

Weingarten Surface Approximation by Curvature Diagram Transformation

Fei Huang^a, Caigui Jiang^{b,*}, Yong-Liang Yang^a

^a*University of Bath, Bath, United Kingdom*

^b*Xi'an Jiaotong University, Xi'an, China*

Abstract

Weingarten surfaces are characterized by a functional relation between their principal curvatures. Such a specialty makes them suitable for building surface paneling in architectural applications, as the curvature relation implies approximate local congruence on the surface thus the molds for paneling can be largely reused. In this work, we aim at a novel task of Weingarten surface approximation. Given a surface mesh with arbitrary topology, we optimize its shape to make it as Weingarten as possible. We devise a curvature-based optimization approach based on the fact that the 2D principal curvature plots of a Weingarten surface comprise a group of 1D curves that encode the curvature relations. Our approach alternatively performs two steps. The first step transforms the principal curvature plots from a 2D region to 1D curves in order to explore the curvature relations. The second step deforms the shape such that its curvatures conform to the corresponding transformed curvature plots. We demonstrate the effectiveness of our work on a variety of shapes with different topologies. Hopefully our work would bring inspiration on the study of general Weingarten surfaces with arbitrary topology and curvature relation.

Keywords: Shape optimization, Shape approximation, Weingarten surface

1. Introduction

Surfaces possessed of specific curvature features have attracted significant attention in the field of computer graphics. Typical cases of these surfaces, such as developable surfaces and minimal surfaces with vanishing Gaussian curvature and mean curvature, respectively, are widely studied and utilized in computer-aided design and manufacturing areas, particular in the prototype design using flat materials and membrane materials [Yang et al. \(2014\)](#); [Kilian et al. \(2017\)](#); [Wang et al. \(2023\)](#).

However, Weingarten surfaces, which are the superset of developable and minimal surfaces, lack graphics research attention. Different from developable or minimal surfaces, Weingarten surfaces do not hold a particular constraint on one type of curvature, but are characterized by a general relation between its two principal curvatures. Weingarten surfaces were first introduced by [Weingarten \(1861, 1863\)](#) and further studied by geometry researchers in the last few decades. They have recently gained attention in the graphics field because it benefits the design and construction of architectural freeform surfaces [Pellis et al. \(2021\)](#). Due to the local shape congruence enabled by the curvature relation, architectural skins modeled with Weingarten surfaces can be segmented into geometrically similar panels and fabricated with the same mold, which largely reduces the engineering budget.

However, approximating a given surface with general Weingarten surfaces is challenging. This is mainly attributed to the wide range of curvature relation allowed by the Weingarten surface. The large degree of freedom makes it difficult to model the curvature characteristics of Weingarten surfaces as specific geometric constraints as those for developable and minimal surfaces. As a result, most prior works studied Weingarten

*Corresponding author

Email addresses: fh463@bath.ac.uk (Fei Huang), cgjiang@xjtu.edu.cn (Caigui Jiang), y.yang@cs.bath.ac.uk (Yong-Liang Yang)

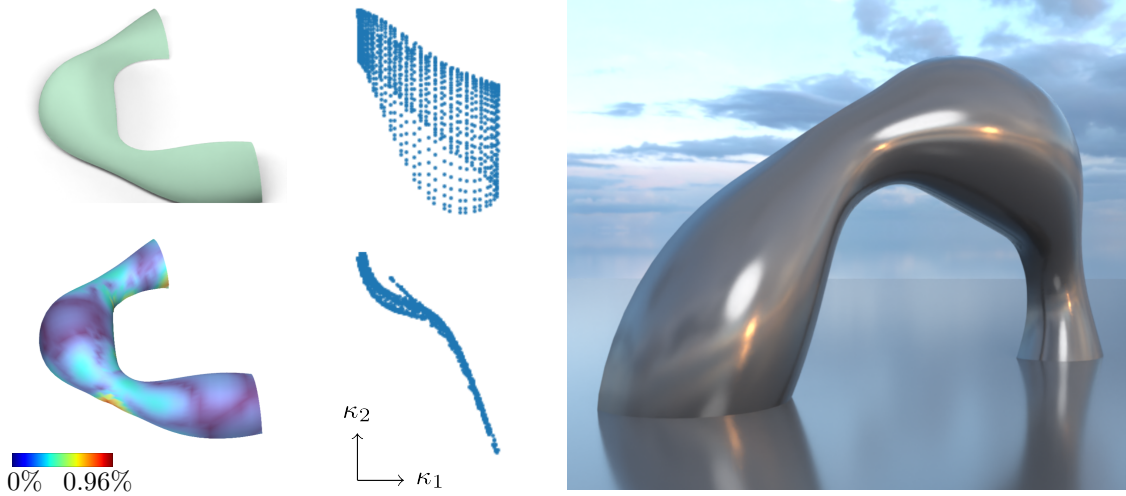


Figure 1: Weingarten surface approximation. *Top left*: The input triangular mesh and its principal curvature plots on (κ_1, κ_2) plane. *Bottom left*: The output mesh and its curvature diagram after optimization. The color-coding shows the deviation from the input, indicated as the percentage of the bounding box diagonal of the mesh. *Right*: Conceptual design and visualization of a façade structure based on the output.

surfaces with specific curvature relations only [Tellier et al. \(2019, 2020b,a, 2023\)](#); [Wang and Pottmann \(2022\)](#). Pellis et al. [Pellis et al. \(2021\)](#) investigated the general Weingarten surface approximation where B-spline surfaces were used to represent the target shape. They implicitly modeled the relation of curvatures as the alignment of curvature isolines where the two gradients of Gaussian and mean curvatures are optimized to be linearly dependent. Although using B-spline surfaces enables exact computations of curvatures and their derivatives, the shapes to be approximated are restricted to having a regular parametric domain.

In this paper, we propose a more generalized Weingarten surface approximation method for mesh surfaces (see Fig. 1). Our method takes advantage of the representation ability of polygon meshes, as well as the availability of abundant mesh models in the field. However, simply following the approach of [Pellis et al. \(2021\)](#) by optimizing its discretized counterpart is not feasible because the optimization therein requires the surface derivatives up to the fourth order, which cannot be easily formulated on mesh surfaces. Inspired by the developable surface approximation method based on manipulating the Gaussian image [Binninger et al. \(2021\)](#), our method leverages the transformation of the principal curvature diagram to drive the shape approximation. We deform the shape geometry in an as-Weingarten-as-possible manner, while keeping it close to the original shape. More specifically, we iteratively update the principal curvatures in the curvature diagram such that the curvature plots transform from a 2D region to a group of curves that encode the curvature relations, while optimizing the mesh vertices to conform to the updated curvatures accordingly until convergence. Extensive experiments demonstrate that our method is effective on meshes with different topologies from different shape categories.

Our work makes the following contributions:

- We attempt a new task of Weingarten surface approximation on general mesh surfaces.
- We devise a novel approach that uses the curvature diagram transformation to guide the mesh deformation for the Weingarten surface approximation.
- We have achieved the Weingarten surface approximation on more complicated shapes that have not been done before.

In the next section, we will introduce the related work of Weingarten surfaces and curvature-based mesh optimization. We will describe the pipeline of our method in the Section 3, and will demonstrate with diverse types of meshes in the Section 4. Finally, the implementation detail, limitations and future work will be discussed in Section 5 and Section 6.

2. Related Work

2.1. Weingarten surfaces

Weingarten surfaces are characterized by a functional curvature relation between two principal curvatures $f(\kappa_1, \kappa_2) = 0$ (or equivalently between Gaussian and mean curvatures $f(H, K) = 0$) [Krivoshapko and Ivanov \(2015\)](#). Mathematically, Weingarten surfaces are classified into the following main groups: surfaces of revolution, curved tube surfaces where one principal curvature is constant, helicoidal surfaces, surfaces of constant Gaussian curvature, and surfaces of constant mean curvature [Kühnel and Steller \(2005\)](#); [Krivoshapko and Ivanov \(2015\)](#). Noted these groups are not exhaustive. The principal curvature diagram of a Weingarten surface lies in the (κ_1, κ_2) plane and comprises a group of curves below the straight line defined by $\kappa_1 = \kappa_2$ (κ_1 is the maximum principal curvature and κ_2 is the minimum one). The diagram only occupies one and two halves quadrants of the whole plane [Hopf \(1989\)](#).

Some research works focus on studying the Weingarten surface by first identifying a mathematical feature of it on the continuous setting, then generating discrete Weingarten surfaces according to this feature and target practical applications. However, most of these works apply to one specific type of Weingarten surfaces only. For instance, [Wang and Pottmann \(2022\)](#) studied Weingarten surfaces with a constant ratio of principal curvatures, named CRPC surfaces. Compared to general Weingarten surfaces, CRPC surfaces have a more specific relation between their principal curvatures, reducing the difficulties for surface modeling and editing. Some works utilize the characteristics of Weingarten surfaces as additional constraints for shape exploration on top of certain discrete mesh structures, such as principal symmetric meshes [Pellis et al. \(2020\)](#) and C-meshes [Liu et al. \(2023\)](#). Although these works are not directly related to the Weingarten surface modeling, they demonstrate that Weingarten surfaces have a potential for structure design.

Works from civil engineering and architectural design fields aim at using Weingarten surfaces in real construction circumstances. Thus, they combined the physical stress based constraints with characteristics of Weingarten surfaces. Similar to the theoretical work mentioned before, these engineering works usually focus on one specific type of Weingarten surface only, for the purpose of making it easier to incorporate into the optimization process. [Tellier et al. \(2023\)](#) studied linear Weingarten surfaces, whose principal curvature relation is a linear function, in the isotropic space. Applications of linear Weingarten surfaces on designing architectural surfaces or special beam structures were also explored in [Tellier et al. \(2019, 2020b,a\)](#).

2.2. Curvature-driven mesh optimization

Mesh optimization has been extensively studied for various applications in shape modeling, such as mesh parametrization, shape deformation, geometric design, etc. [Botsch et al. \(2010\)](#). Here we review the mostly relevant works that optimize meshes directly with the guidance of curvatures. The majority of works manipulate the shape of the input mesh by the curvature flow of different types of curvatures, including mean curvature [Desbrun et al. \(1999\)](#); [Kazhdan et al. \(2012\)](#), Gaussian curvature [Jin et al. \(2008\)](#); [Su et al. \(2019\)](#); [Fang et al. \(2020\)](#), and their variations [Crane et al. \(2013\)](#). [Fang et al. \(2020\)](#) reconstructed mesh metric according to the mixture of mean curvature and Gaussian curvature and their corresponding curvature flows. There are also works that attempt to optimize the shape according to modified curvature magnitudes. For instance, [Eigensatz et al. \(2008\)](#); [Eigensatz and Pauly \(2009\)](#) utilized curvature magnitudes estimated within the local neighborhood as the major constraint to optimize the input mesh. In addition to curvature magnitudes, [Pan et al. \(2015\)](#); [Li et al. \(2017\)](#) employed principal curvature directions defined by user-input sketches to guide surface generation and optimization.

3. Method

The principal curvature diagram of a Weingarten surface comprises a group of 1D curves on the 2D plane. These curves are formed by discrete 2D points whose coordinates correspond to maximum and minimum principal curvatures, implying the curvature relations of their corresponding points on the surface. According to the curvature diagram, the Weingarten surface consists of multiple smoothly connected patches with different principal curvature relations.

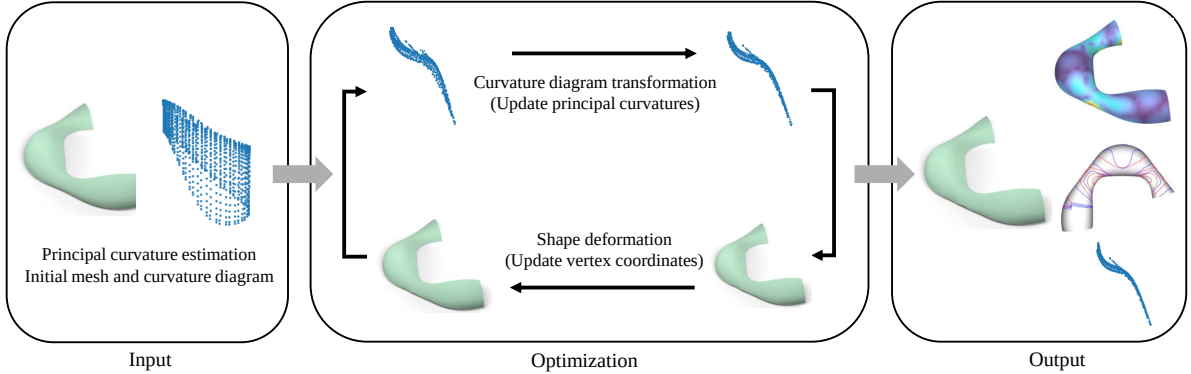


Figure 2: Pipeline of our method. *Input*: We take as input a triangular mesh where we estimate the principal curvatures to generate the initial curvature diagram. *Optimization*: We iteratively perform curvature diagram transformation and shape deformation, where we update curvature values based on local fitting and projection followed by deforming the mesh shape according to the updated curvature diagram in an alternative way. *Output*: The output mesh is as Weingarten as possible, which can be verified by curve-like curvature plots and aligned curvature isolines (deviation from the original shape is also color-coded).

Our main goal is to optimize a given shape to make it as-Weingarten-as-possible. We propose to first transform the 2D principal curvature plots into a group of 1D curves, then use the transformed curvature diagram to guide the deformation of the given shape. We alternately perform curvature diagram transformation and shape deformation in a local-global manner as follows (see also Fig. 2).

In the local step, we move each 2D principal curvature plot in the diagram so that the initial scattered plots can locally contract into an arc. After the contraction, updated principal curvatures are assigned to the corresponding vertex of the shape. In the global step, we deform the shape according to the new principal curvatures under regularizations of shape smoothness and closeness. We choose to alternately edit the curvature and the shape to avoid the unstable deformation result caused by a large change in principal curvatures. Limiting the principal curvature transformation in each iteration makes the shape deformation more robust (see Section 4).

In the following subsections, we describe the local step (curvature diagram transformation) and the global step (shape deformation) in detail. Our algorithm takes as input a discrete triangulated shape, denoted as $\mathcal{M} = \{\mathcal{V}, \mathcal{F}, \mathcal{E}\}$ according to its vertex, face, and edge sets. Noted that the input and output shapes have the same connectivity. The two principal curvatures at vertex $\mathbf{v}_i$ on $\mathcal{M}$ are represented as $\kappa_{i,w}$, where $w = 1$ or $w = 2$ indicates the maximum or minimum principal curvature, respectively. We normalize the input shape by scaling the shape with a uniform bounding box diagonal.

3.1. Curvature diagram transformation

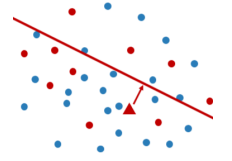
There are various methods to estimate principal curvatures on discrete shapes. We employ a vertex-based principal curvature estimation method for the convenience of shape deformation in the next step. This vertex-based method calculates principal curvatures from a local curvature tensor at each vertex, which is formed by integrating the edge-based principal curvatures across a surrounding region [Cohen-Steiner and Morvan \(2003\)](#); [Alliez et al. \(2003\)](#). In practice, we uniformize the input to have a standard edge length. Then we choose the region by counting the topology n -ring neighborhood area. The minimal region covers the one-ring neighborhood area as in the Laplacian operator.

The curvature diagram of a smooth Weingarten surface is a group of curves located in the half plane under the main diagonal of a (κ_1, κ_2) plane [Hopf \(1989\)](#). In our local step, we first estimate vertex-based principal curvatures and plot them as 2D points (cf. curvature points) in the (κ_1, κ_2) plane to construct the curvature diagram. Then we iteratively contract curvature points into arcs to form a group of curves similar to [Binninger et al. \(2021\)](#). The main idea is that, for each curvature point, we locally perform weighted principal component analysis (PCA) within a local neighborhood, and then project the point onto the principal line from PCA, which is essentially an approximation of curve tangent.

More specifically, the neighbors of a given curvature point are selected by the k -Nearest Neighbor method from the curvature space while also being within the target distance in Euclidean space. We first find k nearest neighbors of the target vertex in the curvature plot. We store the distance of the target vertex plot and the neighbor plot in the curvature space as d_{ij}^c . We remark the maximum curvature distance of each target vertex as $\gamma_i = \max\{d_{ij}^c, j \in [0, k]\}$. We store the distance of the target and the neighbor in the Euclidean space as d_{ij}^e . Then we set a threshold σ to bound Euclidean distance d_{ij}^e between mesh vertices. For each vertex $\mathbf{v}_i$, we denoted the set of neighboring vertices before filtering by σ as $\mathcal{N}_i$. Then we set entries of PCA weight matrix as in the following equation:

$$\forall \mathbf{v}_j \in \mathcal{N}_i, w_{ij} = \begin{cases} 0 & \text{if } d_{ij}^e \geq \sigma \\ \exp[-(\frac{d_{ij}^e}{a\gamma_i})^2] & \text{otherwise} \end{cases} \quad (1)$$

Note that a is a constant to control the weight falloff according to the distance in the curvature diagram. The inset figure gives an example. The current vertex $\mathbf{v}_i$ is marked by a red triangle. All the other vertices in the inset figure belong to $\mathcal{N}_i$. Vertices marked in red are valid vertices and they fit a red line, where $\mathbf{v}_i$ is going to project on. Vertices marked in blue have $w_{ij} = 0$ as defined in Eqn. 1, meaning they are too far from $\mathbf{v}_i$ in Euclidean space. In our implementation, we set σ as $1/5$ bounding box diagonal and the k as $1/20$ number of vertices of the input mesh as the default. To stabilize the contraction of curvature plots into curves, we reduce k in each PCA iteration.



After calculating the weight matrix, we apply weighted PCA to curvature plots to find the best fitting line for each local neighborhood by eigen decomposition of the covariance matrix:

$$\mathbf{A} = \mathbf{K}\mathbf{W}\mathbf{K}^T, \quad (2)$$

where $\mathbf{W}$ denotes the weight matrix as a collection of w_j , and $\mathbf{K}$ denotes the principal curvature matrix. Each row in $\mathbf{K}$ is a 2D vector $\kappa_i = (\kappa_{i,1}, \kappa_{i,2})$. The normal of the best-fitting line (i.e., local curve tangent approximation) is the eigenvector of the covariance matrix corresponding to the minimum eigenvalue. We project each curvature point onto the local tangent to achieve the updated principal curvatures.

Fig. 10 shows the transformation of curvature plots of an input shape. It is easy to see that the scattered curvature plots are contracted into a group of curves after a few iterations.

3.2. Shape deformation

After curvature diagram transformation in the local step, the updated principal curvatures (denoted as $\hat{\kappa}_{i,w}$) are better correlated to meet the requirement of a Weingarten surface. In the global step, we aim to update vertex coordinates such that principal curvatures of the deformed shape conform to those in the curvature diagram as much as possible. We formulate the curvature-guided shape deformation as a nonlinear least-squares optimization with the following objectives.

Curvature matching. The main objective of our optimization is to minimize the difference between the curvatures of the shape and the target principal curvatures. To achieve this, we employ the energy term from Eigensatz et al. (2008); Eigensatz and Pauly (2009) for curvature-based shape deformation which is defined as the sum of squared principal curvature differences:

$$f_{curv} = \frac{1}{2} \omega_{curv} \sum_{v_i \in V} (\kappa_{i,w} - \hat{\kappa}_{i,w})^2 \quad (w = 1, 2) \quad (3)$$

This energy term is highly non-linear because $\kappa_{i,w}$ is solved by eigen decomposition of a 3×3 discrete curvature tensor. Thus we need to compute the derivative of eigenvalues to get the Jacobian concerning the vertex coordinate. We refer to Eigensatz et al. (2008); Eigensatz and Pauly (2009) and Wardetzky et al. (2007) for the detailed numerical computation.

To regularize the highly non-linear principal curvatures objectives and preserve the geometry of the input shape, we add additional objectives to the optimization.

Smoothness. We exploit per-vertex discrete Laplacian to ensure the smoothness of the resultant shape:

$$f_{smooth} = \frac{1}{2} \omega_{smooth} \sum_{\mathbf{v}_i \in \mathcal{V}} \left\| \sum_{\mathbf{v}_j \in \mathcal{N}_i^1} c_{ij} (\mathbf{v}_j - \mathbf{v}_i) \right\|^2, \quad (4)$$

where $\mathcal{N}_i^1$ denotes the one-ring neighborhood of vertex $\mathbf{v}_i$, $\mathbf{v}_j$ denotes one of its one-ring neighbors, c_{ij} is the cotangent weight of edge e_{ij} . We keep c_{ij} constant in our implementation to simplify the optimization process. Note that for a shape with open boundaries, the smoothness is only defined on inner vertices, while boundary smoothness is separately defined as follows.

Boundary smoothness. Given the input shape may have one or more boundaries and each boundary may consist of several smooth boundary segments, we also devise a boundary smoothness term to preserve the boundary quality of the resultant shape:

$$f_{bdry} = \frac{1}{2} \omega_{bdry} \sum_{\mathcal{B}_j \in \mathcal{B}} \sum_{\mathbf{v}_i \in \mathcal{B}_j} \|\mathbf{v}_{i-1} + \mathbf{v}_{i+1} - 2\mathbf{v}_i\|^2, \quad (5)$$

where $\mathcal{B}$ denotes the set of all smooth boundary segments, $\mathcal{B}_j$ is one such boundary segment in $\mathcal{B}$. For instance, a rectangle-like boundary can be divided into four smooth boundary segments by the four boundary corner points.

Closeness. We also define a closeness term as a regularizer to preserve the geometry of the current shape, inspired by Tang et al. (2014). It measures the difference between the current shape and the shape from the last optimization iteration:

$$f_{close} = \omega_{close} \sum_{x_i \in \mathcal{X}} \|x_i - x_i^k\|^2, \quad (6)$$

where x_i denotes a variable in the optimization system $\mathcal{X}$, x is the value of the current iteration, and x^k is the value from the last (k -th) iteration.

We associate weights ω_{curv} , ω_{smooth} , ω_{bdry} and ω_{close} to each energy term and define the final objective $f = f_{curv} + f_{smooth} + f_{bdry} + f_{close}$. We solve the deformed vertex coordinates by minimizing the overall objective using the Levenberg–Marquardt algorithm.

Algorithm 1: Weingarten surface approximation

Data: κ_1, κ_2, V , initial value for $\kappa'_1, \kappa'_2, V', k, f$
 Fix threshold f_{min}, k_{min} , maximum iterations i_{max}, j_{max}
 Initialize $i = 0$
while $k \geq k_{min}$ **and** $i < i_{max}$ **do**
 Update κ'_1 and κ'_2 by PCA
 Initialize $j = 0$
 while $f \geq f_{min}$ **and** $j < j_{max}$ **do**
 $f \leftarrow f_{curv} + f_{smooth} + f_{bdry} + f_{close}$
 Update V'
 $j \leftarrow j + 1$
 end
 $i \leftarrow i + 1$
 $k \leftarrow k * 0.95$
end
Result: V'

3.3. Local-global approach

To make the input shape as Weingarten as possible, we perform the local step and global step alternatively. Note that both local and global steps can undergo an iterative process by themselves, i.e., iterative fitting and projection for the local step while iterative optimization for the global step. The shape deformation usually takes several iterations to converge to a reasonable minimum, while the curvature diagram transformation usually takes fewer steps to converge as PCA is more stable. Therefore, we set different numbers of iterations within local and global steps. In practice, we perform one iteration in the local step while multiple iterations (usually 10-30 till convergence) in the global step. This allows stable curvature-guided shape optimization with fast convergence. The curvature diagram transformation stops when the current number of neighbors k is reduced to a constant minimal number k_{min} . In our implementation, we reduce the k by 5% in the end of each local iteration. The deformation stops when the total energy is below a threshold or the number of iterations is over the maximum value Jiang et al. (2021). We summarize our method in Algorithm 1.

4. Evaluation

In this section, we extensively evaluate our work. First, we demonstrate the results of our method performed on different types of shapes. Then we visualize the behavior of our method using curvature diagrams, isolines, gradient alignment and surface smoothness. We also test the method on the example with different tessellations. Finally, we compare our method with the previous work.

4.1. Results

We evaluate our method on four different types of mesh models, including patches as in Pellis et al. (2021) (Fig. 3), architectural surfaces (Fig. 4), general shapes with varying topologies (Fig. 5), and organic shapes like Bunny (Fig. 6). For each test case, we show the input shape, optimized shape, input curvature diagram, output curvature diagram, and color-coded shape deviation, respectively. The shape deviation is measured by per-vertex distances as the percentage of the bounding box diagonal.

Patches. As shown in Fig. 3, surface patches have smooth geometry and simple disk-like topology. The initial curvature diagrams also exhibit a patch-like distribution with more regularly sampled curvature points. The curvature diagrams after optimization comprise well-established curves. For applications like CAD and architecture design, it is convenient to achieve an as-Weingarten-as-possible output from a simple surface patch, which usually has a regular curvature distribution across the whole surface. The maximal shape deviation is generally on the boundary and the area with significant curvature changes.

Architectural surfaces. Compared to disk-like patches, the architecture models in Fig. 4 have the same topology as an annulus with different numbers of boundaries. They have unstructured curvature diagrams compared to simple surface patches. Curvature diagrams of these models after optimization are usually composed of several different curves that may not be connected altogether, indicating there are multiple principal curvature relations across the model. This is related to shape approximation with piece-wise Weingarten surfaces of specific types, which is an interesting topic to explore.

General shapes. General shapes with a more complicated topology such as high genus are rarely attempted in previous works on Weingarten surface approximation. The proposed mesh-based optimization approach enables Weingarten surface approximation on such shapes. The initial curvature diagram of general shapes such as ‘bumpy’ (first row of Fig. 5) is more unstructured compared to patches. It usually takes more iterations to get a reasonable result.

Organic shapes. As in Fig. 6, organic shapes usually have a sphere-like or torus-like topology but with large geometric variations compared with the above shape categories. This leads to very scattered curvature points in the diagram which even contains many plots like visual outliers, making it challenging to be contracted and optimized in the local-global process. Our method is still effective in terms of simplifying the curvature plots into more curve-like distribution.

4.2. Visualization of curvature isolines

As discussed in Pellis et al. (2021), a Weingarten surface will have only one group of principal curvature isolines based on the curvature relation. We use this property to verify our results as in Fig. 7. It demonstrates that the principal curvature isolines are much better aligned after optimization.

4.3. Visualization of curvature gradient alignment

In addition to the principal curvature isoline visualization, we measure and color-code the alignment of two principal curvature gradients in Fig 8. We first compute and normalize the gradient of the maximum principal curvature as well as the gradient of the minimum principal curvature within each mesh face, and measure their alignment as the absolute value of their dot product. Then the alignment value at each mesh vertex is defined as a weighted average of those values from the neighboring faces, where the face area is used as the weight. We also quantitatively measure the average alignment angle θ between two principal curvature gradients before and after optimization (see Table 1).

4.4. Visualization of Smoothness

To better demonstrate the smoothness of our results, we visualize the input and output of representative examples with flat shading in Fig. 9. It shows that the deformation optimization does not bring noise and unstructured faces into the result, which validates the effectiveness of our smoothness terms (see Eqn. 4 and Eqn. 5).

4.5. Transformation of curvature diagram

Fig. 10 shows the evolution of the curvature diagram of a patch during the optimization. It can be seen that the curvature plots are contracted during the process and the mesh is deformed accordingly. Usually the first few steps make larger deformations. The later steps refine the mesh at a smaller scale such that its principal curvature values better conform to the transformed diagram.

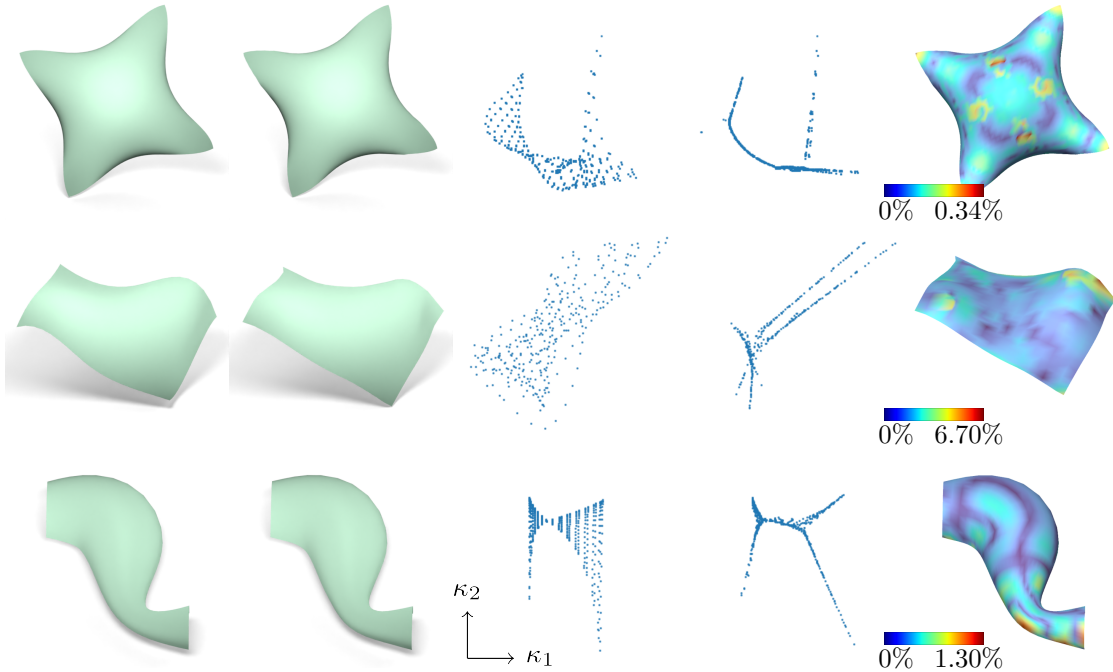


Figure 3: Patches. *From left to right:* Input mesh, output mesh, initial curvature diagram, resultant curvature diagram, output mesh with color-coded shape deviation.

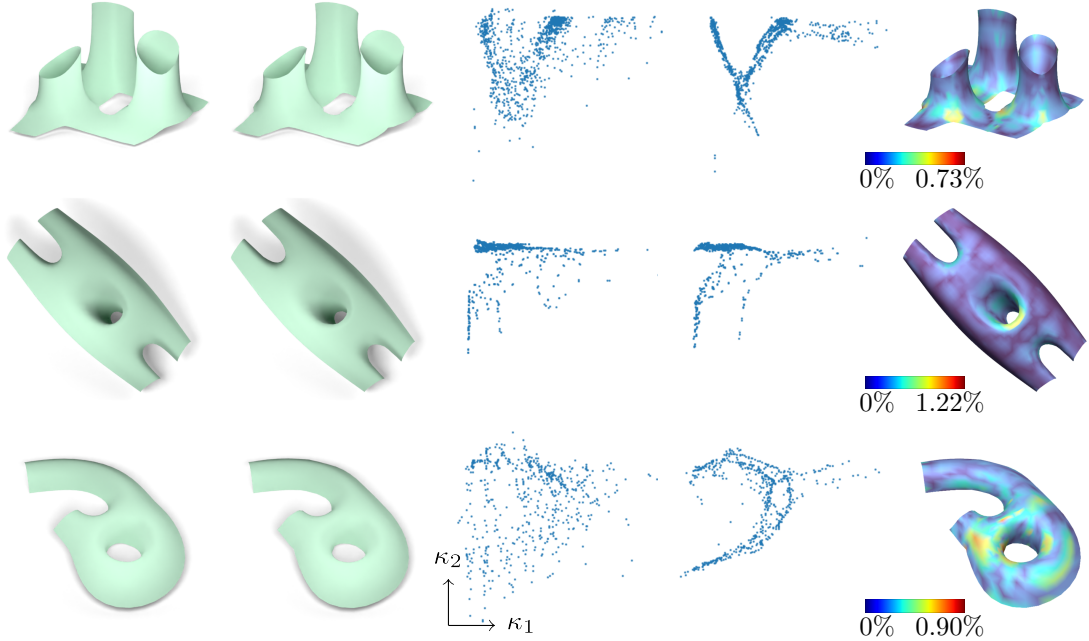


Figure 4: Architectural surfaces. *From left to right*: Input mesh, output mesh, initial curvature diagram, resultant curvature diagram, output mesh with color-coded shape deviation.

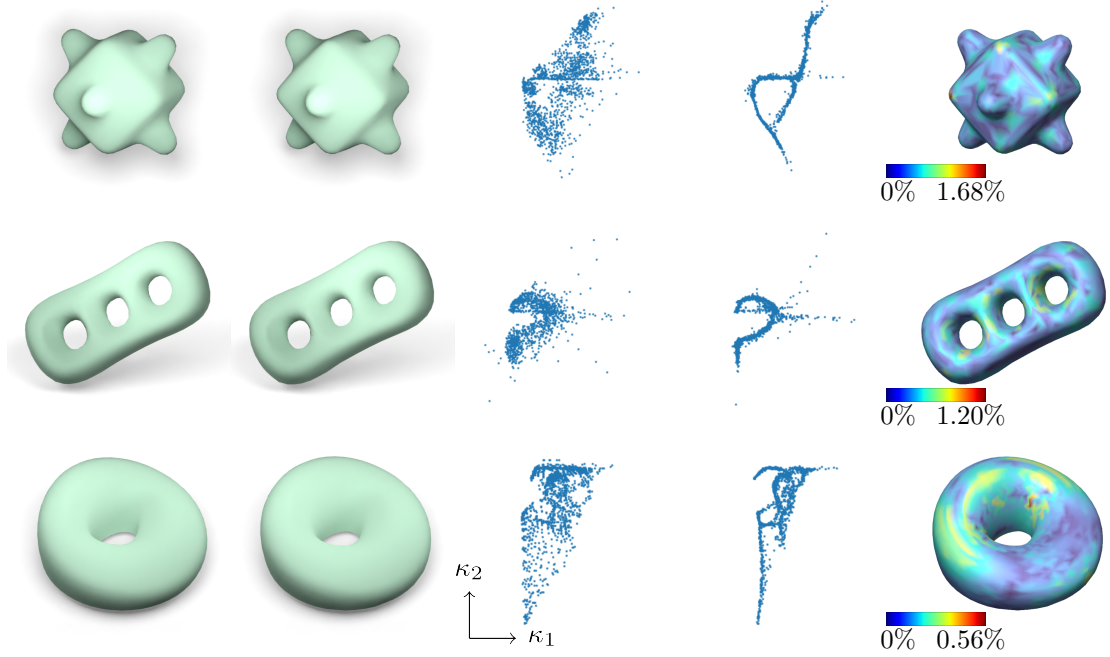


Figure 5: General shapes. *From left to right*: Input mesh, output mesh, initial curvature diagram, resultant curvature diagram, output mesh with color-coded shape deviation.

4.6. Influence of surface tessellation

In Fig. 11 we test on a patch surface that is discretized by three different tessellations: the first row is a regular tessellation used commonly in CAD meshes; the second row is a low-quality tessellation with

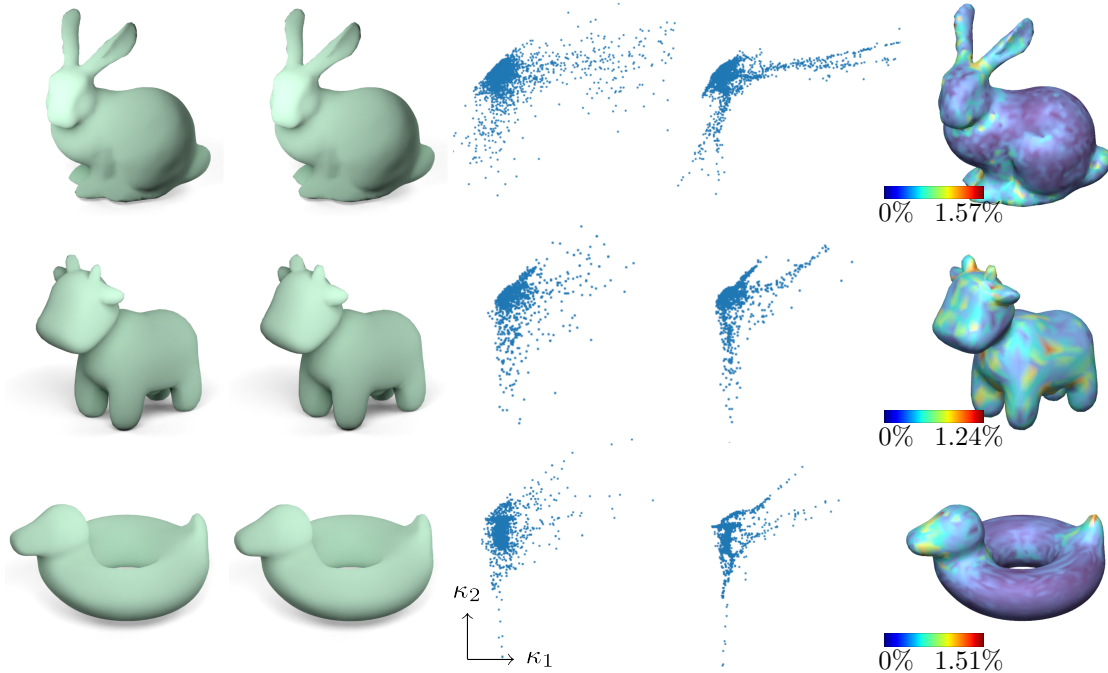


Figure 6: Organic shapes. *From left to right*: Input mesh, output mesh, initial curvature diagram, resultant curvature diagram, initial curvature isolines, resultant curvature isolines, output mesh color-coded shape deviation.

triangles that are nearly degenerate; the third row is another regular tessellation with different connectivities. The tessellations are generated using mesh editing tools. In our experiment, we used the same weights for the three cases. It can be seen that our method can effectively generate as-Weingarten-as-possible results for different tessellations as shown in the contracted curvature diagrams. The input curvature diagrams are slightly different for different tessellations. The input curvature of the second case is noisy due to the low-quality tessellation. The optimized shapes are comparable as demonstrated by the color-coded shape variations. And the output curvature diagrams are almost similar. Our method can generate stable output without regard to the connectivity and the quality of the tessellation.

4.7. Comparison

Now we compare our result with the prior work. To our knowledge, the only work on general Weingarten surface approximation is [Pellis et al. \(2021\)](#). However, their work only applies to B-spline surfaces while our method can work on more general triangular meshes. Also, their optimization is performed on Gaussian and mean curvatures while our optimization based on principal curvatures. Further, the parametric representation of B-splines allows explicit computation of curvatures and their derivatives while we need to estimate curvatures from surface discretizations. For a fair comparison, we generate results on the same patch surfaces represented by meshes and B-splines respectively (see [Fig. 12](#)). The results show that our method can generate comparable results with the previous work. It is also worth noting that our approach is fundamentally different by using mesh optimization driven by curvature distribution without the complication of formulating curvature derivatives as in [Pellis et al. \(2021\)](#). To validate the quality of the results, we demonstrate the curvature diagram and the isoline visualization of both results in [Fig. 12](#). For a fair comparison, we apply the same discretization to their B-spline surfaces to get triangle meshes, where we compute the curvatures in the same way as for our results (i.e., estimate κ_1 and κ_2 using [Cohen-Steiner and Morvan \(2003\)](#) and compose H and K). The comparison shows that the curvature diagrams, particularly for H and K , are also comparable in terms of how the curvature plots are distributed, meaning that both methods can produce effective Weingarten surface optimization. The minor difference is due to different optimization targets (principle curvatures vs. mean and Gaussian curvatures), optimization approaches (alternative curvature

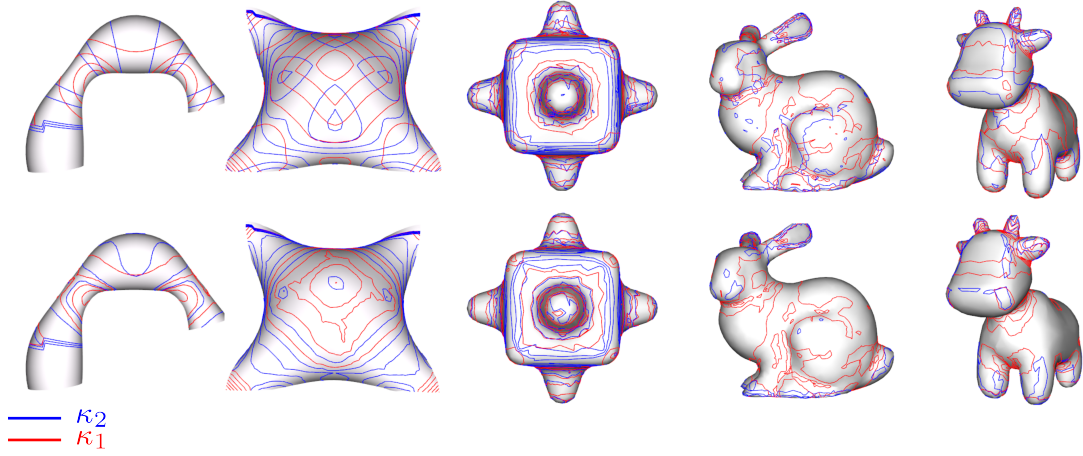


Figure 7: Principal curvature isolines. *Top row*: Principal curvature isolines of the input meshes. *Bottom row*: Principal curvature isolines are aligned after optimization. The red lines are formed by the same maximum principal curvature κ_1 . The blue lines come from minimum principal curvature κ_2 .

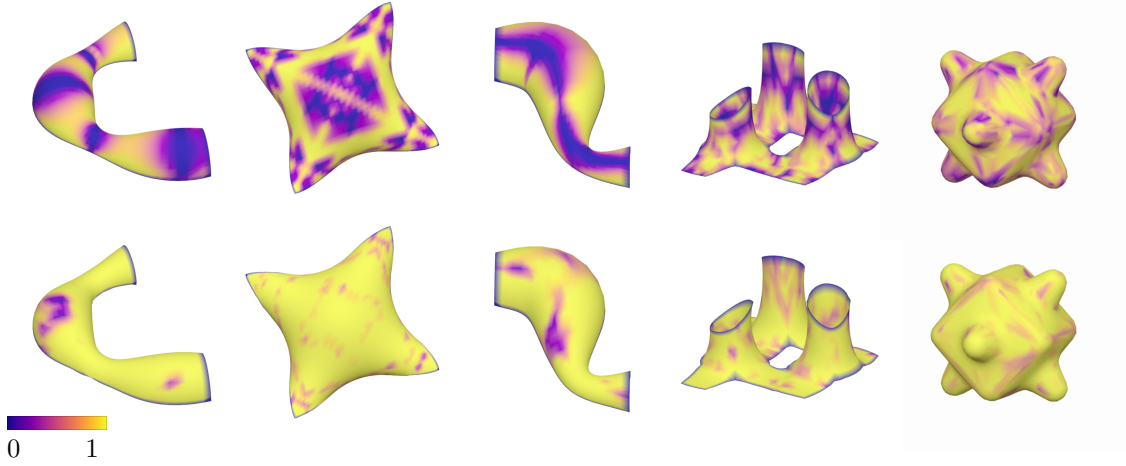


Figure 8: Color-coded principal curvature gradient alignment. *Top row*: Principal curvature gradient alignment of the input meshes. *Bottom row*: Principal curvature gradient alignment of the output meshes. Dot product between gradients of maximum principal curvature κ_1 and minimum principal curvature κ_2 are maximized across the mesh surface after optimization.

and geometry updates vs. direct curvature gradient optimization), and shape representation (discrete mesh vs. continuous B-spline).

5. Discussion

5.1. Implementation details

We implemented our method using C++ based on the geometry processing library LibIGL [Jacobson et al. \(2018\)](#). We used the half-edge data structure from OpenMesh [Bischoff et al. \(2002\)](#). Our experiments were running on a laptop with the Apple M1 processor and 32GB memory. Table 1 shows the detailed statistics of the examples used in this paper. To make it comparable with [Pellis et al. \(2021\)](#), we use seconds per iteration for the running time, which refers to the time for shape deformation, as it is the most time-consuming part of our method.

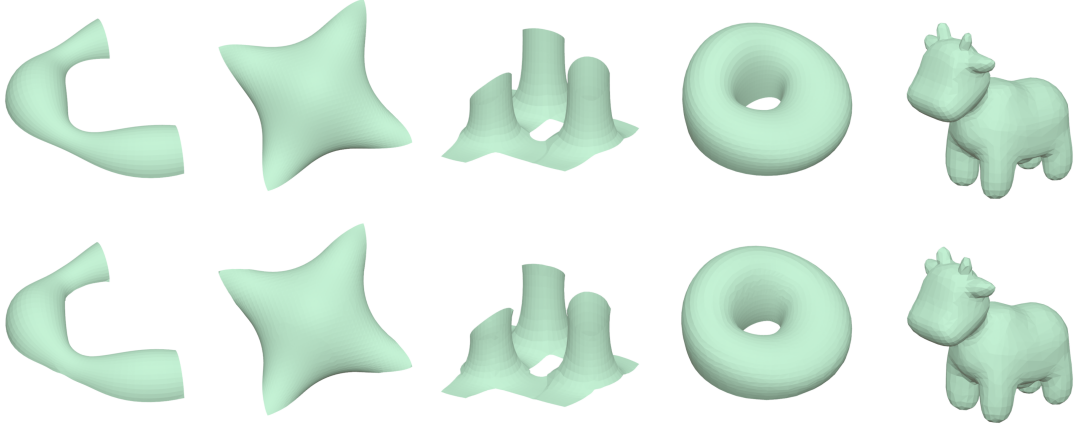


Figure 9: Visualization of smoothness. We show some representative results rendered with flat shading. *Top row*: The input meshes. *Bottom row*: The output meshes after optimization. The mesh smoothness is well preserved after deformation.

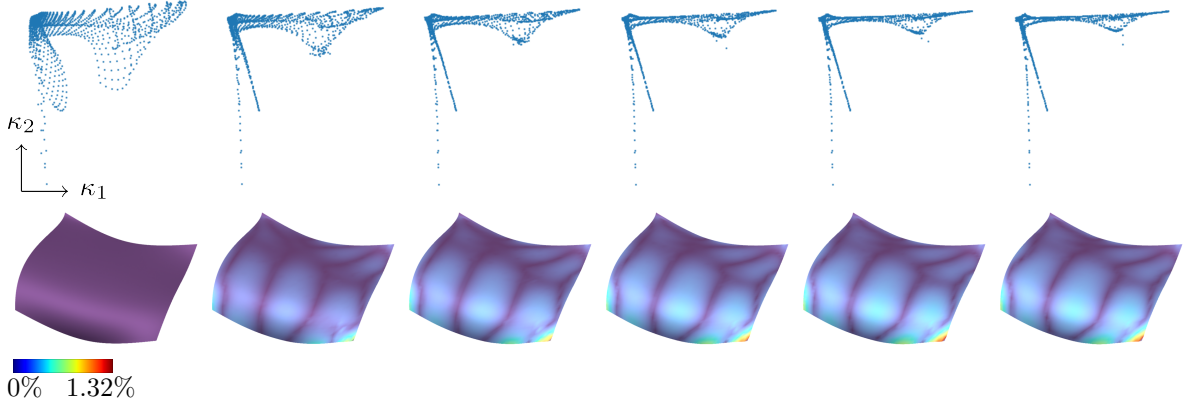


Figure 10: Curvature diagram transformation. *Top row*: The curvature diagram is effectively contracted with the increased number of iterations from left to right. *Bottom row*: The meshes are deformed according to the contracted curvature diagram at each iteration.

5.2. Limitations

The overall Weingarten surface approximation process is computationally expensive. For meshes with around 2k vertices, it usually takes about 2 seconds per local-global step. This is mainly because we need to solve the eigen decomposition (for principal curvatures) in each global step for surface deformation as well as the local step for curvature estimation. Another limitation is the performance on organic shapes like Bunny, which have complex geometry (see Fig. 6). Other than the challenge of contracting scattered curvature diagrams, we also observed that the curvature-driven optimization is likely to fall into a local minimum given the curvatures estimated on discrete meshes are highly non-linear w.r.t. mesh vertices. When the scattered curvature values drastically change during contraction, the mesh vertices cannot be effectively optimized during the shape deformation. In Fig. 13, the curvature diagram from transformation (left) indicated there should be a more obvious functional relation between principal curvatures. However, the curvature diagram of the deformation result (middle) did not reproduce the transformation result well because the curvature-driven optimization falling into a local minimum. Thus there is no guarantee of convergence to an ideal Weingarten surface. However, as noted in Pellis et al. (2021), even curvature plots cannot form a clear group of 1D curves, the ‘more Weingarten’ surface after optimization still benefits the paneling and mold re-use.

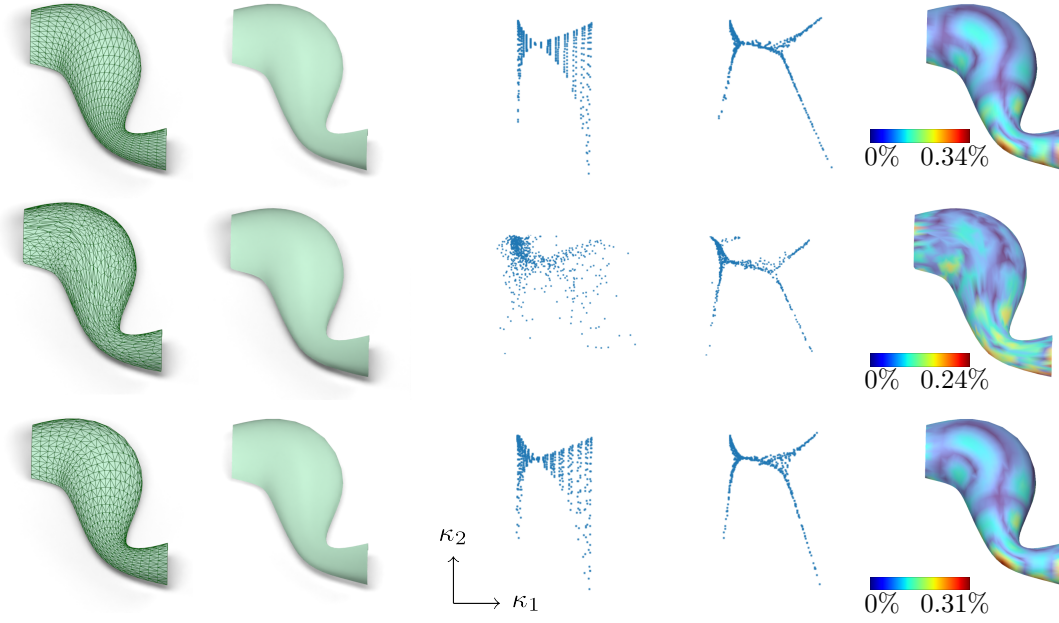


Figure 11: Influence of mesh tessellations. We test on the same shape with three different tessellations. *From left to right:* Input mesh with tessellation highlighted, output mesh, initial curvature diagram, resultant curvature diagram, output mesh with color-coded shape deviation.

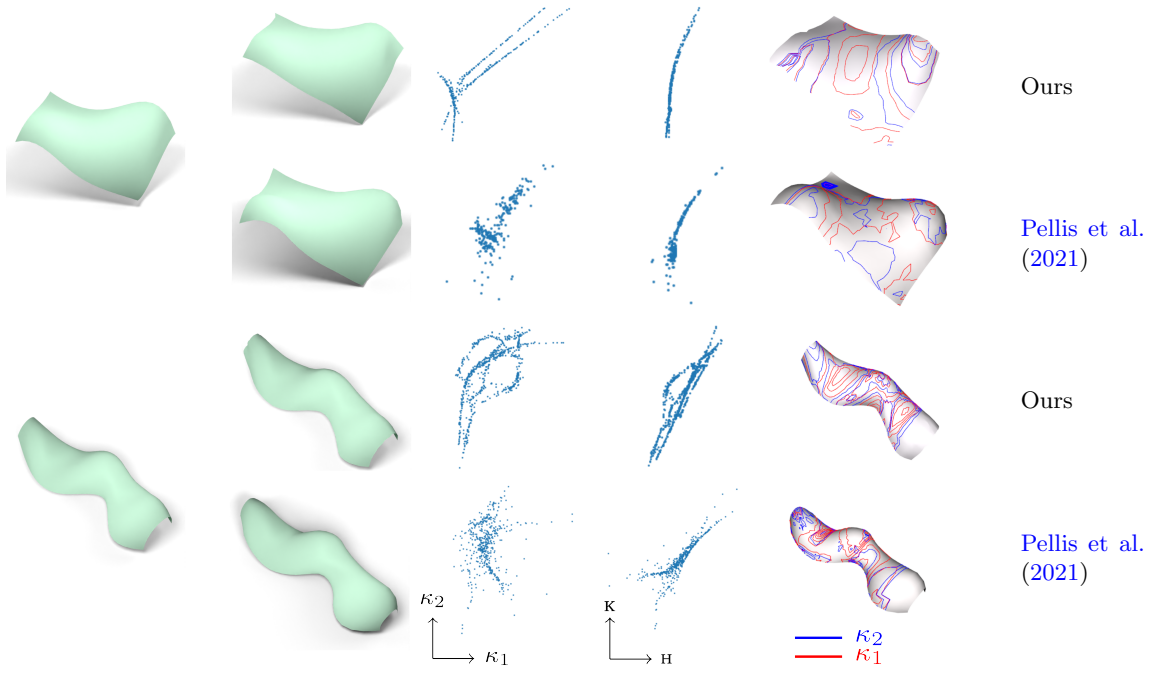


Figure 12: Comparison with [Pellis et al. \(2021\)](#). *Left to right:* Input meshes, outputs meshes, principal curvature diagram of the output, mean-Gaussian curvature diagram of the output, isoline visualization of the output. Noted that our optimization is based on triangular meshes while those from [Pellis et al. \(2021\)](#) are based on B-splines.

5.3. Further discussion

In our method, we use triangle mesh rather than B-spline surface [Pellis et al. \(2021\)](#) as the shape representation for Weingarten surface approximation. This is mainly because mesh representation is more flexible for general shapes with different topologies, and converting a general mesh model to a B-spline surface is inherently challenging and even impossible due to their different representation abilities. Our work would open up new opportunities for general Weingarten surface design and fabrication that is beyond architectural applications. Also, our curvature-driven deformation approach does not require the computation of higher-order curvature gradients for alignment as in [Pellis et al. \(2021\)](#), thus simplifying the optimization formulation on meshes where curvature has no continuous analytical form as for B-splines.

On the other hand, as mentioned in Sec. 5.2, our approach poses new challenges, particularly for shapes with complex geometry. This is mainly because the curvature diagram transformation and shape deformation optimization are operated in two different domains, namely 2D curvature plot and 3D Euclidean space, respectively. For complicated shapes such as Bunny, the transformed curvatures would differ severely from their current values, and thus cannot be easily realized by curvature-driven shape deformation which is a complex non-linear optimization by itself.

6. Conclusion and Future work

In this paper, we propose a method for a new task of Weingarten surface approximation on discrete meshes. Our method is based on a novel approach that optimizes the input mesh shape according to the transformation of its curvature diagram, where the curvature plots are contracted in a way that satisfies the intrinsic property of Weingarten surfaces. We evaluate our method on a variety of shapes with general geometry and topology that have not been attempted before. The results demonstrate the effectiveness of the proposed method and its advantage over prior work. We expect our findings to be inspiring in the field given the challenging nature of the problem.

In the future, we would like to study how to improve the performance of organic shapes by avoiding the local minimum in the optimization. We plan to add a better step size control strategy to the curvature diagram transformation process to prevent curvature plots from changing too much in each step. Also, more robust global optimization methods can be introduced to explore globally optimal solutions. Moreover, we would like to explore the possibilities of using Gaussian and mean curvatures instead of principal curvatures to guide more efficient mesh deformations, which is in line with [Fang et al. \(2020\)](#), where it mainly attempted perturbing one type of curvature while keeping the other fixed, and the (discrete) Gaussian curvatures need to be constrained according to discrete Gauss-Bonnet theorem which could be a limitation for optimizing meshes at different scales. Last but not least, as discussed before, for shapes with complicated geometry like detailed architectural surfaces and organic shapes, we would like to investigate how to approximate piece-wise Weingarten surfaces with specific types, as well as adapting surface types to semantic parts to reduce approximation error. On the other hand, in terms of real fabrication, it would be useful to combine our method with the architectural paneling framework [Eigensatz et al. \(2010\)](#) to generate panels from reused molds for fabrication cost reduction.



Figure 13: Limitation of shape deformation. Here we show the intermediate output from the last iteration of optimizing an organic shape. *From left to right*: Principal curvature diagram from curvature diagram transformation, resultant curvature diagram after shape deformation, output mesh.

Fig.	#V(k)	#F(k)	t(s)/iter	ω_{curv}	ω_{smooth}	ω_{bdry}	ω_{close}	θ_{before}	θ_{after}
Fig 1, 2, 7a, 8a, 9a	1.3k	2.4k	0.95	1.0	10^{-4}	0.5	10^{-6}	48.59	10.74
Fig 3a, 7b, 8b, 9b	1.6k	3.0k	1.36	1.0	10^{-4}	5×10^{-4}	10^{-6}	46.12	14.31
Fig 3b	0.4k	0.7k	0.18	1.0	0.03	0.3	10^{-6}	51.21	10.31
Fig 3c, 8c	0.6k	1.2k	0.33	1.0	10^{-4}	0.001	10^{-6}	48.52	18.44
Fig 4a, 8d, 9c	1.4k	2.5k	0.93	1.0	10^{-4}	0.005	10^{-6}	46.96	19.34
Fig 4b	1.3k	2.5k	0.94	1.0	10^{-4}	10^{-4}	10^{-6}	40.60	31.67
Fig 4c	0.7k	1.3k	0.39	1.0	10^{-4}	10^{-4}	10^{-6}	35.58	20.72
Fig 5a, 7c, 8e	2.5k	5.0k	1.33	1.0	0.001	-	10^{-6}	40.33	18.66
Fig 5b	1.3k	2.5k	1.02	1.0	0.001	-	10^{-6}	41.64	26.01
Fig 5c, 9d	1.7k	3.2k	1.44	1.0	10^{-4}	0.001	10^{-6}	36.19	26.06
Fig 6a, 7d	3.0k	6.0k	5.27	1.0	5×10^{-5}	-	10^{-6}	40.05	32.44
Fig 6b, 7e, 9e	1.4k	2.6k	1.44	1.0	10^{-4}	-	10^{-6}	41.98	32.93
Fig 6c	2.6k	5.3k	4.36	1.0	10^{-4}	-	10^{-6}	37.71	32.43
Fig 10	1.6k	3.0k	1.39	1.0	0.005	0.1	10^{-6}	57.40	18.81
Fig 11a	0.6k	1.2k	0.33	1.0	10^{-4}	0.001	10^{-6}	48.52	18.44
Fig 11b	0.8k	1.5k	0.63	1.0	10^{-4}	0.001	10^{-6}	38.62	18.76
Fig 11c	0.6k	1.2k	0.33	1.0	10^{-4}	0.001	10^{-6}	43.15	20.56
Fig 12a	0.4k	0.7k	0.18	1.0	0.03	0.3	10^{-6}	51.21	10.31
Fig 12b	0.8k	1.5k	0.46	1.0	5×10^{-4}	0.2	10^{-6}	45.68	19.25
Fig 13	2.6k	5.3k	4.36	1.0	10^{-4}	-	10^{-6}	37.71	32.43

Table 1: Detailed statistics of all the example meshes. *From left to right*: figure index, number of vertices and faces, mesh deformation time per iteration, weights in the objectives, and the average angle in degree between principal curvature gradients before and after optimization.

Acknowledgement

The authors would like to thank anonymous reviewers for their suggestive comments. The authors also thank Davide Pellis for sharing models for test and comparison purposes. This paper is partly supported by RCUK grant CAMERA (EP/T022523/1) and National Science and Technology Major Project 2023ZD0121300, NSFC under grant No. 62495092 and 62088102.

References

- Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., Desbrun, M., 2003. Anisotropic polygonal remeshing. *ACM Transactions on Graphics* 22, 485–493. doi:[10.1145/882262.882296](https://doi.org/10.1145/882262.882296).
- Binninger, A., Verhoeven, F., Herholz, P., Sorkine-Hornung, O., 2021. Developable approximation via gauss image thinning. *Computer Graphics Forum* 40, 289–300. doi:[10.1111/cgf.14374](https://doi.org/10.1111/cgf.14374).
- Bischoff, B.S., Botsch, M., Steinberg, S., Bischoff, S., Kobbelt, L., Aachen, R., 2002. Openmesh—a generic and efficient polygon mesh data structure. <https://www.openmesh.org>.
- Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., Lévy, B., 2010. Polygon Mesh Processing. A K Peters.
- Cohen-Steiner, D., Morvan, J.M., 2003. Restricted delaunay triangulations and normal cycle, in: *Proceedings of the nineteenth annual symposium on Computational geometry*, pp. 312–321.
- Crane, K., Pinkall, U., Schröder, P., 2013. Robust fairing via conformal curvature flow. *ACM Transactions on Graphics* 32, 1–10. doi:[10.1145/2461912.2461986](https://doi.org/10.1145/2461912.2461986).
- Desbrun, M., Meyer, M., Schröder, P., Barr, A.H., 1999. Implicit fairing of irregular meshes using diffusion and curvature flow, in: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, p. 317–324.
- Eigensatz, M., Kilian, M., Schiftner, A., Mitra, N.J., Pottmann, H., Pauly, M., 2010. Paneling architectural freeform surfaces 29.
- Eigensatz, M., Pauly, M., 2009. Positional, metric, and curvature control for constraint-based surface deformation. *Computer Graphics Forum* 28, 551–558. doi:[10.1111/j.1467-8659.2009.01395.x](https://doi.org/10.1111/j.1467-8659.2009.01395.x).
- Eigensatz, M., Sumner, R.W., Pauly, M., 2008. Curvature-domain shape processing. *Computer Graphics Forum* 27, 241–250. doi:[10.1111/j.1467-8659.2008.01121.x](https://doi.org/10.1111/j.1467-8659.2008.01121.x).
- Fang, Q., Zhao, Z.Y., Liu, Z.Y., Liu, L., Fu, X.M., 2020. Metric first reconstruction for interactive curvature-aware modeling. *Computer-Aided Design* 126, 102863.
- Hopf, H., 1989. *Differential Geometry in the Large*. Springer Berlin Heidelberg, Berlin. doi:[10.1007/3-540-39482-6](https://doi.org/10.1007/3-540-39482-6).

- Jacobson, A., Panozzo, D., et al., 2018. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>.
- Jiang, C., Wang, H., Ceballos Inza, V., Dellinger, F., Rist, F., Wallner, J., Pottmann, H., 2021. Using isometries for computational design and fabrication. *ACM Transactions on Graphics (TOG)* 40, 1–12.
- Jin, M., Kim, J., Luo, F., Gu, X., 2008. Discrete surface ricci flow. *IEEE Transactions on Visualization and Computer Graphics* 14, 1030–1043. doi:[10.1109/tvcg.2008.57](https://doi.org/10.1109/tvcg.2008.57).
- Kazhdan, M., Solomon, J., Ben-Chen, M., 2012. Can mean-curvature flow be modified to be non-singular? *Computer Graphics Forum* 31, 1745–1754.
- Kilian, M., Monszpart, A., Mitra, N.J., 2017. String actuated curved folded surfaces. *ACM Transactions on Graphics* 36, 1. doi:[10.1145/3072959.3015460](https://doi.org/10.1145/3072959.3015460).
- Krivoshapko, S., Ivanov, V., 2015. *Encyclopedia of Analytical Surfaces*. Springer.
- Kühnel, W., Steller, M., 2005. On closed weingarten surfaces. *Monatshefte für Mathematik* 146, 113–126. doi:[10.1007/s00605-005-0313-4](https://doi.org/10.1007/s00605-005-0313-4).
- Li, C., Pan, H., Liu, Y., Tong, X., Sheffer, A., Wang, W., 2017. Bendsketch: modeling freeform surfaces through 2d sketching. *ACM Transactions on Graphics* 36, 1–14. doi:[10.1145/3072959.3073632](https://doi.org/10.1145/3072959.3073632).
- Liu, D., Pellis, D., Chiang, Y.C., Rist, F., Wallner, J., Pottmann, H., 2023. Deployable strip structures. *ACM Transactions on Graphics* 42, 1–16. doi:[10.1145/3592393](https://doi.org/10.1145/3592393).
- Pan, H., Liu, Y., Sheffer, A., Vining, N., Li, C.J., Wang, W., 2015. Flow aligned surfacing of curve networks. *ACM Transactions on Graphics* 34, 1–10. doi:[10.1145/2766990](https://doi.org/10.1145/2766990).
- Pellis, D., Kilian, M., Pottmann, H., Pauly, M., 2021. Computational design of weingarten surfaces. *ACM Transactions on Graphics* 40, 1–11. doi:[10.1145/3450626.3459939](https://doi.org/10.1145/3450626.3459939).
- Pellis, D., Wang, H., Kilian, M., Rist, F., Pottmann, H., Müller, C., 2020. Principal symmetric meshes. *ACM Transactions on Graphics* 39. doi:[10.1145/3386569.3392446](https://doi.org/10.1145/3386569.3392446).
- Su, K., Li, C., Zhou, Y., Xu, X., Gu, X., 2019. Discrete calabi flow: A unified conformal parameterization method. *Computer Graphics Forum* 38, 707–720. doi:[10.1111/cgf.13873](https://doi.org/10.1111/cgf.13873).
- Tang, C., Sun, X., Gomes, A., Wallner, J., Pottmann, H., 2014. Form-finding with polyhedral meshes made simple. *ACM Transactions on Graphics (TOG)* 33, 1–9.
- Tellier, X., Douthe, C., Baverel, O., Hauswirth, L., 2023. Designing funicular grids with planar quads using isotropic linear-weingarten surfaces. *International Journal of Solids and Structures* 264, 112028. doi:[10.1016/j.ijsolstr.2022.112028](https://doi.org/10.1016/j.ijsolstr.2022.112028).
- Tellier, X., Douthe, C., Hauswirth, L., Baverel, O., 2019. Linear weingarten surfaces for conceptual design. *Proceedings of the International fib Symposium on Conceptual Design of Structures*, 225–232.
- Tellier, X., Douthe, C., Hauswirth, L., Baverel, O., 2020a. Form-finding with isotropic linear weingarten, in: *Advances in Architectural Geometry*.
- Tellier, X., Douthe, C., Hauswirth, L., Baverel, O., 2020b. Linear-weingarten membranes with funicular boundaries. *Structural Concrete* 21, 2293–2306. doi:[10.1002/suco.202000030](https://doi.org/10.1002/suco.202000030).
- Wang, B., Wang, H., Schling, E., Pottmann, H., 2023. Rectifying strip patterns. *ACM Transactions on Graphics* 42, 1–18. doi:[10.1145/3618378](https://doi.org/10.1145/3618378).
- Wang, H., Pottmann, H., 2022. Characteristic parameterizations of surfaces with a constant ratio of principal curvatures. *Computer Aided Geometric Design* 93, 102074. doi:[10.1016/j.cagd.2022.102074](https://doi.org/10.1016/j.cagd.2022.102074).
- Wardetzky, M., Bergou, M., Harmon, D., Zorin, D., Grinspun, E., 2007. Discrete quadratic curvature energies. *Computer Aided Geometric Design* 24, 499–518. doi:[10.1016/j.cagd.2007.07.006](https://doi.org/10.1016/j.cagd.2007.07.006).
- Weingarten, J., 1861. Ueber eine klasse auf einander abwickelbarer flächen. *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1861, 382–393. doi:[10.1515/crll.1861.59.382](https://doi.org/10.1515/crll.1861.59.382).
- Weingarten, J., 1863. Ueber die flächen deren normalen eine gegebene fläche berühren. *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1863, 61–63. doi:[10.1515/crll.1863.62.61](https://doi.org/10.1515/crll.1863.62.61).
- Yang, N., Quan, Z., Zhang, D., Tian, Y., 2014. Multi-morphology transition hybridization cad design of minimal surface porous structures for use in tissue engineering. *Computer-Aided Design* 56, 11–21. doi:[10.1016/j.cad.2014.06.006](https://doi.org/10.1016/j.cad.2014.06.006).