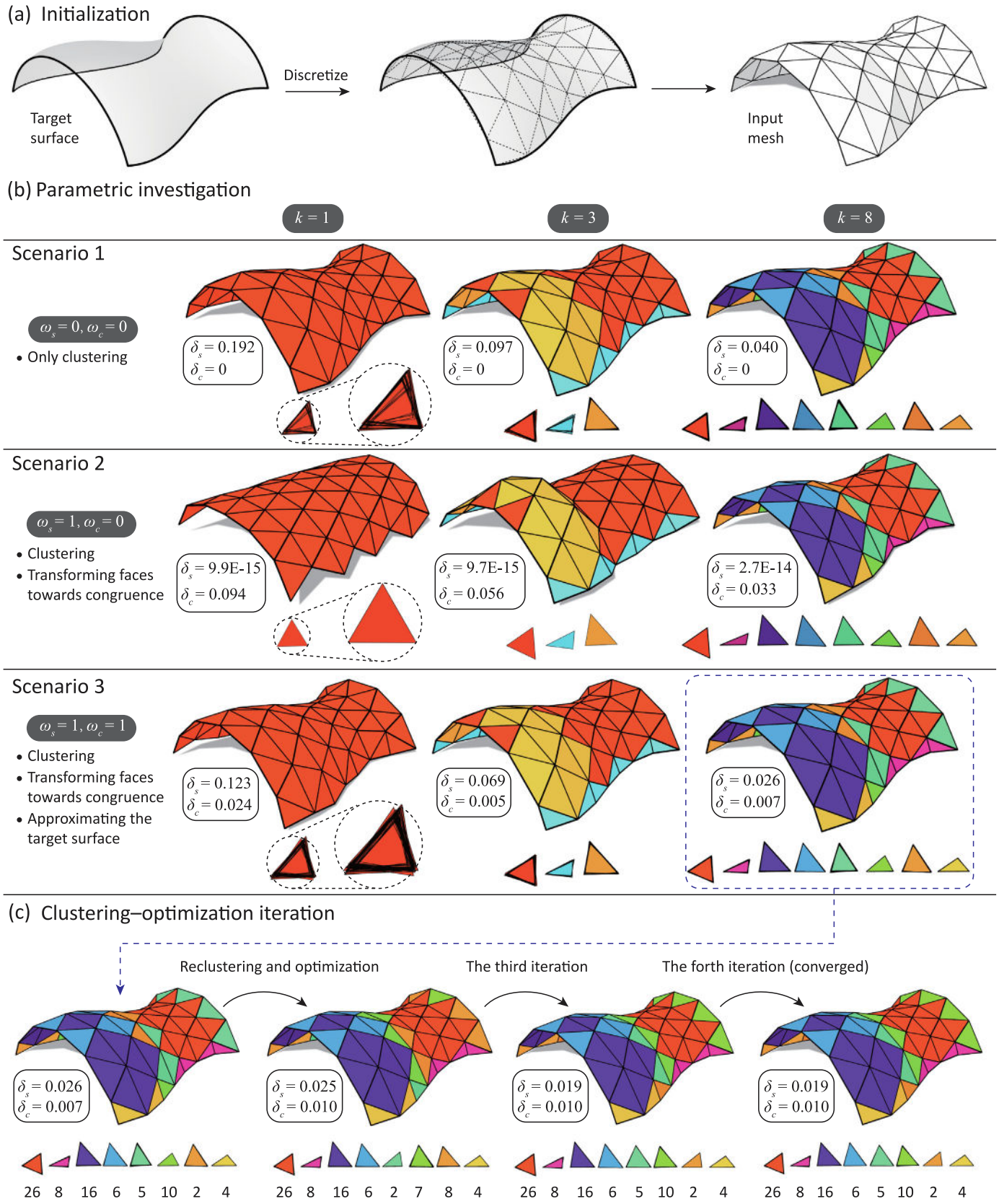


Fig. 8. Numerical investigation. (a) The input mesh generated by discretizing a given free-form surface. (b) Investigating the effects of the input parameters ω_s, ω_c, and k on the generated geometries and errors (δ_s and δ_c). (c) A proposed iterative clustering–optimization strategy, which reclusters the faces based on the optimized configuration of the last step, followed by another optimization until reaching convergence, i.e., the clustering results remain unchanged. All numerical examples in this figure are finished within 15 s.

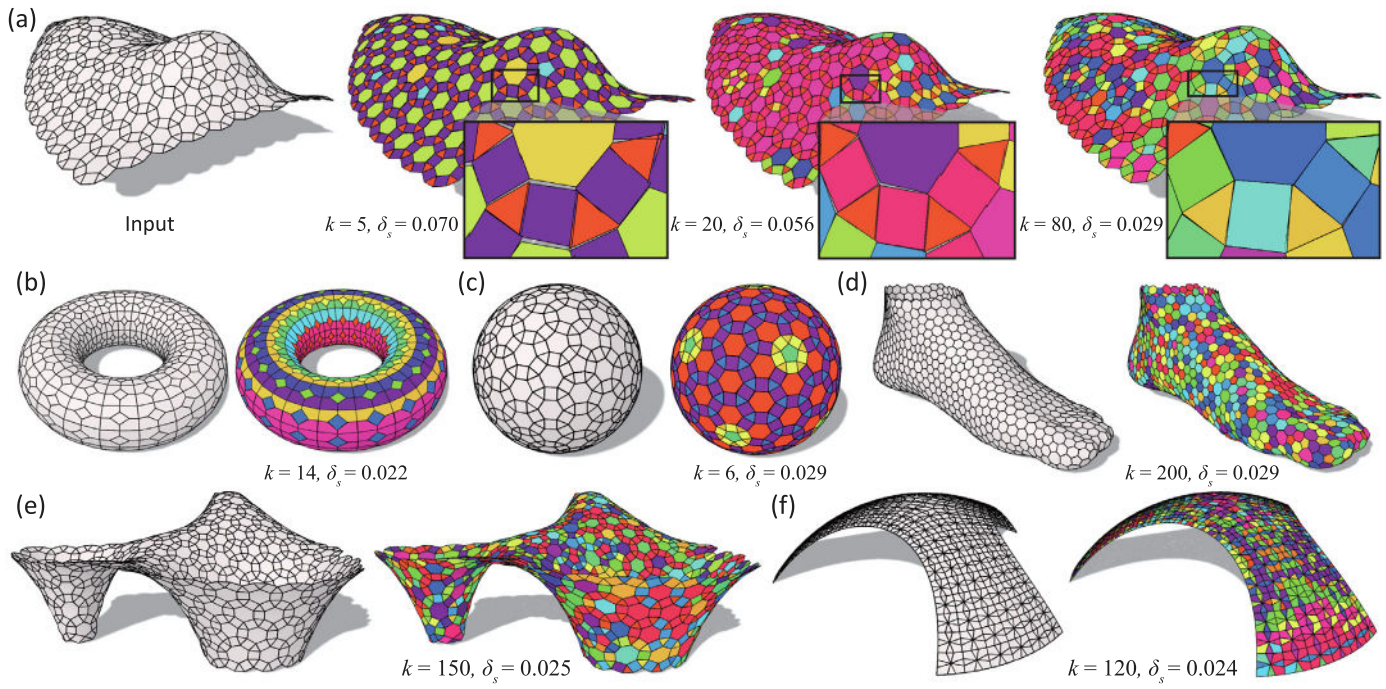


Fig. 9. Our method has wide applicability to meshes composed of general n-gons. (a) By increasing the group number, we can effectively reduce the similarity error, hence mitigating the gaps and overlaps. (b and c) The method can identify the inherent symmetry of input geometries, allowing fair approximations with only a few groups of distinct shapes. (d–f) General free-form geometries typically require a moderate number of groups to achieve a relatively low similarity error.

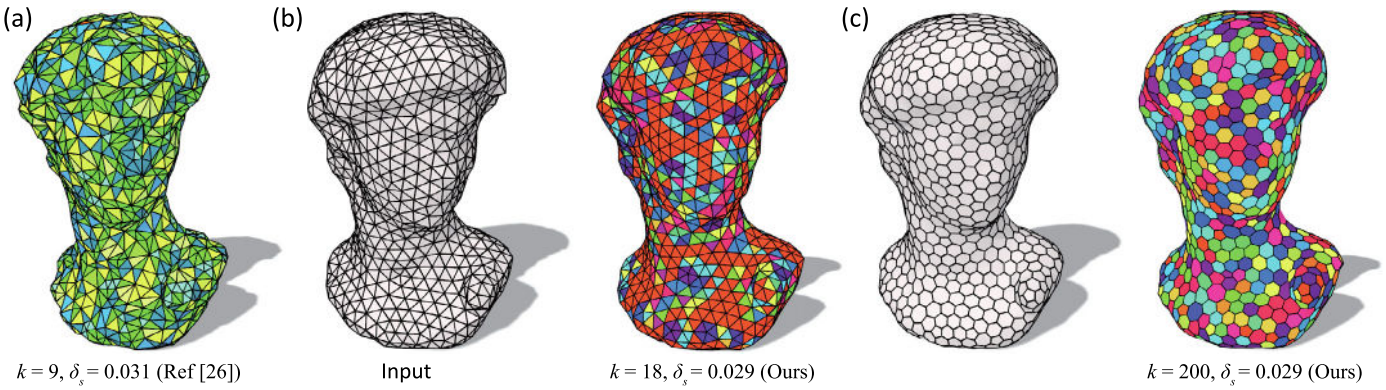


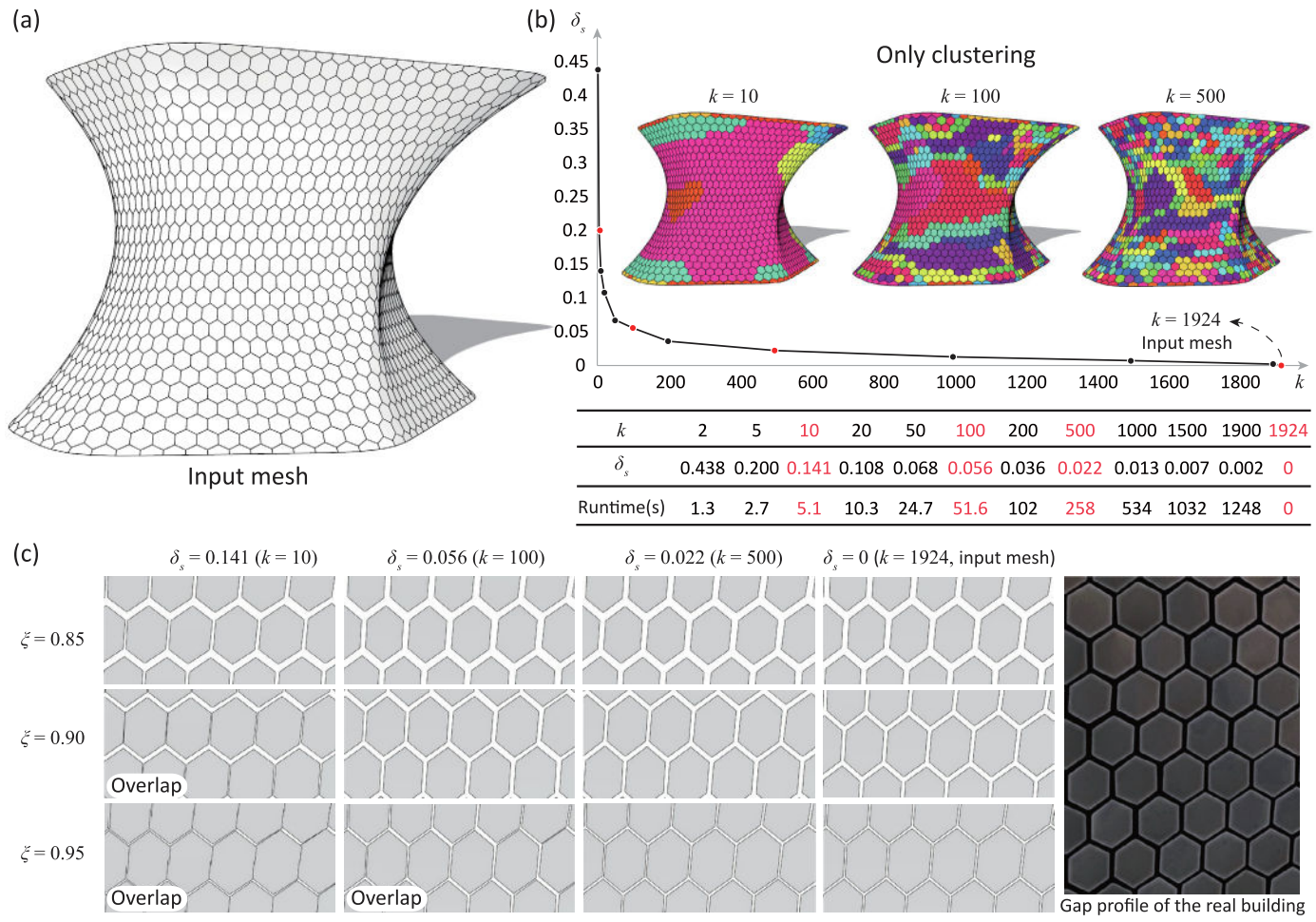
Fig. 10. Comparison results. (a) The triangular mesh created by [26], with the similarity error evaluated using our definition. (b) We create an isotropic mesh based on (a) with a close face amount, and take it as the input for our method to generate a solution with a comparable similarity error. (c) Our method can also deal with the dual of the isotropic mesh, which contains mixed 5, 6, and 7-gons.

Table 2
Detailed data of numerical examples. F is the face number, k is the group number, δ_s is the similarity error, and δ_c is the closeness error.

Model	F	k	δ_s	δ_c	Runtime(min)
9(a)	725	5	0.070	0.006	0.3
9(a)	725	20	0.056	0.004	0.5
9(a)	725	80	0.029	0.002	0.6
9(b)	1,000	14	0.022	0.001	3.1
9(c)	722	6	0.029	0.004	0.3
9(d)	1,121	200	0.029	0.009	2.7
9(e)	1,260	150	0.025	0.001	3.1
9(f)	1,656	120	0.024	0.001	6.8
10(b)	2,068	18	0.029	0.005	0.9
10(c)	1,036	200	0.029	0.015	3.4

3.5. Comparison

We compare our method with a relevant study that focuses on triangular meshes [26], as shown in Fig. 10. Both approaches ([26] and ours) aim to achieve surface approximations with a limited number of different faces. Their method realizes diverse free-form shapes using a specified set of template triangles by reconfiguring mesh topology. Conversely, our method reduces the number of different faces within given input meshes by adjusting vertex positions with preserved topology. For comparison, we start with a mesh created using their method (see Fig. 10(a)), containing 2,074 faces with only 9 distinct shapes. The similarity error evaluated using our definition (see Section 3.1) is δ_s = 0.031. Then, we apply a remeshing technique [44] to create an isotropic mesh with a comparable face amount (i.e., 2,068). The isotropic mesh contains faces that are already similar, making it a suitable high-quality



k	2	5	10	20	50	100	200	500	1000	1500	1900	1924
δ_s	0.438	0.200	0.141	0.108	0.068	0.056	0.036	0.022	0.013	0.007	0.002	0
Runtime(s)	1.3	2.7	5.1	10.3	24.7	51.6	102	258	534	1032	1248	0

Fig. 11. Results obtained by only clustering the faces of the initial mesh, as per the original design approach. (a) The recreated input mesh. (b) The relationship between the similarity error δ_s and the group number k. (c) Local gap profiles of different combinations of δ_s and ζ, and that of the real building.

input for our approach. Based on this input, our method achieves 18 groups of different triangles to attain a comparable similarity error (see Fig. 10(b)). As indicated by the results, their method typically requires fewer groups of distinct shapes, due to the careful design of the mesh topology. However, their method only deals with triangular meshes, while our approach is more versatile, capable of solving general n-gons (see Fig. 10(c)). Moreover, their method tends to substantially modify the input mesh, while our method maintains topology and subtly adjusts geometry, hence serving as a more fitting choice for scenarios demanding specified patterns (like semi-regular, Penrose, Islamic, or other patterns) to achieve intended aesthetic effects.

4. Case study: Soumaya museum

In this study, we revisit the façade design of the Soumaya Museum based on an input mesh (Fig. 11(a)) that contains 1,924 pentagonal and hexagonal faces, each with a distinct shape. The original approach [29, 32] for reducing the shape variety is to cluster all panels into a specified number of groups, with each replaced by the group centroid. Here, we first follow the original approach by performing clustering alone ($\omega_s = \omega_c = 0$). The relationship between the group number k and the similarity error δ_s is detailed in Fig. 11(b). It can be seen that clustering alone can reduce δ_s and is especially effective when the group number is small, as δ_s drops rapidly when k increases. However, after a certain point (around $k = 150$), δ_s decreases much more slowly; a small decrease in δ_s requires a large number of additional groups. This is because faces are already similar in each group, and the most representative shapes are already identified; increasing the group number will only add details, which has a subtle influence on reducing the error. Fig. 11(c) presents local gap profiles after replacing each panel with its corresponding group centroid. To prevent overlaps, a simple approach is to uniformly scale the panels by a factor of ξ based on their average vertex positions. This approach offers an initial estimate of the gaps during the early design stage. However, the actual gap distribution could be highly project-specific and require further fine-tuning of the panel positions based on multiple practical considerations. This is beyond the scope of the present study and will be further discussed in Section 5. For the current case study, we select "$\delta_s = 0.022$ and $\xi = 0.90$" as a reference due to its visual resemblance to the real gap profile. Now, the required group number by clustering alone is $k = 500$. To reach the same similarity error with fewer groups, we incorporate the proposed optimization strategy, as demonstrated below.

Given a specified error threshold, the minimal required group number can be identified using the bisection method [45], as shown in Fig. 12. We start with a group number interval of [200, 500] and gradually reduce it to a tolerance of 10 by adaptively updating the upper or lower bound. The final group number converges to $k = 285$, with a total runtime of 71.7 min. In the obtained mesh, the similarity error is $\delta_s = 0.022$, satisfying the given threshold. Vertices remain close to the target surface ($\delta_c = 0.001$), indicating that the optimized geometry well approximates the intended shape. The final group number ($k = 285$) achieves a 43.0% decrease from 500 (clustering alone) and a significant 85.2% reduction from the original 1,924 input faces, as shown in Fig. 13. Due to the shape optimization, vertices are shifted on the target surface, leading to a slight modification of the panel shapes.

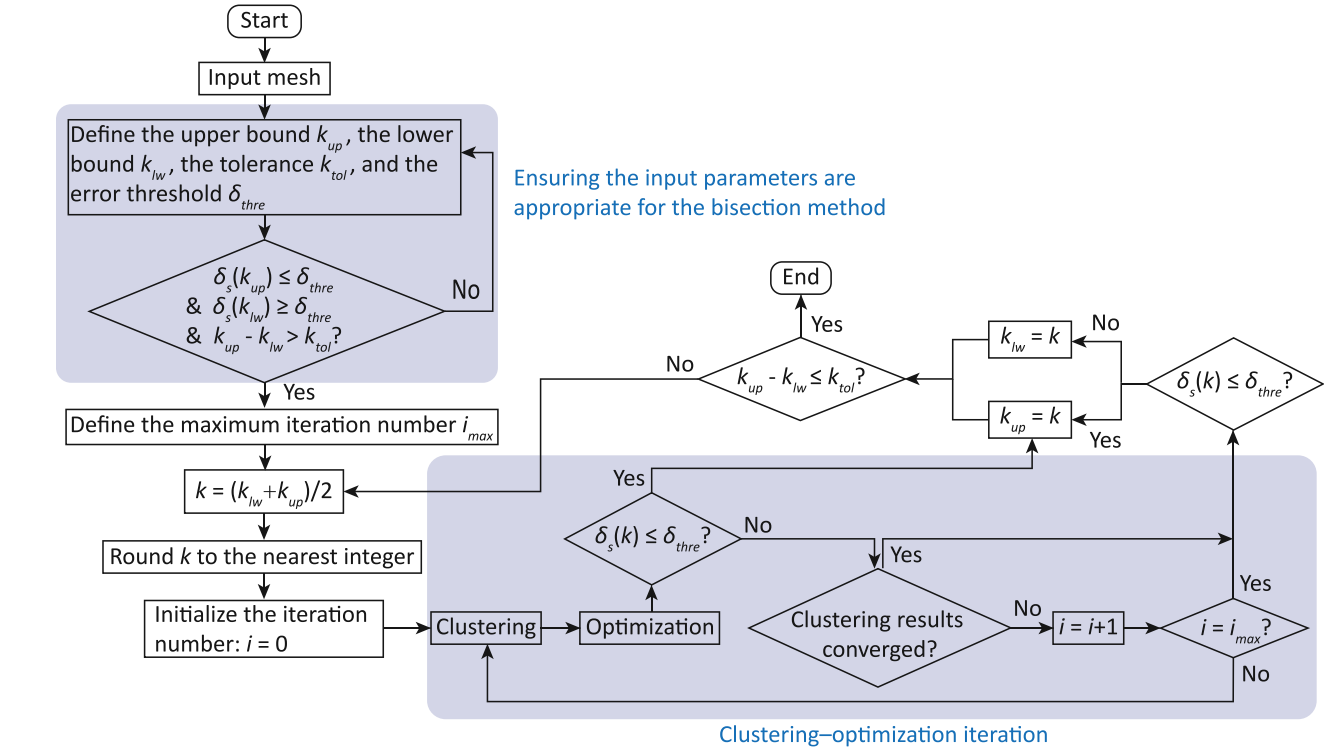


Fig. 12. The proposed framework for determining the required group number with respect to a specified error threshold.

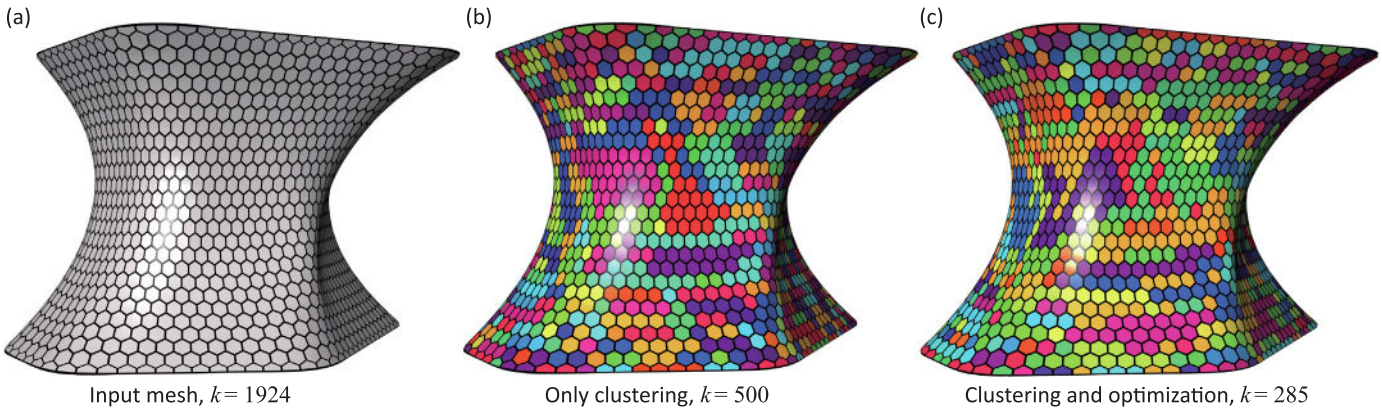


Fig. 13. The comparison among (a) the input mesh, (b) the geometry obtained by only clustering, and (c) the optimized geometry. The final group number ($k = 285$) achieves a 43.0% decrease from 500 (clustering alone) and only accounts for 14.8% of 1,924 (the input mesh).

For cases where the shape modification is considered undesired, users can increase the group number or add additional optimization goals to account for the panel shapes.

5. Discussion

The proposed method has potential practical application in the façade design of free-form structures. Given a user-defined error threshold, our method can create a mesh with a reduced number of different faces while approximating the target surface. Mesh faces can be replaced with their respective group centroids to achieve exactly congruent panels. Uniformly scaling the panels can avoid overlaps and create an initial estimate of the gaps at the early design stage. However, further adjustment of the gaps is still required to meet practical considerations, such as facilitating installation, accommodating thermal expansion, and creating desired aesthetic effects [46,47]. To solve this problem, optimization techniques can be employed. Specifically, the practical considerations first need to be formulated as explicit geometric constraints. The design created using our method can be taken as the initialization. The six degrees of freedom (i.e., translations and rotations) of each panel and the scaling factor of each group could be considered as the design variables. Standard gradient-based methods could be used as the solver, while special techniques may be required for complex constraints that are highly nonlinear or discrete. By solving this optimization problem, the created design could better align with the specific practical needs while maintaining a small number of different panel shapes.

For future research, the clustering–optimization method may be extended in several aspects. First, besides the input parameters discussed in Section 3.3.1, the final result also depends on the input mesh. In this study, the input mesh is assumed to be provided by users, with the topology remaining unchanged. A possible research direction is to improve the input mesh through topological operations, such as specifically designed remeshing techniques [26,48]. Second, a more

sophisticated clustering method may further improve the final outcome. The current strategy is based on the face shapes without considering the subsequent optimization. After clustering, the corresponding edges, angles, and signed distances are expected to be equivalent, and vertices are expected to stay on the target surface. These constraints may easily be in conflict, resulting in a compromising solution. In this regard, pre-considering the imposed constraints during clustering to reduce the possible conflicts may improve the optimization outcome. Finally, in terms of runtime, our method is not highly efficient for large-scale problems with many panels. In the case of the Soumaya Museum, to precisely determine the required group number up to a specific error threshold, a large number of (around 20) iterations of clustering and optimization are involved, leading to a lengthy runtime of 71.7 min. However, for general usage of our algorithm, where users interactively adjust the group number based on the resultant geometry, the runtime for each trial is much shorter. Notably, the simple cases in Figs. 7 and 8 are all finished within 15 s. Even for the complex geometries in Figs. 9 and 10, each case only takes a few minutes, as shown in Table 2. Nevertheless, for future research, it is worth further improving efficiency to enhance the interactive experience, such as by integrating parallel computing techniques [49].

6. Conclusion

In this study, we present a new method that interleaves clustering and optimization to reduce the number of different faces in free-form surface approximations composed of general n-gons. Specifically, a similarity metric is defined to quantify the geometric closeness between different faces. An adaptive clustering method is developed to partition mixed different n-gonal faces into a specified number of groups of similar shapes. An optimization strategy is proposed that can transform geometrically different n-gonal faces toward congruent shapes, as validated through a series of numerical tests. By constraining vertex deviations from the target surface, the generated mesh can remain close to the intended shape, which is often desired in architectural applications. However, when multiple geometric goals are included, the optimization may only converge to a compromising solution, with each goal merely partially achieved. Nevertheless, an iterative clustering–optimization strategy is proposed that may enhance outcomes without increasing the group number. The effectiveness of our method is demonstrated through diverse geometries with various mesh patterns.

Moreover, our method is tested based on the case study of an actual architectural project, the Soumaya Museum. We first revisit the original design approach, where only clustering is performed. We find that clustering alone can reduce the similarity error and is especially effective when the group number is small, but it is generally less useful for achieving very low errors. To enhance the result, we present a framework that further incorporates optimization to determine the minimal required group number while satisfying a user-specified error threshold. By using this approach, the group number can be significantly reduced by 43%, from 500 (clustering alone) to 285 (with optimization), accounting for only 14.8% of 1924 (the total face number). Overall, the proposed method holds potential practical application in the façade design of free-form structures for producing cost-effective solutions.

CRediT authorship contribution statement

Yuanpeng Liu: Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Ting-Uei Lee:** Methodology, Visualization, Writing – review & editing. **Anooshe Rezaee Javan:** Methodology, Writing – review & editing. **Nico Pietroni:** Methodology, Writing – review & editing. **Yi Min Xie:** Supervision, Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Australian Research Council (FL190100014 and DP200102190).

References

[1] Adriaenssens S, Block P, Veenendaal D, Williams C. Shell structures for architecture: form finding and optimization. Routledge; 2014, ISBN 978-0415840606.
[2] Li Q, Su Y, Wu Y, Borgart A, Rots JG. Form-finding of shell structures generated from physical models. Int J Space Struct 2017;32(1):11–33. http://dx.doi.org/10.1177/026635111769657.
[3] Wang H, Pellis D, Rist F, Pottmann H, Müller C. Discrete geodesic parallel coordinates. ACM Trans Graph 2019;38(6):1–13. http://dx.doi.org/10.1145/3355089.3356541.
[4] Austern G, Capeluto IG, Grobman YJ. Rationalization methods in computer aided fabrication: A critical review. Autom Constr 2018;90:281–93. http://dx.doi.org/10.1016/j.autcon.2017.12.027.
[5] Pottmann H, Asperl A, Hofer M, Kilian A. Architectural geometry. Exton, Pennsylvania: Bentley Institute Press; 2007, ISBN 978-1934493045.
[6] Pottmann H, Eigensatz M, Vaxman A, Wallner J. Architectural geometry. Comput Graph 2015;47:145–64. http://dx.doi.org/10.1016/j.cag.2014.11.002.
[7] Liu Y, Pottmann H, Wallner J, Yang Y-L, Wang W. Geometric modeling with conical meshes and developable surfaces. ACM Trans Graph 2006;25(3):681–9. http://dx.doi.org/10.1145/1141911.1141941.
[8] Huang W, Wu C, Hu J, Gao W. Weaving structure: A bending-active gridshell for freeform fabrication. Autom Constr 2022;136:pp. 104184. http://dx.doi.org/10.1016/j.autcon.2022.104184.
[9] Schling E. Repetitive structures [Ph.D. thesis], Technische Universität München; 2018.
[10] Castañeda E, Lauret B, Lirola J, Ovando G. Free-form architectural envelopes: Digital processes opportunities of industrial production at a reasonable price. J Facade Design Eng 2015;3(1):1–13. http://dx.doi.org/10.3233/FDE-150031.
[11] Fu C-W, Lai C-F, He Y, Cohen-Or D. K-set tilable surfaces. ACM Trans Graph 2010;29(4):1–6. http://dx.doi.org/10.1145/1778765.1778781.
[12] Brownell P, Merchant KA. The budgetary and performance influences of product standardization and manufacturing process automation. J Account Res 1990;28(2):388–97. http://dx.doi.org/10.2307/2491156.
[13] Baud-Lavigne B, Agard B, Penz B. Mutual impacts of product standardization and supply chain design. Int J Prod Econ 2012;135(1):50–60. http://dx.doi.org/10.1016/j.ijpe.2010.09.024.
[14] Koronaki A, Shepherd P, Evernden M. Rationalization of freeform space-frame structures: Reducing variability in the joints. Int J Architect Comput 2020;18(1):84–99. http://dx.doi.org/10.1177/14780771198948.
[15] Brütting J, Senatore G, Fivet C. Design and fabrication of a reusable kit of parts for diverse structures. Autom Constr 2021;125:pp. 103614. http://dx.doi.org/10.1016/j.autcon.2021.103614.
[16] Lu H, Xie YM. Reducing the number of different members in truss layout optimization. Struct Multidiscip Optim 2023;66(3):52. http://dx.doi.org/10.1007/s00158-023-03514-y.
[17] Lee T-U, Liu Y, Xie YM. Dividing a sphere hierarchically into a large number of spherical pentagons using equal area or equal length optimization. Comput Aided Des 2022;148:103259. http://dx.doi.org/10.1016/j.cad.2022.103259.
[18] Liu Y, Lee T-U, Rezaee Javan A, Xie YM. Extending goldberg's method to parametrize and control the geometry of goldberg polyhedra. Royal Soc Open Sci 2022;9(8):pp. 220675. http://dx.doi.org/10.1098/rsos.220675.
[19] Zimmer H, Lafarge F, Alliez P, Kobbelt L. Zometool shape approximation. Graph Models 2014;76(5):390–401. http://dx.doi.org/10.1016/j.gmod.2014.03.009.
[20] Liu Y, Lee T-U, Koronaki A, Pietroni N, Xie YM. Reducing the number of different nodes in space frame structures through clustering and optimization. Eng Struct 2023;284:116016. http://dx.doi.org/10.1016/j.engstruct.2023.116016.
[21] Lobel A. Lobel frame. 2005, URL https://www.equilatere.net. [Accessed on 17 February 2023].

[22] Huard M, Eigensatz M, Bompas P. Planar panelization with extreme repetition. In: Advances in architectural geometry 2014. Cham, Switzerland: London, England. Springer; 2015, p. 259–79. http://dx.doi.org/10.1007/978-3-319-11418-7_17, September 2014.
[23] Jiang C, Tang C, Tomičí M, Wallner J, Pottmann H. Interactive modeling of architectural freeform structures: Combining geometry with fabrication and statics. In: Advances in architectural geometry 2014. Cham, Switzerland: London, England. Springer; 2015, p. 95–108. http://dx.doi.org/10.1007/978-3-319-11418-7_7, September 2014.
[24] Eigensatz M, Kilian M, Schiftner A, Mitra NJ, Pottmann H, Pauly M. Paneling architectural freeform surfaces. ACM Trans Graph 2010;29(4):1–10. http://dx.doi.org/10.1145/1778765.1778782.
[25] Singh M, Schaefer S. Triangle surfaces with discrete equivalence classes. ACM Trans Graph 2010;29(4):1–7. http://dx.doi.org/10.1145/1778765.1778783.
[26] Liu Z, Zhang Z, Zhang D, Ye C, Liu L, Fu X-M. Modeling and fabrication with specified discrete equivalence classes. ACM Trans Graph 2021;40(4):1–12. http://dx.doi.org/10.1145/3450626.3459843.
[27] Li Y, Liu Y, Wang W. Planar hexagonal meshing for architecture. IEEE Trans Vis Comput Graphics 2015;21(1):95–106. http://dx.doi.org/10.1109/TVCG.2014.2322367.
[28] Jiang C, Tang C, Vaxman A, Wonka P, Pottmann H. Polyhedral patterns. ACM Trans Graph 2015;34(6):1–12. http://dx.doi.org/10.1145/2816795.2818077.
[29] Sidelko J. Museo Soumaya: facade design to fabrication. Lulu. com; 2013, ISBN 978-1622098507.
[30] Meekes M, Vaxman A. Unconventional patterns on surfaces. ACM Trans Graph 2021;40(4). http://dx.doi.org/10.1145/3450626.3459933.
[31] Vaxman A, Müller C, Weber O. Regular meshes from polygonal patterns. ACM Trans Graph 2017;36(4). http://dx.doi.org/10.1145/3072959.3073593.
[32] Peña de Leon A. Rationalisation of freeform façades: A technique for uniform hexagonal panelling. In: Proceedings of the 17th international conference on computer aided architectural design research in Asia. 2012, p. 243–52. http://dx.doi.org/10.52842/conf.caadria.2012.243.
[33] Sorkine-Hornung O, Rabinovich M. Least-squares rigid motion using SVD. 2017, URL https://igl.ethz.ch/projects/ARAP/svd_rot.pdf. [Accessed on 17 February2023].
[34] Eldar Y, Lindenbaum M, Porat M, Zeevi YY. The farthest point strategy for progressive image sampling. IEEE Trans Image Process 1997;6(9):1305–15. http://dx.doi.org/10.1109/83.623193.
[35] Lloyd S. Least squares quantization in PCM. IEEE Trans Inform Theory 1982;28(2):129–37. http://dx.doi.org/10.1109/TIT.1982.1056489.
[36] Hennig C, Meila M, Murtagh F, Rocci R. Handbook of cluster analysis. CRC Press; 2015, ISBN 978-0429185472.
[37] Maxwell JC. L. on the calculation of the equilibrium and stiffness of frames. Lond Edinb Dublin Philos Mag J Sci 1864;27(182):294–9. http://dx.doi.org/10.1080/14786446408643668.
[38] Pellegrino S, Calladine CR. Matrix analysis of statically and kinematically indeterminate frameworks. Int J Solids Struct 1986;22(4):409–28. http://dx.doi.org/10.1016/0020-7683(86)90014-4.
[39] Rhino FitPlaneToPoints method. 2023, https://developer.rhino3d.com/api/rhinocommon/rhino.geometry.plane/fitplanetopoints. [Accessed 12 August 2023].
[40] Zong C, Xu J, Song J, Chen S, Xin S, Wang W, Tu C. P2M: a fast solver for querying distance from point to mesh surface. ACM Trans Graph 2023;42(4). http://dx.doi.org/10.1145/3592439.
[41] Rhino mesh.ClosestPoint method. 2023, https://developer.rhino3d.com/api/rhinocommon/rhino.geometry.mesh/closestpoint. [Accessed 12 August 2023].
[42] Rhino polyline.ClosestPoint method. 2023, https://developer.rhino3d.com/api/rhinocommon/rhino.geometry.polyline/closestpoint. [Accessed 12 August 2023].
[43] Dai Y-H, Yuan Y. A nonlinear conjugate gradient method with a strong global convergence property. SIAM J Optim 1999;10(1):177–82. http://dx.doi.org/10.1137/S1052623497318992.
[44] Botsch M, Kobbelt L. A remeshing approach to multiresolution modeling. In: Proceedings of the 2004 eurographics/acm SIGGRAPH symposium on geometry processing. New York, NY, USA: Association for Computing Machinery; 2004, p. 185–92. http://dx.doi.org/10.1145/1057432.1057457.
[45] Burden RL, Faires JD, Burden AM. Numerical analysis. Cengage learning; 2015, ISBN 978-1305253667.
[46] Herzog T, Krippner R, Lang W. Facade construction manual. DETAIL; 2017, ISBN 978-3955533694.
[47] Pariafsai F. A review of design considerations in glass buildings. Front Architect Res 2016;5(2):171–93. http://dx.doi.org/10.1016/j.foar.2016.01.006.
[48] Botsch M, Kobbelt L, Pauly M, Alliez P, Lévy B. Polygon mesh processing. New York: CRC Press; 2010, ISBN 978-1568814261.
[49] Bhimani J, Leeser M, Mi N. Accelerating K-means clustering with parallel implementations and GPU computing. In: 2015 IEEE high performance extreme computing conference. 2015, p. 1–6. http://dx.doi.org/10.1109/HPEC.2015.7322467.