Designing Patterns using Triangle-Quad Hybrid Meshes

CHI-HAN PENG, KAUST
HELMUT POTTMANN, TU Wien
PETER WONKA, KAUST

ACM Reference Format:
Chi-Han Peng, Helmut Pottmann, and Peter Wonka. 2018. Designing Patterns using Triangle-Quad Hybrid Meshes. *ACM Trans. Graph.* 37, 4, Article 1 (August 2018), 18 pages. https://doi.org/10.1145/3197517.3201306

ADDITIONAL MATERIAL

Patch graph editing operations. We formulate two kinds of operations - one that works on a 4-sided patch such that the opposite sides have the same length and the other works on a 3-sided patch such that all sides have the same length (see Fig. 13). Here a *side* means a consecutive sequence of patch boundary half-edges between two vertices with corner indices not equal to 6. We parameterize a zigzag as $Z_{a,b,d,x}$, a and b are the corner indices of the two vertices in counter-clockwise order, d is the number of edges between the first vertex and the closest vertex with a non-6 corner index in clock-wise order, and x is the number of edges between the two vertices. (a, b) is either (5,7) or (7,5), d is a non-negative integer number, and d is a positive integer number.

For a 4-sided patch, zigzags are introduced in pairs. The two zigzags of a pair appear on the opposite sides of the patch and they must have the same d and x parameters. For two zig-zag pairs on the two sets of opposite sides to appear at the same time, denoting the first two parameters of the first pair of zigzags as (a_1, b_1), (a_2, b_2) and for the second pair of zigzags as (a_3, b_3), (a_4, b_4), the following condition ensures that the resulting patch boundary is still feasible to be tessellated with a regular hybrid mesh: $a_1 = a_2 = a_3 = a_4$.

For a 3-sided patch, zigzags are introduced in triples. A triple of zigzags, Z_{a_1,b_1,d_1,x_1}, Z_{a_2,b_2,d_2,x_2}, and Z_{a_3,b_3,d_3,x_3}, can be introduced if one of the two following conditions is met: 1) $a_1 = a_2 = a_3 = 5$ and $d_1 + d_2 + d_3 = 2L - 2$, or 2) $a_1 = a_2 = a_3 = 7$ and $d_1 + d_2 + d_3 = L - 1$.

Extension to Equation 11.

The discretization of edges in the surface partition graph structure into both 2-edges and $\sqrt{3}$-edges is done by the following:

$$\begin{aligned}
&\text{find} && e_i^2, e_i^{\sqrt{3}}, 0 \leq i < N^e && \text{such that}\\
&\text{min.} && \sum_i (L_i - 2e_i^2 - \sqrt{3}e_i^{\sqrt{3}})^2, &&\\
&\text{s.t.} && \sum_{0 \leq a < 12} \Big(\sum_{i \in I^{e^2,a,f_j}} |E^{2,a}| e_i^2 + \sum_{i \in I^{e^{\sqrt{3}},a,f_j}} |E^{\sqrt{3},a}| e_i^{\sqrt{3}} \Big) && = 0, \ \forall f_j,
\end{aligned} \tag{12}$$

Here N^e is the number of edges in the graph, e_i^2 and $e_i^{\sqrt{3}}$, $0 \leq i < N^e$, are positive integers indicating the number of 2-edges and $\sqrt{3}$-edges constituting the i-th edge in the graph. L_i, $0 \leq i < N^e$, are the actual lengths of the graph edges in 3D. f_j denotes the j-th face in the graph. a, $0 \leq a < 12$, indicates one of the twelve directions. $|E^{2,a}|$ and $|E^{\sqrt{3},a}|$ are the six-integer vectors of the 2-edge and $\sqrt{3}$-edge in the a-th direction. I^{e^2,a,f_j} and $I^{e^{\sqrt{3}},a,f_j}$ indicate the lists of indices of 2-edges and $\sqrt{3}$-edges circulating f_j in the a-th direction (the first edge of every face is chosen to be at the 0-th direction). In short, the objective function is to minimize the sum of squared errors of the differences between the combinatorial edge lengths $2e_i^2 + \sqrt{3}e_i^{\sqrt{3}}$ and their actual lengths L_i. The constraints are to ensure that for every face, the boundary loop forms a closed loop in the six-integer discrete coordinate system.

Additional results. In Fig. 24 we show the input surfaces as fine polygonal meshes for the examples shown in Fig. 12, Fig. 18, Fig. 19, and Fig. 20. In Fig. 25, we show more results for the British Museum model. In Fig. 26, we show planarity optimization for the Yas Island Hotel results. The 3D details are shown in Fig. 27. In Fig. 28, we show additional rendering for selected results. In Fig. 29, we show the computed edge candidates for selected boundaries.

Authors' addresses: Chi-Han Peng, KAUST, chihan.peng@kaust.edu.sa; Helmut Pottmann, TU Wien, pottmann@geometrie.tuwien.ac.at; Peter Wonka, KAUST, peter.wonka@kaust.edu.sa.

2018. 0730-0301/2018/8-ART1 $15.00
https://doi.org/10.1145/3197517.3201306

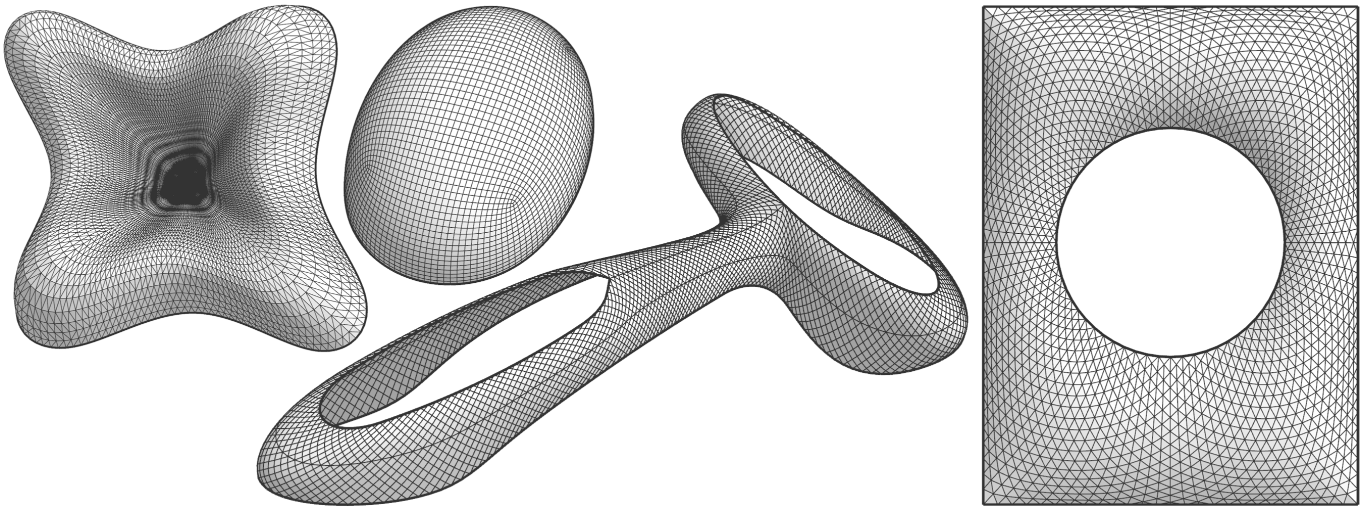

Fig. 24. Input surfaces given as fine polygonal meshes.

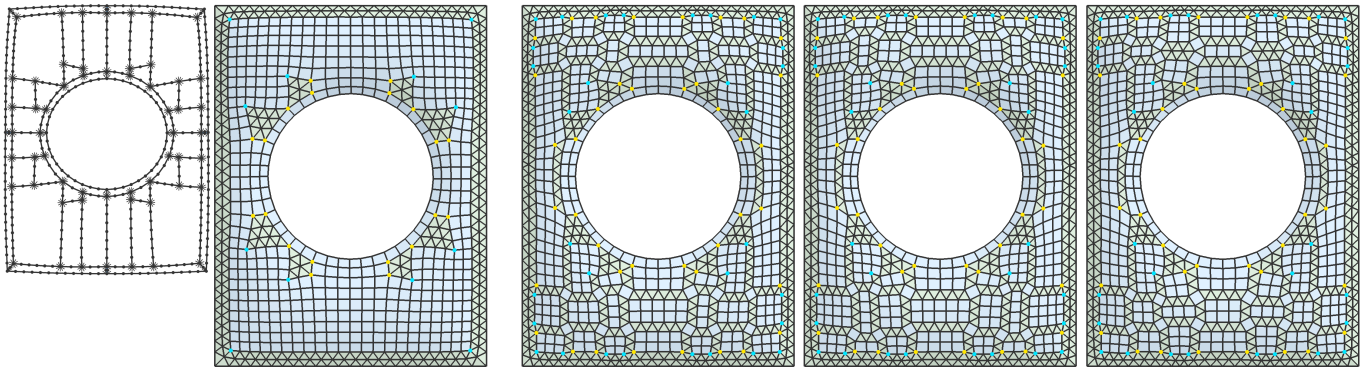

Fig. 25. Left: A 3D hybrid mesh design for the Great Court of British Museum not shown in Fig. 19. Right: Three results of left-right reflective symmetry for the patch graph of Fig. 19 (c).

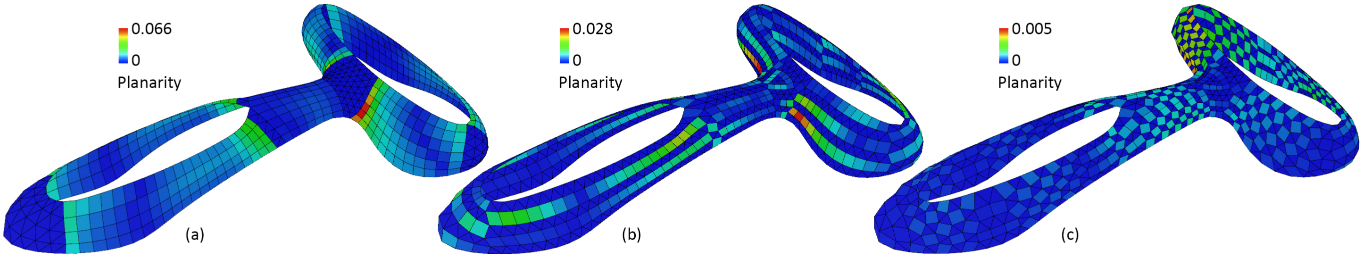

Fig. 26. Meshes (a), (b) and (c) from Fig. 18 after optimization for planarity with the same tolerance for proximity to the reference surface. Similar to the results in Fig. 22, planarity optimization achieved better scores with the fractured patterns in (c). Only pattern (c) could be planarized according to construction requirements.

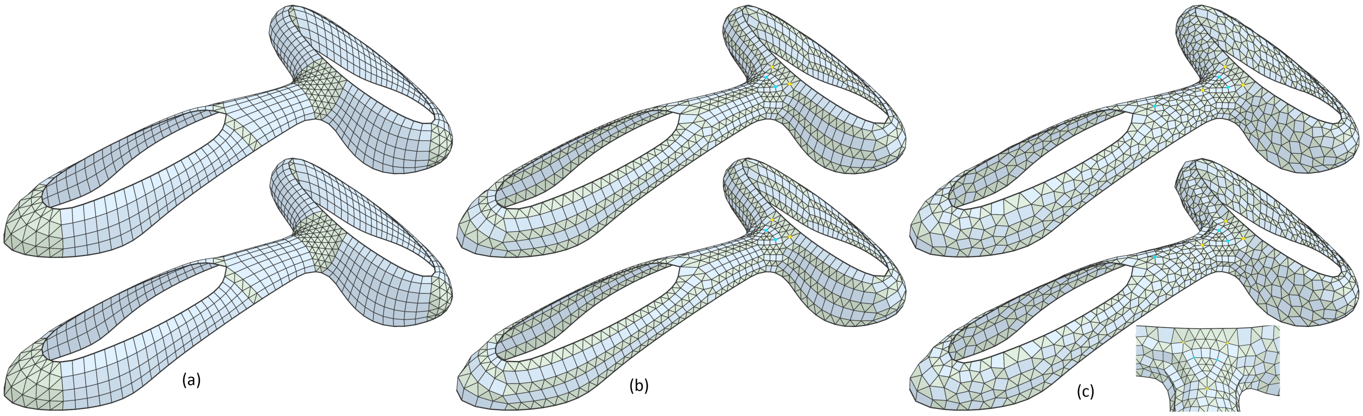

Fig. 27. Interesting 3D details on the planarity optimization results of meshes (a), (b) and (c) from Fig. 18. The top and bottom of each row show the meshes before and after the planarity optimization.

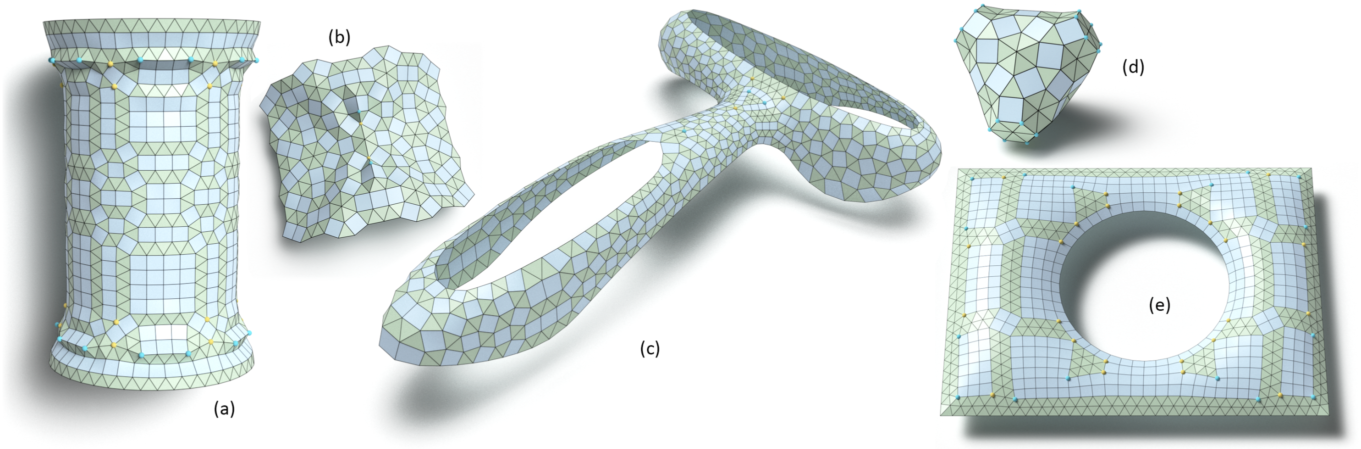

Fig. 28. (a)-(e): Rendering for meshes Fig. 16, Fig. 15 (a), Fig. 18 (c), Fig. 14 tatT, and Fig. 19 (b).

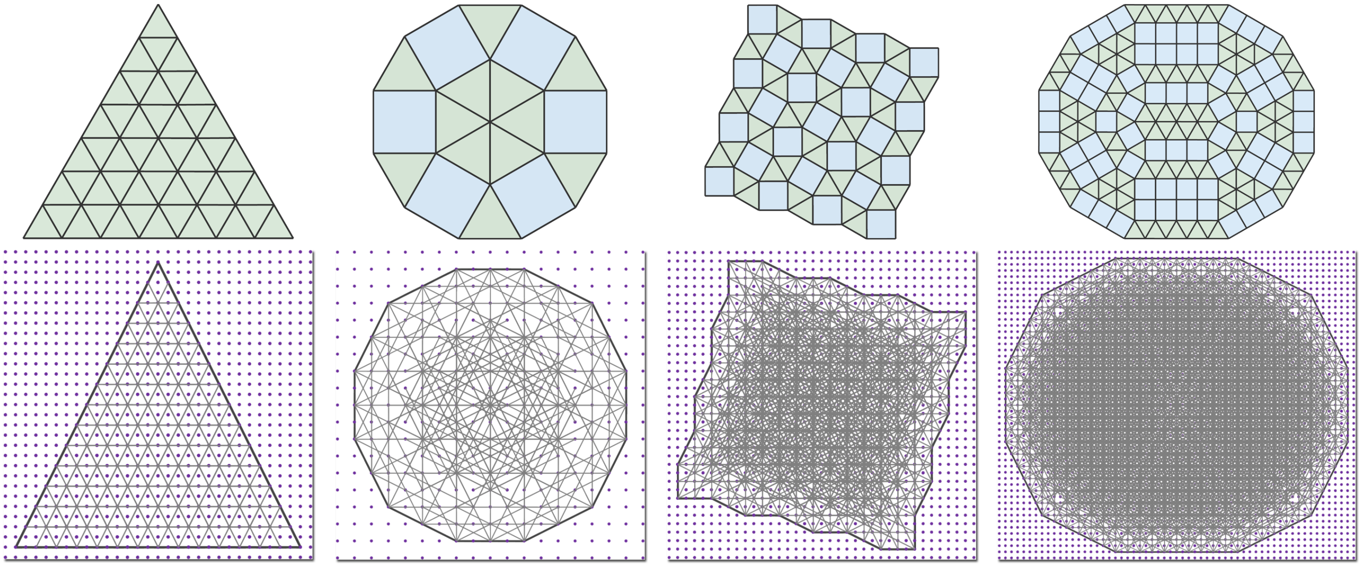

Fig. 29. We visualize the computed edge candidates (interior 2-edges) for four selected boundaries. On the top row we show one possible computed solution inside the boundary. On the bottom row we show all the computed edge candidates. The numbers of generated edge candidates from left to right are: 234, 390, 3946, and 31174.