DOI: 10.1111/cgf.14610
Eurographics Symposium on Geometry Processing 2022
M. Campen and M. Spagnuolo
(Guest Editors)

Volume 41 (2022), Number 5

Fabricable Multi-Scale Wang Tiles

Xiaokang Liu[1], Chenran Li[1], Lin Lu[† 1], Oliver Deussen[2] and Changhe Tu[1]

[1]Shandong University, China
[2]University of Konstanz, Germany

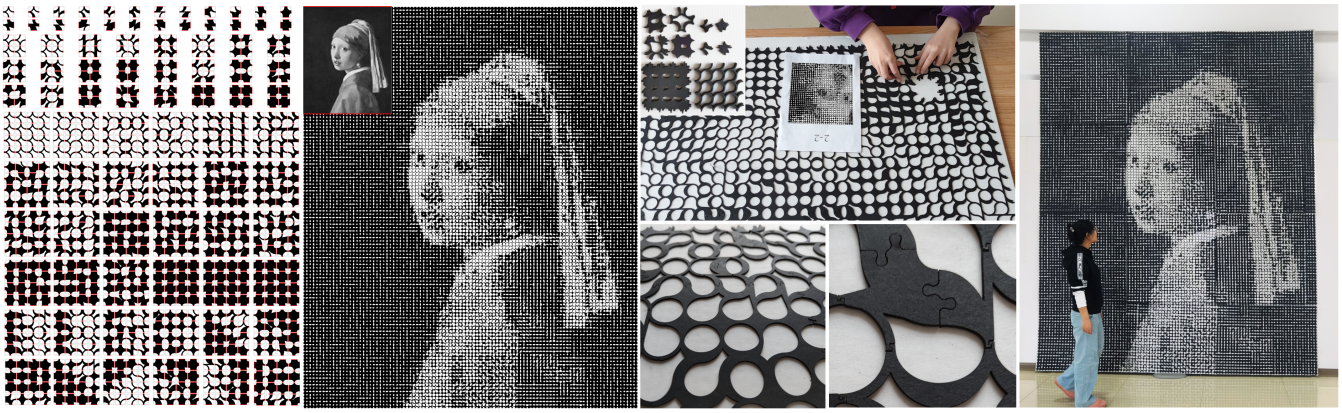


Figure 1: *We design fabricable multi-scale Wang tiles (left), such that they can tile a given image hierarchically with fine details (middle-left). Users can efficiently tile the fabricated tiles following Wang tiling rules (middle-right), and obtain a physical half-toning result (right).*

Abstract

Wang tiles, also known as Wang dominoes, are a jigsaw puzzle system with matching edges. Due to their compactness and expressiveness in representing variations, they have become a popular tool in the procedural synthesis of textures, height fields, 3D printing and representing other large and non-repetitive data. Multi-scale tiles created from low-level tiles allow for a higher tiling efficiency, although they face the problem of combinatorial explosion. In this paper, we propose a generation method for multi-scale Wang tiles that aims at minimizing the amount of needed tiles while still resembling a tiling appearance similar to low-level tiles. Based on a set of representative multi-scale Wang tiles, we use a dynamic generation algorithm for this purpose. Our method can be used for rapid texture synthesis and image halftoning. Respecting physical constraints, our tiles are connected, lightweight, independent of the fabrication scale, able to tile larger areas with image contents and contribute to "mass customization".

CCS Concepts
• ***Computing methodologies*** → ***Shape modeling;*** • ***Applied computing*** → *Computer-aided design;*

1. Introduction

Wang tiles are squared tiles with color-coded edges that have to match. They were first proposed by Wang Hao in 1961 [Wan61] and intended to tile the plane. Due to their ability to aperiodically and stochastically tile the plane with simple rules, they became popular for the procedural synthesis of textures, height fields, and other large and non-repeating datasets [CSHD03, Wei04, KCODL06].

When representing large-size textures or other structures, however, many tiles are needed, which limits their application due to the needed production effort.

† Corresponding author. E-mail: llu@sdu.edu.cn

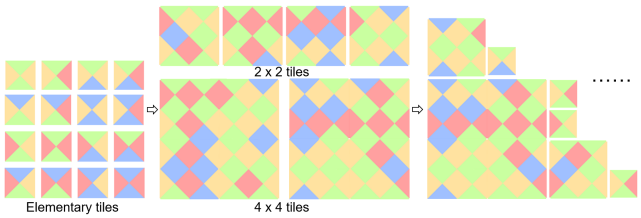


Figure 2: *Generation and tiling procedure for multi-scale tiles: based on a basis tile-set, larger tiles are synthesized. The plane is tiled by using large tiles wherever possible, and filling the gaps with smaller ones.*

Our motivation is therefore to minimize the amount of types and the overall number of pre-fabricated tiles, to reduce the production cost and assembly cost respectively, and meanwhile minimizing visible seams in the results. Moreover, minimizing the density error, to achieve optimal "half-toning" results for given images. To overcome the conflict between four objectives, we need to find an optimal trade-off between tile size and presentation details as well as an efficient algorithm to place them.

Our idea is to introduce the concept of multi-scale Wang tiles, as shown in Figure 2, which allows patches of different scales that are still tileable. Starting from a basic set of Wang tiles, we hierarchically combine the tiles into larger tiles. The essential challenge here is to fight the combinatorial explosion during synthesizing large-scale tiles, since edge constraints and tileability require many different tiles. This is especially the case if tiles have to be physically fabricated, e.g., by 3D printing or laser cutting. We propose a parameterizable method for fabricating tiles that needs such optimization.

Suppose a basic tile edge could have n colors, then the boundary of a $k \times k$ tile would have n^{4k} cases, while the number of combined boundary colors for tiles is $n^{2k(k-1)}$. As a result, the total number of possible $k \times k$ tiles is $n^{4k} \times n^{2k(k-1)} = n^{2k(k+1)}$. Our primary task therefore is to find a smart way for picking “representative” tiles instead of considering all possible tiles. We follow two design principles: 1) We use very limited but "representative" conditions for assigning boundary colors to tiles, i.e., we introduce a representative elementary tile set (RES) and a representative combined tile set (RCS). Tiles of RES and RCS, even though having different sizes, can always be tiled using the Wang tile rules. Both sets are derived from a parametric model and independent to specific input images. 2) We customize multi-scale tiles to represent a given density distribution besides RES and RCS.

Our work is also closely related to half-toning, which deals with representing continuous tones by discrete values. Traditional methods usually optimize the distributions or sizes of picture elements (dots, line segments, etc.) to generate half-toning images. Our method allows to efficiently produce physical half-toning results that can be assembled to large patterns.

Our contributions can be summarized as follows.

- We introduce the concept of multi-scale Wang tiles, targeting for physically synthesizing large scale patterns with variable densities.
- We propose a parametric model for generating a set of representative elementary tiles, and compute a minimal representative multi-scale tile set through matching representative boundaries and densities.
- We demonstrate fabricated multi-scale Wang tiles and their applications.

2. Related Work

Related work was done on Wang tiles and cubes, but also on creating manufacturable structures on the basis of tiles.

2.1. Wang tiles and Wang cubes

Wang tiles were first proposed by Wang Hao in 1961 [Wan61]. While [MRR02] lay the foundations of tiling groups, [CSHD03] is the first to discuss applying Wang tiles in computer graphics. They propose an interactive tool to assist users in designing tiles that meet the constraints of Wang tiles. At the same time, they also provide an automatic generation method, which is a variant of patch-based texture synthesis. Printable structures are not considered.

Since then, many methods to generate textures using Wang tiles have been proposed. In most cases, they use the continuity along the boundaries of the tiles to create stochastic textures. [KCODL06] generate a set of progressive and recursive Wang tiles that contain blue noise sampling patterns. By controlling point density, they can create tiling results representing different input densities, but they do not show structures that can be used for fabrication since the structures on the tiles are not connected. Our method uses smooth curves to generate a set of tiles with different densities, which can be manufactured.

Wei [Wei04] uses Wang tiles for texture synthesis. He generates an arbitrarily large and non-periodic virtual texture map from a small set of stored texture tiles. [FL05] use Wang tiles to achieve textured tilings on arbitrary topological surfaces. They convert an input surface model to a Quad-based geometry by surface mapping. Then they place Wang tiles on all quadrilateral surfaces, and map them back to the input surface. [DJSJ19] generate stochastic wall patterns by designing specific patterns in Wang tiles. [LD06] add corner constraints to the boundary constraints of Wang tiles, which solves the continuity problems near tile corners.

[LEQ*07] extend the non-periodic tiling process of Wang Tiles to Wang Cubes. They automatically generate isotropic Wang Cubes consisting of 3D patterns or textures to simulate various 3D visualization and illustration effects. [DZRN20] generate two-dimensional and three-dimensional microstructures based on Wang tiles, which significantly broaden their applicability.

A very relevant work to us is [BWL18], which also targets physical manufacturing and involves topological constraints in tile design. The proposed method considers interactive design as well as on-the-fly tile synthesis. The difference to our work is that we aim to synthesize tiles with varying structural density to tile various density maps rather than tiles for ornamental usage.

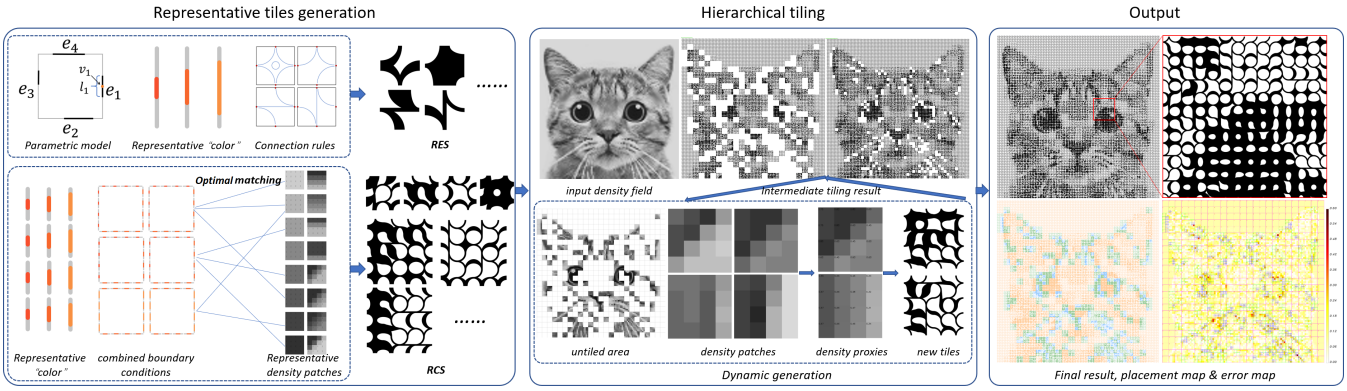


Figure 3: *Overview. Starting from a parametric tile model that satisfies given topological constraints, we first design a set of representative elementary Wang tiles (RES) and then match boundaries and density levels on larger scales to achieve a set of representative combined tiles (RCS, on the left). RES and RCS allow to tile a given density map hierarchically. In addition, during the tiling process we can generate new large-scale tiles on the fly, such that a given density map is represented by an as-small-as-possible set of multi-scale Wang tiles (middle). The placement map (orange for* 4×4 *tiles, green for* 2×2 *tiles, and blue for elementary tiles) and the error map (right) give us control to minimize the error by re-tiling.*

2.2. Manufacturable Tilings

In a number of approaches, researchers tried to create producible tiled structures. Examples are ornaments applied to fabricated 3D objects by synthesizing 1D patterns along curves with restricted topology [ZJL14]. [MDLW15] create flat shapes for laser cutting, while [DLL*15] synthesize 3D patterns by modifying texture synthesis to match a base surface. [YLS*21] embed the geometric tiles on-the-fly for fabrication. [CZX*16] synthesize filigree-like structures along surfaces by using a set of base elements along a target surface. They further synthesize tiled patterns on a base surface [CML*17], such that the planar tiles can be easily printed and assembled. [ZCT16] propose an interactive design method for curve networks as ornamental priors on 3D surfaces. [LLS*20] adapt Escher's dihedral tiling and realize dual shape patterns on the shapes that utilize both solid and hollow spaces. Such tilings can also be applied to B-rep CAD models to achieve 3D jigsaw puzzles [EK21].

Several research works manipulate tiling patterns on the surface to produce nice projected effects via drilling holes [AM12] or tiny tubes [ZLW*16], or to embed the QR codes through reflection effects [PLL*19, PLL*20]. Besides the aesthetics aspect, tiles can also produce specifically defined functionality, like enabling lightweight and thermal comfort in personalized orthopedic casts [ZFD*17, RTY*19]. This is another application with great potential.

3D standard tiles like LEGO®bricks have been studied as a means for improving fabrication efficiency [MMG*14, CLF*18]. Xu et al. [XHFZ19] even propose a method for automatically computing LEGO®assemblies from user sketches. Our work shares the idea to explore the design space with a limited set of fabricable tiles.

2.3. Modeling using discrete equivalence classes

Similar to our work, many other works explore the highly repetitive equivalence classes with the simplest geometric base elements, such as triangles [SS10, HEB15], quads [EKS*10, FLHCO10], triangle-based point-folding structures [ZCBK12] to model a given freeform surfaces, then these repetitive structures will be manufactured and assembled subsequently according to the modeling results, reducing complexity and cost of modeling, manufacturing, and assembly in various applications, such as architectural projects.

However, the above methods generate different equivalence classes for different input models, since they do not modify the topology of the input mesh. In contrast, Liu et al. [LZZ*21] propose a fabrication-error-driven remeshing algorithm, using a set of specified template triangles, which can be easily fabricated and assembled, to explicitly and directly remesh the input meshes. Similar to Liu's work, our method tiles a density map using predefined multi-scale Wang tiles, which are generated according to given densities. And although the equivalence classes are predefined, we both can use different parts of them for different inputs.

2.4. Halftoning and stippling

Halftoning/stippling is a classic technique used in traditional paper printing and digital displays [Kip01], producing well-distributed point sets which play an important role in various computer graphics contexts, such as anti-aliasing, global illumination, non-photorealistic rendering, and so on. Over the past few decades, researchers have worked to improve algorithms to generate better point sets [KCODL06, BSD09, DGBOD12, LLCY15, DSZ17], or stylized lines [Ahm14]. In addition to work on generating halftoning and stippling patterns in 2D, another part of the work [Stu97, ZC09, ZLW*16] focuses on 3D digital halftoning for transforming continuous density objects into binary rendition through additive

manufacturing technologies. In this paper, we replace the points in other work with our multi-scale Wang tiles of different densities and optimize the number of the tile rather than the position to represent a halftoning result of a continuous-density image.

3. Overview

The goal of multi-scale Wang tiles is to efficiently tile a potentially large plane with a small set of tiles, such that the tiled result represents the given density map with high fidelity. Thus, tiles need to have a large coverage of densities and at the same time limited boundary conditions.

Considering physical fabrication requirements, we use binary structures and apply *topological constraints* to the tiles, to guarantee that they can be fabricated and have the necessary stability. Specifically, we create structures with a single connected component for both elementary and larger tiles. Besides, the tiles are allowed to rotate during physical tiling, which is more flexible than classic Wang tiles.

As shown in Figure 3, we propose a two-stage algorithm to generate fabricable multi-scale Wang tiles: First, we introduce a parametric model of elementary tiles, considering topology constraints and density coverage of the structures on the tile. This way, we can represent thinner and thicker structures, or in the case of representing images, lighter and darker parts. From this parametric model, we select representative boundaries, create the RES by an *optimal matching* algorithm, and come up with a solution with representative boundaries and densities. Subsequently, we construct the set of larger combined tiles (RCS) following the same optimal matching between boundaries and densities. To optimize the usage of tiles, we propose a dynamic generation method for producing new multi-scale tiles from the RES customized for a given image or density map.

4. Parametric model of multi-scale Wang tiles

4.1. Parametric model

We define our model on a unit rectangle with binary structures inspired by Truchet tiles [Car18] by restricting the pairwise connection to be arcs and straight lines, as illustrated in Figure 4. Attached to each edge is a black segment with a controllable position and width to represent the *boundary patterns*. The segment is represented by a pair of parameters (v_i, l_i), where $v_i \in (0,1)$ is the center of the segment and $l_i \in (0,1)$ denotes its width. Considering the topological constraints of having only one single black component on each edge, there are four *connection rules* selected out for the boundaries of each tile.

Considering some underlying aesthetic factors, we generate our patterns using elliptic curves, ensuring smooth curvature transition at the junction of the boundary. Two curves are created for each connection between two edges, which, together with the area sandwiched between them, form a black arc with varying widths. Furthermore, all lines are connected end to end to form a single piece, satisfying the single-connected constraints while allowing the pattern to have a certain degree of separation for visual clarity and smoothness.

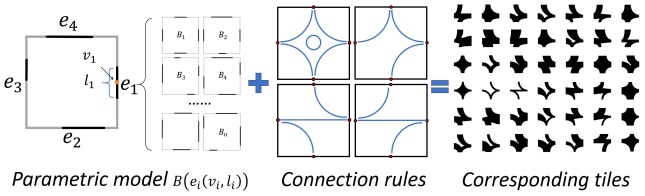


Figure 4: *Elementary tiles are designed based on boundary patterns and connection rules.*

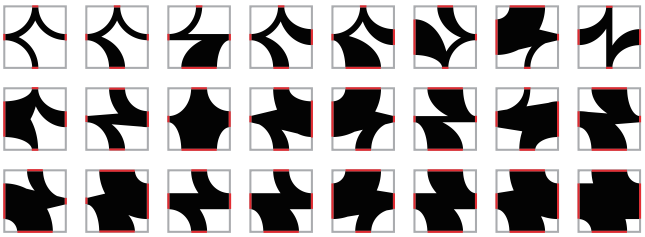


Figure 5: *RES generated according to 24 types of boundaries and target density.*

More complex connection rules can, of course, be used to generate tiles as long as the connection pattern is consistent, but they also make the interior more crowded, reducing the expressive power of the pattern.

Based on this parametric model, we can design a series of elementary tiles. The density of a tile is determined by the ratio of the area covered by the black component.

4.2. Representative elementary tile set (RES)

A set of elementary tiles should possess limited boundary colors, while having a large and uniform coverage of possible density levels. To fulfill this goal, we discretize the boundary patterns into limited colors, and optimize the inner patterns such that the resulted densities are distributed uniformly on the requested range of densities. Practical fabrication limits the minimal density on the tiles, and it depends also on the physical size of the unit tile.

We center the boundary pattern ($v_i = 0.5$) and sample it into three levels of low, medium, high for l_i, representing the trade-off between a high freedom of boundary parameters and the faithfullness of the density representation. Other density levels can be implemented in addition, but a combinatorial explosion cannot be avoided in this case. Suppose a target density in the range of [0.1, 0.9], we solve the parametric model, and reach the density value $\{0.1, 0.5, 0.9\}$.

Since the relationship between edge width and density is nonlinear, we use the bisection method to inversely find the three width values of $l_i = \{0.034, 0.26, 0.627\}$, which represent three different colors of unit edges. As the boundary pattern is set at the center ($v_i = 0.5$), our tiles are allowed to rotate without introducing new

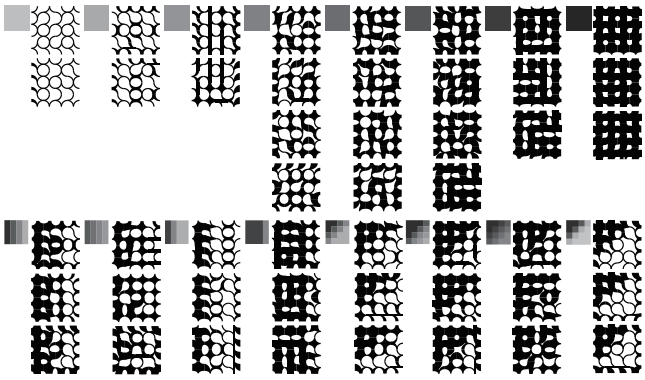


Figure 6: *The representative density patches (top left corners) and the RCS of* 4×4 *tiles.*

edges. However, many different cases of the total $3^4 = 81$ boundaries, such as "1211", "1121" and "1112", will become rotationally symmetric, leading to a lot of redundant tiles in the next generation step. After eliminating such boundaries, we get 24 cases left, as shown in Figure 5.

To generate optimal inner patterns according to the boundaries, we formulate our problem as an optimal matching problem to match each boundary with its "favorite" pattern. With 24 representative boundaries, we sample 24 target densities from o.1 to 0.9 to construct a weighted bipartite graph. For each pair of boundary and density, we generate tiles using the above connection rules. The one whose density is closest to the target is regarded as the corresponding tile of such a pair; then, the density error between it and the target is the weight of this pair. With the weighted bipartite graphs, we employ the Hungarian algorithm, also called Kuhn–Munkres algorithm [Kuh55] to solve this matching problem between the boundaries and representative densities by finding a perfect matching with the minimal sum of weights of all matched edges. Based on the perfect matching, we construct a RES, as shown in Figure 5.

4.3. Representative combined tile set (RCS)

To reduce the total number of tiles, we compose representative $k \times k$ tiles by composing elementary tiles that satisfying the Wang tile rules. The major issue here is to pick representative boundary colors and assign proper inner structures.

4.3.1. Representative boundary colors

For a $k \times k$ tile, one of its edges can be seen as a "combined edge" composed of k unit edges. Since a unit edge has three colors of "low, medium, high," the number of possible edge colors of a $k \times k$ tile is 3^k. Here, to avoid too few colors covering an insufficient density range and too many colors bringing huge calculations for potential tiles, we still select three representative colors of "low, medium, high" from these 3^k possibilities to keep a minimal amount of boundary colors for the RCS, as shown in the bottom left of Figure 3.

We additionally restrict the combined edge with the selected color to be composed of k unit edges with the same color, so that the combined edge is rotationally symmetric as well as its unit edge. Then, in total $3^4 = 81$ boundary cases of $k \times k$ tiles, the rotationally symmetric cases would be eliminated, leaving 24 representative boundaries, the number of which is the same as that of RES.

4.3.2. Representative density patches

In contrast to the RES, we want the RCS to additionally cover the most prominent gray-scale variations of input images, thus we predefine uniform and gradient-varying patches. We first determine 8 uniform representative density distributions, enough to cover a density range from 0.15 to 0.85 at a sampling spacing of 0.1. Then, for balance, we choose 8 gradient-varying densities, which are required to concentrate on linear density variations so that we could empirically sample gradient variations horizontally and diagonally. Finding the optimal target densities is impossible because of the limited number of pre-defined tiles and the wide variety of target patterns. In total, 16 density patches are created for the highest levels, as shown in Figure 6. Note that we only represent uniform densities for 2×2 tiles, since we are always able to use uniform smaller tiles for high frequency density changes, and large tiles should be applied for smooth changes or uniform densities.

4.3.3. RCS generation through optimal matching

Once we have 24 representative boundaries and 16 density patches, the key step is to assign boundaries with densities, such that $k \times k$ tiles can be generated and form the RCS (denoted by $k-$RCS).

We first compute the optimal match of 24 boundaries with 8 uniform densities. Similar to the generation of RES, we construct the weighted bipartite graph and then employ the Hungarian algorithm [Kuh55] to find the optimal matching. To construct the weighted bipartite graph, we first expand the number of densities to equal that of boundaries by replication. Since the tile with thickest edges and small density would bring obvious streaks in the tiling result, whereas the boundary cases with thickest edges make up the majority, we have to adjust the duplication multiples for different density patches. As shown in the top row of Figure 6, the first three are only doubled, the middle three are quadrupled, and the last two are tripled, for a total of 24.

Then, two components need to be settled to apply the Hungarian algorithm: the $k \times k$ tile generation and an error function for calculating the weight of each pair of density and boundary, as discussed in detail in the following two subsections. With the help of the error obtained by $k \times k$ tile generation, we obtain the complete bipartite graph and thus compute the optimal matching of 24 boundaries with 8 uniform densities using the Hungarian algorithm, obtaining 24 tiles for k-RCS. Thereafter we again combined the 24 boundaries with 8 gradient-varying densities, as we have just done, to obtain another 24 tiles, which, together with the previous 24 make up the whole |k-RCS|=48, as illustrated in Figure 6.

Since we only represent uniform densities for 2×2 tiles, there are only 24 tiles for 2-RCS, i.e., |2-RCS|=24.

4.3.4. $k \times k$ tile generation

To generate tiles for each pair of density and boundary, we apply a two-step strategy, including a heuristic iterative search trial, and a

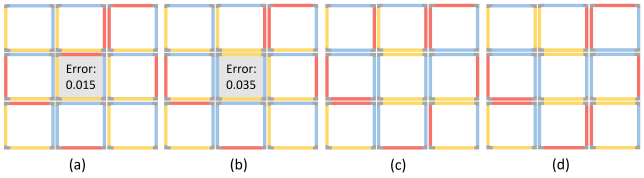


Figure 7: *The generation process of $k \times k$ tiles. (a) is the initial tiling; (b) shows one replacement where the central tile is replaced by another with a larger error but fewer mismatched edges; (c) shows the result after first iteration; (d) is the final result with a threshold of 0.2 at termination.*

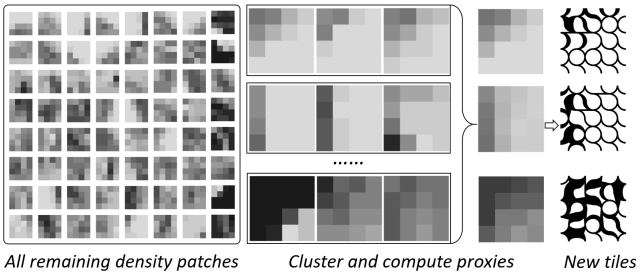


Figure 8: *Density patches in so far untiled regions are clustered into groups, a series of representative 4×4 tiles are generated for each group.*

random search combinatorial optimization, to find the best matching tile to the given density.

Initially, the $k \times k$ area is tiled using a density-first strategy: place unit tiles with a density closest to the target density patch regardless of the boundary constraints, as shown in Figure 7(a). Then, we randomly select one tile with mismatched edge(s) with the adjacent tile and replace it with another with fewer mismatched edges from the entire 1×1 tile set, as shown in Figure 7(b). Moreover, the latter is required to possess a density error less than the threshold set initially as 0.05 to avoid a large error increment due to the replacement. We search and replace until no new ones are replaced (Figure 7(c)), then we increase the threshold by 0.05 and perform the above steps iteratively. There is no upper limit to the threshold, and the process will not stop until all boundaries are matched, shown in Figure 7(d).

To optimize our results, we set different initial states and perform the above steps 50 times, then choose the one with the least error as the final result of $k \times k$ tile generation.

4.3.5. Matching error function

To evaluate the error between a density patch D and a $k \times k$ tile T, we propose a combined error function from the average density error, the maximal density error and the gradient error:

$$Error(D,T) = \alpha \cdot E_{avg} + \beta \cdot E_{max} + \gamma \cdot E_{grad}. \quad (1)$$

The average density error is:

$$E_{avg} = \frac{1}{k^2} \sum_{i,j=1}^{k} |D_{ij} - T_{ij}|, \quad (2)$$

where D_{ij} and T_{ij} denote the density value on (i, j) for the density patch and the tile, respectively, and the same below. The maximal density error is:

$$E_{max} = \max_{i,j}(|D_{ij} - T_{ij}|),\ i,j = 1,2,...,k. \quad (3)$$

The gradient error is used to emphasize features of the density map or the sharpness of the image:

$$E_{grad} = \frac{1}{(k-1)^2} \sum_{i,j=1}^{k-1} |G_{D_{ij}} - G_{T_{ij}}|, \quad (4)$$

where $G_{D_{ij}}$ and $G_{T_{ij}}$ denote the gradient magnitude on (i, j) for the density patch and the tile, respectively.

Normalizing weights ($\alpha+\beta+\gamma = 1$) can help finding boundary-matched patches whose density distribution fits best to the given target density. We use two weight sets for uniform-density tiles and non-uniform-density tiles. For uniform-density tiles, we want to approximate uniform density levels as well as possible without large representation errors for single tiles. Thus, here we raise the weight for the max error term, i.e., $\alpha = 0.2, \beta = 0.6, \gamma = 0.2$. For non-uniform tiles, we intend to represent the density with tiles of similar patterns and preserve density features as much as possible. Thus, average and gradient error are emphasised, i.e., $\alpha = 0.4, \beta = 0.2, \gamma = 0.4$.

5. Image Tiling and Dynamic Tile Generation

5.1. Hierarchical Tiling

A hierarchical tiling process is now performed to tile a given image using RES and RCS. The process starts from large tiles and continues to finer ones. For each scale level k, we traverse the corresponding superpixels of the image and search for a suitable tile from the $k-$RCS that satisfies the boundary constraints and has a minimal matching error (Eq. 1) within a given matching threshold ψ. If no satisfactory tile exists in the $k-$RCS, the current position is skipped and subsequently filled with smaller tiles.

5.1.1. Parameters of matching function

As mentioned in Section 4.3, we have two weight sets for the matching error function. When matching the combined tiles, we calculate the density gradient of the target region, and pick a weight set according to the gradient. For low gradient patches, we use a weight set as shown above; otherwise, we use the set that focuses on average and gradient error. Moreover, another option is to assign adaptive matching error thresholds, i.e., a small threshold for salient regions and a large threshold for non-salient regions, which helps optimize the utilization of combined tiles. We have also developed a user interface that helps to capture these small salient features more accurately, such as the earring in Figure 1. Then these areas are forced to be tiled only using 1×1 tiles, which results in performance improvement while increasing the total number of tiles. Only the result in Figure 1 in this paper is of manual annotation.

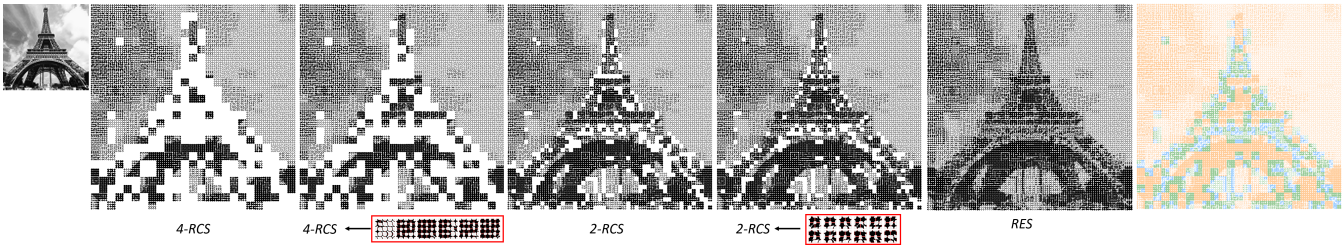


Figure 9: *Hierarchical tiling process for an image of the Eiffel tower. From left to right: tiling with* $4-RCS$ *and generation of 6 new* 4×4 *tiles on-the-fly (red box); the remaining part of the image is tiled using* $2-RCS$*, where 12 new* 2×2 *tiles are generated; the full half-toning image is finished after filling with RES. The rightmost figure shows the placement of tiles in the three scales.*

Since the proposed RES and RCS are complete, there is always a valid solution regardless of starting tiles and tiling orders, and the results do not differ much in terms of used number of tiles and produced errors. The reason is that we consider the multiple boundaries for each density level (Figure 6) and set the matching error bound.

5.2. Dynamic multi-scale tile generation

The RCS with preset densities is able to handle most of the target area; however, there might be particular patterns that can be customized for a given image. Thus, we propose a dynamic generation strategy to generate additional combined tiles.

As illustrated in Figure 8, for each density patch in the remaining regions that were not tiled so far, we calculate an averaged gradient vector to describe the overall density variation. In detail, we compute the density-changing vectors between a cell and its eight neighbors and sum up a directional vector of this cell. All the directional vectors are summed up. Then, we cluster the density patches into several groups according to their directional vectors and average density. We compute a representative density patch as the proxy for each group by averaging all members. Based on these density proxies and boundaries in RCS, we generate new combined tiles by the optimal matching algorithm. These customized tiles are then appended to $k-$RCS, and the tiling process for the untiled region is performed. Figure 9 shows the complete process of tile generation and tiling, which elaborates coupled tilings and a dynamic generation strategy for combined tiles. Six 4×4 tiles and twelve 2×2 tiles are produced on the fly during tiling. Compared to tiling with preset RES, 2-RCS, and 4-RCS, the total number of tiles reduced from 1978 to 1744.

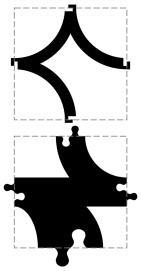
Note that the complete set composed of two sets RES and RCS has a good performance with respect to matching errors on various density maps, as shown in Figure 1 and 12, and customized tiles generated dynamically for a specific input image can further improve the utilization of larger tiles, but do not affect the validity of a solution produced only from RES and RCS.

6. Results and Discussions

Our method was implemented in C++, running on a PC with an Intel Core^{TM} i7-7700K CPU @3.6GHz and 16GB RAM.

Performance. The complexity for generating RES and RCS depends on the number of representative boundaries and density cases. We recommend three boundary colors for each edge, resulting in a limited number of RES and RCS tiles. In our experiments, we work with a set |RES|=|2-RCS|=24, |4-RCS|=48. It takes less than 30 minutes to generate these sets, mainly due to the $k \times k$ tile generation. The tiling process depends on the tiling resolution. For a resolution of 100×100, direct tiling based on RES and RCS takes less than 1 minute. The dynamic generation of combined tiles takes 3-5 minutes without optimization.

Although our halftoning result focuses more on manufacturing and assembly, the visual effects are not worse than traditional halftoning using only element tile and multi-scale rectangles. As shown in Figure 11, we compare our results with the modular line-based halftoning and stylized dot pattern results, and we can see that our results are much more realistic. While compared with the traditional method [DGBOD12], our tones are a bit darker in high brightness areas considering the fabrication constraints.



Fabrication. Considering the manufacturing constraints, a minimum density value is required to keep the result fabricable. We designed two kinds of joints for shape interlocking during assembly like for jigsaw puzzles (see the inset figure). The "hook" style joint is for thin edges, and rounded tabs are designed for edges with enough space. The tiles can be fabricated through 3D printing or laser cutting. The tiles in both Figure 1 and 10 are laser cut with the minimal feature size of 0.1mm. The elementary tile size is $30mm \times 30mm$, and the thinnest boundary edge is set to 3mm with the parameters $(0.5, 0.1)$, such that the joint can be fabricated. The image in Figure 1 has a resolution of 100×120 and is tiled by 1656 tiles in total with a size of $3m \times 3.6m$, with detailed information in the second row of "Girl" in Table 1. The result density error is 3.55%, which is measured by the average density error for each pixel and its corresponding tile.

Parameters. There is always a trade-off between the number of tiles and error tolerance ψ. A small ψ indicates more tiles and thus higher matching quality. More large-scale tiles are generated and involved in the results when the ψ value increases. In practice, we may set the upper bound of the global matching error, or the total

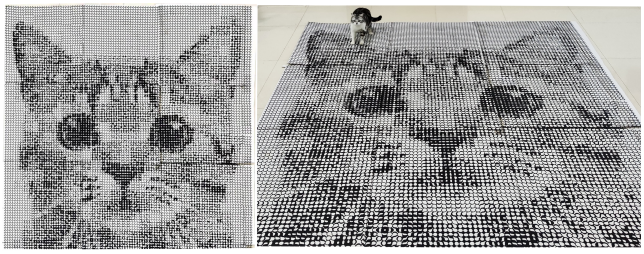


Figure 10: *Photos of the cat image tiled in* $3m \times 3m$*, by 1696 fabricated multi-scale tiles, shown in two views. Detail information is in the second row of Table 1.*

number of tiles, and obtain the tiling results via binary searching the ψ values. Generally, since a piece takes about 15s or less to assemble in practice, we take 20% of the tiling resolution as the reference number of total tiles to control the assembly cost, which saves 80% tiling effort.

Figure 12 shows tiling results for three example images (cat, lion, and rings). The results are from the same sets, RES, 2-RCS, and 4-RCS, but differ in their utilization of combined and elementary tiles. For the cat and lion image, we set the result density error to 5%, and the results are still good (b-column), with around 2000 tiles used, which reduces the needed tiling effort by about 80%. If we constrain the total number of tiles to 1000, the result incorporates a bigger number of large tiles and thus has reduced quality (c-column). The "rings" image is of high frequency and therefore sensitive to the matching error, while we see that the result is reasonable with 625 4×4 pieces (c-column). We compare these results with a result from elementary tiles only, as shown in a-column. Table 1 lists statistical data for results in this paper.

Density maps. Our method also works well for density maps besides images. Figure 13 shows an example of a density map from topology optimization of a cantilever beam. Tilings can also help to create sophisticated structures adapted to the local truss level efficiently.

Potential applications. In large-scale application scenarios, we believe that our method can contribute to "mass customization" and leverages the customization of an arbitrarily given density map and mass production of pre-computed tiles. In addition to the application of art display as shown in Figure 1, our tiles are easy to package, transport, and assemble on-site, so that additional functionality such as ventilation or heat dissipation can be added and then applied to support structures or protective devices in specific scenarios. Besides large-scale application scenarios, our results themselves have a specific artistic aesthetic. Users can choose to tile an image using only 1×1 tiles for minimal errors and then manufacture the results as decorative paintings by laser cutting printing. Moreover, we can also input patterns that encode information, such as QR codes, and thus combine the manufactured result with encoding information.

Limitations. Sometimes it is difficult for our algorithm to balance quantity and representation quality for some patterns or density fields. As shown in Figure 14, we set the error tolerance to $\psi = 0.16$, the total number of 46 types of 4×4 tiles used to tile the zebra is only 356. At this point, the error reaches up to 7.14%, due to the fine diagonal details of the fur – if we would raise the tolerance further, too much error would make the results completely incomprehensible.

Table 1: *Statistics of the results. The first four images are in* 100×100 *resolution, the "Girl with a Pearl Earring" is in* 100×120 *resolution. The number of types (#Types) and number of pieces (#Pieces) for* 4×4*,* 2×2*,* 1×1*, respectively, are listed. Reduction is the percentage of saved tiles in comparison to just using elementary tiles. Error is the average density error for the tiling result. Some results are shown in the supplementary files.*

Image	#Types	#Pieces	Red.	Error
Cat	24 (0, 0, 24)	10000 (0, 0, 10000)	0	2.42%
	84 (36, 24, 24)	1696 (467, 433, 796)	83%	4.93%
	86 (42, 22, 22)	934 (578, 132, 224)	90%	6.13%
Lion	24 (0, 0, 24)	10000 (0, 0, 10000)	0	2.24%
	80 (33, 23, 24)	2671 (352, 683, 1636)	73%	4.89%
	82 (39, 21, 22)	970 (572, 150, 248)	90%	6.81%
Rings	24 (0, 0, 24)	10000 (0, 0, 10000)	0	3.22%
	61 (16, 21, 24)	3979 (164, 1187, 2628)	60%	6.43%
	39 (39, 0, 0)	625 (625, 0, 0)	94%	11.2%
Eiffel	24 (0, 0, 24)	10000 (0, 0, 10000)	0	2.65%
	89 (41, 24, 24)	2173 (429, 464, 1280)	78%	5%
	92 (46, 23, 23)	988 (565, 179, 244)	91%	6.48%
Tiger	24 (0, 0, 24)	10000 (0, 0, 10000)	0	2.11%
	85 (38, 23, 24)	2800 (357, 615, 1828)	72%	4.91%
	86 (41, 22, 23)	964 (572, 152, 240)	94%	7.29%
Girl	24 (0, 0, 24)	12000 (0, 0, 12000)	0	1.86%
	84 (36, 24, 24)	1656 (642, 238, 776)	86%	3.55%
	87 (39, 24, 24)	1209 (690, 125, 460)	89%	3.82%

7. Conclusion and Future work

In this paper, we introduce the concept of multi-scale Wang tiles and customize it to digital fabrication, targeting synthesizing large-scale scenarios with variable densities. To facilitate a binary tile design, we propose a parametric model for designing unit tiles with topological constraints. Then, we develop a two-stage search strategy for generating a small set of combined Wang tiles with optimized density representations. Results show that using such Wang tiles allows us to achieve both high tiling efficiency and fine resolutions.

In the future, we plan to integrate more functionalities into tile design beyond density variation, such as kinematics, permeability, etc. To extend the unit tile structures from binary tones to varying densities and add properties such as roughness, slip resistance, etc., is also an exciting topic. Furthermore, we would like to extend our multi-scale tiles towards the generative design of meta-materials both 2D and 3D, and establish a computational framework for designing such materials with defined physical properties and aesthetics.

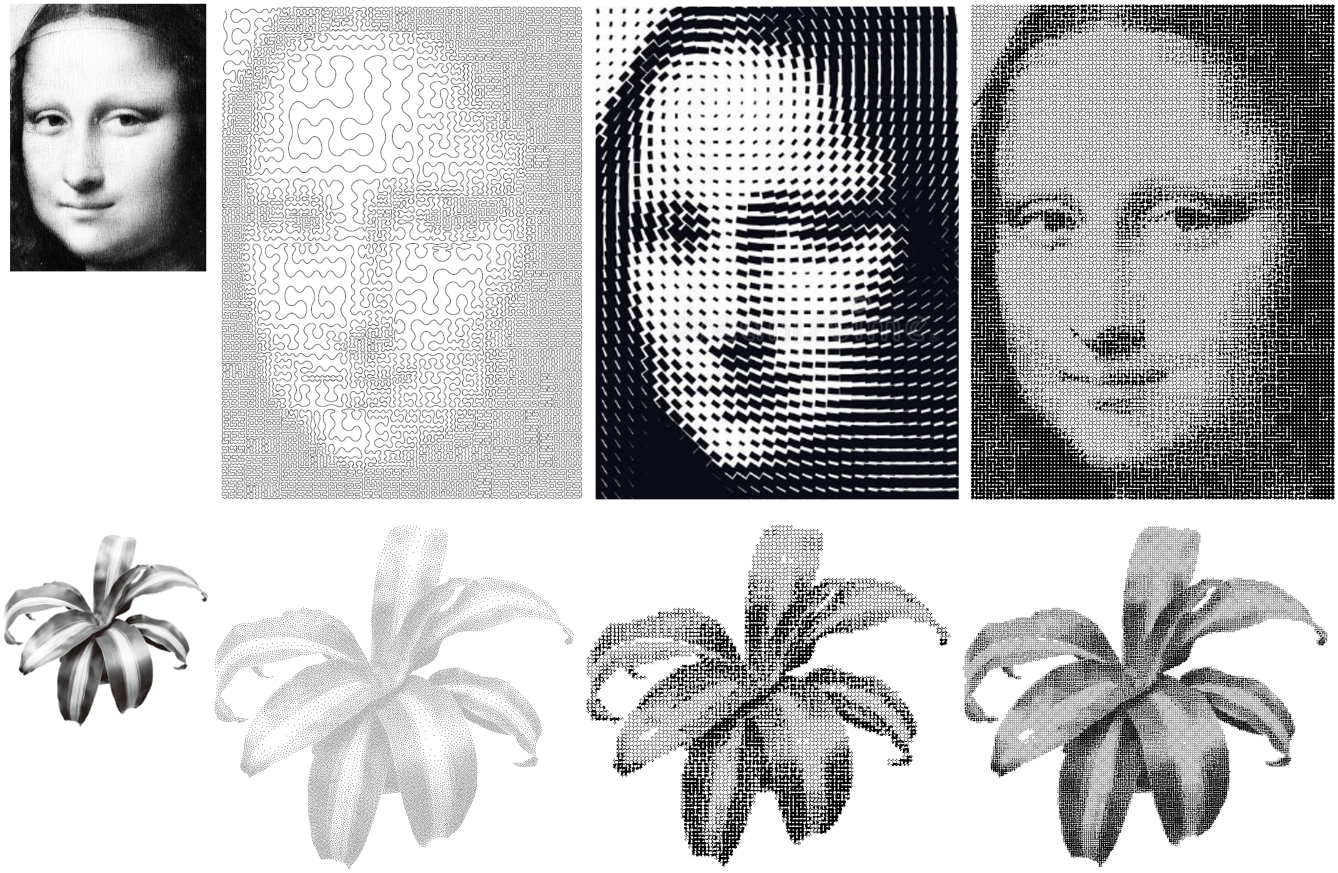


Figure 11: *Top row: from left to right are modular line-based halftoning result [Ahm14], stylized dot pattern [Mon], and our result with 2128 tiles of* 100×120 *resolution, respectively. Bottom row: from left to right are the stippling result [DGBOD12], our result with 2062 tiles of* 100×100 *resolution, and our result with 4953 tiles of* 200×200 *resolution, respectively.*

Acknowledgements

We thank all the anonymous reviewers for their constructive suggestions. This work is supported by the grant No. 61972232 from National Natural Science Foundation of China (NSFC), Science and Technology Program of Shenzhen, China (CJGJZD20200617102202007). Oliver Deussen was funded by the German Research Foundation (DFG)-Project-ID 251654672-TRR 161.

References

[Ahm14] AHMED A. G. M.: Modular line-based halftoning via recursive division. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering - NPAR '14* (2014), ACM Press. doi:10.1145/2630397.2630403. 3, 9

[AM12] ALEXA M., MATUSIK W.: Irregular pit placement for dithering images by self-occlusion. *Computers Graphics 36*, 6 (2012), 635–641. 2011 Joint Symposium on Computational Aesthetics (CAe), Non-Photorealistic Animation and Rendering (NPAR), and Sketch-Based Interfaces and Modeling (SBIM). URL: https://www.sciencedirect.com/science/article/pii/S0097849312000258, doi:https://doi.org/10.1016/j.cag.2012.02.008. 3

[BSD09] BALZER M., SCHLÖMER T., DEUSSEN O.: Capacity-constrained point distributions: A variant of lloyd's method. *ACM Transactions on Graphics (TOG) 28*, 3 (2009), 1–8. 3

[BWL18] BIAN X., WEI L.-Y., LEFEBVRE S.: Tile-based pattern design with topology control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 1*, 1 (jul 2018), 1–15. doi:10.1145/3203204. 2

[Car18] CARLSON C.: Multi-scale truchet patterns. In *Bridges 2018 Conference Proceedings* (2018), Tessellations Publishing, pp. 39–44. 4

[CLF*18] CHEN X., LI H., FU C.-W., ZHANG H., COHEN-OR D., CHEN B.: 3d fabrication with universal building blocks and pyramidal shells. *ACM Trans. Graph. 37*, 6 (dec 2018). URL: https://doi.org/10.1145/3272127.3275033, doi:10.1145/3272127.3275033. 3

[CML*17] CHEN W., MA Y., LEFEBVRE S., XIN S., MARTÍNEZ J., WANG W.: Fabricable tile decors. *ACM Transactions on Graphics 36*, 6 (nov 2017), 1–15. doi:10.1145/3130800.3130817. 3

[CSHD03] COHEN M. F., SHADE J., HILLER S., DEUSSEN O.: Wang tiles for image and texture generation. *ACM Transactions on Graphics 22*, 3 (jul 2003), 287–294. doi:10.1145/882262.882265. 1, 2

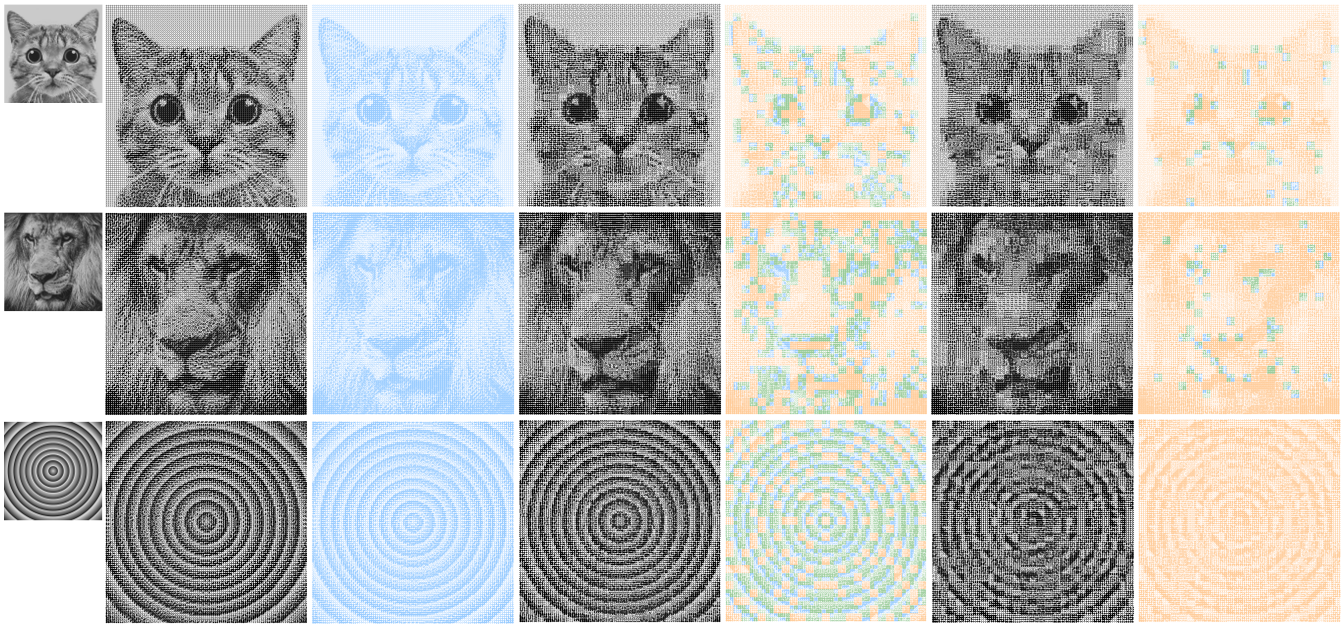


Figure 12: *Results of the cat, lion, and rings image (first column) for different scale tiles, shown in both tiling results (left) and placement maps (right). Data of (a-c) is in the three sub-rows of the corresponding images in Table 1, respectively.*

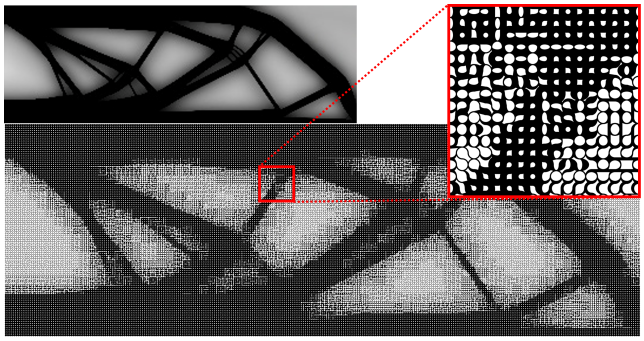


Figure 13: *Given a density field from topology optimization, we apply multi-scale tiling and achieve microstructures that conform to the required densities.*

[CZX*16] CHEN W., ZHANG X., XIN S., XIA Y., LEFEBVRE S., WANG W.: Synthesis of filigrees for digital fabrication. *ACM Transactions on Graphics 35*, 4 (jul 2016), 1–13. doi:10.1145/2897824.2925911. 3

[DGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue noise through optimal transport. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 1–11. 3, 7, 9

[DJSJ19] DEROUET-JOURDAN A., SALVATI M., JONCHIER T.: Generating stochastic wall patterns on-the-fly with wang tiles. *Computer Graphics Forum 38*, 2 (may 2019), 255–264. doi:10.1111/cgf.13635. 2

[DLL*15] DUMAS J., LU A., LEFEBVRE S., WU J., DICK C.: By-example synthesis of structurally sound patterns. *ACM Trans. Graph. 34*, 4 (jul 2015). URL: https://doi.org/10.1145/2766984, doi:10.1145/2766984. 3

[DSZ17] DEUSSEN O., SPICKER M., ZHENG Q.: Weighted linde-buzo-gray stippling. *ACM Transactions on Graphics (TOG) 36*, 6 (2017), 1–12. 3

[DZRN20] DOŠKÁŘ M., ZEMAN J., RYPL D., NOVÁK J.: Level-set based design of wang tiles for modelling complex microstructures. *Computer-Aided Design 123* (jun 2020), 102827. doi:10.1016/j.cad.2020.102827. 2

[EK21] ELBER G., KIM M.-S.: Synthesis of 3d jigsaw puzzles over freeform 2-manifolds. *Computers Graphics* (2021). URL: https://www.sciencedirect.com/science/article/pii/S0097849321002259, doi:https://doi.org/10.1016/j.cag.2021.10.014. 3

[EKS*10] EIGENSATZ M., KILIAN M., SCHIFTNER A., MITRA N. J., POTTMANN H., PAULY M.: Paneling architectural freeform surfaces. In *ACM SIGGRAPH 2010 papers*. 2010, pp. 1–10. 3

[FL05] FU C.-W., LEUNG M.-K.: Texture tiling on arbitrary topological surfaces using wang tiles. The Eurographics Association. doi:10.2312/EGWR/EGSR05/099-104. 2

[FLHCO10] FU C.-W., LAI C.-F., HE Y., COHEN-OR D.: K-set tilable surfaces. *ACM transactions on graphics (TOG) 29*, 4 (2010), 1–6. 3

[HEB15] HUARD M., EIGENSATZ M., BOMPAS P.: Planar panelization with extreme repetition. In *Advances in architectural geometry 2014*. Springer, 2015, pp. 259–279. 3

[KCODL06] KOPF J., COHEN-OR D., DEUSSEN O., LISCHINSKI D.: Recursive wang tiles for real-time blue noise. *ACM Trans. Graph. 25*, 3 (jul 2006), 509–518. URL: https://doi.org/10.1145/1141911.1141916, doi:10.1145/1141911.1141916. 1, 2, 3

[Kip01] KIPPHAN H.: *Handbook of print media: technologies and production methods*. Springer Science & Business Media, 2001. 3

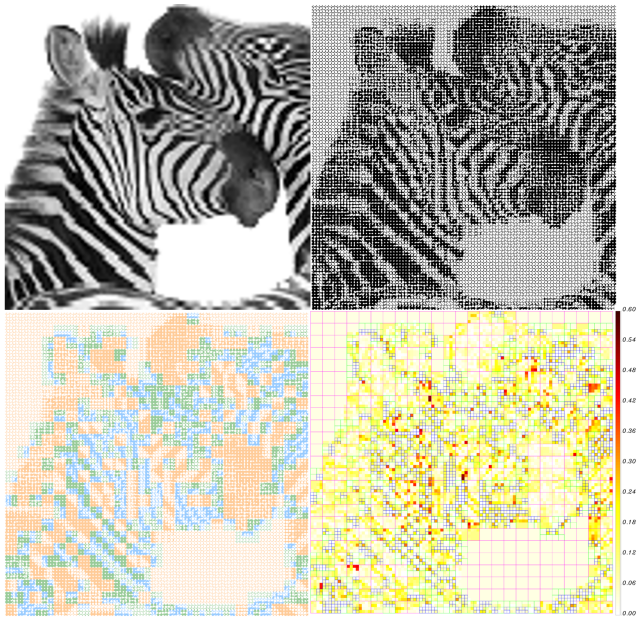


Figure 14: *Tiling the zebra image with $\psi = 0.16$, including 46 types of 4×4 tiles (356 pieces), 24 types of 2×2 tiles (563 pieces), and 24 types of elementary tiles (2052 pieces), resulting in an error of 7.14%.*

[Kuh55] KUHN H. W.: The hungarian method for the assignment problem. *Naval research logistics quarterly 2*, 1-2 (1955), 83–97. 5

[LD06] LAGAE A., DUTRÉ P.: An alternative for wang tiles: Colored edges versus colored corners. *ACM Trans. Graph. 25*, 4 (oct 2006), 1442–1459. URL: https://doi.org/10.1145/1183287.1183296, doi:10.1145/1183287.1183296. 2

[LEQ*07] LU A., EBERT D. S., QIAO W., KRAUS M., MORA B.: Volume illustration using wang cubes. *ACM Trans. Graph. 26*, 2 (jun 2007), 11–es. URL: https://doi.org/10.1145/1243980.1243985, doi:10.1145/1243980.1243985. 2

[LLCY15] LIANG G., LU L., CHEN Z., YANG C.: Poisson disk sampling through disk packing. *Computational Visual Media 1*, 1 (mar 2015), 17–26. doi:10.1007/s41095-015-0003-7. 3

[LLS*20] LIU X., LU L., SHARF A., YAN X., LISCHINSKI D., TU C.: Fabricable dihedral escher tessellations. *Computer-Aided Design 127* (2020), 102853. URL: https://www.sciencedirect.com/science/article/pii/S0010448520300464, doi:https://doi.org/10.1016/j.cad.2020.102853. 3

[LZZ*21] LIU Z.-Y., ZHANG Z., ZHANG D., YE C., LIU L., FU X.-M.: Modeling and fabrication with specified discrete equivalence classes. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–12. 3

[MDLW15] MARTÍNEZ J., DUMAS J., LEFEBVRE S., WEI L.-Y.: Structure and appearance optimization for controllable shape design. *ACM Transactions on Graphics 34*, 6 (nov 2015), 1–11. doi:10.1145/2816795.2818101. 3

[MMG*14] MUELLER S., MOHR T., GUENTHER K., FROHNHOFEN J., BAUDISCH P.: Fabrickation: Fast 3d printing of functional objects by integrating construction kit building blocks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI '14, Association for Computing Machinery, p. 3827–3834. URL: https://doi.org/10.1145/2556288.2557005, doi:10.1145/2556288.2557005. 3

[Mon] Mona lisa stylized dot pattern. https://www.dreamstime.com/mona-lisa-stylized-dot-pattern-gioconda-image237499692. Accessed: 2022-06-08. 9

[MRR02] MOORE C., RAPAPORT I., RÉMILA E.: Tiling groups for wang tiles. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (USA, 2002), SODA '02, Society for Industrial and Applied Mathematics, p. 402–411. 2

[PLL*19] PENG H., LU L., LIU L., SHARF A., CHEN B.: Fabricating QR codes on 3d objects using self-shadows. *Computer-Aided Design 114* (sep 2019), 91–100. doi:10.1016/j.cad.2019.05.029. 3

[PLL*20] PENG H., LIU P., LU L., SHARF A., LIU L., LISCHINSKI D., CHEN B.: Fabricable unobtrusive 3d-QR-codes with directional light. *Computer Graphics Forum 39*, 5 (aug 2020), 15–27. doi:10.1111/cgf.14065. 3

[RTY*19] RAO C., TIAN L., YAN D.-M., LIAO S., DEUSSEN O., LU L.: Consistently fitting orthopedic casts. *Computer Aided Geometric Design 71* (2019), 130–141. URL: https://www.sciencedirect.com/science/article/pii/S0167839619300342, doi:j.cagd.2019.04.018. 3

[SS10] SINGH M., SCHAEFER S.: Triangle surfaces with discrete equivalence classes. In *ACM SIGGRAPH 2010 papers*. 2010, pp. 1–7. 3

[Stu97] STUCKI P.: 3d halftoning. In *Imaging Sciences and Display Technologies* (1997), vol. 2949, SPIE, pp. 314–317. 3

[Wan61] WANG H.: Proving theorems by pattern recognition — ii. *The Bell System Technical Journal 40*, 1 (1961), 1–41. doi:10.1002/j.1538-7305.1961.tb03975.x. 1, 2

[Wei04] WEI L.-Y.: Tile-based texture mapping on graphics hardware. In *ACM SIGGRAPH 2004 Sketches on - SIGGRAPH '04* (2004), ACM Press. doi:10.1145/1186223.1186307. 1, 2

[XHFZ19] XU H., HUI K.-H., FU C.-W., ZHANG H.: Computational lego technic design. *ACM Trans. Graph. 38*, 6 (nov 2019). URL: https://doi.org/10.1145/3355089.3356504, doi:10.1145/3355089.3356504. 3

[YLS*21] YAN X., LU L., SHARF A., YU X., SUN Y.: Man-made by computer: On-the-fly fine texture 3d printing. In *Symposium on Computational Fabrication* (oct 2021), ACM. doi:10.1145/3485114.3485119. 3

[ZC09] ZHOU C., CHEN Y.: Three-dimensional digital halftoning for layered manufacturing based on droplets. *Transactions of the North American Manufacturing Research Institution of SME 37* (2009), 175–182. 3

[ZCBK12] ZIMMER H., CAMPEN M., BOMMES D., KOBBELT L.: Rationalization of triangle-based point-folding structures. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 611–620. 3

[ZCT16] ZEHNDER J., COROS S., THOMASZEWSKI B.: Designing structurally-sound ornamental curve networks. *ACM Trans. Graph. 35*, 4 (jul 2016). URL: https://doi.org/10.1145/2897824.2925888, doi:10.1145/2897824.2925888. 3

[ZFD*17] ZHANG X., FANG G., DAI C., VERLINDEN J., WU J., WHITING E., WANG C. C.: Thermal-comfort design of personalized casts. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2017), UIST '17, Association for Computing Machinery, p. 243–254. URL: https://doi.org/10.1145/3126594.3126600, doi:10.1145/3126594.3126600. 3

[ZJL14] ZHOU S., JIANG C., LEFEBVRE S.: Topology-constrained synthesis of vector patterns. *ACM Trans. Graph. 33*, 6 (nov 2014). URL: https://doi.org/10.1145/2661229.2661238, doi:10.1145/2661229.2661238. 3

[ZLW*16] ZHAO H., LU L., WEI Y., LISCHINSKI D., SHARF A., COHEN-OR D., CHEN B.: Printed perforated lampshades for continuous projective images. *ACM Trans. Graph. 35*, 5 (jun 2016). URL: https://doi.org/10.1145/2907049, doi:10.1145/2907049. 3