Content-aware Tile Generation using Exterior Boundary Inpainting – Supplemental

SAM SARTOR, College of William & Mary, USA
PIETER PEERS, College of William & Mary, USA

ACM Reference Format:
Sam Sartor and Pieter Peers. 2024. Content-aware Tile Generation using Exterior Boundary Inpainting – Supplemental. *ACM Trans. Graph.* 1, 1 (September 2024), 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 CORNER WANG TILES

Wang tiles suffer from the so-called *"corner problem"* [Cohen et al. 2003; Lagae and Dutré 2005], i.e., each corner texel is shared between all tiles because the tile placed diagonally across a corner is unconstrained and thus each corner needs to match to the corners of any other tile. Lagae and Dutré [2005] introduced Corner Wang tiles as a solution. Corner Wang tiles associate a color with each corner instead of an edge of the tile.

We can also adapt our content-aware tile generation procedure to Corner Wang tiles (Figure 1) following a similar process as Lagae and Dutré [2005] but with the graph-cut based generation of the interior diamond replaced with inpainting. In short, instead of selecting $2C$ square template patches as for Wang tiles, we select C diamond shaped template patches. Each template is cut in four parts by splitting the diamond horizontally and vertically. Each triangular template patch is then copied to the matching corners of the Corner tiles and the interior diamond is inpainted. This process, however, breaks the cardinal rule that we do not verbatim copy patches from the exemplar texture. As discussed in the main paper, this introduces two practical problems:

(1) by verbatim copying patches from the exemplar, the diversity for the corner regions of the tiles exhibit less diversity than the centers that are uniquely synthesized for each tile; and
(2) as also observed by Lagae and Dutré [Lagae and Dutré 2005], the centers of each edge often contain artifacts and/or small discontinuities as these texels are very weakly constrained by the surrounding triangular patches from the exemplar texture.

Figure 2 shows an example of a content-aware generated Corner Wang tiling.

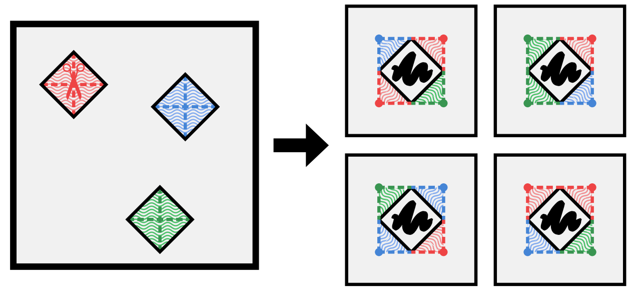

Fig. 1. **Corner Wang Generation:** We follow a very similar generation process as Lagae and Dutré [2005]. Instead of using square template patches, we select C diamond shaped template patches, and cut them both horizontally and vertically. Next, we copy the triangular patches into the corner tiles based on corner colors, and inpaint the interior. Unlike the other tiling schemes, Corner Wang tiles include (parts of) the template patches in the final tiled texture.

Fig. 2. **Content-aware Corner Wang Tiling:** An example of a Corner Wang tiling [Lagae and Dutré 2005] generated with an adaptation of our content-aware tile synthesis method. Unlike regular Wang tiles and Dual Corner Wang tiles, the resulting tiles contain verbatim copied texels from the exemplar image, resulting in a less diverse tiling.

2 PRACTICAL CONSIDERATION: LATENT INPAINTING

At a high level, our technique treats the inpainting model as a black box, which accepts a pixel-space image and mask. However, special care must be taken when the underlying diffusion model operates in a (VAE) latent space. Because we treat the diffusion model as a black box, such a latent diffusion model encodes the boundary conditions before each inpaint step, and then decodes the result

Authors' addresses: Sam Sartor, College of William & Mary, Williamsburg, USA, slsartor@wm.edu; Pieter Peers, College of William & Mary, Williamsburg, USA, ppeers@siggraph.org.

ACM 0730-0301/2024/9-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

again to pixel-space when inpainting is complete. This causes two potential issues:

- Current VAEs are quite lossy, which can cause some color shift over the course of the tile generation process. A potential solution would be to performing tile-generation completely in latent space (i.e., only encode and decode once for the whole tile set). We did not implement this, as this generally requires customization per diffusion model, thereby reducing flexibility.
- VAE encoding also requires care in specifying the boundary conditions, as the encoding/decoding of boundary pixels might be affected by pixels *inside* the mask. We therefore extend the template patch far enough past the actual boundary into the masked area to not influence the latents in the boundary region. Values copied inside the masked area will be overwritten by the inpainting step.

3 NECESSITY OF 3-STEP DUAL TILE GENERATION

Both regular Wang tile textures as well as Corner Wang tile textures can be generated directly from the template patches. This raises the question whether it would be possible to generate Dual Wang Tiles in a single pass with inpainting? For argument's sake, lets assume that we only need infinitely thin boundary conditions in order to generate the interior of a tile. These boundary conditions need to be free of unwanted discontinuities along each edge. This is currently enforced by the construction of template patches for regular Wang tiles by picking a contiguous boundary in the source image. This is a sufficient condition for regular Wang tiles, and hence direct synthesis is possible. However, for Dual Wang tiles we also require that there are no unwanted discontinuities on the corners of the generated tiles, even at a single point. Hence, direct synthesis would therefore require that the boundary conditions would also match where two edges meet. This is difficult to achieve with template patches selected from an exemplar texture. Thus, we must first synthesize a boundary condition with repetition where those boundaries can meet without discontinuity. We resolve this by computing templates which in turn are computed from Wang tiles; in each step we expand the continuity along the edges and corners. Hence the need for a multi-stage generation process.

4 DUAL WANG TILE PACKING

A Dual Wang tile packing is a $C^2 \times C^2$ Wang tiling such that each interior and cross Dual Wang tile occurs exactly once, which ensures minimum storage requirements as well as no visible seams. A Dual Wang tile packing is a specialization of a regular Wang tile packing as illustrated in Figure 3 (left), where the white diamonds are the *interior* Dual Wang tiles that occupy the center of each regular Wang tile, and the black diamonds are the *cross* Dual Wang tiles that straddle the boundaries of four regular Wang tiles. Note that the cross tiles at the boundaries of the packing continue on the opposite side (i.e.*wrap-around* texture), and thus the packing itself is also tileable. Each interior Dual Wang tile is identified by the 4 colors of the encompassing regular Wang tile, and each cross Dual Wang tile is identified by the colors of the four edges incident at the cross tile's center.

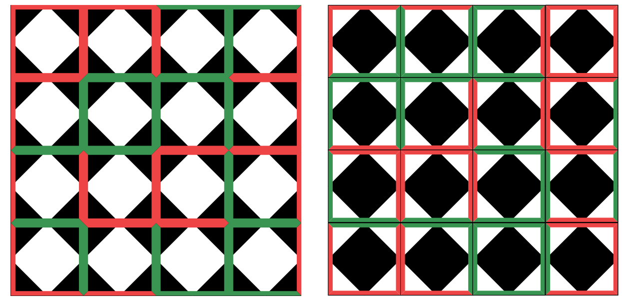

Fig. 3. Left: an example of a 2-color Dual Wang tile packing that includes all possible interior tiles (white) and cross tiles (black). Note, the cross tiles at the edges continue on the opposite side (i.e., wrap around edges). Right: the corresponding complement of the 2-color Dual Wang tile packing where each edge is replaced by its perpendicular edge. Note that the complement is shifted by half a tile, and that the role of interior and cross tiles have swapped. For example, the colors of the edges incident on the first complete black diamond in the Dual Wang tile packing (left) are *red, green, green, red* (N-E-S-W), which corresponds to the first tile in the complement (right).

Brute force computation of a (Dual) Wang tile packing is NP-complete, and only practical for tilings with few colors. For $C = 2$, we found that there exist 9,408 possible packings. For tilings with more colors, even more Dual Wang tile packings exists. However, we are not interested in enumerating each possible packing, but only desire a fast algorithm for generating a single Dual Wang tile packing for any given number of colors. Wei [2004] showed that such algorithm exists for regular Wang tile sets, but while Wei's packing algorithm includes all interior Dual Wang tiles, does not guarantee a complete packing for cross Dual Wang tiles.

To derive a Dual Wang tile packing algorithm, we will first introduce some necessary definitions and tools (Section 5) to aid in defining a generative algorithm. Next, we introduce a packing algorithm for Dual Wang tiles with an odd number of colors (Section 6), followed by a algorithm for Dual Wang tiles with an even number of colors (Section 7).

5 DEFINITIONS

We will denote a 2D tiling as $\mathcal{T}(x, y)$ that for each 2D coordinate (x, y) returns four colors c_N, c_E, c_S, c_W for the north, east, south, and west edge respectively. Furthermore, we assume the tiling wraps around at the edges. For a 2D tiling with C colors, we therefore have:

$$\mathcal{T}(x, y) = \mathcal{T}(x \ mod\ C^2, y \ mod\ C^2). \tag{1}$$

A tiling $\mathcal{T}$ is complete if it contains all C^4 tiles that can be formed with C colored edges (i.e., each tile occurs exactly once), and that are tiled such that the edge colors of neighboring tiles match.

We define the complement $\bar{\mathcal{T}}$ of a 2D Wang tiling as the 2D Wang tiling where each edge is replaced by its perpendicular edge going through the edge's center. The complement is related to a regular tiling by:

$$\bar{\mathcal{T}}(x, y) = \{\mathcal{T}(x, y)_E, \mathcal{T}(x + 1, y)_S, \mathcal{T}(x, y + 1)_E, \mathcal{T}(x, y)_S\}. \tag{2}$$

Figure 3 (right) shows the complement of a 2-color Dual Wang tile packing. Note that the role of interior and cross tiles switches

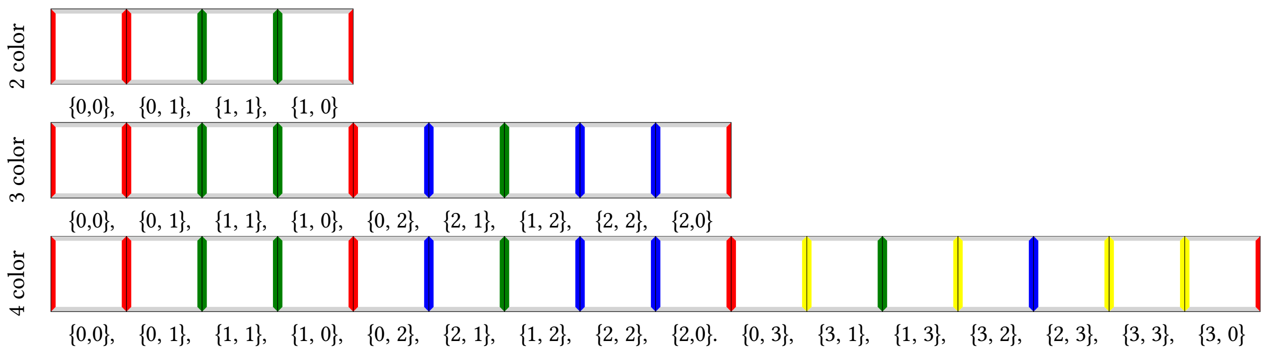

Fig. 4. Domino strings generated using algorithm 1 for 2, 3 and 4 colors. For clarity, we also list the (numerical representation of) color-pairs for each domino tile. Note that each sequence extends the previous sequence.

between the regular and complement. Consequently, a $C^2 \times C^2$ tiling $\mathcal{T}$ is a C-color Dual Wang tile packing if both $\mathcal{T}$ and $\bar{\mathcal{T}}$ are complete (i.e., it contains exactly all C^4 interior and C^4 cross tiles once).

To aid in defining the generative algorithms we will also consider a 1D domino string (i.e., a tiling with only 2 edges) denoted by $\mathcal{D}(x) = \{c_W(x), c_E(x)\}$. We will also assume that these 1D domino strings wrap around at the outer edges. We will differentiate between horizontal and vertical domino strings when tiled in a 2D domain as $\mathcal{D}(x)$ and $\mathcal{D}(y)$ respectively. A C^2-length domino string with C colors is complete if it contains all C^2 possible domino tiles (i.e., each tile occurs exactly once).

Finally, each $C^2 \times C^2$ tiling can be written as a combination of C^2 different horizontal and vertical C^2-length domino strings:

$$\mathcal{T}(x, y) = \{\mathcal{D}_x(y)_N, \mathcal{D}_y(x)_E, \mathcal{D}_x(y)_S, \mathcal{D}_y(x)_W\}, \tag{3}$$

where $\mathcal{D}_x$ and $\mathcal{D}_y$ indicate the x-th vertical domino string and y-th horizontal domino string respectively. Given the tiling $\mathcal{T}$, we can define the complementary domino strings $\bar{\mathcal{D}}_x$ and $\bar{\mathcal{D}}_y$ as the corresponding horizontal and vertical domino strings that constitute the complement tiling $\bar{\mathcal{T}}$.

Our goal for developing a generative algorithm is to derive a fast algorithm for computing a valid sequence $\mathcal{D}_x$ and $\mathcal{D}_y$ such that the resulting tiling is a Dual Wang tile packing.

6 ODD COLOR DUAL WANG TILE PACKING

We will first start by deriving a generative algorithm for a Dual Wang tile packing with an odd number of colors C.

We start with the following key observation. If $\mathcal{D}(x)$ is a complete C^2-length domino string and given a tiling $\mathcal{T}(x, y)$, where $\mathcal{D}_y(x) = \mathcal{D}(x \pm y)$, then each $\bar{\mathcal{D}}_x(y)$ is also a complete domino string, and:

$$\bar{\mathcal{D}}_x(y) = \bar{\mathcal{D}}(y \pm x) = \mathcal{D}'(x \pm y), \tag{4}$$

where $\mathcal{D}'$ is identical to $\mathcal{D}$ when shifting in the positive direction, and mirrored when shifting in the negative direction. Equation (4) follows trivially from the fact that $\mathcal{D}(x)_E = \mathcal{D}(x+1)_W$, and $\mathcal{D}(x-1)_E = \mathcal{D}(x)_W$, and thus the domino formed by the colors from the corresponding edges in neighboring domino strings (shifted by 1) corresponds to the domino at the corresponding position in the original domino string. Since the domino string is complete, it follows that the string formed by corresponding edges must therefore also be complete (Figure 5).

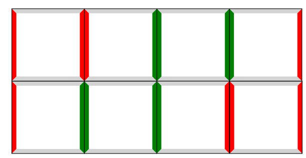

Fig. 5. Shifting the top row domino string one position to the left in the subsequent row, produces a complete sequence of vertical edge combinations. Note, the *red-red* combination occurs at the edges, followed by (left to right), *red-green*, *green-green*, and *green-red*, which exactly matches the domino-string in the top-row.

However, complete horizontal and vertical domino strings are not a sufficient condition for the resulting tiling to be complete. In order for the resulting (shifted) tiling to be complete, each element from the horizontal string $\mathcal{D}(y)$ needs to be combined with each element in the vertical string $\mathcal{D}(x)$. This can be achieved by shifting the horizonal and vertical domino strings in opposite directions:

$$\mathcal{T}(x, y) = \{\mathcal{D}(y - x)_N, \mathcal{D}(x + y)_E, \mathcal{D}(y - x)_S, \mathcal{D}(x + y)_W\}. \tag{5}$$

Note if both the horizontal and vertical domino string are shifted in the same direction, the north/east and south/west edge would always be the same.

Combining both observations, yields that the resulting tiling $\mathcal{T}$ and its complement $\bar{\mathcal{T}}$ are complete, and thus $\mathcal{T}$ forms a valid Dual Wang tile packing.

There exist many possible complete domino strings for C colors. Algorithm 1 provides a generative algorithm for generating a complete C-colored domino string. This algorithm exploits the idea that interleaving all colors one by one in between a fixed color, yields all combinations for that given fixed color on either the north/east and south/west edge. By ensuring that each sequence of tiles (for a given fixed color) starts with the same color, we can concatenate the sequences without loss of a tile combination. The algorithm for generating domino strings is not limited to odd colored domino strings only; Figure 4 shows complete domino strings generated using algorithm 1 for 2, 3 and 4 colors.

ALGORITHM 1: Produce a complete C^2-length domino string that contains all C-color domino tiles.

Data: colors C
Result: C^2-length domino string $\mathcal{D}(y)$
$y \leftarrow 0$
for $c \leftarrow 0$ *to* $(C-1)$ **do**
  for $i \leftarrow 0$ *to* $(c-1)$ **do**
    $\mathcal{D}(y+0) \leftarrow \{i, c\}$
    $\mathcal{D}(y+1) \leftarrow \{c, i+1\}$
    $y \leftarrow y+2$
  end
  $\mathcal{D}(y) \leftarrow \{c, 0\}$
  $y \leftarrow y+1$
end

ALGORITHM 2: Produce two C^2-length domino strings containing all C-color dominoes twice, and meeting the conditions in Section 7.

Data: colors C
Result: C^2-length domino strings $\mathcal{D}_0(y)$ and $\mathcal{D}_1(y)$.
$y \leftarrow 0$
for $c \leftarrow 0$ *to* $(C-1)$ *by 2* **do**
  for $i \leftarrow 0$ *to* $(C-1)$ **do**
    `// First domino sequence`
    `// Fixed color 'c' at even edges`
    $\mathcal{D}_0(y-1) \leftarrow \{(i+C-1) \; mod \; C, c\}$
    $\mathcal{D}_0(y+0) \leftarrow \{c, i\}$
    `// Second domino sequence`
    `// Fixed color 'c+1' at odd edges`
    $\mathcal{D}_1(y+0) \leftarrow \{i, c+1\}$
    $\mathcal{D}_1(y+1) \leftarrow \{c+1, (i+1) \; mod \; C\}$
    $y \leftarrow y+2$
  end
end

7 EVEN COLOR DUAL WANG TILE PACKING

Applying the generative algorithm for odd color Dual Wang tile packings to Dual Wang tiles with an even number of colors does not yield a valid packing. As shown in Figure 6, only half of the tiles occur twice and the other half is excluded. When tracking a single domino (e.g., the first domino tile in the horizontal domino string), we observe that for each column, the relative distance traveled in the vertical domino string equals two. Hence, after $C^2/2$ steps (of visiting odd numbered dominoes in the string), we wrap around. Due to the even number of colors, C^2 is also even, and thus, we arrive (after wrap around) at the first domino tile again. In the odd colored case, C^2 is also odd, and thus after wrap around, we arrive at the second domino tile, and subsequently visit the odd dominoes, thus visiting all possible combinations.

For even color Dual Wang tiles, a possible solution would be to use a different domino string template for the vertical domino strings that would be shifted by 2, i.e., $\mathcal{D}_x(y) = \mathcal{D}(y \pm 2x)$. However, this is not possible for complete domino strings. This can easily be proven for the case where $C = 2$. Assign the first color to the first edge. As the string needs to be complete, we will need to include the first color also on the second edge (in order to include the domino with both edges the same first color). However, if we also want to fulfill the shift-by-two condition, then the third edge also needs to be assigned the first color (for including the same combination with both edges the same color), yielding a sequence of three consecutive first colors and only leaving one more edge to be assigned. Consequently, the resulting sequence will never be able to include the domino with both edges assigned the second color. We have also computationally verified that no such sequence exists for $C = 4$.

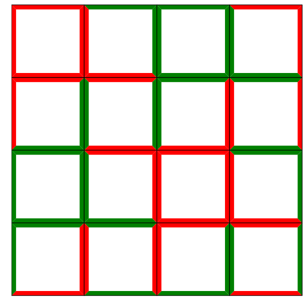

Fig. 6. Invalid packing for an even (2) color tiling generated by shifting a template domino string (algorithm 1) in opposite horizontal and vertical directions. For example, the top-left tile (all red) also occurs at position (2,2).

The previous failed experiment also indicates that in order to obtain a packing that we cannot rely on complete domino strings if using the shifted domino template for the other dimension. Hence, instead of utilizing a single domino string, we opt for using a template of two neighboring domino strings that together are (twice) complete. Consider the case where the colors of the vertical edges are determined by the double domino string template without shifting, and the horizontal edge colors by a shifted single string as before:

$$\mathcal{T}(x,y) = \{\mathcal{D}(x+y)_N, \mathcal{D}_{(x \; mod \; 2)}(y)_E, \mathcal{D}(x+y)_S, \mathcal{D}_{(x \; mod \; 2)}(y)_W\}, \tag{6}$$

where $\mathcal{D}_0$ and $\mathcal{D}_1$ (i.e., the result of $(x \; mod \; 2)$ is either 0 or 1) are the first and second domino string respectively, and $\mathcal{D}_0$ is copied in even number rows and $\mathcal{D}_1$ in odd numbered rows.

To derive the necessary conditions for $\mathcal{D}_0$ and $\mathcal{D}_1$, observe that the first domino $\mathcal{D}(0)$ in the vertical domino string is combined (after shifting) with the even number dominoes in $\mathcal{D}_0$ and the odd dominoes in $\mathcal{D}_1$. Consequently, the union of both: $\{\mathcal{D}_0(2i), \mathcal{D}_1(2i+1)\}_i$ must form a complete sequence. Similarly, the second domino $\mathcal{D}(1)$ visits the complementary sequence $\{\mathcal{D}_1(2i), \mathcal{D}_0(2i+1)\}_i$ which must also be complete.

The above conditions ensure that the resulting tiling is complete, however, it does not ensure that the resulting complementary tiling is also complete. The necessary condition for a complete complementary tiling can be derived in a similar manner. First observe that by copying the template without shift, that each complementary row will be a repetition of the same two dominoes, namely $\{\mathcal{D}_0(y)_N, \mathcal{D}_1(y)_N\}$ and its mirror $\{\mathcal{D}_1(y)_N, \mathcal{D}_0(y)_N\}$. Similarly following the the first element from the complementary vertical domino string, yields that it only visits the even number rows in $\mathcal{D}_0$ and odd rows in $\mathcal{D}_1$. Consequently, by requiring that the sequence $\{\mathcal{D}_{(i \; mod \; 2)}(i)_N, \mathcal{D}_{((i+1) \; mod \; 2)}(i)_N\}_i$ is complete ensures a complete complementary packing.

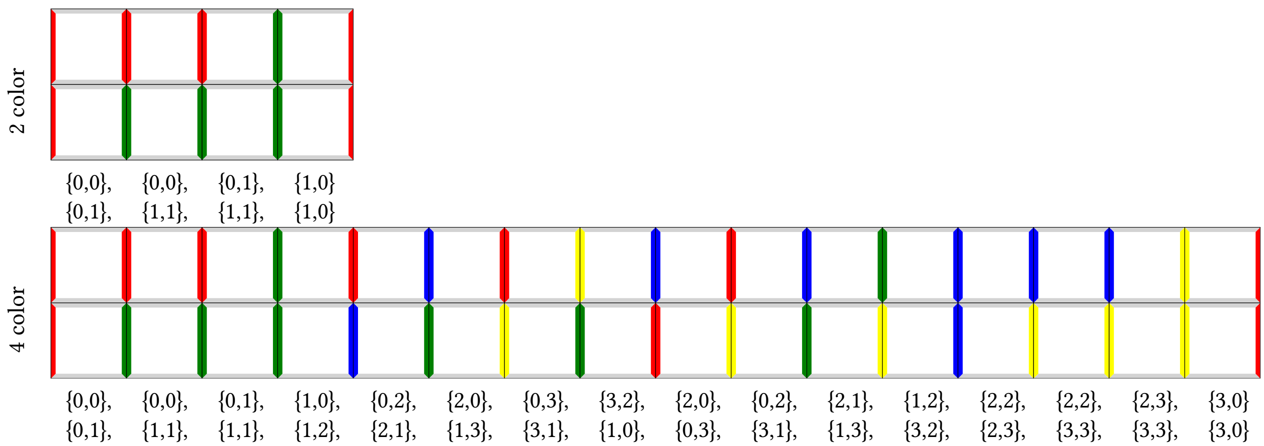

Fig. 7. Double domino strings generated using algorithm 2 for 2 and 4 colors. For clarity, we also list the (numerical representation of) color-pairs for each domino tile. Note that each sequence extends the previous sequence (albeit with the last domino's outer edge's color altered).

Similar as for the complete single domino string, many possible sequences exist that meet both above conditions for the double domino strings. Algorithm 2 provides a generative algorithm to produce a duo of domino strings that meets the above conditions. Intuitively, we place every domino with one edge fixed to any of the even colors in $\mathcal{D}_0$ and the combinations with one edge fixed with an odd color in $\mathcal{D}_1$ as well as switching which edge will be fixed for each (odd vs even edges). The latter ensures that the complement will be complete, whereas the former ensures that the regular (zig-zag) sequences are complete. Figure 7 shows examples of 2 and 4 color template strings.

Figure 8 shows examples of generated Dual Wang tile packings for 2, 3, 4, and 5 colors. We refer to https://github.com/samsartor/content_aware_tiles for a reference implementation of both packing algorithms.

REFERENCES

Michael F Cohen, Jonathan Shade, Stefan Hiller, and Oliver Deussen. 2003. Wang tiles for image and texture generation. *ACM Trans. Graph.* 22, 3 (2003), 287–294.

Ares Lagae and Philip Dutré. 2005. A procedural object distribution function. *ACM Trans. Graph.* 24, 4 (2005), 1442–1461.

Li-Yi Wei. 2004. Tile-based texture mapping on graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*. 55–63.

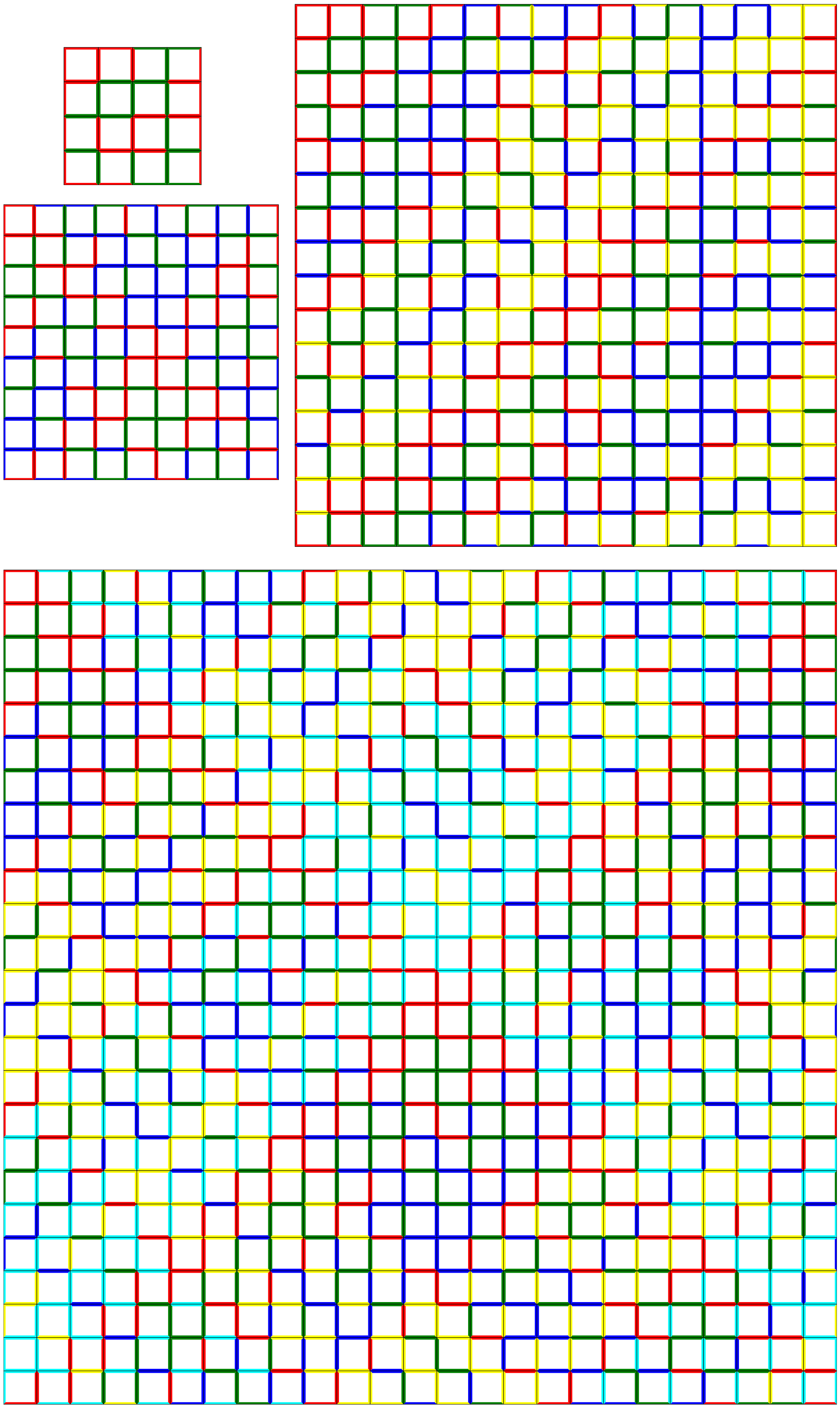

Fig. 8. Dual Wang tile packing for 2, 3, 4, and 5 colors.