

Animated Map Colourings of Hinged Squares

Craig S. Kaplan

School of Computer Science, University of Waterloo, Ontario, Canada; csk@uwaterloo.ca

Abstract

The regular tiling by squares can be unfolded by placing hinges at the vertices of tiles, connecting squares to their neighbours. The hinges open continuously to reveal rhombs, which grow into squares and then close up again. The intermediate square tiling offers an opportunity to create an animated loop, but colouring the tiles in that loop so that adjacent tiles never share a colour offers some interesting mathematical challenges. I discuss how many colours are required to permit a looping animation of unfolding squares, and demonstrate a few such loops.

Introduction

The creation of looping mathematical GIFs is an engaging interdisciplinary pastime [2]. An artist might imagine a satisfying, surprising, or dramatic transformation of an abstract geometric design. They must then deduce mathematical rules that express the design and the transformation, and reduce those rules to a computer program that renders a sequence of images that can be stitched together into a looping GIF.

Throughout fall 2020, I undertook a challenge of creating looping GIFs based on checkerboards. An animation would begin with a checkerboard of black-and-white, axis-aligned squares, undergo some sort of geometric transformation, and end with the same checkerboard, thereby closing the loop. The humble checkerboard served as an excellent vehicle for experimentation and play. It provided a common theme for a collection of animations, and a constrained aesthetic domain that rewards creative new ideas.

I shared many of these creations on social media, and others contributed their own loops. I was particularly intrigued by an animation created by Robert Fathauer; I recreate a selection of frames in Figure 1. The animation has a natural physical interpretation in terms of a tiling of squares with hinges at their corners. One square rotates in place at the centre of the frame, and every hinge opens from 0 degrees through 180 degrees and then closes again to complete a cycle.

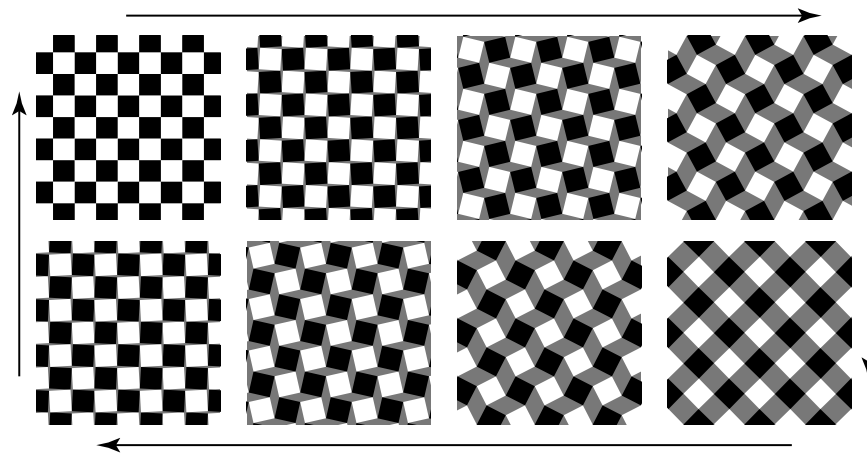


Figure 1: A recreation of eight frames from an animation by Robert Fathauer showing hinged squares. The frames are to be read in a circle, as indicated by the arrows surrounding them.

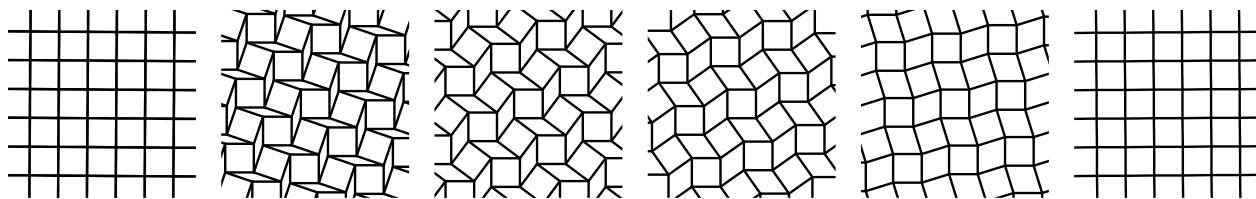


Figure 2: Half of the hinged square loop, showing only the outlines of tiles. The tiling as a whole rotates smoothly, so that when we reach the frame shown on the right, we apply a rotation of 45 degrees. The resulting tiling is therefore identical to the starting point, permitting an animated loop.

The implied static tiling by squares and rhombs, and this hinged motion, have a long history in mathematics and art. The tiling appears in architecture and ornamentation at least as far back as Roman mosaics. The hinged motion has been considered by numerous mathematicians and designers [1, 5], and a polyhedral analogue was famously christened the “Jitterbug” by Buckminster Fuller [6]. Most recently, these hinged structures have become popular in geometry processing and fabrication, where they can act as a macroscopic simulation of auxetic materials [3].

The bottom-right image in Figure 1 shows the halfway point of Fathauer’s loop, when all hinges are open to 90 degrees. Notice that if we ignore colour, the halfway point is itself a tiling by squares, rotated by 45 degrees relative to the start of the loop. We can therefore imagine adding a global rotation to the animation, about the central square, which reaches 45 degrees just as the hinges open to 90 degrees, yielding a loop that is closed geometrically. I visualize this loop in Figure 2, discarding colour and drawing only the outlines of tiles. The central white square stays completely fixed as the rest of the tiling rotates and translates around it.

I refer to one iteration of the cycle shown in Figure 2 as an “unfolding”. This shortened unfolding loop is somewhat paradoxical. The squares of the original checkerboard move to positions in which they occupy exactly half the plane, with new squares filling in the gaps. But at the point where we restart the loop, the new configuration of squares is remapped to the original one. With Fathauer’s colouring, this remapping step would be visually discontinuous, because colours are not preserved.

In this paper, I investigate the question of whether there are other colourings of the square tiling that can form closed loops after a finite number of unfoldings. I am especially interested in *map colourings*, colourings in which squares and/or rhombs that share an edge always have distinct colours. We might study several different mathematical questions in this context, notably the number of colours required, the number of unfoldings required for a given colouring, and the difference between a colouring that works on a finite region of the plane versus a colouring that is defined on the infinite checkerboard. In the following sections, I will present some preliminary results that address these questions.

Please visit isohedral.ca/hinged-squares for animated versions of many of the diagrams and results shown in this paper.

Minimizing the number of colours

In this section I argue that at least five colours are necessary in any map-coloured animation of hinged squares. In subsequent sections I will show that five colours are also sufficient.

We begin with a simple observation about the relationship between old and new squares in an unfolding, illustrated in Figure 3. An unfolding opens up rhombs next to every edge of every square in the initial tiling, which then grow into new squares. In particular, if we pick any vertex in the initial square tiling, unfolding will introduce a new square that is adjacent to the four squares that initially surrounded that vertex.

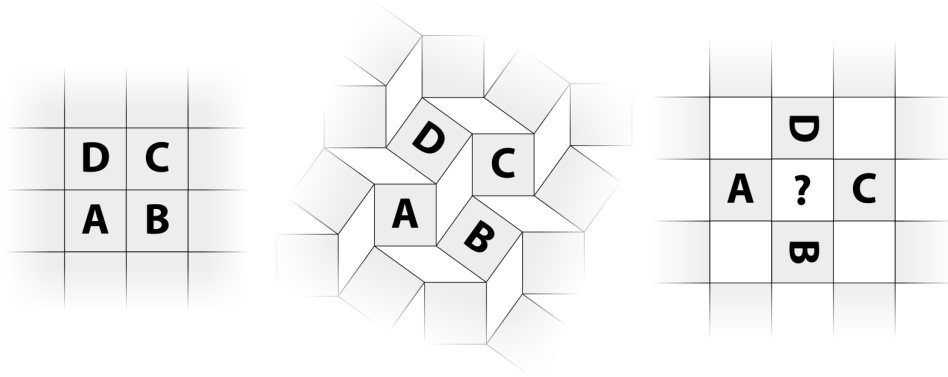


Figure 3: A visualization of the local effect of unfolding on four squares around a vertex. The four labelled squares move to the positions (and orientations) indicated on the right, leaving behind a new square tile, indicated by a question mark, that is adjacent to all of them.

Using this observation, we can determine the minimum number of colours required by examining candidate colourings of local neighbourhoods of squares under repeated unfoldings. For example, Fathauer’s original loop demonstrates that two colours are never sufficient. The only map colouring of the square tiling by two colours is the alternating checkerboard shown in Figure 1; a single unfolding requires a third colour.

A similar argument, visualized in Figure 4, shows that three colours do not suffice. Select a vertex in the tiling surrounded by squares of three different colours. (If no vertex uses three colours, then we must have an alternating checkerboard, and a single unfolding will produce a candidate vertex as above.) After one more unfolding we will create a new square adjacent to all three initial squares, necessitating a fourth colour distinct from the three we started with.

We can also show that four colours do not suffice, though the argument requires two unfolding steps rather than one. Let us assume that we have four colours available. If any vertex is adjacent to squares of all four colours, then a fifth colour will clearly be needed after one unfolding, so we assume that no such vertex exists. Select a vertex surrounded by squares of three different colours (possibly unfolding once until such a vertex exists). As we saw in the three-colour case, one unfolding will create a square requiring the fourth colour, as shown on the right in Figure 4. Now we unfold this larger neighbourhood, as shown in Figure 5. Here we have two distinct vertices with three known colours around them, forcing us to choose the fourth colour deterministically in both cases. These choices will produce a vertex surrounded by squares of all four colours, which we know cannot be unfolded without introducing a fifth.

Five-colour loops with finite extent

The previous section showed that five colours are necessary, but we have not yet established that five colours are sufficient—perhaps six or more are needed in any looping animation. On the other hand, there is good reason to believe that five should be enough. After each round of unfolding, we are faced with the problem of colouring all the new squares we have introduced. Each such square shares an edge with exactly four neighbours, all squares from the tiling at the start of this round. Because there are only four neighbours to contend with, there is always at least one free colour available for each new square. This argument shows that five colours are always enough to unfold as many times as we wish, but it does not yet prove that there exist *closed loops*, sequences of colourings that repeat after a finite number of unfoldings.

In this section I offer a simple algorithm for constructing closed five-coloured loops limited to a fixed grid of any finite size. Obviously, a finite grid suffices in any practical animation context. In the next section

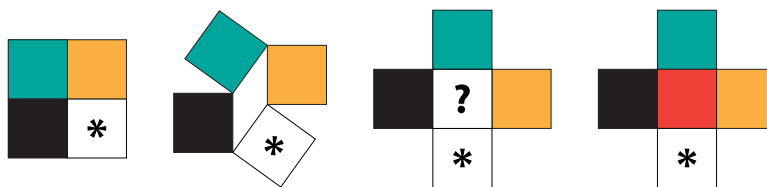


Figure 4: A visual proof that three colours are not sufficient in a map-coloured unfolding. The drawing on the left shows a vertex with three distinct colours, with the asterisk denoting a “wild card” whose colour does not matter. Unfolding creates a new square indicated by a question mark, which must use a fourth colour to maintain map colouring.

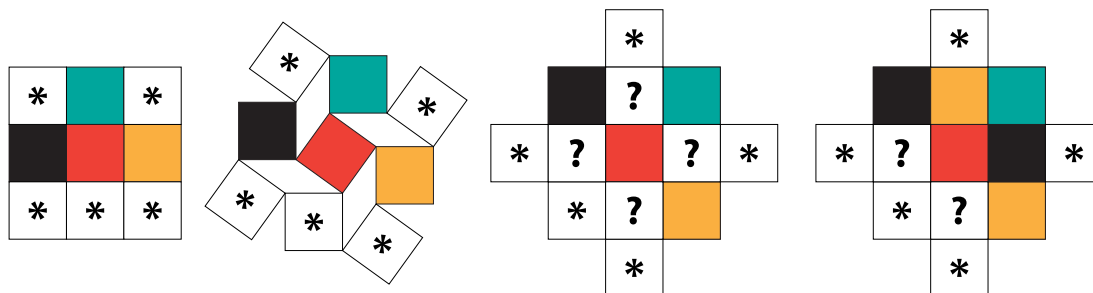


Figure 5: A visual proof that four colours are not sufficient in a map-coloured unfolding. A vertex surrounded by three colours unfolds into a configuration like the one on the right in Figure 4. Unfolding that configuration opens up two new squares whose colours are fully determined by their neighbours. But filling those two squares produces a vertex surrounded by four distinct colours, requiring a fifth to maintain map colouring.

I offer a mathematical construction that is periodic in space and time.

Construct a finite $n \times n$ grid of squares, and assume that the unfolding will take place around an “origin” square near the centre of the grid. Each unfolding step will move some squares off the edge of the grid, at which point we can simply forget about them.

Now choose a set of five colours $\{c_1, c_2, c_3, c_4, c_5\}$. Construct any initial colouring for the grid; for simplicity, I begin by assigning c_1 to every square. Label this initial coloured grid G_1 . Using G_1 as a starting point, we iteratively construct a sequence of coloured grids, stopping when we produce a grid that we have already seen. At each step we have some grid G_i . Unfold G_i , leaving behind a set of squares in need of colours. For each such square, one or more colours in the set will be distinct from all the colours used in that square’s neighbours; we simply choose the free colour with the lowest index. Label the newly coloured grid G_{i+1} .

This iterative process will produce a sequence of colourings G_1, G_2, G_3, \dots , where every grid from G_2 onward is necessarily map coloured. We halt the iteration when we produce a grid G_e that is equal to some previous grid G_s with $2 \leq s < e$. We know that we must arrive at G_e eventually, because there are only finitely many distinct map colourings of an $n \times n$ grid with five colours. We construct an animation by concatenating the unfoldings from G_s to G_{e-1} . I refer to this colouring method as the “lowest legal colour” algorithm.

Figure 6 shows an example of a looping animation on a finite grid, in this case of size 7×7 . This animation has period 6: six unfolding steps are required to return to the initial colouring. The figure shows

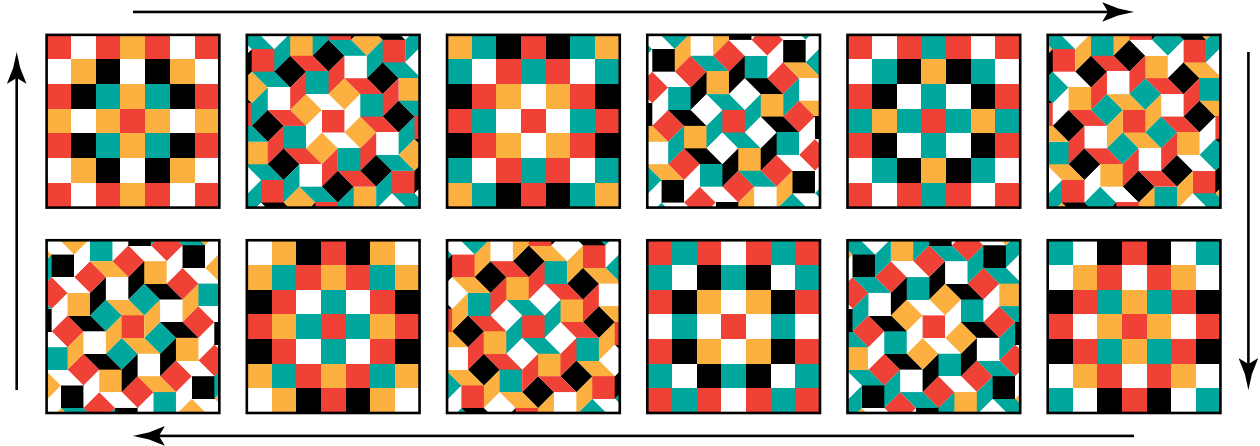


Figure 6: An animated loop that repeats after six unfoldings of a 7×7 grid, created using the lowest legal colour algorithm. The loop is arranged in a circle, as indicated by the arrows.

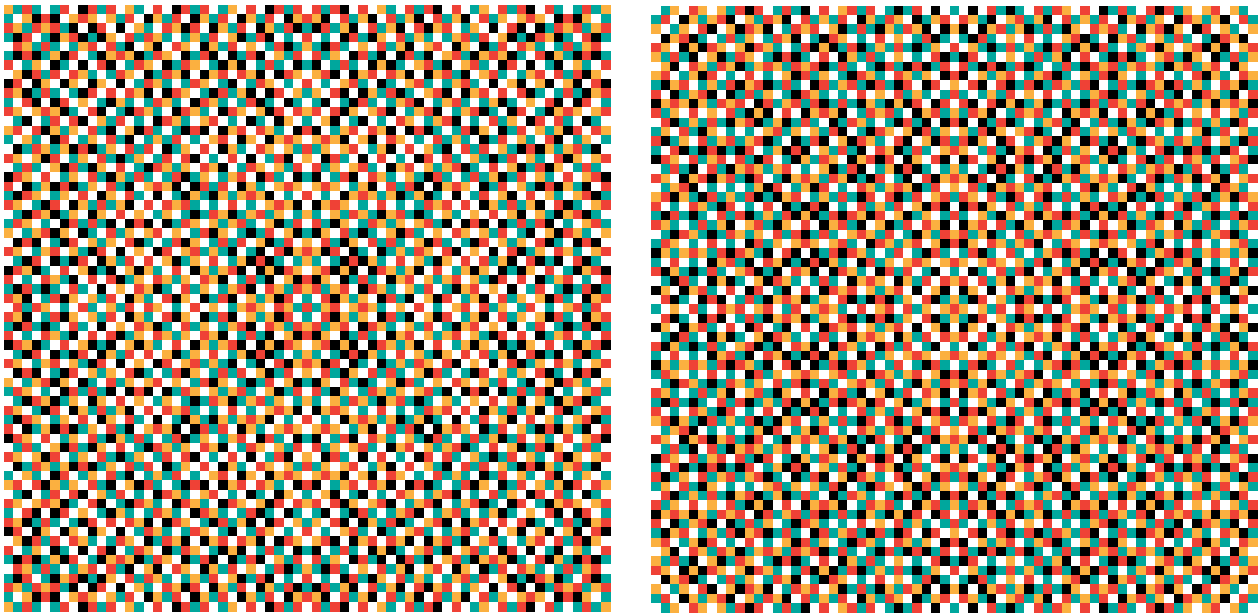


Figure 7: Two colourings from a looping animation of a 65×65 grid, created using the lowest legal colour algorithm.

six square grids alternating with six frames from the middle of each unfolding. Careful observation shows that the red and black squares alternate between two configurations and the other three colours are permuted in a cycle, which might offer a starting point for explaining the period. Indeed, I have found that this rule always leads to a cycle of period 6, regardless of the dimension of the grid (though larger grids require more initial steps to arrive at the cycle).

Notice that the grids all have d_4 symmetry (the rotations and reflections of a square), and the intermediate frames have c_4 symmetry (just the rotations). If G_1 is coloured symmetrically, as in the case of a simple uniform colouring, those symmetries will be preserved throughout the unfolding sequence. On the other

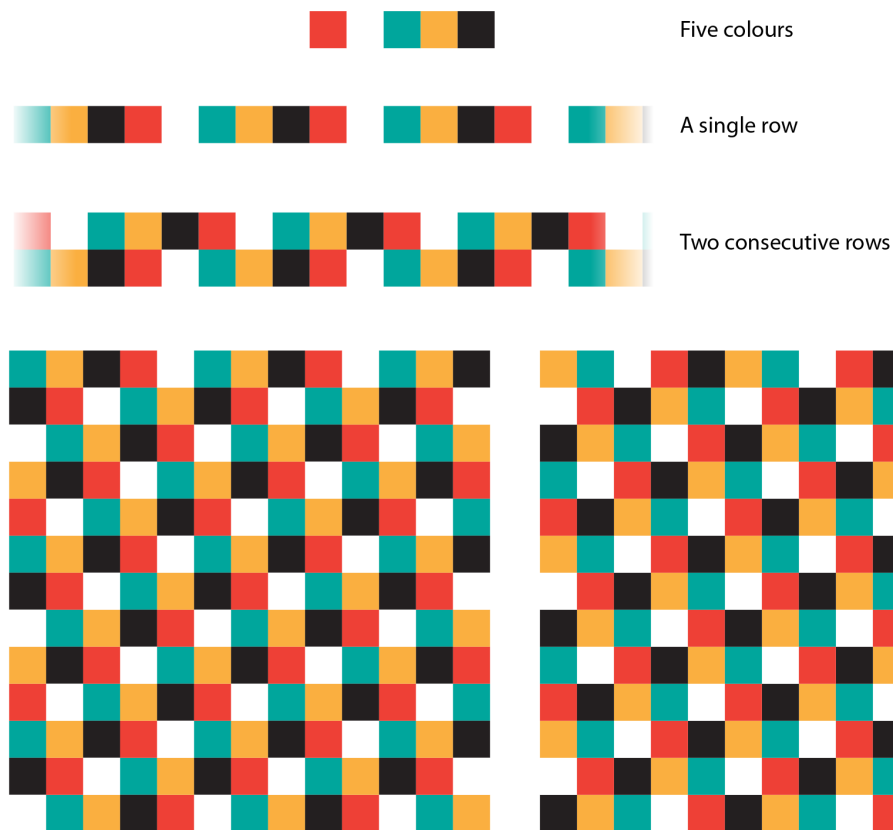


Figure 8: A periodic colouring that produces an unfolding animation with period 2. Five colours are repeated to form each row of the colouring, and adjacent rows are offset horizontally by two squares. Every vertex in the resulting colouring is surrounded by squares of four distinct colours.

hand, these colourings cannot easily be extended to periodic colourings of the whole plane. Figure 7 shows two colourings from a 65×65 grid cycle, a scale at which the lack of periodic repetition becomes more apparent.

A simple periodic loop

The lowest legal colour rule is not particularly principled, and so there is no reason to expect it to produce map-coloured loops that are spatially or temporally simple, even if such loops exist in principle. In this section I present the simplest possible animated map colouring, which uses five colours periodically across the whole plane and repeats after two unfoldings. I arrived at this colouring through trial and error.

The construction is shown in Figure 8. Choose five colours $\{c_1, c_2, c_3, c_4, c_5\}$ and place them in five consecutive horizontal squares. Repeat that five-square unit to create an infinite horizontal row of coloured squares. Now displace that row by multiples of the vector $(2, 1)$ to fill the plane. That is, a row is shifted one square up and two squares to the right to produce the row above it. In the resulting colouring of the plane, each colour is arranged periodically in a “knight’s grid”, where closest neighbours of the same colour are always offset by a knight’s move. All rotational symmetries of the tiling permute the colours, yielding what Rigby called a “chirally perfect colouring” [4].

In this colouring, shown at the bottom left of Figure 8, every vertex is surrounded by squares of four distinct colours. Therefore, after one unfolding, the colours of all new squares are fully determined, and we

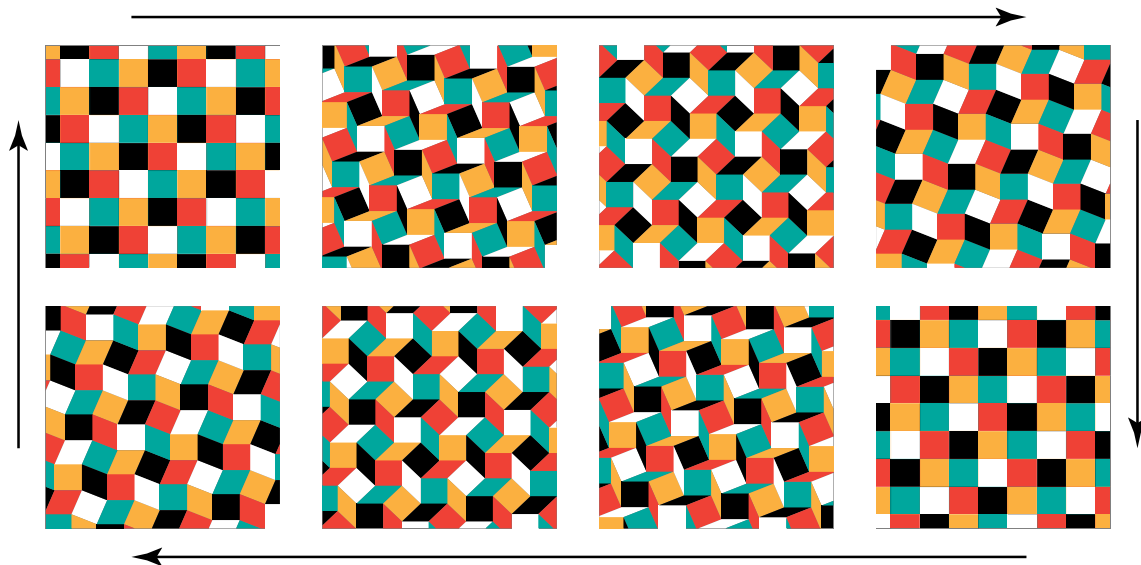


Figure 9: Selected frames from the animated unfolding of the colouring shown in Figure 8.

necessarily obtain the colouring shown on the bottom right of Figure 8. This new colouring has the same mathematical properties as the initial one, but permutes the colours. Happily, the permutation has order two, and another unfolding returns us to the initial colouring. Figure 9 shows eight frames from the complete animated loop.

This colouring is probably “ideal” in a mathematical sense, as a minimal fixed point of the unfolding process. As evidence, consider Figure 10, which shows a small excerpt from the upper left corner of a 257×257 grid coloured via the lowest legal colour algorithm. We see large patches of exact or approximate knight’s grid colouring arising organically.

Conclusion

Fathauer’s original hinged tiling animation naturally avoids the need for the techniques introduced in this paper. Rather than opening the hinges to 90 degrees and using the new square tiling to close the loop, he continues to 180 degrees, collapsing the rhombs and returning to the original checkerboard. By identifying the squares of the fully open tiling with those of the original, I uncover a collection of new challenges to be overcome in the construction of looping animations.

It would be interesting to explore what other numbers of colours and sizes of periods are possible with animated unfoldings of squares. On finite grids, it should be possible to use variations of the lowest legal colour rule to produce new loops with any number of colours beyond five. However, that does not help to determine which numbers of colours are possible if the colourings are required to be periodic in space.

Another direction for future work is to extend the idea of looping coloured unfoldings to other tilings. Here the path forward is less clear. The square tiling is the only regular or Archimedean tiling that unfolds to a transformed copy of itself. There may exist other more complex tilings that unfold to themselves using a similar hinged mechanism, but some new tiling-theoretic ideas may be needed to find them.

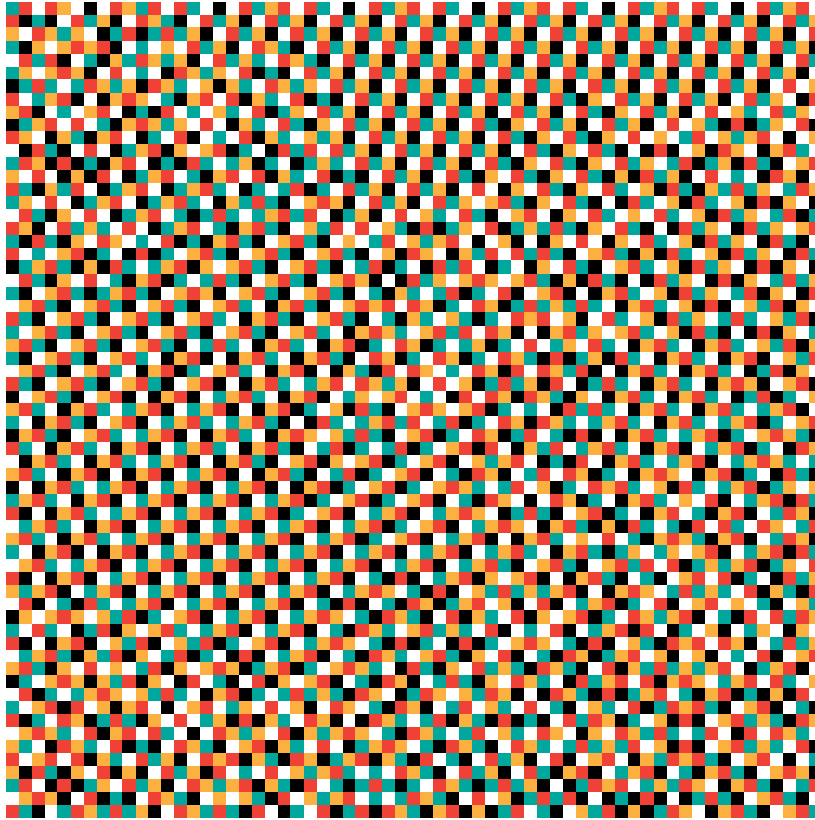


Figure 10: An excerpt from a 257×257 grid coloured using the lowest legal colour method. Large regions of the excerpt approximate the ideal colouring shown in Figure 8.

References

- [1] J. D. Clinton. A geometric transformation concept for expanding rigid structures. Technical Report NASA CR-1735, Southern Illinois University, 1971.
<https://ntrs.nasa.gov/api/citations/19710028061/downloads/19710028061.pdf>.
- [2] Craig S. Kaplan. Animated isohedral tilings. In Susan Goldstine, Douglas McKenna, and Kristóf Fenyvesi, editors, *Proceedings of Bridges 2019: Mathematics, Art, Music, Architecture, Education, Culture*, pages 99–106, Phoenix, Arizona, 2019. Tessellations Publishing. Available online at <http://archive.bridgesmathart.org/2019/bridges2019-99.pdf>.
- [3] Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. Beyond developable: Computational design and fabrication with auxetic materials. *ACM Trans. Graph.*, 35(4), July 2016.
- [4] John F. Rigby. Fun with tessellations. In Richard K. Guy and Robert E. Woodrow, editors, *The Lighter Side of Mathematics: Proceedings of the Eugène Strens Memorial Conference on Recreational Mathematics and Its History*, pages 73–90. Cambridge University Press, 1994.
- [5] Duncan Stuart. Polyhedral and mosaic transformations. *North Carolina State University, College of Design Student Publications*, 12(1), 1963.
https://d.lib.ncsu.edu/collections/catalog/ua110_200-003-cn0027-v12n1.
- [6] H. F. Verheyen. The complete set of Jitterbug transformations and the analysis of their motion. *Computers Math. Applic.*, 17(1–3):203–250, 1989.