

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository:<https://orca.cardiff.ac.uk/id/eprint/160032/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Chen, Rulin, Qiu, Pengyun, Song, Peng, Deng, Bailin , Wang, Ziqi and He, Ying 2023. Masonry shell structures with discrete equivalence classes. *ACM Transactions on Graphics* 42 (4) , 115. 10.1145/3592095

Publishers page: <https://doi.org/10.1145/3592095>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



# Supplementary Material for Masonry Shell Structures with Discrete Equivalence Classes

This supplementary material is composed of two parts. The first part provides implementation details about our base mesh optimization in Section 5.1 of the paper. The second part shows the input surface and optimized base mesh for each result shown in Figure 13 of the paper.

## 1 Implementation of Base Mesh Optimization

We solve the base mesh optimization problem using ShapeOp library [Bouaziz et al. 2012; Bouaziz et al. 2014]. In this optimization problem, the search space is the vertices  $\{\mathbf{v}_i\}$  of the base mesh, and the objective functions  $E_1$  and  $E_2$  consist of six terms, i.e.,  $E_{\text{edge}}$ ,  $E_{\text{dihed}}$ ,  $E_{\text{planar}}$ ,  $E_{\text{surf}}$ ,  $E_{\text{smth}}$ , and  $E_{\text{polygon}}$ . Among the six terms, we have provided formulas of  $E_{\text{surf}}$  and  $E_{\text{smth}}$  expressed as functions of mesh vertices  $\{\mathbf{v}_i\}$  in the paper.

We also provided the following formulas for  $E_{\text{edge}}$ ,  $E_{\text{dihed}}$ ,  $E_{\text{planar}}$ , and  $E_{\text{polygon}}$  in the paper:

$$\begin{aligned} E_{\text{edge}} &= \sum_k \sum_i \|e_k^i - \bar{e}_k\|^2, \quad 1 \leq k \leq K_E, \\ E_{\text{polygon}} &= \sum_k \sum_i s(F_k^i, \bar{F}_k), \quad 1 \leq k \leq K_F, \\ E_{\text{planar}} &= \sum_i P(F_i), \\ E_{\text{dihed}} &= \sum_k \sum_i \|\alpha_k^i - \bar{\alpha}_k\|^2, \quad 1 \leq k \leq K_D, \end{aligned}$$

However, some of the above formulas are not expressed as functions of mesh vertices  $\{\mathbf{v}_i\}$ . This section explains how  $E_{\text{edge}}$ ,  $E_{\text{polygon}}$ ,  $E_{\text{planar}}$  and  $E_{\text{dihed}}$  can be reformulated using mesh vertices  $\mathbf{v}_i$  such that the optimization problem can be solved using ShapeOp library.

### 1.1 $E_{\text{edge}}$ , $E_{\text{polygon}}$ and $E_{\text{planar}}$

The terms  $E_{\text{edge}}$ ,  $E_{\text{polygon}}$  and  $E_{\text{planar}}$  involve a set of constraints  $\{C_i\}$  regarding the mesh vertex positions. Following the solving strategy in [Bouaziz et al. 2012], in each iteration we update the definition of each term according to the projections of the vertex positions to the feasible set of the constraints. Specifically,  $E_{\text{edge}}$ ,  $E_{\text{polygon}}$  and  $E_{\text{planar}}$  can each be written in the following form in the  $k$ -th iteration:

$$\sum_i \|\mathbf{Q}_i \mathbf{V} - P_{C_i}(\mathbf{Q}_i \mathbf{V}^{(k)})\|^2,$$

where  $\mathbf{V}$  concatenates the vertex position variables,  $\mathbf{V}^{(k)}$  are the vertex positions in the current iteration, the matrix  $\mathbf{Q}_i$  selects the relevant vertices for the constraint  $C_i$  and performs mean-centering on their positions, and  $P_{C_i}(\cdot)$  is the closest projection of point positions onto the feasible set of  $C_i$ . The computation of the projections is explained below.

#### 1.1.1 Point-to-point distance Projection

Recall  $E_{\text{edge}}$  and  $E_{\text{polygon}}$  involve constraints regarding the lengths of boundary edges and diagonals of the polygons, respectively. Specifically, each constraint specifies a target value  $d_j^*$  for the distance between two vertices  $\mathbf{v}_{j_1}, \mathbf{v}_{j_2}$ . Given two mean-centered positions  $\bar{\mathbf{v}}_{j_1}$  and  $\bar{\mathbf{v}}_{j_2}$ , the projections onto the feasible set are computed as

$$P(\bar{\mathbf{v}}_{jk}) = \frac{d_{ij}^*}{\|\bar{\mathbf{v}}_{j_1} - \bar{\mathbf{v}}_{j_2}\|} \bar{\mathbf{v}}_{jk}, \quad k = 1, 2.$$

#### 1.1.2 Planar Projection

For  $E_{\text{planar}}$ , each constraint require the vertices  $\mathbf{v}_{j_1}, \dots, \mathbf{v}_{j_m}$  from the same polygon to lie on a common plane. Given the mean-centered positions  $\bar{\mathbf{v}}_{j_1}, \dots, \bar{\mathbf{v}}_{j_m}$ , the projections onto the feasible set are computed as

$$P(\bar{\mathbf{v}}_{jk}) = \bar{\mathbf{v}}_{jk} - \mathbf{n}(\mathbf{n} \cdot \bar{\mathbf{v}}_{jk}), \quad k = 1, \dots, m,$$

where  $\mathbf{n}$  is a unit normal vector for the best-fitting plane and is computed as a right singular vector for the matrix  $[\bar{\mathbf{v}}_{j_1}, \dots, \bar{\mathbf{v}}_{j_m}]^T \in \mathbb{R}^{m \times 3}$  corresponding to the smallest singular value.

### 1.2 $E_{\text{dihed}}$

[Bouaziz et al. 2014] introduced a bending energy measuring the squared difference of absolute mean curvatures

$$E_{\text{bending}} = \frac{w}{2} \int_S (|H_f| - |H_g|)^2 dA, \quad (1)$$

where  $H_f$  and  $H_g$  are the mean curvature functions of deformed and undeformed surface respectively. The dihedral angles for each connected faces can be represented as discretized bending energy proposed by [Bouaziz et al. 2014]. Given a mesh with  $n$  dihedral angles,  $E_{\text{dihed}}$  can be formulated by introducing an auxiliary rotation matrices  $\mathbf{R}$  as

$$E_{\text{dihed}} = \sum_{i=1}^n \frac{w}{2} A \|\mathbf{X}_f \mathbf{c} - \mathbf{R} \mathbf{X}_g \mathbf{c}\|_2^2 + \delta_{SO(3)}(\mathbf{R}), \quad (2)$$

where  $A$  is the Voronoi area of the vertex,  $\mathbf{X}_f \mathbf{c}$  and  $\mathbf{X}_g \mathbf{c}$  derived from  $\alpha_k^i$  and  $\bar{\alpha}_k$ , and  $\mathbf{c}$  is the cotangent weights in the Voronoi area, see details in Section 5.4 of [Bouaziz et al. 2014]

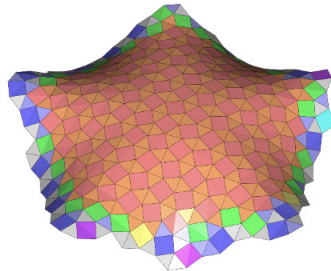
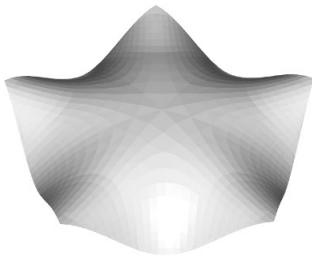
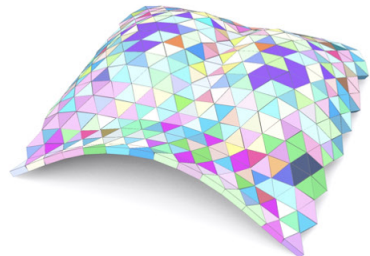
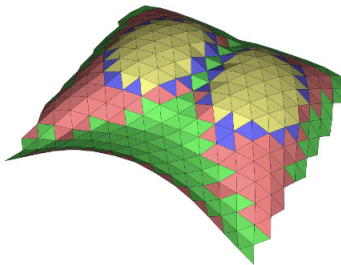
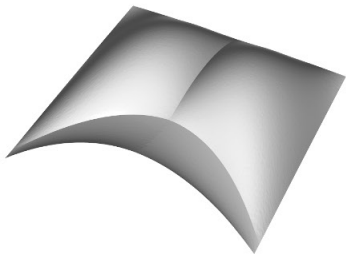
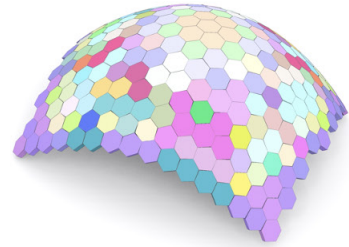
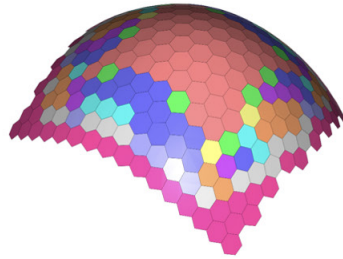
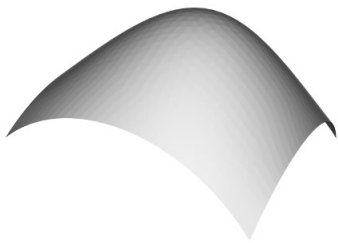
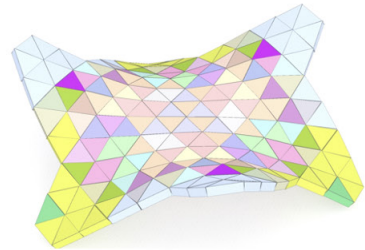
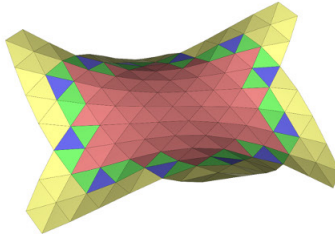
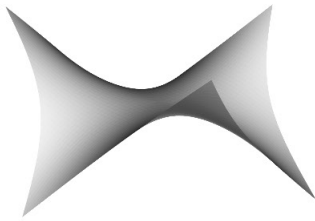
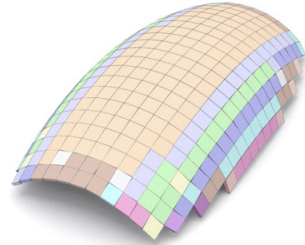
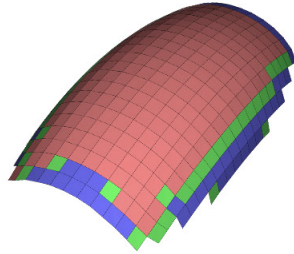
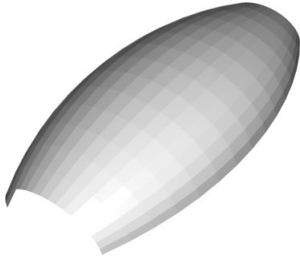
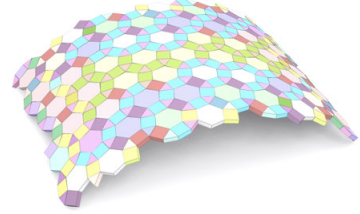
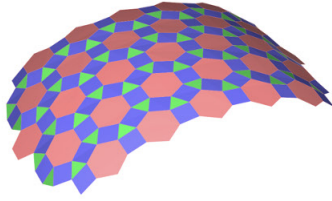
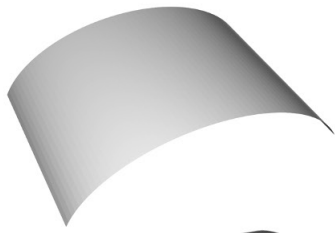
## 2 Input Surfaces and Optimized Meshes

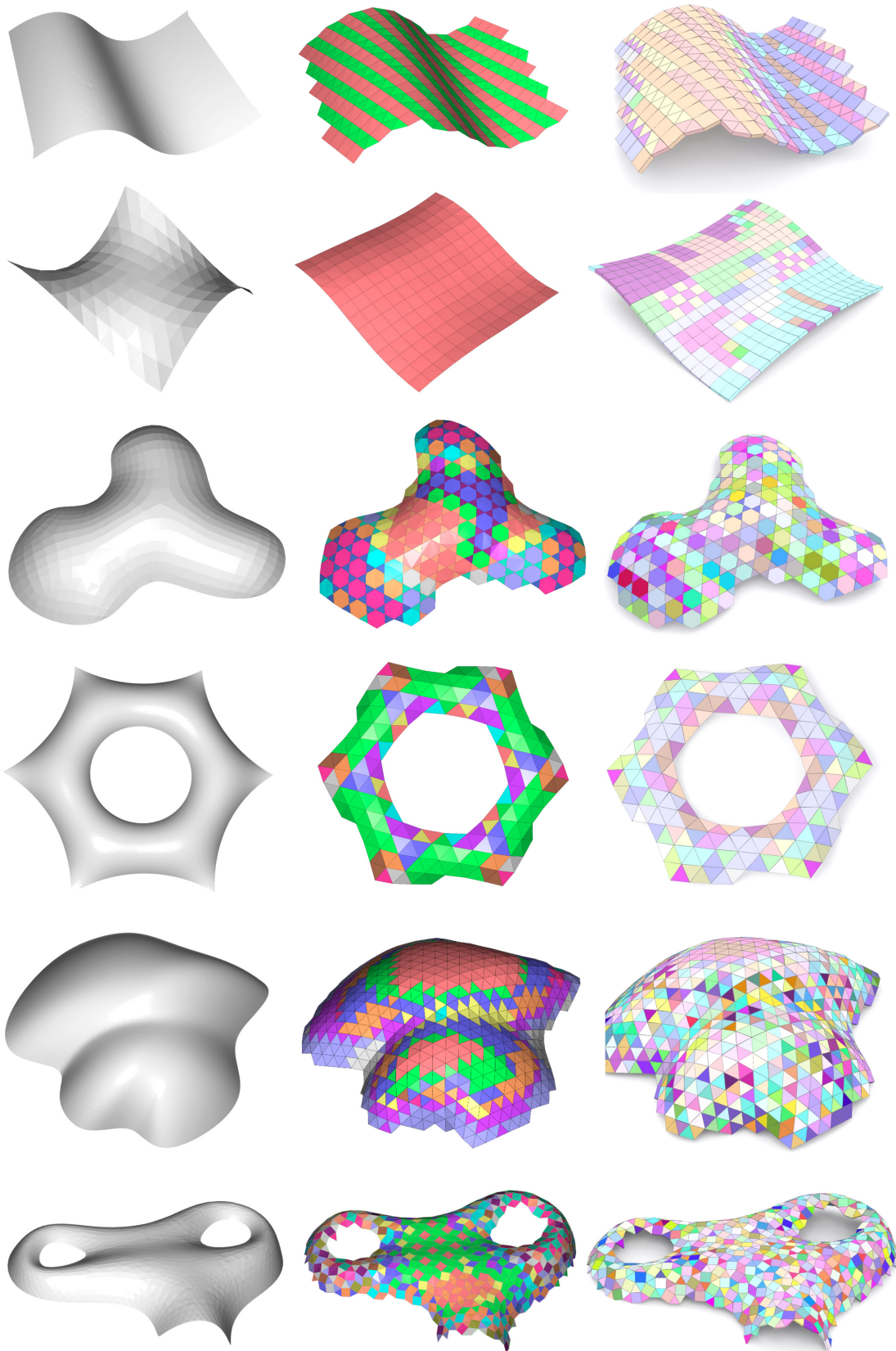
This section provides the input guiding surface and optimized base mesh for each result shown in Figure 13 of our paper. In Figure 1, we can see how input surface is deformed by our mesh optimization in order to improve performance in clustering edges, dihedral angles and polygons. Some surfaces are flattened obviously such as MONKEY SADDLE and ROOF due to the sharp curvature changes of the input surface.

Thanks to our base mesh optimization, the number ( $K_F$ ) of clusters of polygons in the optimized mesh is quite small, usually less than 10; see Table 1 in the paper for details. For the optimized MONKEY SADDLE, we can even use a single polygon to tile the whole surface. In our results, the number ( $K_S$ ) of clusters of elements is much larger than the number ( $K_F$ ) of clusters of polygons, even if we have optimized the augmented angles to reduce  $K_S$  as much as possible, showing that tiling a freeform surface with shell elements is much more challenging than tiling the surface with polygons.

## References

- BOUAZIZ, S., DEUSS, M., SCHWARTZBURG, Y., WEISE, T., AND PAULY, M. 2012. Shape-Up: Shaping discrete geometry with projections. *Comp. Graph. Forum (SGP)* 31, 5, 1657–1667.
- BOUAZIZ, S., MARTIN, S., LIU, T., KAVAN, L., AND PAULY, M. 2014. Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. on Graph. (SIGGRAPH)* 33, 4, 154:1–154:11.





**Figure 1:** We provide the input surface and optimized base mesh for each result shown in Figure 13 of the paper. Polygons in the same cluster of the optimized mesh are rendered with the same color.