Aspects of Symmetry in Bobbin Lace

Veronika Irvine
Cheriton School of Computer Science,
University of Waterloo
Waterloo, ONT, Canada N2L 3G1
virvine@uwaterloo.ca

Frank Ruskey
Computer Science,
University of Victoria
Victoria, BC, Canada V8W 2Y2

Abstract

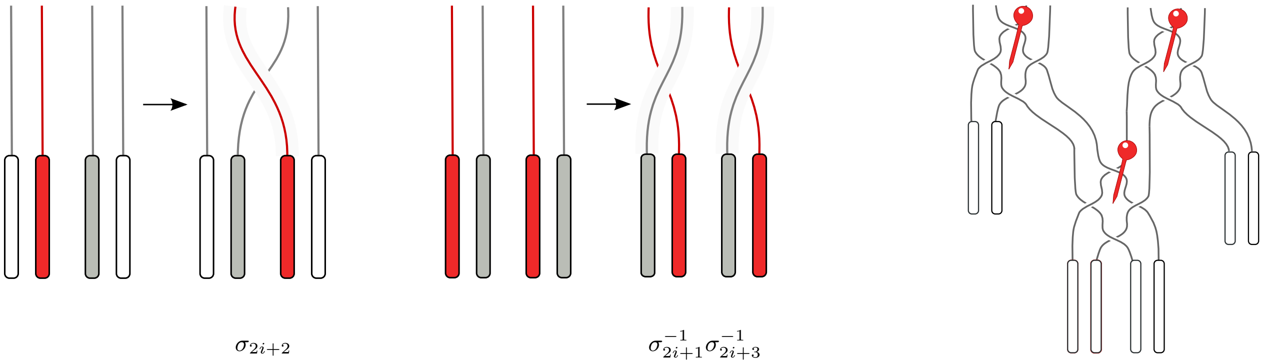
Bobbin lace is a 500-year-old fiber art form created by braiding together fine threads. In its design, lacemakers employ doubly periodic textures to create contrast and interest in a predominantly monochromatic fabric. In the past we have created a model for these doubly periodic patterns which employs graph drawings to describe the flow of threads. In this paper we demonstrate that these graph drawings, which we call 'tesselace patterns', exist for each of the 17 planar periodic symmetry groups. We provide an algorithm for exhaustively generating patterns with a particular symmetry on a grid of fixed size. We also explore the symmetry of the interlaced fabric resulting from these patterns.

Introduction

Bobbin lace is a traditional fiber art dating back to the mid-16th century which was used to decorate clothing and furniture. In previous work we introduced a mathematical model for the doubly periodic patterns used in bobbin lace designs [5, 6]. Since that initial foray, we have been exploring algorithms for generating new and interesting periodic patterns based on the model; patterns which we will refer to as 'tesselace'. We start by giving a brief review of the technique used in the craft and summarize the key elements of our mathematical model. In the following sections, our goal is to emulate the aesthetic appeal of traditional patterns. To that end, we present an algorithm for generating tesselace patterns with a specific symmetry type and demonstrate that the algorithm produces representatives from each of the 17 planar periodic symmetry groups. We conclude by discussing symmetry at the level of individual threads and how this can be explored using the two-colour symmetries of the plane.

Bobbin lace is an alternating braid constructed using four threads at a time. There are just two actions used by lacemakers to form the braids: the cross (C) and the twist (T). These actions, illustrated in Figure 1, can be combined in any order including a simple CT which gives an open weave (also known as a triaxial weave), $CTCTCTCT$ which gives a dense plait, and $CTTTCT$ which creates a small hole. Regardless of which combination of actions is used, the result is always an alternating braid. The alternating structure gives stability to the seemingly fragile fabric by having each thread locked in place by its neighbours.

A common component of bobbin lace designs is a doubly periodic planar pattern called a "ground" or a "filling". These repeating patterns can be modelled as the pair $(\Delta(G), \zeta(v))$ in which G is a 2-regular digraph (a directed graph with two incoming and two outgoing arcs at each vertex), $\Delta(G)$ is a drawing of G and $\zeta(v)$ is a mapping from the vertices of G to a braid word formed from any combination of cross and twist. Each arc in $\Delta(G)$ represents a pair of threads and the graph drawing describes the flow of threads from one braid on four threads to the next. As illustrated in Figure 6, a single graph drawing will result in several lace grounds when paired with different $\zeta(v)$ mappings.



(a) Cross (C) *(b) Twist (T)* *(c) Pin (p)*

Figure 1: *Bobbin lace is created from two basic actions: a) cross and b) twist. c) During construction, braids are temporarily held in place with pins.*

In order to produce workable lace patterns, the graph drawings $\Delta(G)$ must possess certain properties. Each property is derived from attributes of the lace itself. For example, the alternating braid formed by bobbin lace is just like a mathematical braid. As a consequence, all thread crossings must happen in the forward direction; strands cannot loop backwards. In the model, this corresponds to a combinatorial embedding that is free of directed cycles. A full explanation of each of the properties can be found in our previous work [6]; what follows is just a summary. The doubly periodic pattern corresponds to a planar graph with an infinite number of edges and vertices. However, the translational symmetry of the pattern allows us to represent the infinite graph by a finite graph drawn on a torus. In the combinatorial embedding of $\Delta(G)$, the finite 2-regular digraph must have a genus of one, be free of contractible directed cycles and at each vertex the outgoing arcs must be arranged in a rotationally consecutive order (out-out-in-in as opposed to in-out-in-out in clockwise order). Finally, the finite graph drawing on the torus has a unique partition into osculating circuits (i.e., it can be partitioned into a set of closed curves whose elements can touch but no pair of curves can cross transversely) each of which, when smoothed, is homotopic to a $(1, 0)$ torus knot.

In the past, using combinatorial algorithms to generate graph drawings consistent with our model, we have discovered over 5 million tesselace patterns, each of which can be combined with a variety of cross and twist actions. The end result is more bobbin lace grounds then it would be possible to process by an "army" of lacemakers. We therefore turn our attention to finding a set of patterns which warrant our consideration first. To do this we will look at the symmetry of these fabrics.

Algorithmically Generated Patterns with Symmetry

Flipping through a catalogue of traditional bobbin lace grounds, one quickly gets the sense that symmetry is a common, if not dominant, quality of these periodic patterns. It is also important to recall that the definition of lace is an open fabric, one with many holes. In our approach, the holes in bobbin lace correspond (roughly) to faces in the tesselace pattern. We therefore turned our attention to symmetry in the size and position of faces in tesselace patterns and propose the following theorem:

Theorem. *There exist tesselace graph drawings for each of the seventeen periodic symmetry groups in the plane.*

By way of proof, we present a combinatorial algorithm for generating tesselace graph drawings of each symmetry group and example results.

Overview of Algorithm

Although G is a digraph, we will ignore the direction of edges when discussing the symmetry of $\Delta(G)$. The direction of the edges is very important in the making of lace but, from the perspective of a lace admirer, the threads themselves have no orientation and it is difficult to determine in which direction a piece of lace was worked. From a mathematical perspective, disregarding the direction of edges results in a more interesting solution domain; when symmetry generators are applied to directed edges, any transformation other than parallel reflections and parallel glides will result in a contractible cycle and therefore cannot be a braid.

Consider one of the 17 planar periodic symmetry types which we will refer to here as γ. Our algorithm is broken into two parts: (1) Exhaustively generate straight-line graph drawings on the integer lattice for 4-regular undirected graphs such that the closed line segments of the drawing form a motif that is mapped onto itself by the symmetry operations of γ; (2) Assign a direction to the edges of the undirected graph consistent with the properties of a tesselace pattern. As in our previous work, we use backtracking for both parts and generate patterns on various grid sizes. In backtracking, a conceptual tree is created in which internal nodes are "partial" solutions and every solution is a leaf, but not all leaves are solutions. As a node x is created, it is evaluated by a heuristic that, if it fails, implies that it is not a solution and none of its children can lead to a solution; thus x is a failed leaf. On the other hand, if the heuristic succeeds, then either x is a solution or the children of x must be created to determine whether some descendant of x is a solution.

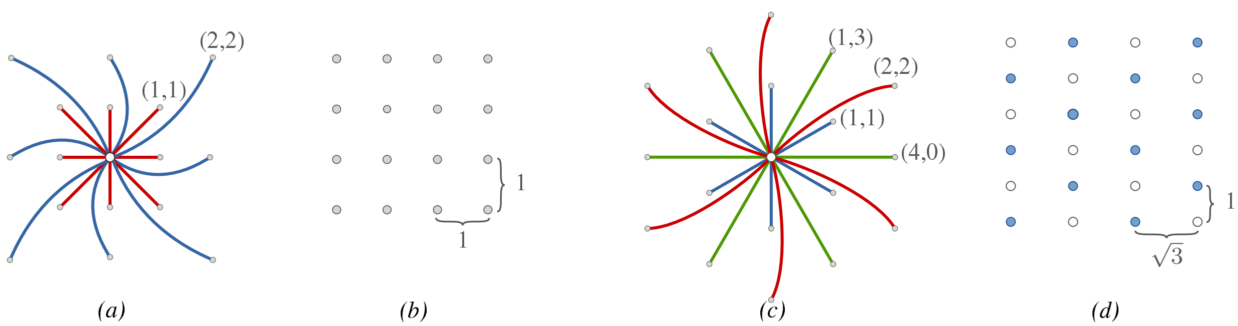


Figure 2: *Square lattice: (a) Near neighbours (b) Lattice points. Hexagonal lattice: (c) Near neighbours (d) Lattice points.*

In part (1), solutions are represented as a set of vertices in which each vertex has an adjacency list and an integer lattice position. The algorithm starts by positioning a minimal set of symmetry generators within a fixed size lattice grid (details provided in next section) and creating isolated vertices at each lattice position. At each node in the backtracking tree, an edge e connecting near neighbours (as defined in Figure 2) is inserted into the graph. Also at each backtracking node, the set of edges representing the images of e under the isometric transformations of the symmetry generators is inserted into the graph drawing.

Rules. *In part (1) of the algorithm, the following conditions must hold at each node of the backtracking tree:*

1. *Each vertex has degree* ≤ 4.
2. *Edges only intersect at end-vertices.*
3. *Vertices are mapped to lattice points.*
4. *All transformed copies of an edge under symmetry group* γ *are present in the embedding.*

When all vertices either have degree 4 or degree 0 and there is at least one vertex of degree 4, the graph is complete. Isolated vertices are removed and the graph embedding is tested to determine if it has the correct genus (i.e., it is a torus graph).

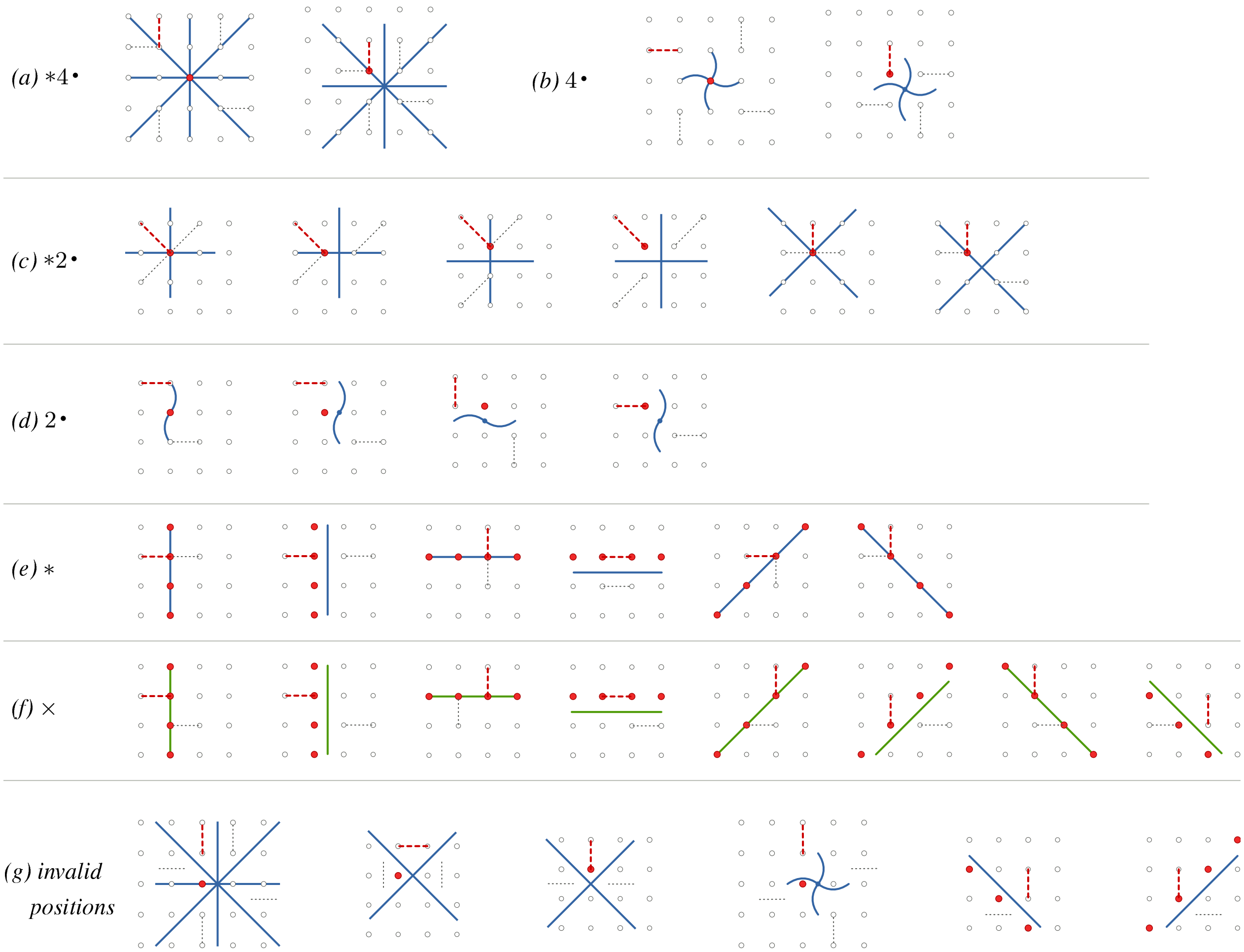


Figure 3 : *All possible locations for symmetry generators on a square lattice*

For part (2), the data from part (1) is extended to include direction information for each edge. A new backtracking tree is created. At each node in the backtracking tree, a vertex v is selected and all edges incident to v are assigned a direction. Vertices are processed in order of the extent to which direction has been assigned to its edges; vertices with a higher number of directed incident edges will be processed first. A lightweight triage of the vertices is all the sorting that is required; vertices are placed in one of five buckets: 0, 1, 2, 3, *done*. All vertices start out in bucket 0. When an edge is oriented, the two end-vertices of the edge move up one bucket. The process stops when either we have reached a state that cannot produce a complete solution or all edges are in the *done* bucket. The triage approach significantly reduces the number of nodes in the backtracking tree. If none of the edges incident to a vertex have a direction, then there are four direction assignments to explore ($\{i, i, o, o\}$, $\{o, i, i, o\}$, $\{o, o, i, i\}$, $\{i, o, o, i\}$ where i is *in* and o is *out*) giving four branch points in the backtracking tree for the node. If one or two edges incident to a vertex have been assigned a direction, the backtracking tree will have at most two branch points at the node.

Rules. *The following conditions must hold at each node of the backtracking tree in part (2):*

1. *Each vertex has in-degree* ≤ 2 *and out-degree* ≤ 2
2. *Outgoing edges are consecutive in rotational order*
3. *An oriented edge contributes to the out-degree of one of its end-vertices and the in-degree of the other*

When all edges have been successfully oriented, the digraph is tested for the presence of contractible cycles and the wrapping index of its osculating circuits.

Algorithm Details

The previous section gives an overview of the algorithm but there are several details to consider, many of which impact the size of the backtracking tree and therefore the performance of the algorithm.

In a tesselace graph drawing, vertices of the graph are mapped to integer lattice points. Transforming an edge of a tesselace graph drawing is permissable only if it takes the end-vertices of an edge from one pair of integer lattice points to another. As a consequence, there are only a small number of locations within a grid of fixed size where each type of symmetry group generator can be placed. Figure 3 shows all allowed positions per generator on a square lattice. For example, in Figure 3(a) the $*4^\bullet$ generator can be placed at (col, row) or at $(col + \frac{1}{2}, row + \frac{1}{2})$ where $col, row \in \mathbb{Z}$. As shown in Figure 3(g), the $*4^\bullet$ generator cannot be at position $(col + \frac{1}{2}, row)$. In Figure 3, bold blue curved lines meet at the center of a point group; bold blue lines are mirror lines and bold green lines are glides lines. Thin gray dotted line segments are the images of the bolder red dashed line segments under symmetry operation.

In the following discussion, a lattice point (col, row) is *near* a Cartesian point (x, y) (where $x, y \in \mathbb{R}$) if $(col, row) = (\lfloor x \rfloor, \lfloor y \rfloor)$. Similarly, a lattice point (col, row) is near a non-horizontal line ℓ if given a point (x, row) on ℓ then $col = \lfloor x \rfloor$. If ℓ is horizontal then given a point (col, y) on ℓ, a lattice point (col, row) is near ℓ if $row = \lfloor y \rfloor$.

Several notations exist for labelling the planar symmetry groups. We will use the orbifold notation of Conway [1] which provides a topological description of the symmetry. The set of all points that are the same under a symmetry operation is called an *orbit*. An *orbifold* is "folded" by identifying all points in the same orbit. In orbifold notation, unique rotational (or gyrational, to use Conway's term) symmetries are listed first by specifying the number of repetitions required to rotate a point back to its original position. The point group $n^\bullet$ (which corresponds to C_n in Schönflies notation) is the group of rotations σ such that $\sigma^n = 1$, where 1 is the identity and n is the smallest number of rotations that returns a point to its original position. Reflection (or kaleidoscope) symmetries are preceded by a $*$ and are represented by the number of mirror lines that meet at a point. The point group $*n^\bullet$ (D_n in Schönflies notation) is the dihedral group of order n. A single mirror line is represented by a solo $*$; a glide line is indicated by $\times$. For example, $*632$ indicates a pattern with three distinct reflection point subgroups of types $*6^\bullet$, $*3^\bullet$ and $*2^\bullet$; 2222 indicates four distinct rotation point subgroups each of type $2^\bullet$; $4{*}2$ represents a mix of rotations and reflections, namely, $4^\bullet$ and $*2^\bullet$; and $*\times$ indicates a symmetry group with single mirror reflection and a glide reflection.

For improved performance, we shall restrict the placement of symmetry group generators to configurations that generate the smallest number of graphs from the same isomorphism class. This is done by choosing the least common, ideally unique, generator from the symmetry group and placing it at or near the lattice point $(0, 0)$. In cases where there is a single unique point group generator, such as $*2^\bullet$ in $*442$, the choice is clear. In cases where there are multiple unique point group generators, such as $*6^\bullet$, $*3^\bullet$ and $*2^\bullet$ in $*632$, we choose the generator with the fewest number of valid positions on the lattice; in the case of $*632$ this is $*6^\bullet$. In cases where there are multiple point group generators of the same type, such as 2222, each one of the generators is placed at or near $(0, 0)$ in turn and lexicographic comparison of the corresponding labels is used to find the representative of the class. For 2222, the worst case scenario, this is a comparison of four different labels.

For symmetry groups with single mirror reflections or glide reflections, we must compare the labels rooted at any lattice point $a \in A$ where A is the set of lattice points in the period rectangle that intersect the mirror or glide line l. If l does not intersect any lattice points, then A is the set of lattice points in the period rectangle that are near l. In Figure 3, lattice points in A are indicated by red dots.

One final consideration for the algorithm is the subgroup relationships between symmetry groups. When generating a pattern with 632 symmetry, a possible outcome induced by the generators is a pattern with symmetry $*632$. The pattern produced by the algorithm must be tested for additional "unwanted" symmetry.

Results

Figures 4 and 5 show example tesselace patterns for all 17 planar periodic symmetry types along with some worked examples.

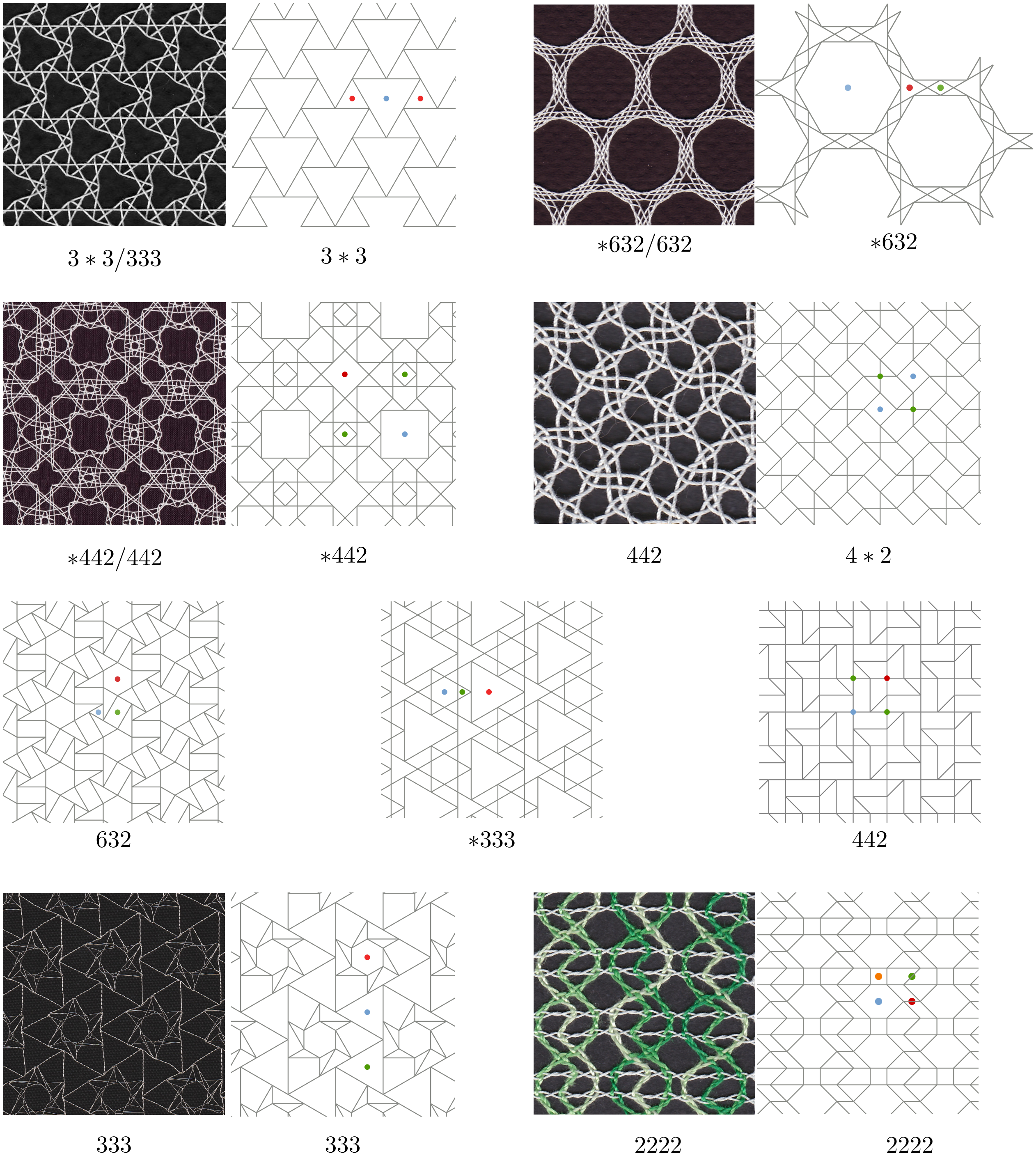


Figure 4: *Examples of tesselace patterns from each of the 17 periodic planar symmetry groups*

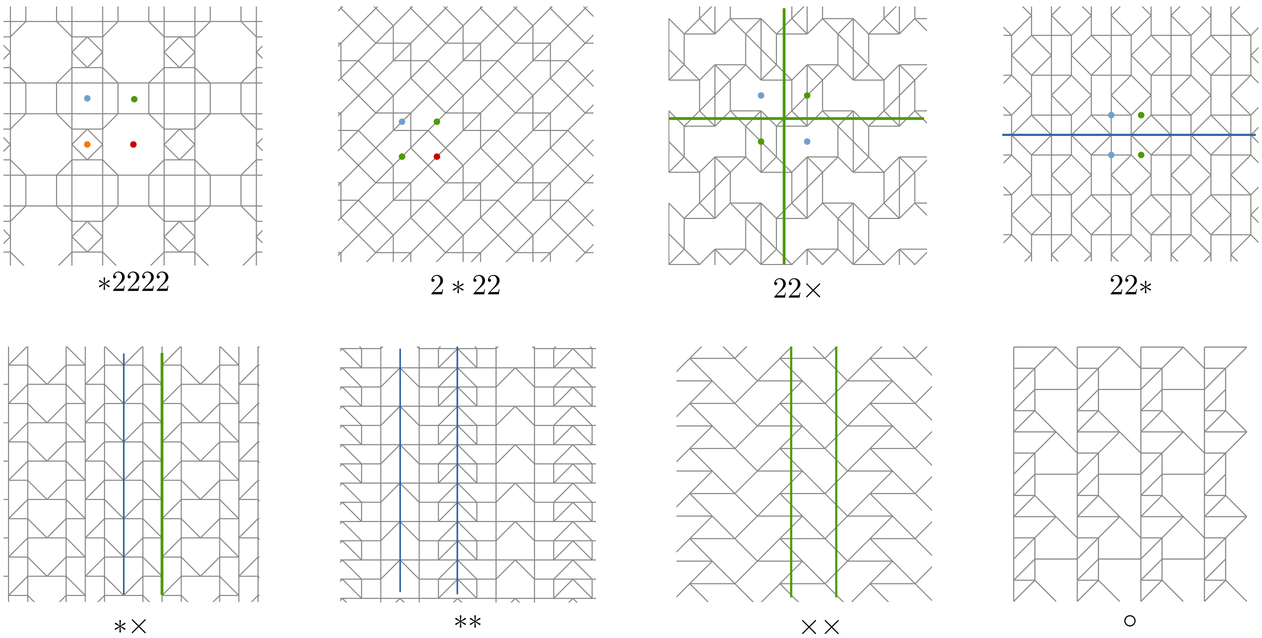


Figure 5 : *Examples of tesselace patterns, continued*

In addition to finding patterns having a particular symmetry type, this algorithm allowed us to explore much larger grid sizes. Using the previous lattice path approach, our exploration was restricted to grids less than 5×5 in size [6]. Using the symmetry approach, we were able to explore grids as large as 10×10 and 18×6 with partial results for even larger grids. The increased size of the tesselace patterns is due in large part to the repetition of the data in the solution. For example, in a pattern with $*2222$ symmetry, only a quarter of the grid needs to be explored, the rest of the solution data is populated by reflection.

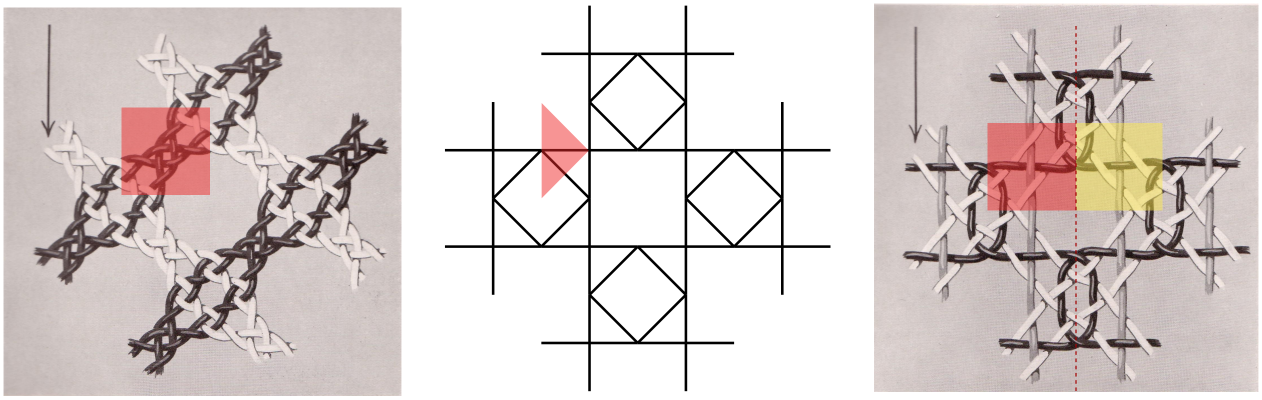
Work in Progress: Interlaced Symmetry

We can also consider the symmetry of bobbin lace at the level of individual threads. Because lace is a very thin fabric, the symmetry of its threads can be described in terms of a two-sided plane. Modeling a thin layer as a two-sided plane can be traced to research on thin films and mono-molecular layers conducted by chemists and physicists in the 1930s [4]. This approach was also used by Cromwell to describe the interlaced patterns of both frieze [2] and doubly periodic [3] celtic knots.

In bobbin lace, as in celtic knots, the top and bottom of the two-sided plane are very closely related: each thread is visible from above and below with an over-crossing on one side corresponding to an under-crossing on the other. To capture the polarity of the thread crossings, we visualize the motif of a pattern as a tile that is white on one side and black on the other. If all of the motifs are oriented the same way (e.g., all are white side up) then the possible periodic symmetries of the two-sided plane are the same as the well known 17 wallpaper groups for a one-sided plane. If the polarities are not completely aligned (some tiles are white side up while others are black side up) then there are an additional 46 symmetry arrangements which map bijectively to the two-colour symmetries of the one-sided plane [4].

In orbifold notation, two-colour symmetry types are labelled as F/K where F is the symmetry of the tiling without regard to colour (referred to as the "full group") and K is the symmetry that maps tiles of the same colour to one another (referred to as the "kernel") [1][1].

[1] For $**/**$ there are two distinct symmetry groups with the same F/K label which Conway et al. distinguish as $**/**(1)$ and $**/**(2)$



(a) *(b)* *(c)*

Figure 6 : *Two traditional patterns with the same pair traversal graph drawing: (a)* Maille flamands *(b) Tesselace graph drawing (c)* Fond à la vierge. *Hand drawn images of lace grounds from* The Art and Craft of Old Lace *[7].*

In Figure 6, we classify two traditional patterns which are both created from the same tesselace pattern using different braid word mappings. In *Maille flamands*, a 90° rotation of the red tile through a vertex of the tile around an axis perpendicular to the plane of the lace leaves the pattern invariant (up to thread colour). The tiles all have the same polarity orientation and the pattern possesses 442 symmetry. In contrast, the motifs in *Fond à la vierge* are oriented in two different ways represented by the red and yellow squares. A yellow square maps onto a red square by a 180° rotation through a side shared by the two tiles around an axis in the plane of the lace. *Fond à la vierge* possesses $*2222/2222$ symmetry. The $*2222$ full group symmetry can be observed by flattening the over and under crossings – that is, consider the shadow projected by this piece of lace. The 2222 kernel symmetry group takes red squares to red and yellow squares to yellow. Finally, the tesselace graph drawing itself has $*442$ symmetry. We note that 442 and $*2222$ are subgroups of $*442$.

We are currently in the process of going through catalogues of traditional lace patterns to identify which of the 63 possible two-sided plane symmetries have been used historically. As part of the process, we are ranking the symmetries by how often they appear and looking for symmetry types that should be possible but are not currently part of the lacemaker's repertoire.

References

[1] J. H. Conway, H. Burgiel, and C. Goodman-Strauss. The symmetries of things. *Wellesley, MA, A K Peters*, 2008.

[2] P. R. Cromwell. Celtic knotwork: Mathematical art. *The Mathematical Intelligencer*, 15(1):36–47, 1993.

[3] P. R. Cromwell. The distribution of knot types in Celtic interlaced ornament. *Journal of Mathematics and the Arts*, 2(2):61–68, 2008.

[4] B. Grünbaum and G.C. Shephard. *Tilings and patterns*. Freeman, 1987.

[5] V. Irvine. Broadening the palette for bobbin lace: A combinatorial approach. In *Proceedings of Bridges 2012: Mathematics, Music, Art, Architecture, Culture*, pages 191–198. Tessellations Publishing, 2012.

[6] V. Irvine and F. Ruskey. Developing a mathematical model for bobbin lace. *Journal of Mathematics and the Arts*, 8(3-4):95–110, 2014.

[7] A. von Henneberg. *The Art & Craft of Old Lace*. B.T. Batsford, 1931.