andrew_glassner@yahoo.com

Hierarchical Textures

Andrew Glassner

Working with visual patterns is one of my delights in working with computer graphics. As I've mentioned here before, I believe that rich, interesting patterns create an oasis of aesthetic pleasure between the arid lands of complete randomness and complete order.

In addition to their inherent beauty, patterns prove useful in graphics when we use them as textures. Textures were originally used to add color detail to shapes, like putting a decal on a model plane or wallpaper in a new apartment. But textures were too useful to be contained, and their uses quickly exploded. Today, textures control everything from shape details to the placement of extras in digital crowd scenes. In modern graphics, textures are used liberally, creating surface appearances as diverse as wood, brushed aluminum, cork, or wet leather; optical effects like fake caustics on the bottom of swimming pools; and even surface details like the rounded-off corners on a wooden box.

The first step in working with textures is creating them. Often we start with a photograph or a hand-drawn image. I also like creating textures in some algorithmic or procedural way, which lets us make infinite amounts of seamless but interesting patterns. Though not appropriate for everything, such patterns can be used for applications from ornamentation to creating complex 3D structures and motion.

One easy and powerful way to create rich textures is to start with something simple, and then use that seed along with some rules to "grow" something more complicated. Lots of growing schemes produce both repeating and nonrepeating patterns. One of the simplest techniques is called "tiling," where you place down multiple copies of a single starting shape, or tile, according to some (usually simple) rules. You can create rich tilings in surprisingly many ways. This issue, I'll look at a tiling technique based on using a set of simple symmetry operations.

Reflecting for a moment

Our goal is to create arbitrary amounts of texture from a simple starting shape. Only three regular polygons cover the plane without gaps or overlaps: the equilateral triangle, the square, and the regular hexagon. Of course, an infinite number of other shapes can do the job, but these are the simplest. To make things easy, let's focus exclusively on the square.

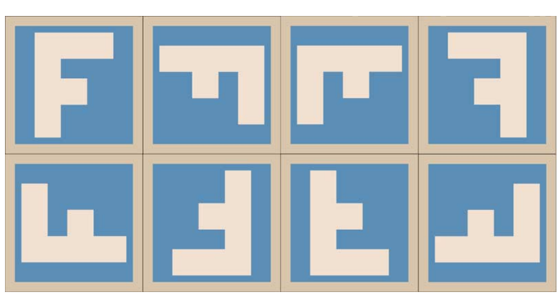
How many ways are there to place a square on the page, so that the sides of the square are parallel to the sides of the page? Figure 1 shows that the answer is eight. Starting from the original orientation, you can rotate the square 90, 180, or 270 degrees. Then you can repeat all four orientations with a flipped-over square, giving a total of eight possibilities. As long as you keep the sides of the square from rotating, this exhausts the ways you can place a square tile onto the plane.

To build patterns, I'll start with a square tile, then make a few copies of it in specific orientations and positions, creating a new cluster of tiles. I'll only build up square- or rectangular-shaped clusters. To keep things tidy, if the result of any of these replacement steps creates a cluster that's shaped like a rectangle, I'll scale it as needed to turn it into a square. That way we're always dealing with square shapes.

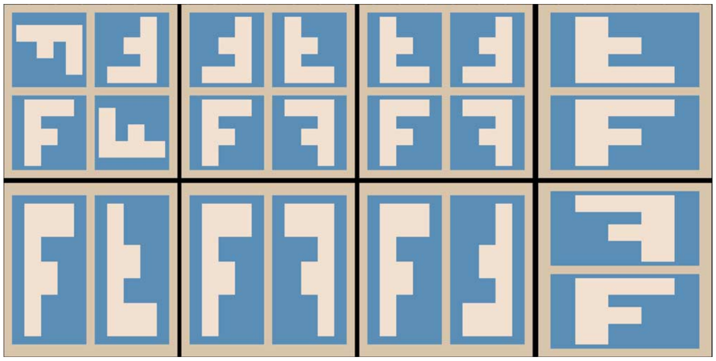
Suppose that we want to make a tiny, 2-by-2 grid of squares. How many possible grids can we make? Let's start by thinking about the upper-left square—as we've seen, that tile can be placed in any of eight orientations. Now let's move to the upper-right square—it also has eight possibilities, so there are a total so far of $8 \times 8 = 64$ pairings. Including the other two squares we find that we can create $8^4 = 4{,}096$ different four-square patterns.

Mathematically, that's perfectly acceptable, but for artistic purposes that's going to be way too many possibilities to work with in any reasonable way. Worse, most of them will look pretty wild and random. My goal here is to create interesting patterns, so I'd like to make sure that the arrangements we use for building up those patterns have some interesting structure themselves. Symmetry operations are a nice way to get that structure.

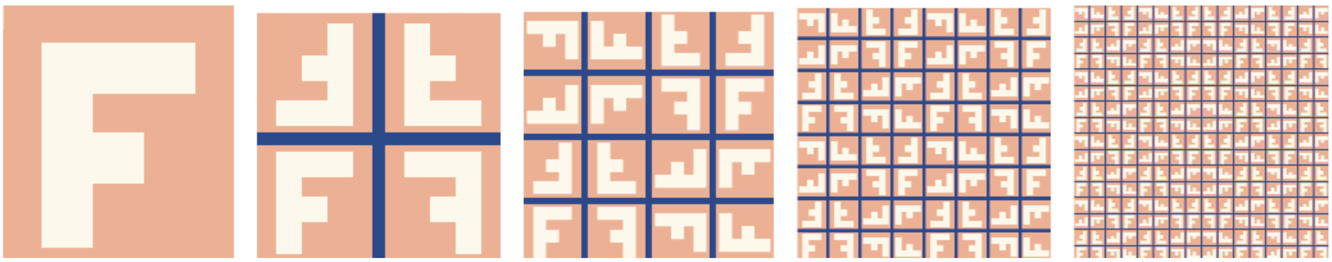
I'll focus on four types of symmetry operations: translation, rotation, reflection, and glide translation. We saw all of these in May 1996 in my



1 **The eight ways to place a square tile. The left half of the figure shows the normal orientation of the F tile including rotations of 0, 90, 180, and 270 degrees. The right half repeats those rotations, after first reflecting (or flipping) the tile.**

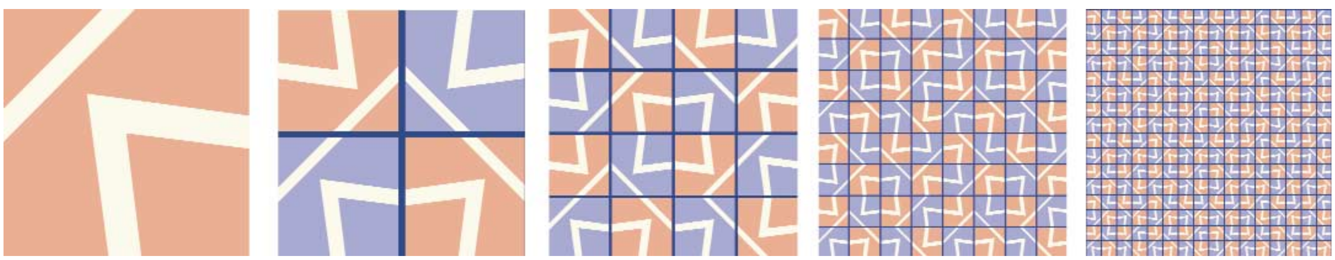


2 Some symmetry patterns applied to the square F tile, named with traditional symmetry pattern symbols. The first three cluster together four tiles, the rest combine two tiles, and then the result is scaled to fit back into a square. Top row, from left to right: p4, p2mg, p2mm, reflectY. Bottom row, from left to right: halfglideX, pm, r2, pg.



(a) (b) (c) (d) (e)

3 (a) The F motif. (b) Applying p2mg. (c) Applying p4, then p2mg. (d) Applying p4, p4, p2mg. (e) Final pattern: p2mg, p4, p4, p2mg. I call this hierarchy pattern h4.



(a) (b) (c) (d) (e)

4 The same process as in Figure 3, except with a different motif.

column on frieze groups. To review, translation means just placing copies side by side, rotation means turning the tile 90 degrees, reflection means flipping it over, and glide translation is a translation followed by a reflection (the axis of this reflection is perpendicular to the direction of movement).

Figure 2 shows a collection of some patterns built up from these symmetry operations, along with the notation that I'll use for them. The cryptic symbols in the caption are a minor variation of the standard shorthand used by crystallographers to describe different symmetry patterns. I've also included some useful pairs in addition to the foursomes, and you can see that I've had to scale those results to make them into squares again. Figure 2 will be our working set of patterns, selected for their symmetry properties from the potential pool of thousands. Let's use these basic patterns to build something interesting.

A square deal

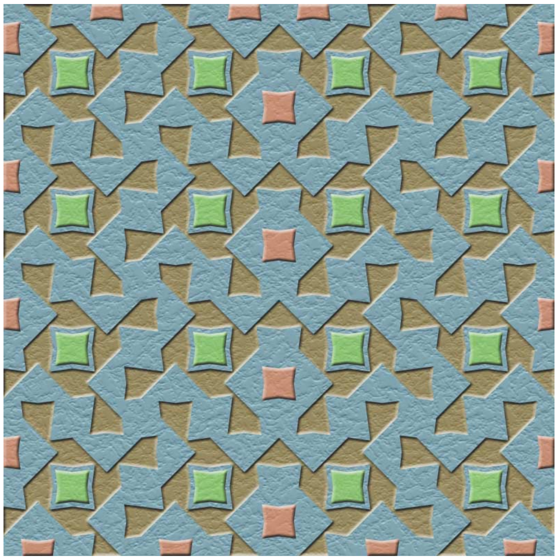
Let's get going by first placing down a single F motif, as in Figure 3a. I'll now replace the F tile with the p2mg pattern that we saw in Figure 2, resulting in Figure 3b. Since the original F was in the "normal" position (that is, looking like the letter F), we didn't have to rotate or flip the p2mg cluster before putting it down, but otherwise we would. For example, if the original F was rotated 90 degrees, then we'd rotate the p2mg cluster 90 degrees as well before putting it down.

We now have four new tiles. We'll make things more interesting by adding some more detail. One way to go is to replace each of the tiles with one of the patterns from before. That's a straightforward substitution technique, and I've written about it before. This time let's turn the process around a little bit.

To make things more interesting, I'll add in the p4 pattern, but I'll apply that before the p2mg we already have. Figure 3c shows the result. If you look carefully—and it can be a little mind-bending at first—you can see that the pattern of Figure 3b appears in each of the quadrants, after it has first been appropriately rotated by p4.

Now I'll apply another p4, as in Figure 3d, and another p2mg, resulting in the 16-by-16 pattern in Figure 3e. I admit that this arrangement of letter F's isn't the most lovely thing to look at, but it forms the scaffolding for something more interesting.

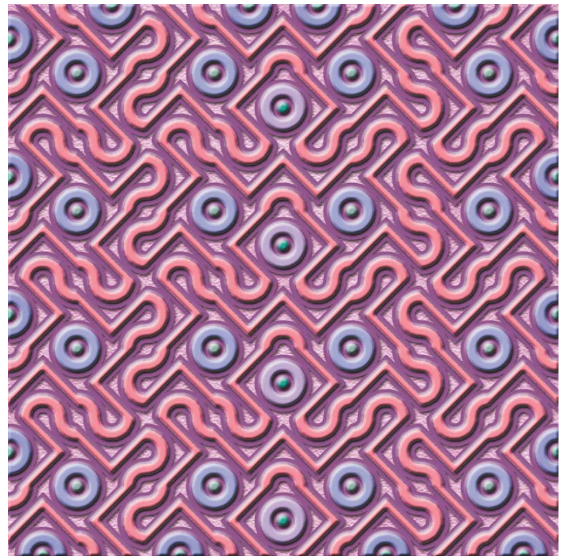
Figure 4a shows a new motif in place of the F shape.



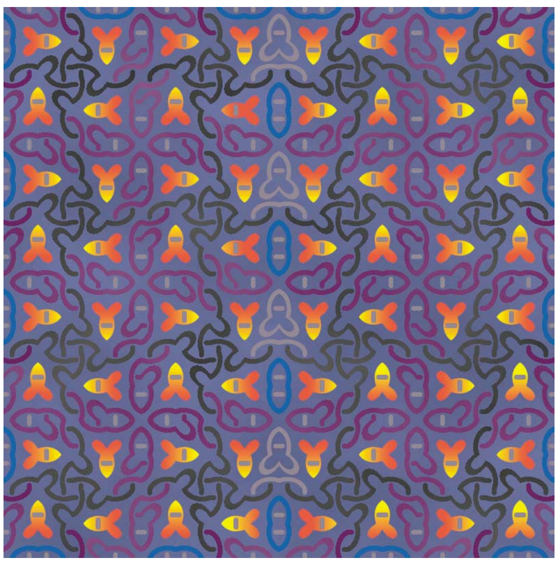
5 **A hand-decorated version of Figure 4e.**

6 **Four square motifs.**

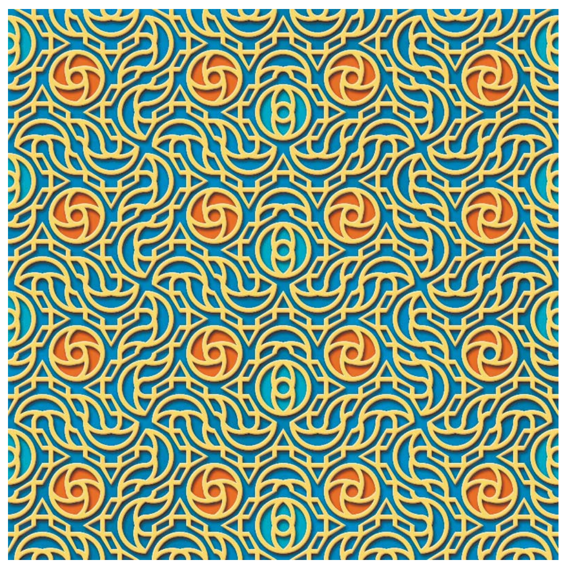
(a) (b) (c) (d)



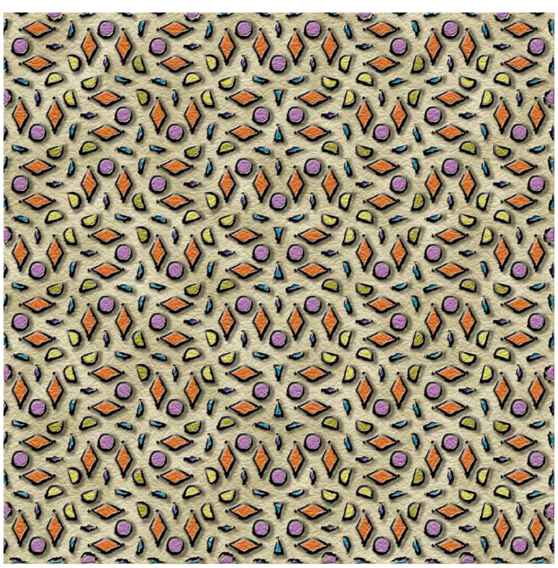
7 **Motif of Figure 6a under hierarchy h4.**



8 **Motif of Figure 6b under hierarchy h4.**



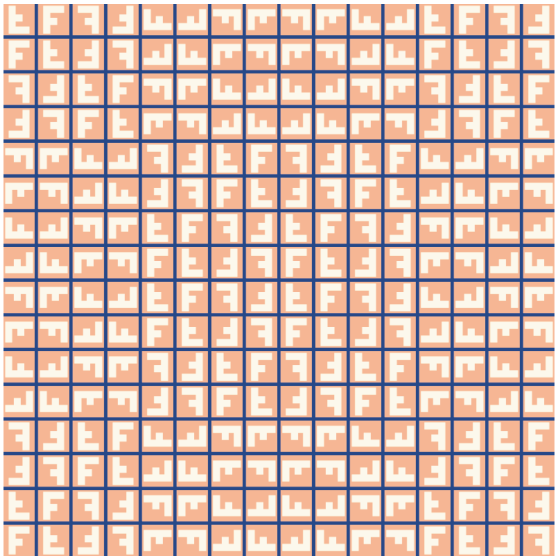
9 **Motif of Figure 6c under hierarchy h4.**



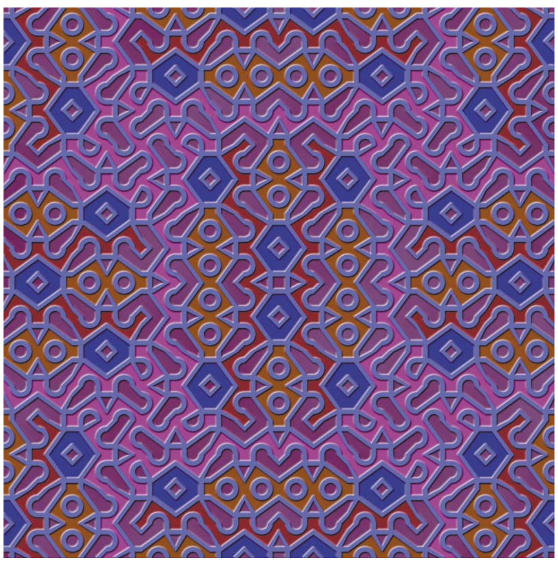
10 **Motif of Figure 6d under hierarchy h4.**

Now I'll complete Figure 4 by applying exactly the same transformations to this little geometric figure that I did in Figure 3. For clarity, I've colored alternating squares with different colors so that it's easier to see the tiles that make up the grid (I added the coloring by hand after I built the pattern). This coloring isn't part of the pattern per se, but is just to help keep our eyes from blurring over. Figure 4e shows the resulting geometric pattern.

Now that we have this template, we can color it by



11 **Pattern h5: p2mm, p4, p2mg, reflectY, halfglideX.**

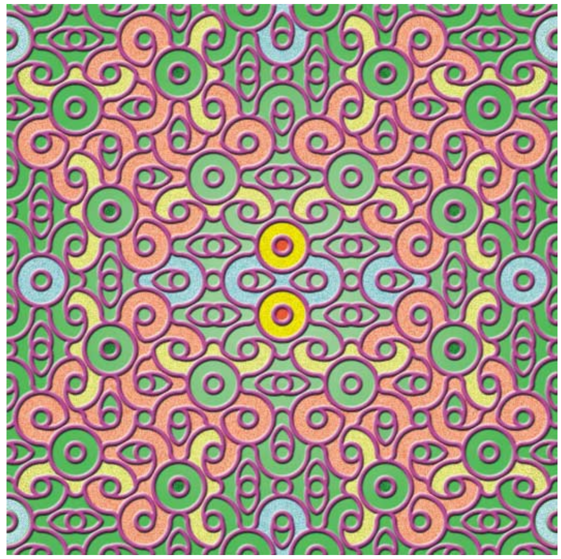


14 **Motif of Figure 12b under hierarchy h5.**

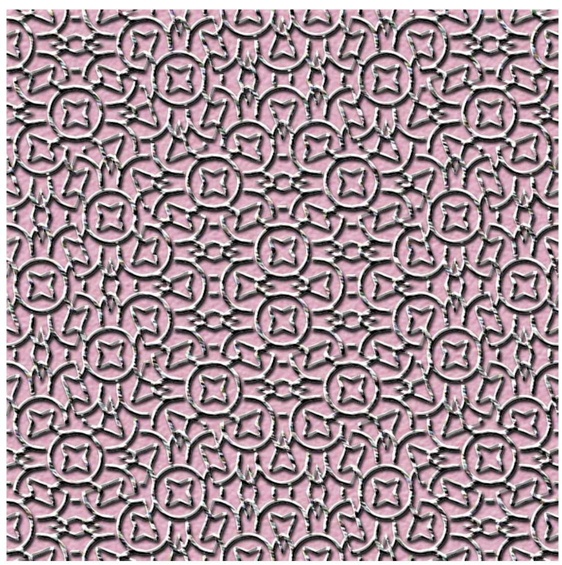
hand and make some interesting patterns. Although it's perfectly possible to hand-paint these patterns, like hand-tinting a black-and-white postcard, I'm thinking of something a little more general. In many programs, you can create a *macro*, or little prerecorded sequence of operations, that you want to apply to an image. Once you've designed and recorded this sequence, you can play it back on top of any image and get back the sort of result you were originally after—this is how we can algorithmically generate lots of interesting texture from simple starting patterns. In this column I used Photoshop to decorate my patterns, storing my procedures in what Photoshop calls *actions*. Figure 5 shows the result of Figure 4e after I've dressed it up a bit in this way with an action that I cooked up (actually it was run on a version of Figure 4e that didn't include the alternating colors and dark-blue grid between the tiles).

The hierarchical sequence of transformations (which

12 **Three square motifs.**



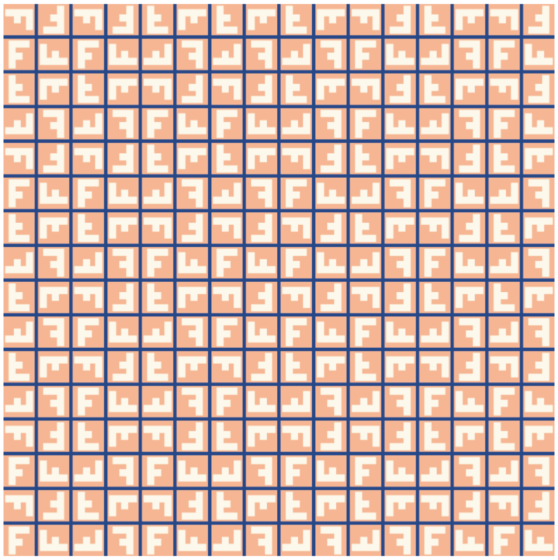
13 **Motif of Figure 12a under hierarchy h5.**



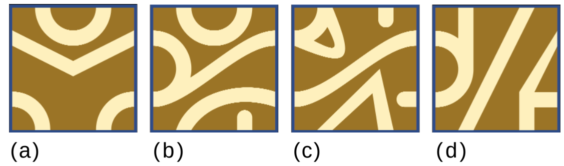
15 **Motif of Figure 12c under hierarchy h5.**

I call *h4*) used for Figures 4 and 5 can be applied to other starting tiles. Figure 6 shows four such tiles. Applying h4 to each of the tiles creates surprising results. Figures 7, 8, 9, and 10 show four more abstract tiles and the result of building them into a hierarchy using pattern h4 (and then decorating the results with a variety of custom actions).

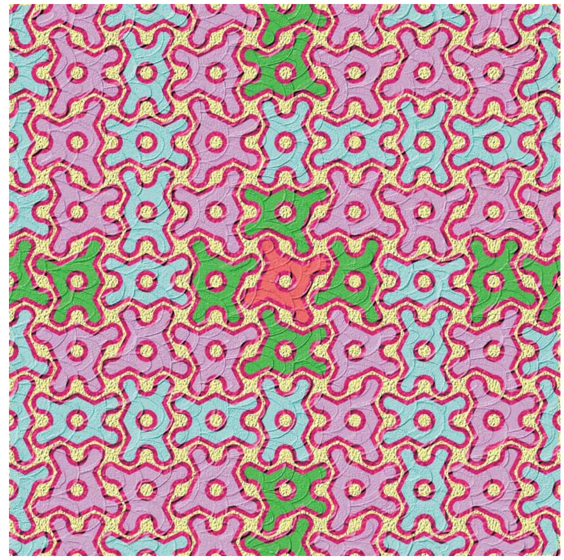
Other symmetry hierarchies yield different kinds of looks. Figure 11 shows a hierarchy I call *h5*, consisting of the sequence p2mm, p4, p2mg, reflectY, halfglideX (compare this to Figure 4e). Figure 12 shows three motifs, and Figures 13 through 15 show some textures



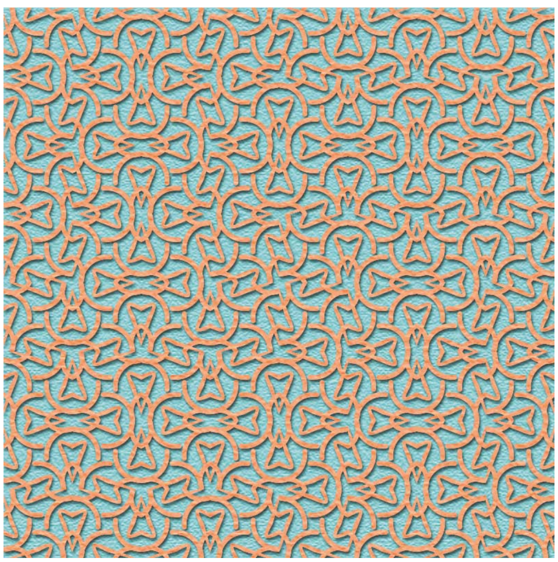
16 **Pattern h1: p4, p2mm, p2mg, p4.**



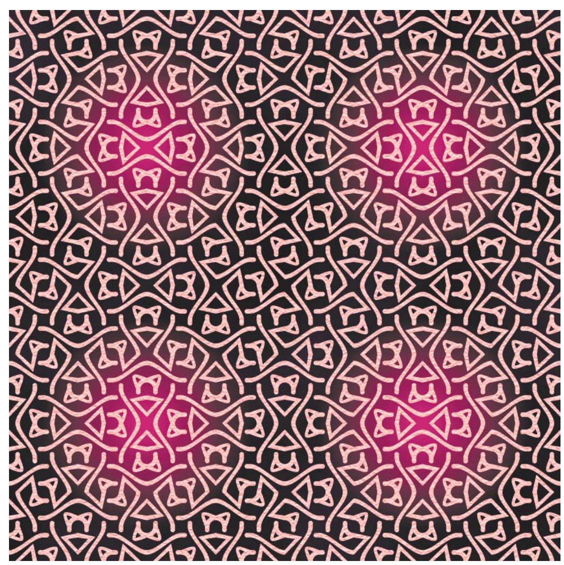
17 **Four square motifs.**



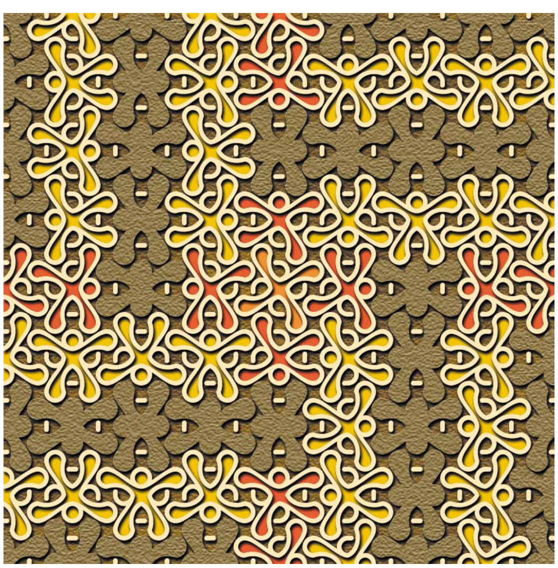
19 **Motif of Figure 17a under hierarchy h1.**



18 **Motif of Figure 12c under hierarchy h1.**



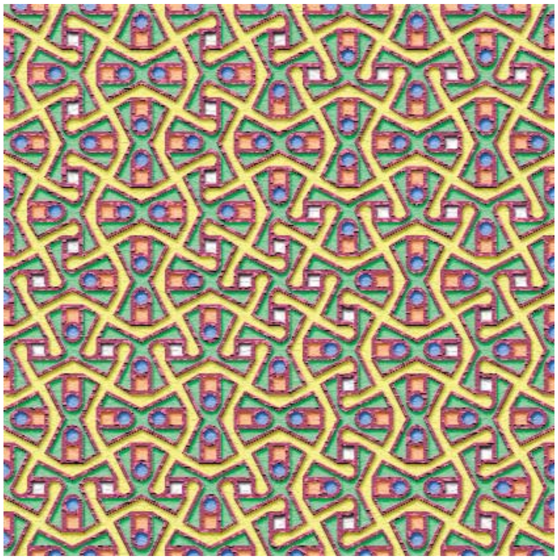
21 **Motif of Figure 17c under hierarchy h1.**



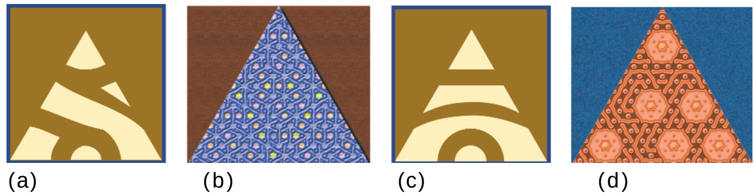
20 **Motif of Figure 17b under hierarchy h1.**

I made by applying h5 to those tiles, and then algorithmically decorating the results.

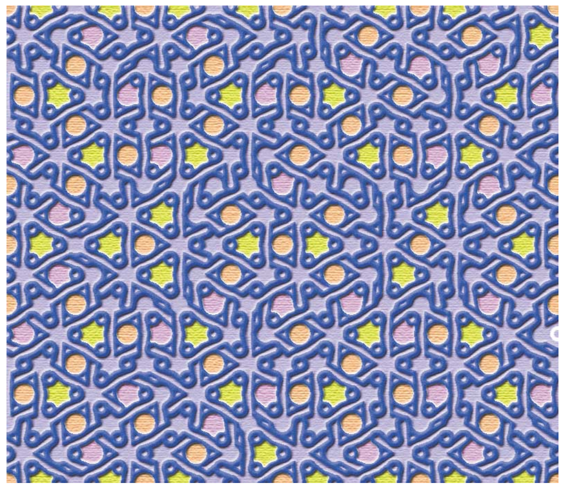
Hierarchy h1 is almost h4: it's p4, p2mm, p2mg, p4. Figure 16 shows the pattern, and Figure 17 shows a new set of motifs to apply it to. Although I've been using a new starting tile for each figure to show some variety, the same tile can create different textures under different hierarchies. Figure 18 shows the same tile used to make Figure 13, but using hierarchy h1 instead of h5.



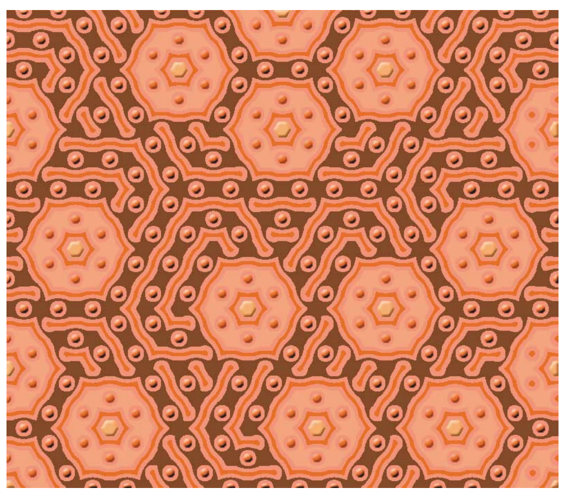
22 **Motif of Figure 17d under hierarchy h1.**



23 **Creating triangular patterns. (a) A triangular motif. (b) A pattern based on Figure 23a. (c) Another triangular motif. (d) A pattern based on Figure 23c.**



24 **Pattern of Figure 23b extended to cover the plane by rotation and tiling.**



25 **Pattern of Figure 23d extended to cover the plane by rotation and tiling.**

Figures 19 through 22 show the tiles of Figure 17 applied to this hierarchy.

Tri, tri again

Squares are surprisingly versatile, but they're just one of many shapes that tile the plane. You can build textures using this technique with any shape that can be broken down into copies of itself—such shapes are called *reptiles*, because they're tiles that repeat or replace themselves. Figure 23 shows the result of applying symmetry hierarchies to equilateral triangles. The resulting triangular patterns can be used to tile the plane by rotating the result and placing the copies side-by-side; Figure 24 and Figure 25 show the result.

Although I enjoyed exploring the flexibility of the humble square, there's no other reason to restrict yourself to regular polygons, or even reptiles. For example, you can create a transformation that replaces a square with a diamond and four small triangles, then apply new transformations to those.

There are endless ways to make beautiful and interesting patterns, useful for everything from creating carpets and wallpaper to arranging city buildings. Working with textures is fun, and many texture-building programs are easy to write. (The Visual Basic program I wrote to make all the square and triangular patterns in this column, including a simple user interface, is about 800 lines of easy code.) Lots of sophisticated ways exist to create texture and patterns, but I still love finding the power inherent in even the simplest approaches. ■

Readers may contact Glassner by e-mail at andrew_glassner@yahoo.com.