

# Polygonal Tiles from Hitomezashi Patterns and Heat Equations

Bjorn Vermeersch

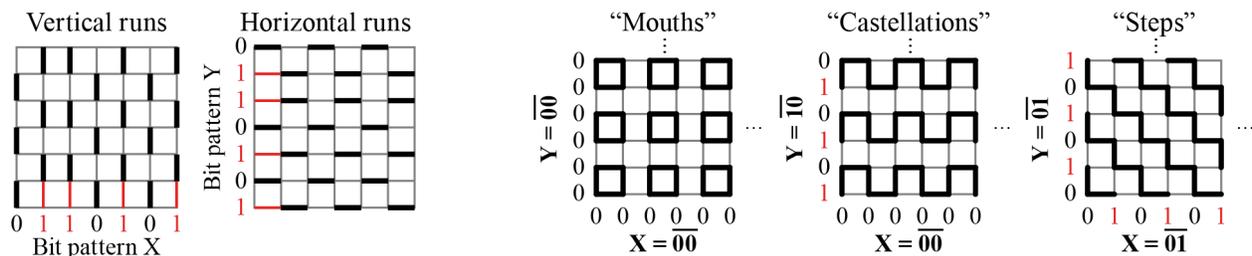
Heverlee (Leuven), Belgium; vermeersch.publications@gmail.com

## Abstract

This paper presents a method to transform procedural sashiko embroidery into intricately coloured polygonal tiles. Two-coloured hitomezashi stitch patterns are used as time-periodic inputs to the heat diffusion equation. Rendering the resulting temperature field with a photograph as colour wheel produces decorative rectangular tiles. Conformal Schwarz-Christoffel mappings extend the recipe to both regular and irregular polygons of arbitrary vertex count. A range of tile designs and examples of an Archimedean and Cairo tessellation thereof are showcased.

## Introduction: Hitomezashi Stitch Patterns

This work was inspired by a Numberphile video [2] in which Aylean MacDonald creates procedural line drawings reminiscent of *hitomezashi* patterns. We will consider sashiko stitches of unit length aligned on a uniform square grid. A binary code on the  $x|y$  axis governs for each column|row whether the first stitch in the vertical|horizontal run touches the axis ('0' bit) or is offset by 1 unit ('1' bit), as depicted in Figure 1.



**Figure 1:** Procedural generation of hitomezashi patterns (left). All 16 combinations of two-bit periodic codes produce possibly offset/reflected versions of three prototypical designs (right).

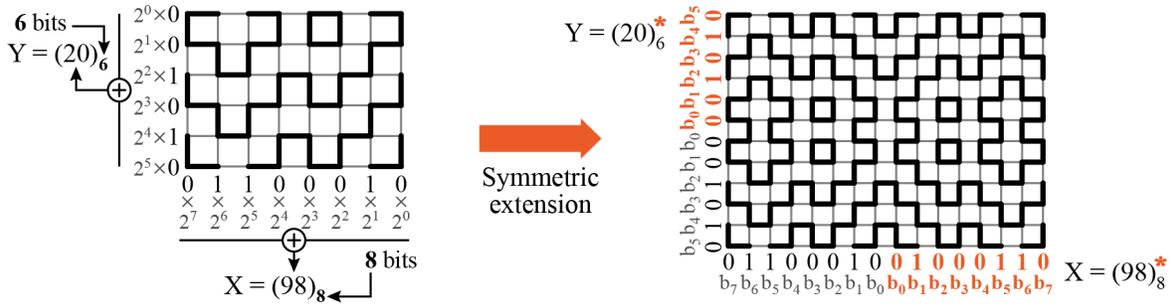
Codes built from repeating two-bit numbers  $\overline{b_1 b_2} \equiv b_1 b_2 b_1 b_2 b_1 b_2 \dots$  induce isolated squares and turret-like or diagonal zigzag lines (Figure 1, right) but irregular codes can quickly produce more intricate patterns (Figure 2). Designs can be uniquely labelled by a pair of identifiers  $\{X, Y\}$  both of the form  $(d)_n^*$ , where the binary stitching code for the corresponding axis is the  $n$ -digit representation of integer  $d$  in base 2 (Figure 2a). The optional superscript  $*$  signals that the binary code for  $x|y$  is to be appended by itself in reverse bit order so as to induce output patterns with horizontal|vertical mirror symmetry (Figure 2b).

Hitomezashi patterns are well known to the math art community [3][5][6][7] and have plenty of decorative appeal in their own right. Here, I use them as inputs for further processing and artistic manipulation.

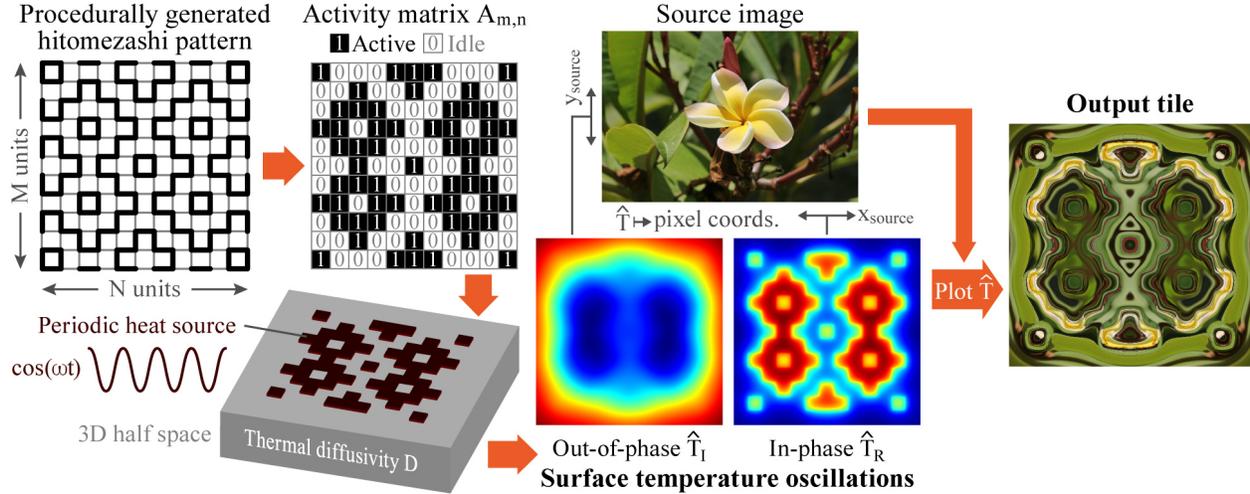
## Generation of Rectangular Decorative Tiles from Stitch Patterns and Heat Equations

The overall workflow, illustrated schematically in Figure 3, is built from the following four steps.

**1. Pseudo-random generation of hitomezashi pattern** We draw binary sequences of length  $N + 1$  and  $M + 1$  for the  $x$  and  $y$  axes respectively. Enforcing correlation between successive bits, i.e. bit flips  $b_i = 1 - b_{i-1}$  ( $i \geq 2$ ) occur with a specific probability  $0 \leq p_{\text{flip}} \leq 1$ , can nudge the output towards aesthetically favourable directions. While highly subjective, the procedure with suitable constraints on  $M, N$  and  $p_{\text{flip}}$  (discussed later) tends to yield a visually pleasing pattern and output tile for one out of every 3 or 4 random trials.



**Figure 2:** Example of a bit-encoded stitch pattern (left) and its symmetric extension (right).



**Figure 3:** Core workflow to generate rectangular decorative tiles.

**2. Two-colouring to obtain a cell activity matrix** The stitch pattern partitions its rectangular bounding box into distinct zones that can be two-coloured, turning it into an  $M \times N$  activity matrix  $[A_{m,n}]$  of cells that are either ‘active’ (black = 1) or ‘idle’ (white = 0). Throughout this work I always assign the bottom left cell to be active, i.e.  $A_{M,1} = 1$  serves as ‘seed’ for the colouring. This choice, combined with symmetrically extended stitch patterns, can offer artistic advantage as it ensures a splash of colour or other visual feature near the corners of the output tile. That said, dual colouring ( $A_{M,1} = 0$ ) could certainly be used as well.

**3. Computing surface temperature  $\hat{T}$**  The cell activity matrix can now be interpreted as  $N \times M$  “electronic circuits” atop a semi-infinite “semiconductor substrate” with thermal diffusivity  $D$ , for which the surface temperature distribution can be computed from the heat diffusion equation  $D\nabla^2 T(\vec{r}, t) - \partial T(\vec{r}, t)/\partial t = 0$ . This seemingly haphazard leap of imagination is rooted in my day job as a thermal modelling engineer at a nano-electronics research facility but should be quite suited to mathematical art, since diffusion equations produce well-behaved unique solutions and tend to soften hard features in their inputs much like a blurring filter does.

For reasons that will become fully clear during step 4 (rendering of the output tile), we assume that the circuit power oscillates sinusoidally in time at angular frequency  $\omega$ . In practice, one does not need to decide on numerical values for  $\omega$  or  $D$  individually. Rather, both parameters combine into a single and more intuitive *diffusion length*  $\lambda$  that can be used to control the artwork appearance, as will be discussed shortly.

Switching to complex phasor representation  $T(\vec{r}, t) \leftrightarrow \text{Re} [\hat{T}(\vec{r}) \exp(i\omega t)]$  turns the partial differential equation for  $T$  into an ordinary one for  $\hat{T}$  that can be solved analytically. In principle, one could forgo the 3D half space and instead opt to solve the 2D heat equation directly on the square or rectangular domain comprising the cell activity pattern. However, this 2D configuration is computationally costly and thus less favourable for generating artwork. The boundary conditions on the rectangle perimeter induce infinite

summations over image heat sources that may converge rather slowly. In addition, the fundamental 2D solution for  $\hat{T}$  involves Bessel functions  $K_0(a\|\vec{r}\|)$  while the 3D case admits the simpler kernel  $\exp(-a\|\vec{r}\|)/\|\vec{r}\|$ .

For 3D half space no image sources are needed, and it suffices to derive the fundamental response  $\hat{G}(x, y)$  to a single square cell  $\text{rect}(x)\text{rect}(y)\delta(z)$  centred at the origin. The desired solution  $\hat{T}(x, y, z = 0)$  induced by the activity pattern  $A_{m,n}$  then immediately follows by superimposing shifted copies of  $\hat{G}(x, y)$ . A fast approximate solution for  $\hat{G}$  can be obtained by evenly subdividing the unit square heat source into  $K \times K$  smaller squares centred around  $(x_k, y_k)$  which are then each treated as infinitesimal point sources, yielding

$$\hat{G}(x, y) \propto \sum_{k=1}^{K^2} \frac{1}{\sqrt{(x-x_k)^2 + (y-y_k)^2}} \exp\left[-\frac{1+i}{\lambda} \sqrt{(x-x_k)^2 + (y-y_k)^2}\right], \quad \lambda \equiv \sqrt{\frac{2D}{\omega}}. \quad (1)$$

Some prefactors have been omitted without loss of generality because the output artwork is invariant to global scaling of the temperature field. The singular terms at  $(x, y) = (x_k, y_k)$  that attempt to give the contribution of a subsquare to the temperature response in its own centre must be replaced by  $2\pi\lambda \{1 - \exp[-(1+i)/(\sqrt{\pi} K \lambda)]\}/(1+i)$ . Physically, the diffusion length  $\lambda$  in Equation (1) is a metric for how deep thermal oscillations at angular frequency  $\omega$  can penetrate into a material with diffusivity  $D$ . Artistically, it can be wielded as an independent parameter for tuning the artwork appearance, as will be illustrated in a later Figure.  $\hat{G}$  is evaluated on the centres of a uniform square grid with unit size  $1/K$  covering the region  $|x| \leq N - 1/2, |y| \leq M - 1/2$ . Superimposing shifted copies then yields  $\hat{T}(z = 0)$  at a resolution of  $K$  pixels per stitch, producing a rectangular output tile that is  $K \times N$  pixels wide by  $K \times M$  pixels tall.

**4. Rendering  $\hat{\mathbf{T}}$  with a source image as ‘colour wheel’** The complex field  $\hat{T} \equiv \hat{T}_R + i\hat{T}_I$  is a smooth function that bundles the in-phase and out-of-phase components of the periodic temperature oscillations. Taking a page from Frank Farris’s delightful playbook [1], we can use a source photograph (measuring  $W_{\text{source}}$  pixels wide by  $H_{\text{source}}$  pixels tall) to serve as a ‘colour wheel’ with which to render  $\hat{T}$ . A variety of schemes could be conceived to map field values to source pixel coordinates; I opted for simple linear transformations of the real and imaginary parts to respectively pixel columns and rows in the source image, i.e.

$$x_{\text{source}} = \text{ceil}\left(\frac{\hat{T}_R - \hat{T}_R^{\min}}{\hat{T}_R^{\max} - \hat{T}_R^{\min}} \cdot W_{\text{source}}\right), \quad y_{\text{source}} = \text{ceil}\left(\frac{\hat{T}_I - \hat{T}_I^{\min}}{\hat{T}_I^{\max} - \hat{T}_I^{\min}} \cdot H_{\text{source}}\right). \quad (2)$$

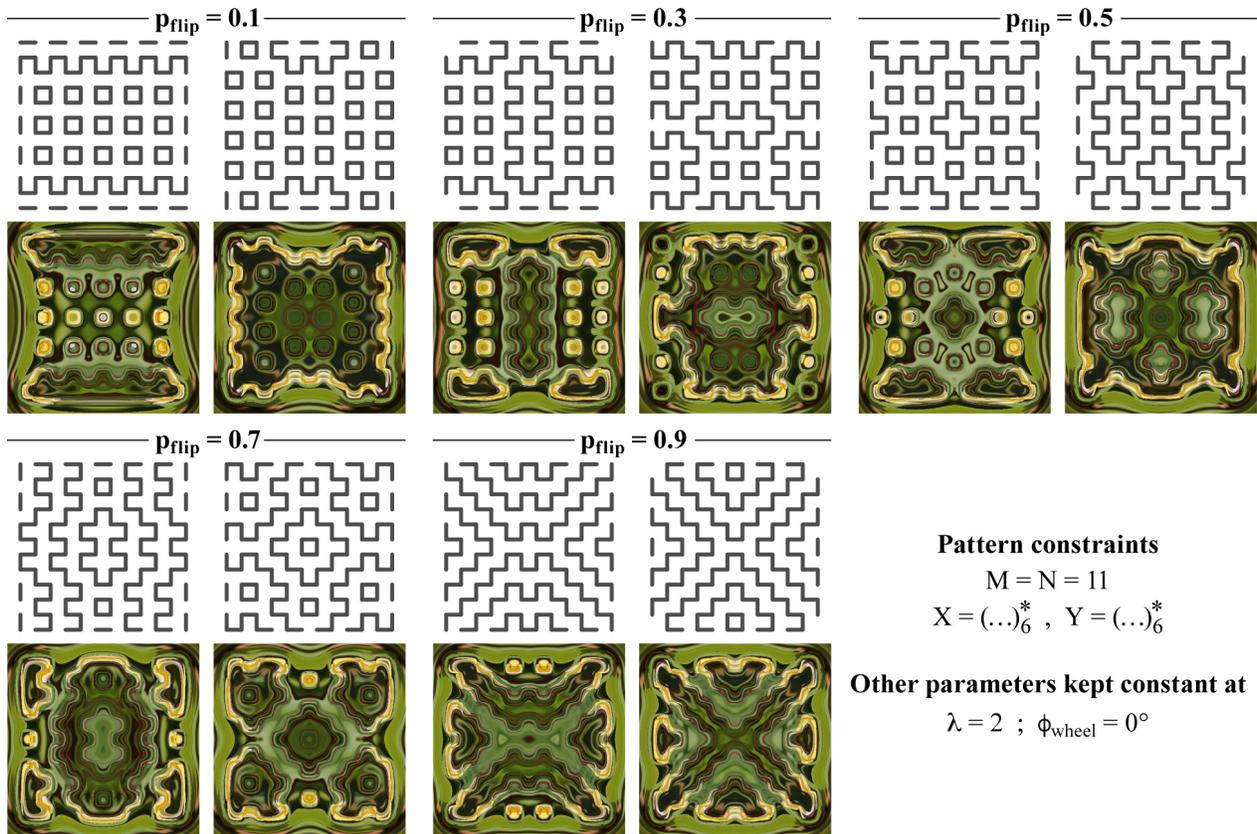
Swapping the min and max bounds of  $\hat{T}_R$  and/or  $\hat{T}_I$  in Equation (2) provides a total of 4 distinct rendering options that correspond to flipping the source image horizontally and/or vertically. Tile appearance can be additionally manipulated by an optional phase rotation  $\hat{T} \leftarrow \hat{T} \exp(i\phi_{\text{wheel}})$  before source image mapping.

### Using the Workflow Parameters to Tune Tile Appearance

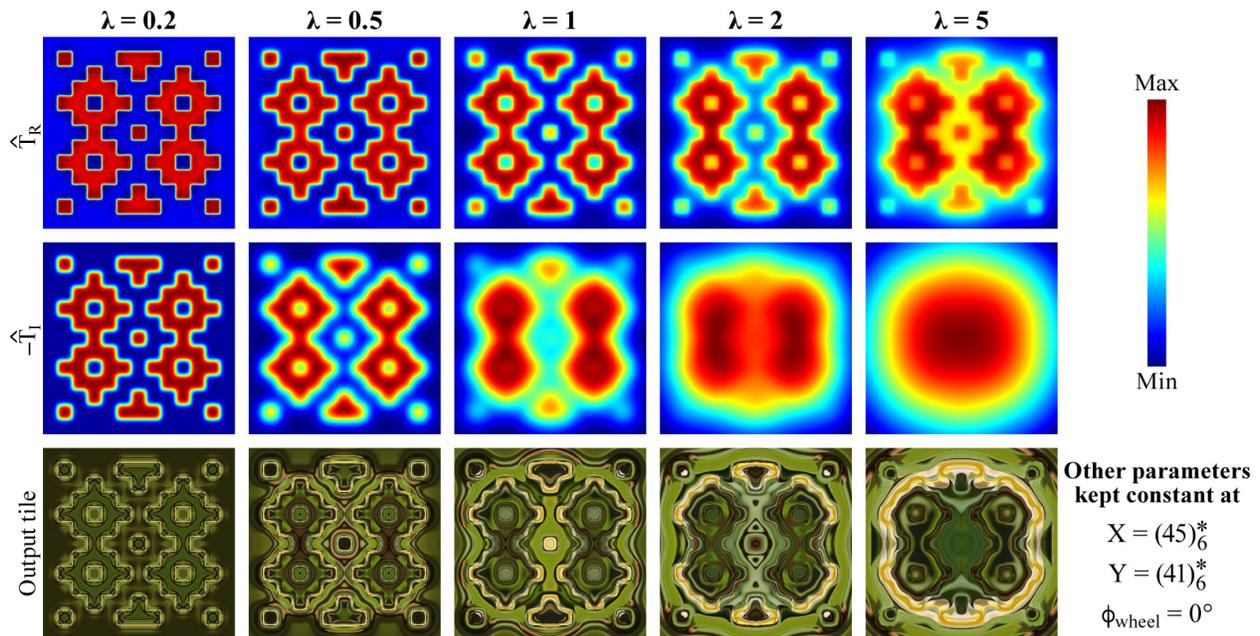
The presented method contains several parameters that can be adjusted to aesthetic advantage. It is instructive to discuss and visually inspect the impact that each of these knobs has on the tile appearance. Obvious key contributors, besides the source image being used, are the size and bit sequences of the stitch pattern.

**Activity matrix size  $M \times N$**  Very small patterns induce somewhat dull and simplistic tiles, while very large patterns can lead to overcrowded designs that lack overall coherence. I found happy middle ground in the ranges  $9 \leq M, N \leq 17$ . Symmetrically extended  $x$  and  $y$  bit sequences (indicated with superscript \* in the ID codes) tend to be aesthetically preferable as they create tiles with twofold reflectional symmetry.

**Bit flip probability  $p_{\text{flip}}$**  Binary sequences generated with either very low ( $p_{\text{flip}} \leq 0.1$ ) or very high ( $p_{\text{flip}} \geq 0.9$ ) bit flip probabilities induce stitch patterns that strongly evoke the simple dotted or diagonally striped appearances of the respective two-bit periodic limits from Figure 1, which may not be desirable (Figure 4). I found  $p_{\text{flip}} = 0.7$  to provide a pleasant blend of global coherence and local complexity.



**Figure 4:** Impact of bit flip probability  $p_{\text{flip}}$  on resulting stitch patterns and tile appearance.

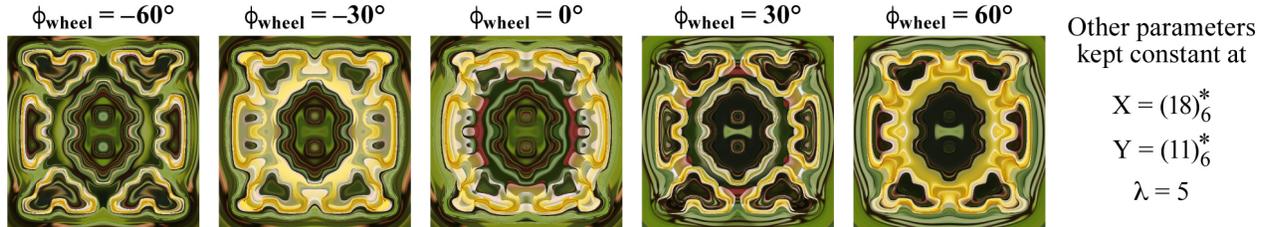


**Figure 5:** Impact of diffusion length parameter  $\lambda$  on an example thermal field and resulting tile appearance.

**Diffusion length  $\lambda$**  Because  $\hat{G}$  is exponentially damped, relatively small diffusion lengths ( $\lambda \leq 0.5$ ) thermally decouple active cells from one another (Figure 5, left two columns). This in turn induces tiles

with sharply delineated regions of quasi-uniform colour. Longer diffusion lengths ( $\lambda \geq 2$ ) yield more gentle temperature transitions and bring partial thermal coupling between neighbouring cells into play, giving a more balanced tile appearance with much more intricate detail (Figure 5, right two columns).

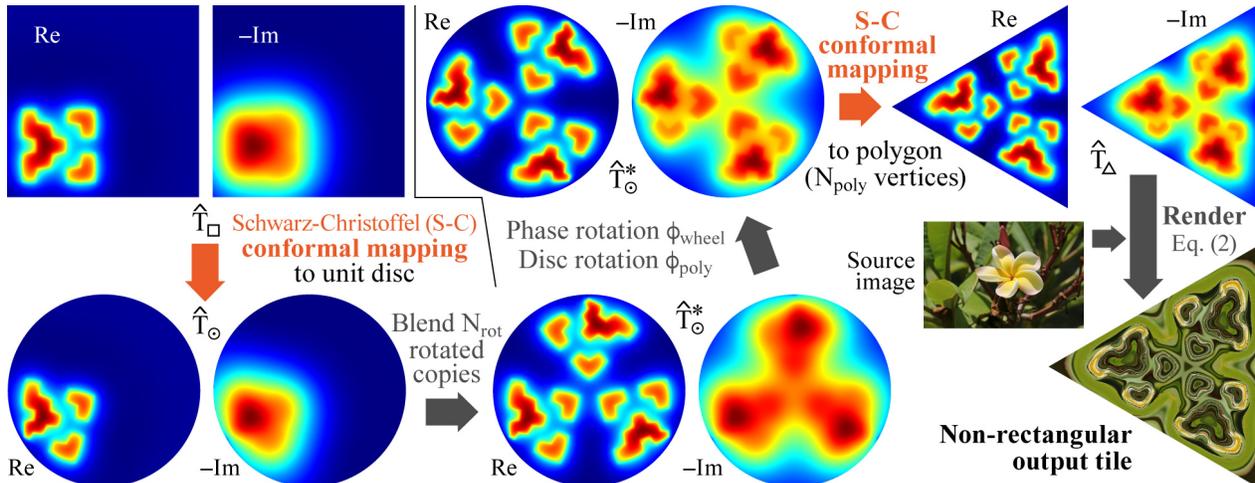
**Phase rotation  $\phi_{\text{wheel}}$**  This parameter does not alter the geometric pattern as such, but strongly affects the tile colourisation and can thus be used to help accentuate certain regions or edge features (Figure 6). The impact co-depends on diffusion length. At typical settings  $\lambda \geq 2$ ,  $|\hat{T}_I|$  tends to be smaller in both magnitude and range compared to  $|\hat{T}_R|$ , and consequently  $\hat{T}_I$  displays a more pronounced sensitivity to phase rotation.



**Figure 6:** Impact of phase rotation parameter  $\phi_{\text{wheel}}$  on the appearance of an example tile.

### Extension to Non-Rectangular Tiles via Schwarz-Christoffel Mappings

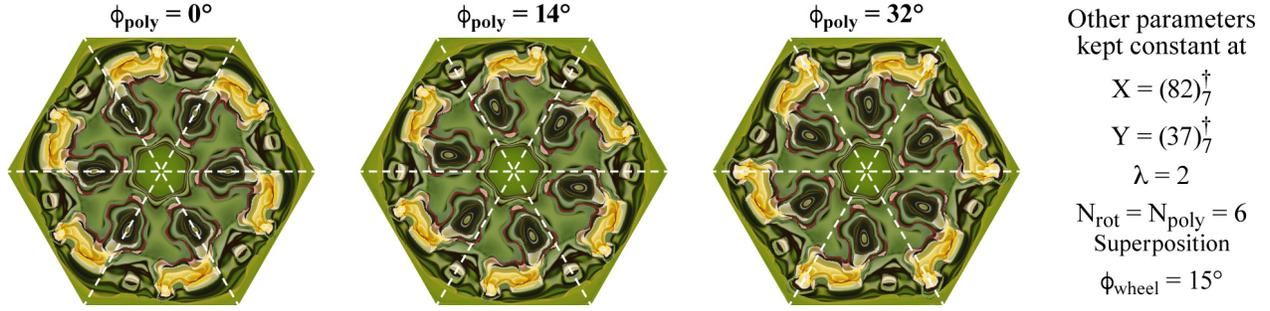
The recipe from Figure 3 can only produce square ( $M = N$ ) and rectangular ( $M \neq N$ ) tiles. However, polygonal tiles with any number  $N_{\text{poly}} \geq 3$  of vertices can be obtained through two consecutive conformal transformations (Figure 7). I carried out the mappings with the Schwarz-Christoffel Toolbox in Octave [4].



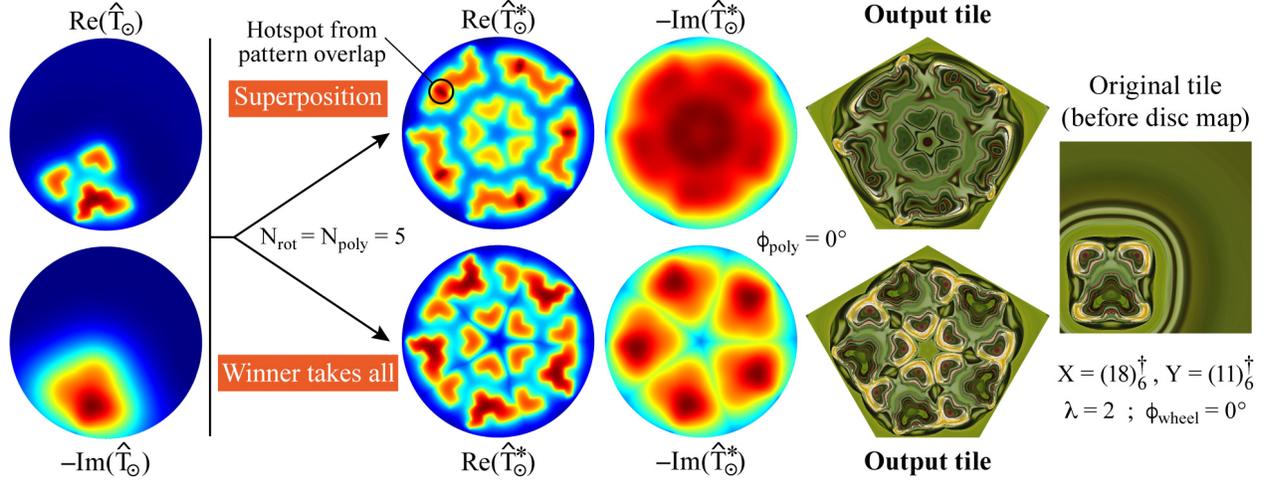
**Figure 7:** Workflow to generate non-rectangular polygonal tiles through conformal mappings.

First, the field  $\hat{T}_{\square}$  computed for a square tile is mapped conformally to its image field  $\hat{T}_{\circ}$  on the unit disc. Combining rotated copies of  $\hat{T}_{\circ}$  (discussed in more detail below) produces a new field  $\hat{T}_{\circ}^*$  on the disc with  $N_{\text{rot}}$ -fold rotational symmetry, which in turn is mapped conformally to the output field  $\hat{T}_{\Delta}$  on the target polygon. Optional pre-rotation of  $\hat{T}_{\circ}^*$  by a judiciously chosen angle  $\phi_{\text{poly}}$  can help align prominent features in the artwork with the edge midpoints or vertices of the non-rectangular output tile (Figure 8).

The rotational symmetry extension  $\hat{T}_{\circ} \rightarrow \hat{T}_{\circ}^*$  can in principle be achieved through domain colouring of complex plane maps  $z_{\circ} \mapsto z_{\circ}^{N_{\text{rot}}}$ , but these severely distort the input patterns. I instead opted for two alternate methods with precautions to avoid excessive self-overlaps of the input pattern. Confining the main features of  $\hat{T}_{\square}$  to a corner of the square tile by overriding 3 quadrants of the cell activity matrix to 0 (indicated by a † in the ID codes) goes a long way. Even so, some remaining subtleties need to be taken into consideration.



**Figure 8:** Example impact of disc prerotation parameter  $\phi_{\text{poly}}$  on a regular hexagonal tile.



**Figure 9:** Example impact of the strategy with which rotated copies of  $\hat{T}_\odot$  are blended together into  $\hat{T}_\odot^*$ .

Additive *superposition* (Figure 9, upper pathway) creates hotspots where rotated copies of the input pattern do happen to overlap. The accumulation of temperature tails tends to affect  $\text{Im}(\hat{T}_\odot^*)$  more prominently due to its narrower range. The resulting alterations of the thermal fields can cause the appearance of the output design to deviate quite strongly from the originating square tile, which may be undesirable.

A *winner takes all* strategy (Figure 9, lower pathway) provides an alternative approach where the real and imaginary parts of  $\hat{T}_\odot^*$  are constructed separately: for each pixel, we maintain whichever of the rotated  $\hat{T}_\odot$  copies gives the largest absolute value for the respective field component. This preserves a much closer resemblance to the patterns of the square tile, but can come at the cost of potentially jarring discontinuities.

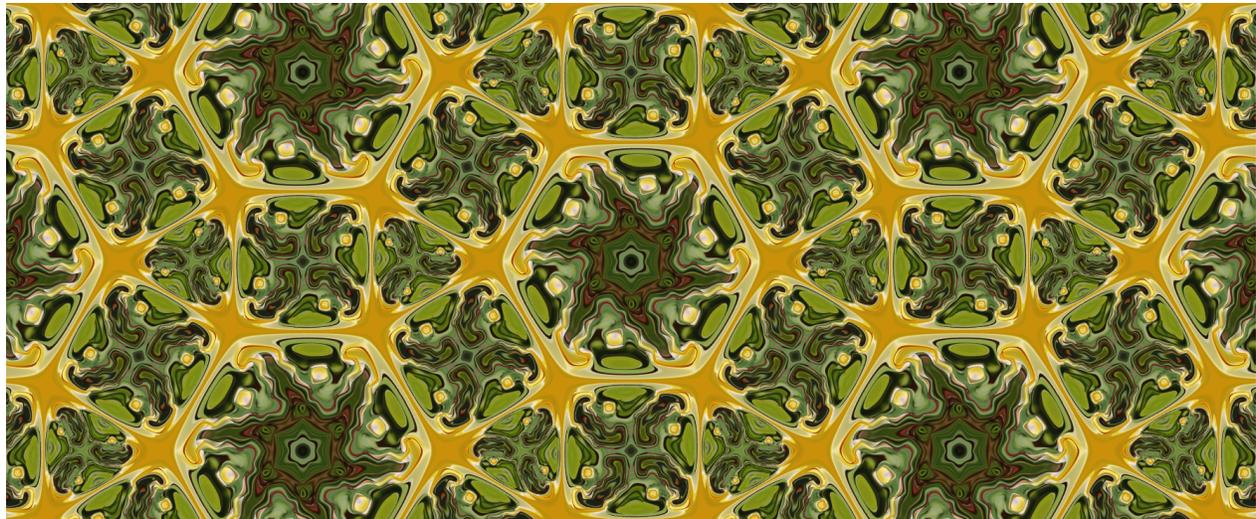
Speaking of discontinuities: we can ensure uniform colourisation around the polygon tile edges through a window tapering function  $f_w$  on the unit disc for which I adopted the form (with  $r_{\text{window}}$  a tuneable parameter):

$$f_w(r_\odot \leq r_{\text{window}}) = 1 \quad \text{and} \quad f_w(r_\odot > r_{\text{window}}) = 3u^2 - 2u^3 \quad \text{with} \quad u = (1 - r_\odot)/(1 - r_{\text{window}}). \quad (3)$$

For standalone tiles, we can simply apply  $f_w$  directly to the disc field  $\hat{T}_\odot^*$ . For tessellations, where a common edge colour must be enforced among distinct tile designs, I instead interpolate the source image coordinates into the colour wheel towards a target source pixel  $\vec{r}_{\text{edge}} = (x_{\text{edge}}, y_{\text{edge}})$  for points near the polygon edges:

$$\vec{r}'_{\text{source}} = \text{ceil}(f_w(r_\odot) \vec{r}_{\text{source}} + [1 - f_w(r_\odot)] \vec{r}_{\text{edge}}) \quad \text{with e.g. } \vec{r}_{\text{edge}} = (W_{\text{source}}/2, H_{\text{source}}/2). \quad (4)$$

Figure 10 shows an Archimedean tessellation of a square, triangular and hexagonal tile constructed in this fashion. Note however that the workflow from Figure 7 is by no means restricted to regular output polygons. Figure 11 presents non-convex flower-shaped polygonal tiles (*florygons*) comprising 90 vertices. Finally, an example of a Cairo tiling with 4 differently coloured irregular pentagons is shown in Figure 12.



Source image



Frangipani • Arona, Tenerife  
Photograph by Author  
(September 2024)

**Tiles**

$$X = (26)_6^\dagger$$

$$Y = (21)_6^\dagger$$

$$\lambda = 2$$

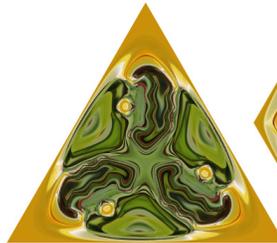
$$N_{\text{rot}} = N_{\text{poly}}$$

(Superposition)

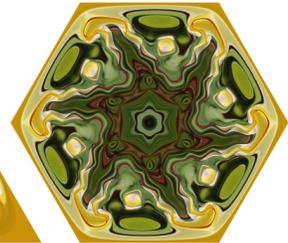
$$\Gamma_{\text{window}} = 0.8$$



$$\phi_{\text{wheel}} = 7^\circ; \phi_{\text{poly}} = 0^\circ$$



$$\phi_{\text{wheel}} = 9^\circ; \phi_{\text{poly}} = 30^\circ$$

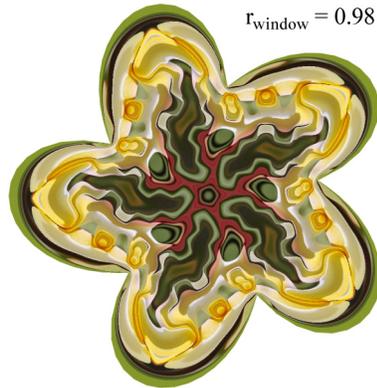


$$\phi_{\text{wheel}} = 1^\circ; \phi_{\text{poly}} = 30^\circ$$

**Figure 10:** Rhombitrihexagonal tessellation of tiles created by the workflows in Figures 3 and 7.

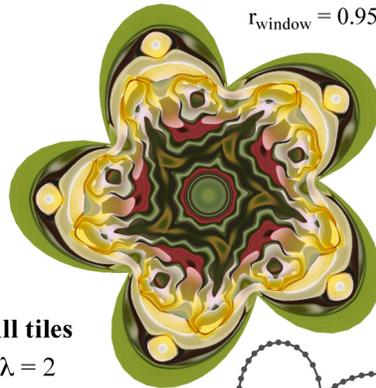
$$X = (53)_7^\dagger, Y = (34)_7^\dagger; \phi_{\text{wheel}} = 9^\circ$$

$$\Gamma_{\text{window}} = 0.98$$



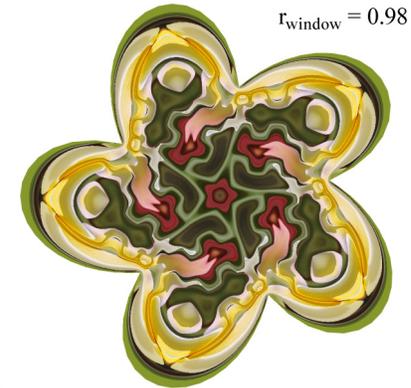
$$X = (82)_7^\dagger, Y = (10)_7^\dagger; \phi_{\text{wheel}} = 9^\circ$$

$$\Gamma_{\text{window}} = 0.95$$



$$X = (53)_7^\dagger, Y = (112)_7^\dagger; \phi_{\text{wheel}} = 10^\circ$$

$$\Gamma_{\text{window}} = 0.98$$



**All tiles**

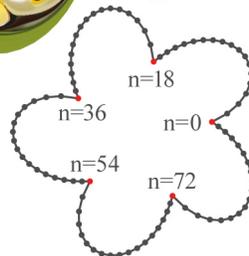
$$\lambda = 2$$

$$N_{\text{poly}} = 90$$

$$N_{\text{rot}} = 5$$

Superposition

$$\phi_{\text{poly}} = 10^\circ$$



**Polygon vertices**  $0 \leq n \leq 89$

$$x_n = f(u_n) \cos[(n+2)\pi/45] + 0.06$$

$$y_n = f(u_n) \sin[(n+2)\pi/45]$$

$$u_n = (4n \bmod 72) \pi / 180$$

$$f(u) = 0.88u^3 - 2.73u^2 + 2.04u + 0.56$$

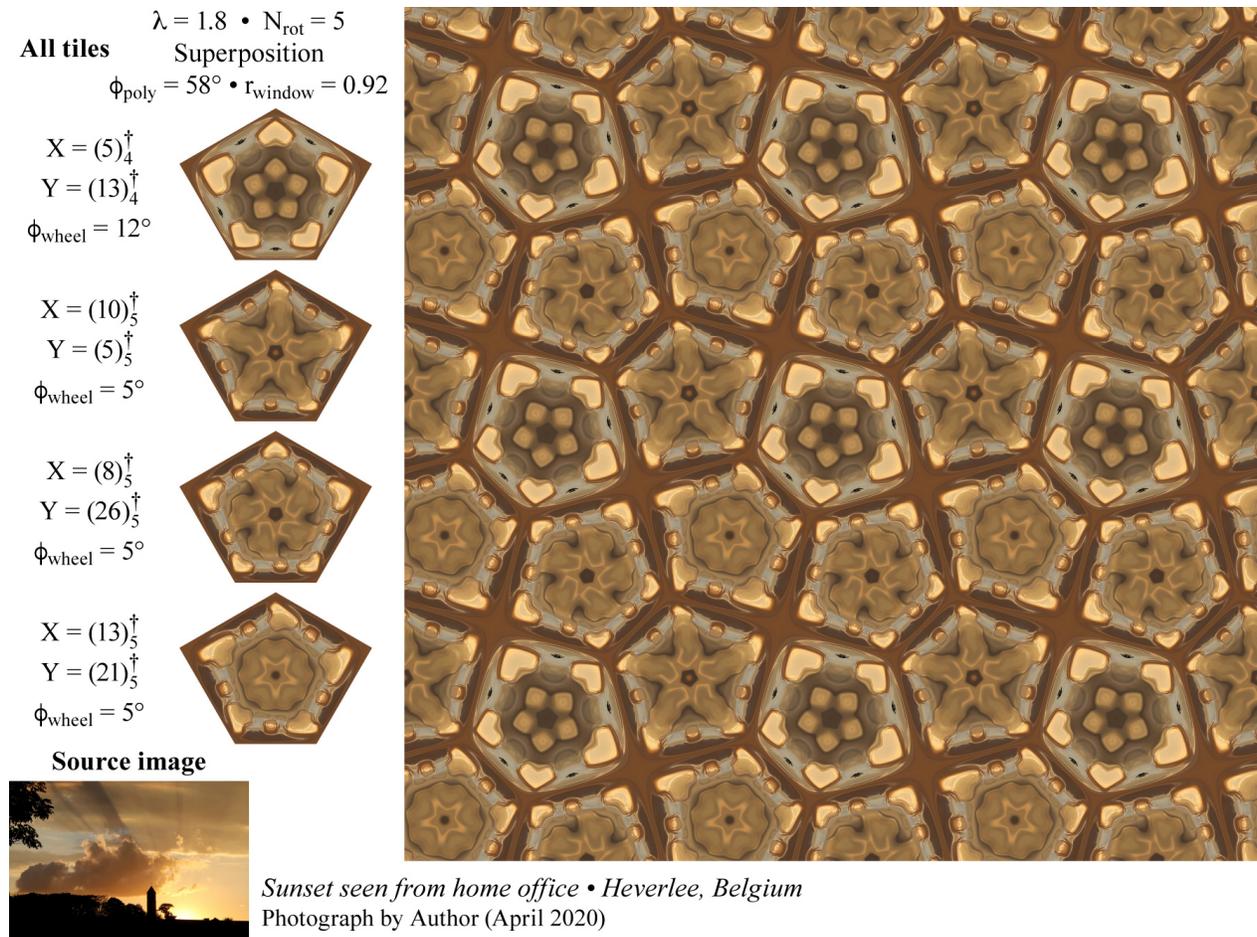
Source image



**Figure 11:** Deconstructed frangipani flower: Decorative non-convex polygonal tiles with 90 vertices.

**Acknowledgements • Disclaimer for Employer Purposes**

I warmly thank Prof. Frank Farris for motivating discussions and insightful feedback. The presented research and manuscript preparation were carried out in the author's personal free time on a privately owned computer.



**Figure 12:** Cairo tiling built from 4 differently coloured irregular pentagons.

## References

- [1] F. A. Farris. *Creating Symmetry: The Artful Mathematics of Wallpaper Patterns*. Princeton University Press, 2015.
- [2] B. Haran. *Numberphile: Hitomezashi Stitch Patterns*. 2021.  
<https://www.youtube.com/watch?v=JbfhziMk2eY>
- [3] C. Hayes and K. A. Seaton. “A Two-Dimensional Introduction to Sashiko.” *Bridges Conference Proceedings*, Online, Aug.1–5, 2020, pp. 517–524. <http://www.archive.bridgesmathart.org/2020/bridges2020-517.html>
- [4] J. Malkoun. *Octave port for Toby Driscoll’s Schwarz-Christoffel Toolbox in Matlab*. 2018.  
[https://github.com/joemalk/sc\\_toolbox\\_octave](https://github.com/joemalk/sc_toolbox_octave)
- [5] D. A. Reimann. “Connections between Hitomezashi Patterns and Truchet Tiling.” *Bridges Conference Proceedings*, Halifax, Nova Scotia, Jul. 27–31, 2023, pp. 457–460.  
<http://archive.bridgesmathart.org/2023/bridges2023-457.html>
- [6] A. Sen and M. Martinez. “Attainable Symmetry in Generalized Hitomezashi Patterns.” *Bridges Conference Proceedings*, Richmond, Virginia USA, Aug. 1–5, 2024, pp. 227–234.  
<http://archive.bridgesmathart.org/2024/bridges2024-227.html>
- [7] D. J. Wildstrom. “Enclave Depth in Hitomezashi Stitchery.” *Bridges Conference Proceedings*, Richmond, Virginia USA, Aug. 1–5, 2024, pp. 357–360. <http://archive.bridgesmathart.org/2024/bridges2024-357.html>