

## ADDITIONAL MATERIALS

### 7.1 Boundary Editing

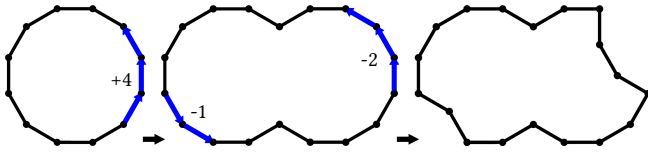


Fig. 26. Boundary editing example. The first edit expands by a magnitude of 4 and with the additional "circular" constraint. The second edit made two subtractions at two different locations with magnitudes of  $-2$  and  $-1$ .

An initial boundary can be given as an arbitrary 2D piece-wise linear loop using the approximation method described in Section 3.4 in [Peng et al. 2018]. In addition, we propose an interactive tool to design admissible boundaries. Starting from an admissible boundary, the tool allows users to iteratively edit the boundary while keeping it admissible.

An editing operation replaces a subset of the current boundary with a new sequence of half-edges. For an operation, the user specifies: 1) the subset of the current boundary to replace and 2) the *magnitude* of the operation, which is a signed non-zero integer. The sign of the magnitude denotes whether the operation is an *expansion* or a *subtraction*. For simplicity, we assume the new sequence of half-edges is a subset of a convex loop (i.e., an admissible boundary

without turning angles  $> 180^\circ$ ). We denote the directions of the first and last half-edges of the current subset as  $d_0$  and  $d_1$ . After the replacement, the directions of the first and last half-edges becomes  $(d_0 - x) \bmod 12$  and  $(d_1 + x) \bmod 12$ , shortened as  $D_0$  and  $D_1$ . If  $D_0 \leq D_1$ , we encode the directions of a subset of a convex loop's boundary in counterclockwise order as  $\{E_{D_0}, E_{D_0+1}, E_{D_0+2}, \dots, E_{D_1}\}$  where these  $E_i$  vectors,  $i \in [0, 11]$ , are one of the twelve 4D direction vectors for half-edges (see Eq. 5.1). Otherwise, we encode it in clockwise order as  $\{E_{D_0}, E_{D_0-1}, E_{D_0-2}, \dots, E_{D_1}\}$ . We then solve the following IP problem for the counterclockwise case:

$$\sum X_{D_0} E_{D_0} + X_{D_0+1} E_{D_0+1} + \dots + X_{D_1} E_{D_1} = Z, \quad (12)$$

and for the clockwise case change the order. Length variables  $X_{D_0}$  to  $X_{D_1}$  denote the numbers of half-edges in the convex loop at the specific directions.  $Z$  is the 4D offset vector from the first to the last vertices of the boundary subset to be replaced. Additionally, a *circular* constraint can be added such that all the length variables are non-zero. In summary, solving Eq. 12 gives us a subset of half-edges of a convex loop that seamlessly replace the original subset of boundary. See Fig. 26 and Fig. 27 for examples.

### 7.2 Additional Figures

In Fig. 28, we show all of our variations for the Tokyo 2020 logo design. In Fig. 29, we show the quad mesh of the bunny example. In Fig. 30, we show the control meshes of the 2D pattern results shown in the paper.

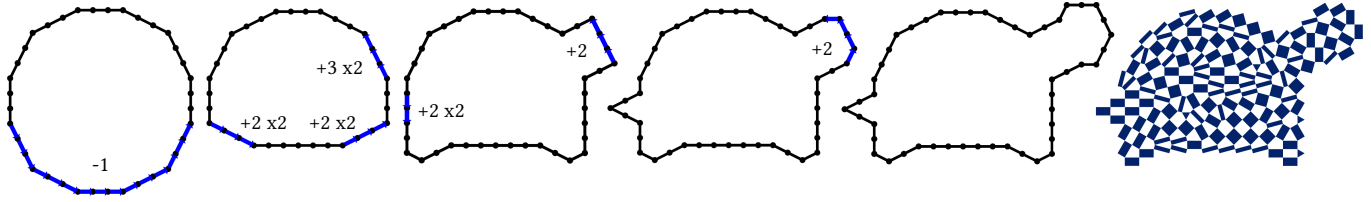


Fig. 27. Making a turtle-shaped boundary using a series of editing operations. "x2" means applies an operation two times.

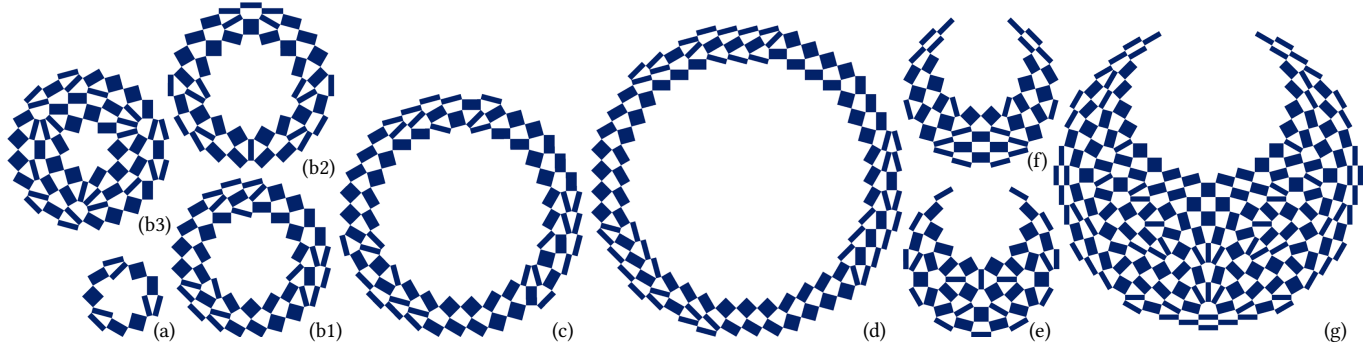


Fig. 28. All of our variations of the Tokyo 2020 logo design. (a), (b1), (c), and (d): we create a series of ring-shaped patterns, similar to the Olympics logo (Fig. 2 (a)), with the same 3-way rotational symmetry but in different scales (measured by the width of the outer boundary). (b1), (b2), and (b3) have the same scale as the original Tokyo design, of which (b2) is a ring-shaped pattern with a left-right reflective symmetry and (b3) is a thicker ring-shaped pattern. (e): a remake of the Paralympic Games logo (Fig. 2 (b)) with a "fractured" global style. (f): a design with the same scale but with a slightly different boundary. (g): a bigger design with a left-right reflective symmetry.

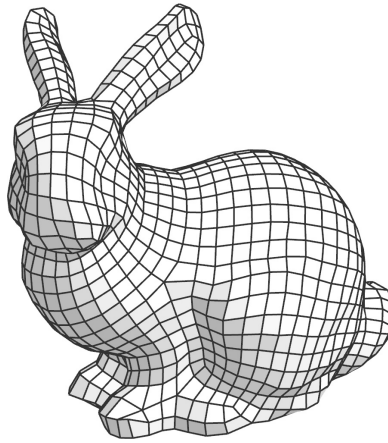


Fig. 29. The quad mesh of the bunny model.

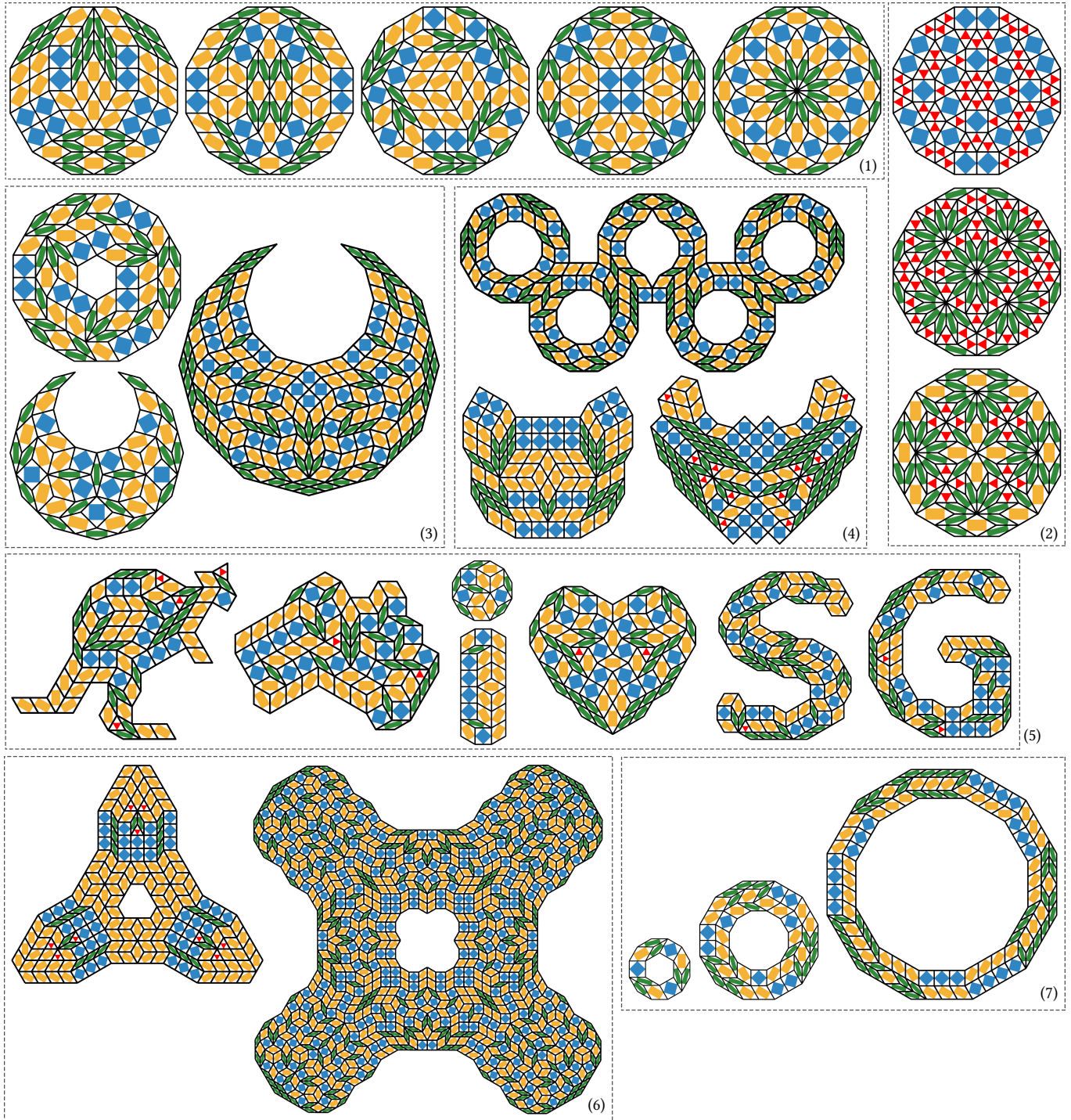


Fig. 30. Control meshes of 2D pattern results shown in the paper. (1): Fig. 14. (2): Fig. 15. (3): Fig. 17. (4): Fig. 18. (5): Fig. 20. (6): Fig. 21. (7): Fig. 28 (a), (b1), and (c). Each type of faces is colored with a different color.