Line-Based Rendering with Truchet-Like Tiles

Abdalla G. M. Ahmed
abdalla_gafar@hotmail.com

Abstract

We explore the use of Truchet-like tiles (tiles with half area shaded) for single-tone (monochrome) line-based rendering of maps. We borrow concepts from information theory to develop some qualitative and quantitative measures, and use them to discuss four rendering styles which use Truchet-like tiles. We first present a style based on the original Truchet tiles, then we review the tile-based approach used by Inglis and Kaplan for Op Art rendering of 2-color maps, and extend the concept to 3- and 4-color input maps. We present two more concepts capable of rendering maps with up to 6 colors. We highlight some relationships between the four rendering styles, and utilize these relationships to generate Op Art/labyrinth renderings from input maps of up to 6 colors.

CR Categories: I.3.0 [Computer Graphics]: General—;

Keywords: Op Art, non-photorealistic rendering, line art, Truchet tiles, tiling, maze, interactive maze

1 Introduction

A Truchet tile is a square tile diagonally sectioned and one half shaded. See Figure 1(a). It is named after French clergyman Sebastian Truchet (1657–1729), who inspected the large variety of ornamental designs that can be obtained by combining different orientations of such tile in an 8×8 layout. Many references to decorated tiles like Truchet's appeared over time in papers, patents, and games [Krawczyk 2011]. In 1987 C. Smith published a paper that included a translation to English of the original Truchet paper, and he introduced another decorated tile which also came to be known as a Truchet tile (or a Smith-Truchet tile) [Smith and Boucher 1987]. See Figure 1(b). Since the time Smith published his paper, Truchet-like tiles have found their way into many aesthetic and analytic applications. To name a few, Pickover suggested a use in visualizing randomness [Pickover 1989], Gale *et al.* used them to study an automaton known as Langton Ant [Gale et al. 1995], Browne demonstrated their use in creating gaming characters [Browne 2008], and Reimann used them for rendering text [Reimann 2009] and for modular knot design [Reimann 2012].

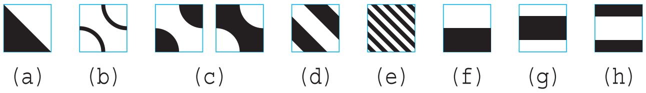


Figure 1: *Examples of Truchet-like tiles: (a) The original Truchet tile, (b) Smith's variant, (c) two shaded Smith variants used by Bosch [Bosch 2011] for image rendering, (d) a modified Smith variant used by Inglis and Kaplan for Op Art rendering, (e) a tile used by a font designer cited by Inglis and Kaplan to create an Op Art typeface, (f) the shadow tile we will use in Section 6, (g) the weave tile we will use in Section 7, and (h) the negative weave tile. Note that the original Truchet tile and the shadow tile have 4 distinct orientations each, whereas all others have only 2 distinct orientations each.*

Recently, the use of Truchet-like tiles was extended to the area of stylized rendering of images and maps. Specifically, Bosch [Bosch 2011] demonstrated how a shaded version (Figure 1(c)) of the Smith variant can be used for rendering an arbitrary grayscale image, and he also applied his own modifications to original Truchet tile to produce multi-tonal tiles [Bosch and Colley 2013] for image rendering. Inglis and Kaplan [Inglis and Kaplan 2013] used a modified Smith variant (arcs replaced by line segments, as in Figure 1(d)), to generate (and animate) an Op Art rendering of an arbitrary monochrome drawing. The implementation is similar in both cases: the color of the pixels in the underlying bitmap decides the shape or orientation of the tiles used to pave these pixels. The concept, however, is quite different: Bosch's approach uses duotone tiles to convey tone information of the underlying image, whereas Inglis and Kaplan's approach uses a single tile, and rely on line bends between differently oriented tiles to convey shape information.

In this paper we explore other possibilities of the Inglis and Kaplan approach to tile-based rendering (pixels paved with decorated tiles). More specifically, we focus on tiles which produce parallel equi-spaced lines for region interiors. We will investigate 4 distinct styles, and show basis for comparing them, highlighting weaknesses and constraints of each style. The paper is organized as follows: In Section 2 we review previous related work, then in Section 3 we briefly outline important aspects of tile-based rendering. We then go through four possible tile designs and discuss the associated rendering styles. We start with the very original Truchet tile in Section 4. In Section 5 we extend the Inglis and Kaplan tile to work with 3- and 4-color input maps. In Section 6 and Section 7 we introduce two new tiles and their associated rendering styles: shadow and weave renderings. In Section 8 we highlight relationships between the 3 styles: weave rendering, Op Art, and shadow rendering, and we use these relationships to gain better insight of Op Art, and to introduce more extensions and improvements. We stop in Section 9 to give a brief idea about how tile-based shape rendering can be adapted for line-based halftoning. Finally we present a conclusion in Section 10, and give suggestions for future work.

2 Related Work

This work in inspired by earlier works by Inglis and Kaplan [Inglis and Kaplan 2011; Inglis et al. 2012], who described a simple (but non-modular) algorithm for producing line based Op Art like the one in Figure 2(d), taking an arbitrary 2-colour drawing as input. Their algorithm instructs to rasterize the input map onto a square grid, fill its region interiors with lines, then draw alternate boundary edges. These steps were shown to guarantee artifact-free output, where artifacts were defined as line breaks, T-junctions, and crosses. For 3-color input maps they suggested applying the same steps to a triangular grid. The 3-color algorithm, however, could not guarantee artifact-free output for all input maps, and they concluded that it is impossible to find an artifact-free solution to a general 3-color problem; so they described sophisticated optimization

---

CAe 2014, August 08 – 10, 2014, Vancouver, British Columbia, Canada.

ACM 978-1-4503-3019-0/14/08 $15.00

techniques to minimize artifacts. Inglis and Kaplan then described how to generate Op Art from 4-color input maps by adding diagonal lines in the square grid. Besides not guaranteeing the elimination of artifacts, this approach also makes unequal spaces between lines in the four directions.

Inglis and Kaplan also made a mention of the modular approach for producing Op Art from 2-color maps, citing an earlier use by a font designer, who used a tile similar to that in Figure 1(e), oriented in one direction or another, depending on the color of replaced pixels. Earlier, Inglis and Kaplan considered this approach disadvantageous for its tendency to produce "block-like" output, but they resorted to it in a later work for animating line-based Op Art [Inglis and Kaplan 2013]; after refining the tile to use a minimal number of lines. See Figure 1(d).

3 Some Considerations in Tile-Based Rendering

Figure 2(a) shows a simple monochrome drawing of a house, rendered using solid pixels (natural way of rendering). In Figure 2(b) pixels are replaced (paved) by Smith's variant of Truchet tile (see Figure 1(b)), oriented according to pixel color, and in Figure 2(c) the Inglis-Kaplan adaptation of Smith's variant (see Figure 1(d)) is used instead. Even though rendering with the original Smith tile manages to show what is there, it is a bit obscure compared to rendering with the Inglis-Kaplan tile. This poses a question about what makes a tile-based rendering looks "good"? To answer this questions we borrow some concepts from information and communications theory, where information is valued in one approach by the "surprise" it adds, and in another approach by the complexity it takes to algorithmically describe it [Salomon 2004]. The original plain rendering in Figure 2(a) ideally shows how much information is there in the rasterized drawing[1]. It also reflects where information is concentrated: we do not pay much attention to the monotonous runs of same-color pixels inside regions of the image; we rather focus more on the sharp transition of color on crossing edges between regions. Therefore edges bear most of the image information. To see this, remember that such drawing can well be sketched by drawing only edges, leaving regions blank.

Looking at Figure 2(b), we find it a bit more complex than necessary, compared to the original image, to describe what is going on inside regions. Indeed, "wavy lines" is not enough here, as one would ask about details of how wavy it is (waveform, frequency, etc.). By delivering irrelevant information, such detail would actually pose "noise" which distracts viewer's perception of the underlying image. Moving on, we notice that Smith's tile does not perform well, either, in conveying edge information: window details get almost lost, and a viewer's eye might step a few pixels before realizing that it has entered a new region. This time, however, the weak performance is due to conveying *less* information than necessary. It is worth noting here that Smith's Truchet tile was specifically designed to tile seamlessly regardless of orientation [Smith and Boucher 1987]. In contrary, the Inglis-Kaplan tile is specifically designed to capture changes in orientation.

For a tile-based rendering to match the distribution of information in an image, then, it should

1. convey a simple and uniform message inside regions with uniform color, and
2. bear a distinct and consistent message on the edges between regions,

Where the word "message" as we use it here means a modular piece of information. Let us see how this applies to the tile-based Op Art rendering in Figure 2(c). "A set of parallel alternating black and white lines" is all that it takes to describe what is rendered inside same-color regions; which is quite simple a message. We should also describe the orientation of lines; but this information is not redundant at all: each direction of lines represents a specific color of the rendered image; which is an advantage of Op Art rendering compared to simple sketching of edges. Moving on to edges, this Op Art rendering relies on sharp corners to convey edge information: Corners align on a horizontal edge to induce an "illusory contour" which is interpreted as a line [Inglis and Kaplan 2011]. The same applies to vertical edges. When we look at the roof, however, we find that instead of a stack of corners we see rows of U-shaped line bends. Op Artist Reginald Neal used stacks of corners to convey edge information in his work "Square of Two", which was the original inspiration of Inglis and Kaplan's research [Inglis and Kaplan 2011], so we take a stack of corners as the primary edge message of Op Art rendering. Rows of U-shaped line bends, then, represent a secondary, perhaps undesirable, edge message. A notable problem with U-shaped edges is that they might be more visually emphasized than stacks of corners (we will give a possible explanation in Section 8.1), and should therefore be avoided in Op Art rendering. Secondary edge messages appear on ±45° edges in renderings with the tile of Figure 1(d), so these angles should be avoided if possible. Similarly, 0° and 90° edges reveal secondary edge messages in the original Inglis and Kaplan's Op Art rendering algorithm, as apparent in Figure 2(d). Note that these angles are relative to the directions of pixels, so one way to avoid ugly edges is to rotate the sampling grid in a way which tries to avoid edges bearing these slopes.

It is worth noting that Op Art rendering with the tile of Figure 1(e) significantly reduces the chance of having secondary edge messages, confining them to corners, so it seems to had been a conscious decision by the font designer to use this tile instead of the simpler one in Figure 1(d). The downside, however, is that the former tile uses 4-times more lines to represent a single pixel. Essentially, our eyes, as well as the devices we render on, have limited resolutions for the number of lines they can see/display per unit area. Thus, for a given area, rendering with the former tile can admit only quarter the resolution of input bitmaps compared to the latter; hence the apparent block-like look due to "pixelation". This aspect of repetition, lines per tile in this case, represents an important thing to consider in a tile-based rendering. Note, however, that lines can be at different angles relative to the tile, and tiles themselves need not be square. It would also be a good idea to be able to compare a tile-based approach (e. g. Figure 2(c)), to a non-modular approach (e. g. Figure 2(d)) which produces similar results. For these considerations we define a measure for line density by comparing the area of a pixel of the input image (be it a square or any otherwise shaped pixel) to the perpendicular distance between parallel lines. To make our measure dimensionless we take the square root of the pixel area. Thus, we have a line density of $\sqrt{2}$ (≈ 1.41) lines per pixel in renderings with the tile of Figure 1(d), $4\sqrt{2}$ (≈ 5.66) in renderings with the tile of Figure 1(e), compared to a density of only 1 line per pixel in the non-modular Inglis and Kaplan Op Art algorithm. The reader is advised to compare density of lines in Figure 2(c, d) to get a feeling of what a 41% increase in line density looks like.

There is still one more important thing to consider: A good tile-based rendering must have an artistic touch. After all, the best conveyance of image information is found in the original input image itself, so why bother rendering it in a different way? Again, information theory might suggest an answer; that in addition to the underlying image information, this kind of stylized rendering aims at

---

[1] As we are using information theoretic concepts, it is worth recalling here that the word "pixel" itself stands for "picture element".

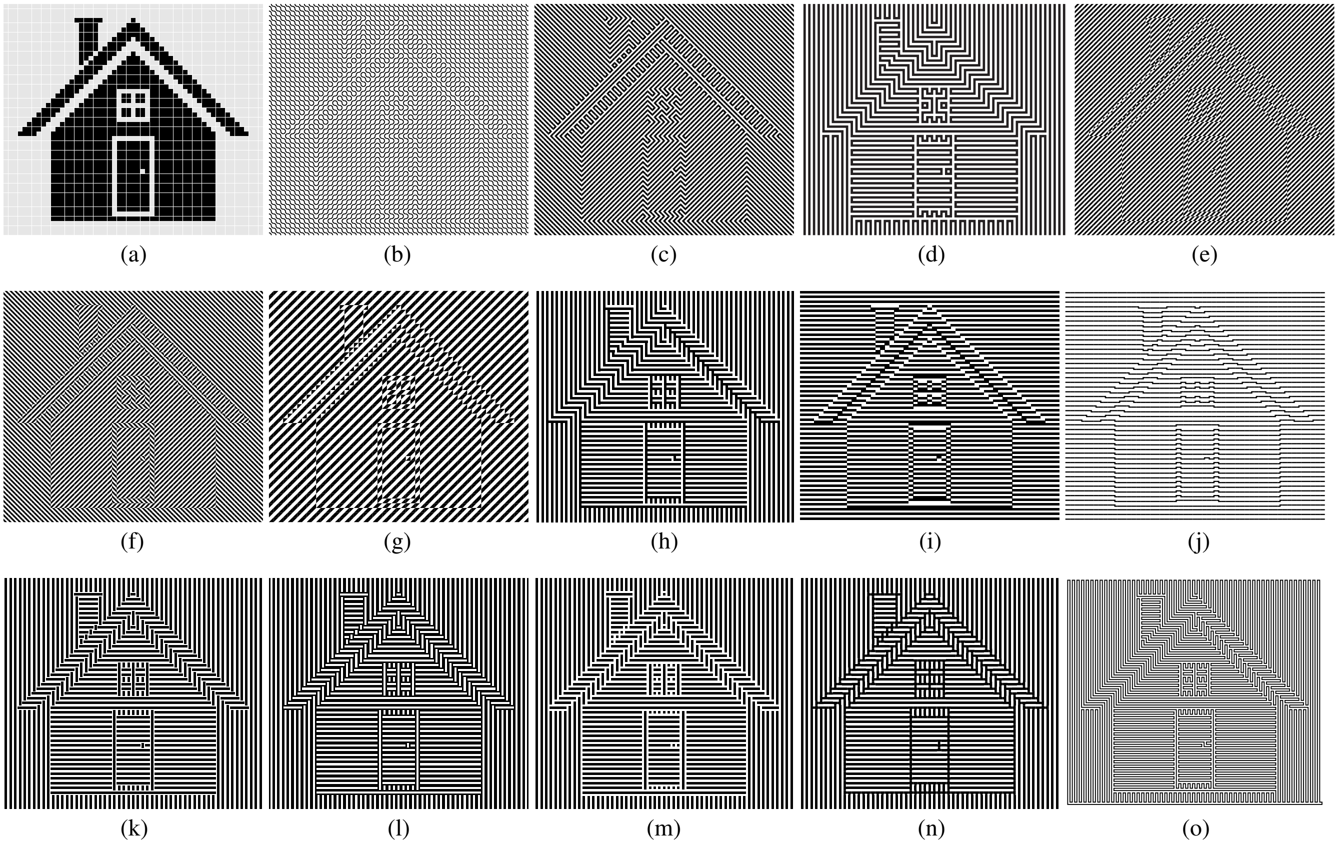


Figure 2: *An illustration of various rendering styles discussed in this paper. (a) A rasterized simple monochrome drawing of a house, rendered using: (b) the Smith variant of Truchet tile (Figure 1(b)), (c) the Inglis-Kaplan adaptation of Smith variant (see Figure 1(d)), (d) Inglis and Kaplan's non-modular Op Art algorithm, (e) modular inversion tiles of Figure 3, (f) the Inglis-Kaplan tile and its negative, (g) position-dependent Truchet tile, (h) shadow tile (Figure 1(f)) and its 90° rotation, (i) shadow tile and its 180° rotation, (j) context-aware (CA) offset tiles, (k) weave tile (Figure 1(g)), (l) negative weave tile (Figure 1(h)), (m) CA white-ink tiles, (n) CA black-ink tiles, and (o) CA outlined shadow tiles.*

conveying other information: the surprise in the non-conventional way of revealing an image on top of an underlying pattern (e. g. parallel lines). The algorithmic description concept of information theory suggests that for the rendering style to be identifiable and memorable, it should have a consistent "signature" that is easy to describe (but not necessarily easy to produce). In the case of Op Art, for example, the artistic touch is found in the scintillating effect of parallel lines, the confusing labyrinth structure, and the surprise of how stacks of corners look like lines. As Inglis and Kaplan identified, artifacts, such as line breaks, crosses, and T-junctions, significantly affect the quality of Op Art. We add that secondary edge messages also deteriorate it.

In the rest of this paper we are going to discuss four tile-based rendering styles which use the same underlying pattern of parallel lines, and vary in the way they present edges. For each style we will discuss how consistent and recognizable the overall style is, the message it conveys on edges, secondary edge messages, if any, and line density of the rendering. How beautiful is the style itself is, of course, the most important, but this is difficult to evaluate analytically.

4 Inversion Rendering

As a tribute to the old work of Truchet we first present a rendering concept based on the very original Truchet tile (Figure 1(a)). Plate 1 of Truchet's work demonstrated how such tiles can be laid to generate parallel diagonal lines [Smith and Boucher 1987, Figure 4(C)]. A 2×2 slice of this layout can be taken as a seamless super-tile. See Figure 3(a). Thus, Truchet tiles can readily provide the required underlying parallel lines pattern, and we just need to find a working edge message. An idea came on noting that color is inverted on crossing the edge between two adjacent similarly oriented Truchet tiles. This represents a simple and clear message that can be used to convey edge information. The parallel lines layout of the block in Figure 3(a) avoids color inversion by maintaining a 180° rotation between adjacent tiles; so 180° rotation is the key for producing the counterpart of the super-tile. See Figure 3(b). We call these super-tiles "inversion tiles". Figure 2(e) shows an example rendering with these tiles. The overall message of this rendering is a set of parallel

Figure 3: *(a) A 2×2 block of Truchet tiles which tile seamlessly to generate parallel diagonal lines. (b) A 2×2 block for paving the other color of inversion rendering, obtained by rotating the one in (a) by 180°.*

diagonal black and white lines which get inverted, like a negative, for black regions of the underlying map. The edge message is essentially inversion of color, and although the message is technically consistent at all angles, it is evidently not visually consistent at an angle of 45°, owing to the ugly standalone triangles. It is notable that edges at -45° do not cause a problem as they do with the Op Art tile discussed in Section 3. A "neat" inversion rendering, one that is consistent in all edges, can be obtained by combining Op Art with inversion rendering, using the Inglis-Kaplan tile (Figure 1(d)) and its 90° rotated negative. See Figure 2(f).

Negative tiles have a line density of $\sqrt{2}$ ($\approx$ 1.41), same as the Inglis-Kaplan tile. Fortunately, it is possible to halve this to $1/\sqrt{2}$ ($\approx$ 0.71), which is the lowest line density we know of so far, by using what we call "position-dependent tiles": Instead of using a 2×2 block of Truchet tiles, we use granular tiles, and set their orientation based on both color and position of the underlying pixels; a tile is 180° rotated for pixels where the sum $(x + y + color)$ is even. See Figure 2(g).

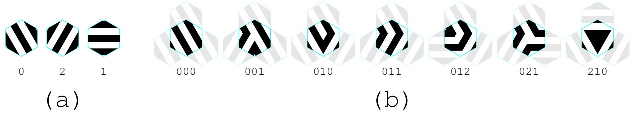
5 Op Art Rendering

Tile-based Op Art rendering of 2-color input maps, studied earlier by Inglis and Kaplan [Inglis and Kaplan 2013], was already discussed in Section 3 of this paper. In this section we would like to extend the concept to render 3- and 4-color input maps using 3 and 4 line directions, respectively.

5.1 3-Way Op Art

In an attempt to solve the standing artifacts problem in 3-color Op Art rendering, inspired by the tile of Figure 1(d) which guaranteed artifact-free output in the 2-color case, we asked a question of whether it is possible to design a similar tile, or a set of tiles, decorated with straight line segments, to be used for generating line-based Op Art guided by 3-color maps. Such a tile needs to have three distinct orientations, so a triangular or a hexagonal tile suggest themselves. Triangular tiles have an odd number of sides, and it is therefore not possible to connect mid-points of sides as in the model square tile. This leaves us with hexagonal tiles, and we are lucky that connecting mid-points of sides produces equi-spaced lines when tiles are tessellated. See Figure 4(a). Rendering a 3-color map with these tiles is simple and straightforward, analogous to the 2-color case; just that the input map is sampled to hexagonal pixels, and that there are 3 colors mapped to the 3 tile orientations.

A problem, however, appears on matching the edges of tiles: although all lines route correctly, a side-line does not miter properly with a middle-line, because the former, being cut at an angle, has twice as long an edge as the latter. One possible solution to this problem is to "weld" line ends before rendering, so that joints are treated as bends in one line. This approach also gives an opportunity to replace a train of middle-line segments with a single long line segment. Details of this "welding" are implementation dependent, and we have verified it using C to generate Postscript. A more modular approach to solving the mitering problem is to use what we call "context-aware" (CA) tiles. The idea is to recenter tiles at vertices between pixels so that every join is contained within a single tile, instead of being split between 2 tiles. See Figure 4(b). A CA tile is drawn and stored for each of the 27 possible combinations of tile orientations around a vertex. Later, on rendering, the appropriate CA tile to be laid on each vertex is selected based on the color of the 3 surrounding pixels. Taking advantage of symmetry reduces the 27 tiles set to only the 7 distinct layouts shown in Figure 4(b).



(a) (b)

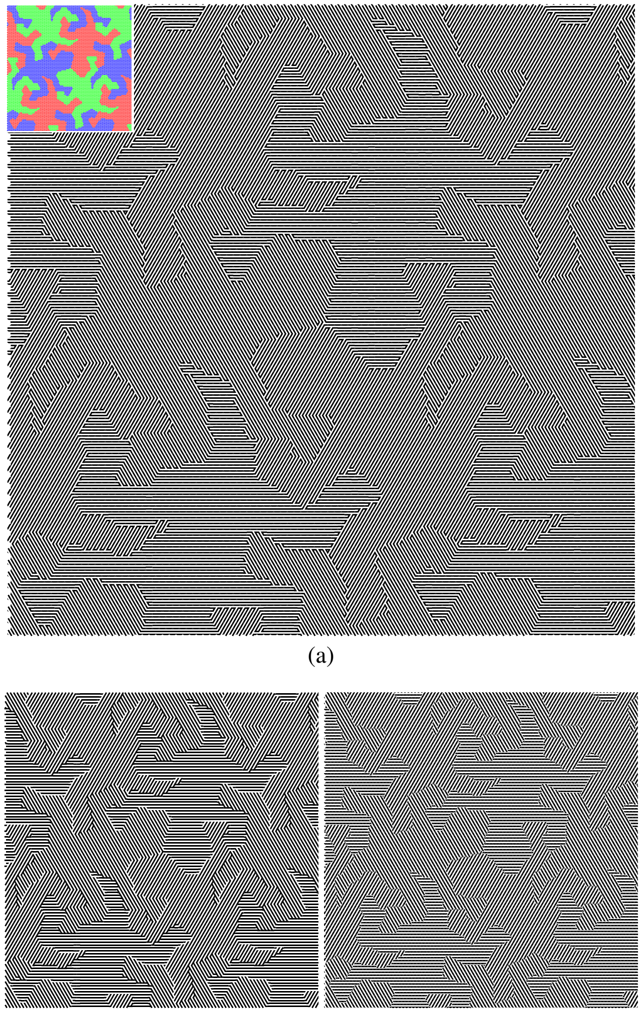
Figure 4: *Hexagonal tiles for 3-way Op Art rendering: (a) The 3 distinct orientations of the raw tile. (b) The 7 distinct layouts of context-aware tiles which capture details from the top, bottom-right, and bottom-left constituent tiles. The digit beneath each CA tile indicate orientations of its constituent tiles.*

3-way Op Art bears the same primary and secondary edge messages as its 2-way counterpart; just that tiles can be aligned in 3-directions instead of 2. A notable difference is that secondary edge messages share the same edge directions as primary messages. Figure 5(a) shows an example 3-way tile-based Op Art rendering. For easier comparison we chose to render Escher lizards similar to Inglis and Kaplan's example. A great advantage of the modular approach is guaranteeing an artifact-free output for an arbitrary input map. That is, it contains no line breaks, T-junctions, or crosses, except possible line breaks on the image border. This is easy to see by noting that all vertices occur at mid-points of tile edges, and each edge (except at the drawing border) is shared between two tiles, each tile has one and only one line segment terminated at the midpoint. So each vertex not on the image border has degree 2. Another way to see this is by looking at the CA tiles of Figure 4(b), which list all the possible vertex configurations that can be found in the drawing. The artifact-free guarantee, however, does not come at no cost: This approach uses a much higher line density: $(64/3)^{1/4}$ ($\approx$ 2.15), compared to only $(1/3)^{1/4}$ ($\approx$ 0.76) in the 3-color Inglis and Kaplan algorithm. Another substantial downside is that having bands of lines in tiles eliminates any chance of having stacks of cute-angled corners, which in turn means less chances of having primary edges.

5.2 4-Way Op Art

After the successful implementation of tile-based artifact-free Op Art from 3-color maps, it was tempting to think of extending the concept to the 4-color case. If a square tile worked for 2-colors and a hexagonal tile worked for 3-colors, then an octagonal tile would be the candidate for 4-color maps. There are, however, two issues with octagons: First, octagons can not tile the plane; they rather leave square gaps. Second, if midpoints of sides of an octagon are connected the same way as for the square and the hexagon, then the resulting lines would not be equi-spaced. Each one of these issues would have been a problem, but after careful thinking and experimenting we found that together they make a solution, as we will explain now in steps.

Let us start with octagonal tiles decorated with lines connecting midpoints, similar to the square and hexagonal tiles mentioned earlier. See Figure 6(a). Octagonal tiles can make a semi-regular Archimedian tiling which includes square gaps [Glassner 1998]. The octagons meet side-to-side at only 4 edges: those facing North, East, South, and West. Between every four tiles there is a square gap, and there is a line coming from each of these tiles and terminated at the midpoint of a side of the square gap. The trick is that we can manually couple pairs of these lines inside the square gap. There are four ends, which makes it always possible to route without crossings, T junctions, or line breaks. Further, manual coupling implies that lines do not have to be terminated at the exact midpoints of sides of the square gaps. This room of freedom can be exploited to solve the spacing problem of the lines: we fix the four end points at midpoints of {N, E, S, W} sides of the octagon, and start to slide the other end points on the {SW, SE, NE, NW} sides, respectively, to make the four lines equi-spaced. See Figure6(b). We are safe as long as the moving ends remain within the respec-



(a)

(b)

(c)

Figure 5: *Escher-style interlocked lizards (inset of (a)) rendered in three modular line-based styles: (a) Op Art rendering, (b) shadow rendering, and (c) weave rendering. Note that the three drawings use the same line width, but the Op Art rendering needed twice as much area as the others due to its doubled line density.*

tive sides. The correct adjustment is obtained at a slope of 3-in-1 (≈ 71.6°). See Figure 6(c). This is going to be our primary tile, and a secondary tile is a reflection of this. Two more tiles are obtained by rotating the primary and secondary tiles 90°. The complete set of tiles is shown in Figure 6(d).

Now that we have 4 tiles with lines in 4 different orientations, our next task is to decide how to route lines in the square gaps between tiles. Every square is surrounded by four tiles, and each tile has four possible orientations, which count to 256 possible configurations. By employing symmetry, the 256 configurations can be reduced to only 39 distinct layouts (not 32 as one might expect, because some configurations map to themselves on applying a symmetry operation). We inspected all these layouts, one by one, and chose a possible routing for lines in each case, as demonstrated in the example of Figure 7. Note that there are always two possible routings in the gaps: {NE–NW, SE–SW} and {NE–SE, NW–SW}. Although it should be possible to develop an algorithm for routing, we see no point in doing this for a finite set of 39 tiles: less than characters of a font; so we defined routes by hand. As illustrated in Figure 8,

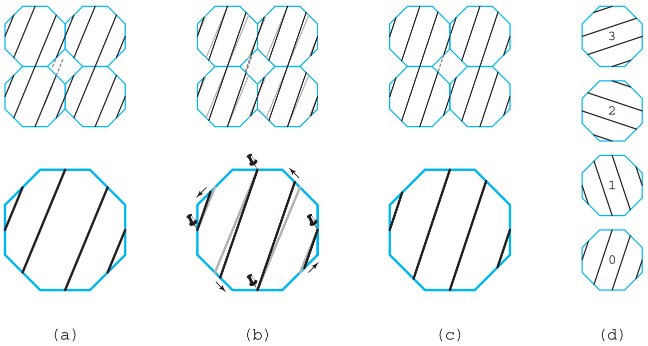


Figure 6: *Developing decorated octagonal tiles to be used for Op Art rendering of 4-color input maps: (a) Lines connecting midpoints of sides, analogous to the 2- and 3- color cases. (b) Fix line ends at midpoints in N, E, S, and W sides, and slide the other ends in the counter-clockwise direction to bring middle lines closer. (c) Lines are equi-spaced and correctly aligned at a slope of 3-in-1, when the S–NE line in a tile aligns with the SE–E line of the tile above. (d) Three more tiles are obtained by reflecting and/or rotating the primary tile.*

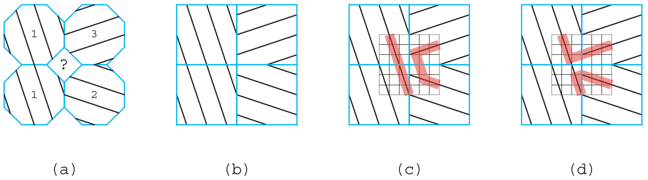


Figure 7: *Routing lines in the gaps between octagonal tiles: (a) Configuration 1321 with octagonal tiles. (b) Octagons replaced by squares for better visibility of possible routes. (c, d) the two possible routings for lines. Note that in (d) it was also possible to chamfer the upper join instead of the lower one. We used the thin line graph in the middle of (c) and (d) to read coordinates, but it also marks the area of CA tiles.*

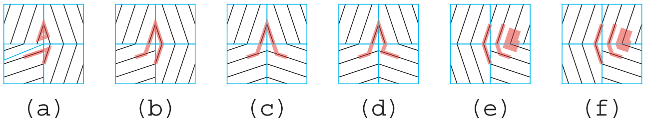


Figure 8: *Examples of things to consider in routing lines between octagonal tiles: (a) A routing that makes a small closed area, which would appear as a big spot after thickening lines, (b) an alternative routing which avoids the small closed area, (c) a simple routing which makes non-uniform white space, (d) a sophisticated routing which makes more uniform white space, (e) a projecting miter joint facing an edge, which would "throttle" the white path when lines are thickened, and (f) the miter joint chamfered parallel to the edge, which makes the white path more uniform, and also align line bents which make the illusory contour. We over-thickened the corner line to make the chamfer clearly visible.*

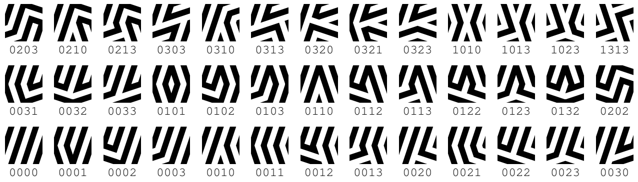


Figure 9: *A working set of the 39 distinct CA tiles for 4-way Op Art rendering. Digits indicate orientation of top-left, top-right, bottom-right, and bottom-left constituent tiles.*

one routing might sometimes be evidently preferable; but in other times we have to flip a coin. It could also be a good idea to give the end user the freedom to toggle between the two possibilities on a vertex-by-vertex bases to fine-tune the rendering. This property of alternative routing could also prove valuable in application where the rendered map is used as a tour puzzle.

As in the 3-color case, a fully modular implementation can be attained by defining a set of CA tiles, which can be made square to encompass the whole routing area plus a quarter of each of the surrounding octagonal tiles. Figure 9 shows a working set of the 39 distinct layouts of CA tiles. As with the 3-color case, rendering with the CA tiles is straightforward: the appropriate CA tile, selected according to the colors of 4 pixels, is placed on the vertex between these 4 pixels. Figure 10(a) shows an example Op Art rendering of a 4-color drawing. Like its 2- and 3-color counterparts, this modular approach guarantees artifact-free output for an arbitrary 4-color input map, as can be seen by inspecting the CA tile set. It also outperforms Inglis and Kaplan's 4-color algorithm in producing uniform line thickness in all 4 directions. Once again, the cost is a considerably high line density: $\sqrt{10}$ (≈ 3.16) lines per pixel; compared to average line density of $\frac{(1+\sqrt{2})}{2}$ (≈ 1.21) in Inglis and Kaplan's 4-color algorithm.

5.3 Overall Message

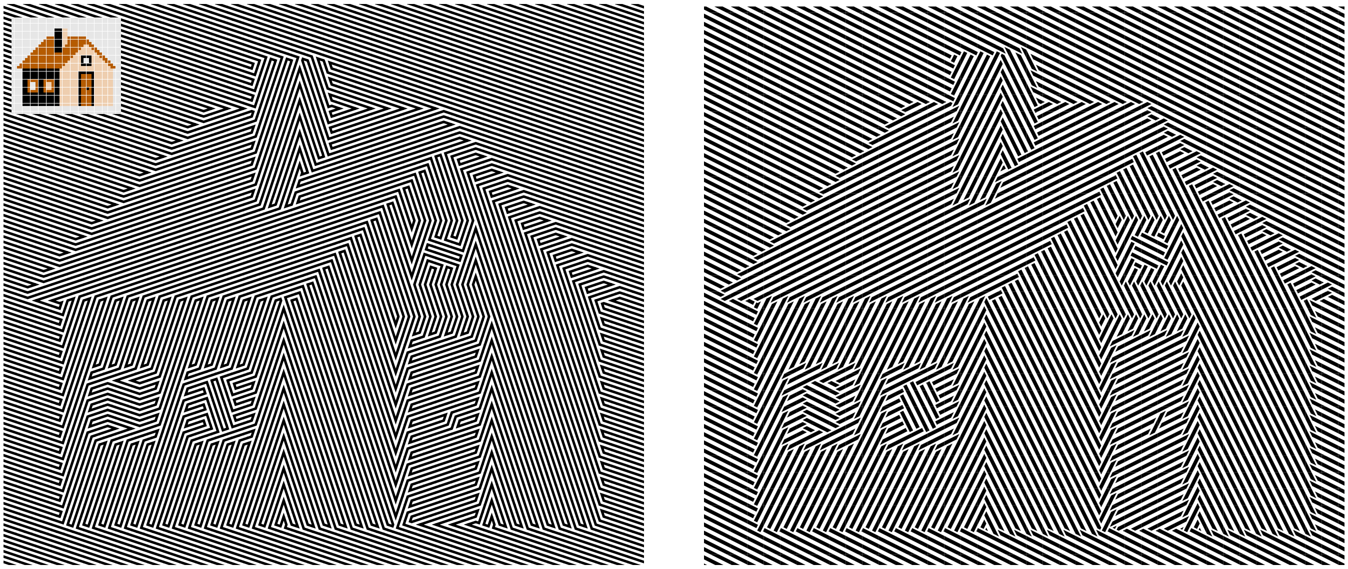
We have seen that by using a modular approach it is possible to generate artifact-free Op Art renderings from arbitrary 3- and 4-color input maps; which was previously deemed impossible [Inglis et al. 2012]. The question we ask is: is it true Op Art; compared to, say, Reginald Neal's "Square of Two"? Well, in any of 3-possible directions of edges in a 3-way rendering, two out of three color combinations on sides produce secondary edge messages, and only one shows the primary message. Coincidently, the ratio is the same in the 4-way case. Two thirds of edges of a randomly picked map, therefore, are likely to render into "U-shaped" edges. Thus, whereas our tile-based approach successfully fulfills two important elements of line-based Op Art (parallelism and elimination of artifacts), it shows poor performance in the third requirement of a consistent edge message based on illusory contours. The problem is not totally caused by our algorithm: part of the problem seems to be inherent in the Op Art rendering concept itself. Thus, unless the input map is carefully designed to reveal primary edges, the result we obtain is more of a labyrinth-like rendering than Op Art. We would therefore suggest adapting the approach to relax the primary edge message altogether and emphasize on secondary message; renaming it "labyrinth rendering". One way to do so is to use narrower lines so that stacks of corners are not quite distinct from rows of line bends.

6 Shadow and Offset Rendering

Having used a known tile for a new rendering style, and new tiles for a known rendering style, we will now present a new tile capable of rendering a new distinct style. The tile, which we will call "shadow tile", is similar to Truchet tile, but is bisected parallel to a side instead of diagonally. See Figure 1(f). It is certainly "simpler" than Truchet tile in more than one aspect. Indeed, a spatial resolution of 2×2 pixels is enough to produce this tile precisely, even in a monochrome device; whereas a Truchet tile can not be produced precisely using square pixels. Further, the tile is also easy to produce physically by aligning two differently colored bricks of 2:1 aspect ratio, which is the aspect ratio of common bricks and most wall tiles. This tile is perhaps the simplest possible Truchet-like tile, and is still capable of producing two interesting rendering styles, as we will see next. Shadow tile has 4 distinct orientations, and enjoys a line density of 1; same as the Inglis-Kaplan 2-way non-modular Op Art.

Using horizontally and vertically bisected tiles to pave black and white pixels, respectively, produces what we call "shadow rendering", as illustrated in Figure 2(h). The name is borrowed from a well-known weaving style which looks similar, although it is not produced in the modular way discussed here. See [Muller 1998], for example. The primary edge message in shadow rendering is a combination of "shadows" in the West and South sides of black regions of the underlying image, and "highlights" in the North and East sides; as if light is spot from the NE direction. A secondary edge message, which resembles the primary edge message of Op Art (stacks of corners), appears on edges at 45° (those running NE to SW); but it does not seem to be detrimental to the perception of the overall style, possibly because these edges run along the perceived direction of light, and should therefore neither reveal shadows nor highlights. The overall style bears the impression that the drawing is "embossed". With other combinations of tile orientations for paving black and white pixels, it is possible to switch between embossing and debossing, and to change the direction of highlights and shadows, but the style remains similar as long as one tile is bisected vertically and the other horizontally. We will return to shadows and highlights later in Section 8.2, but besides them, another way to explain how this style works is through rasterizing tiles into 2×2 pixels. On rasterizing the rendering of Figure 2(h), we find that pixels at even x and y coordinates are always black, and those at odd x and y coordinates are always white. These pixels are, therefore, "neutral": they just maintain the 50% tone of gray. We then find that pixels with odd x but even y coordinates have the same color as pixels of the underlying image, and those at even x and odd y coordinates have inverted pixel colors relative to the underlying image. This resembles a commonly used embossing technique, where a copy of a drawing is offset in space and inverted in color. See Figure 11.

A slightly different rendering style is obtained by using 180° ro-



(a) (b)

Figure 10: *An example tile-based rendering of a 4-color drawing (inset) using (a) Op Art and (b) weave rendering.*

(a) (b) (c)

Figure 11: *(a) The word "EMBOSS" in a stylized typeface. (b) A copy of the word is offset up and to the right. (c) The topmost copy painted in white to give the illusion of white text projecting black shadows.*

tated shadow tiles to pave black and white pixels. See Figure 2(i). This time, highlights and shadows are in the North and South edges, respectively, while the East and West edges appear as if the parallel lines are "cracked" and offset downwards. The overall style, which we call "offset rendering", again has a 3D depth, but this time it bears the impression of the drawing being pulled out and downwards. This rendering style can be seen in some of Victor Vasarely's artworks. The offset illusion becomes more apparent if narrower lines are used so that they do not touch; otherwise it is not possible to tell which side of vertical edges is above or below. See Figure 2(j). Narrower lines can easily be modularly implemented using CA tiles re-centered on the vertical edges between adjacent pixels.

It is certainly possible to extend shadow rendering to work with the 3-color input maps, and since we are relieved from the line-end matching constraint of Op Art, we are free to use either triangular or hexagonal tiles, and have a substantially lower line densities by using a single line to decorate tiles. On the other hand, there seems to be no readily usable tile shape for 4-color input maps, but we will discuss in Section 8.2 how to synthetically create CA tiles for 4-, and even 6-way shadow rendering. Figure 5(b) shows an example shadows rendering with hexagonal CA tiles. We are going to skip details, as they are quite similar to weave rendering below, and, as we will see, the latter represents a more general case.

7 Weave Rendering

Even though shadow tiles described in the previous section preserve the 50% shading in each individual tile, and therefore convey a 50% ambient level of gray, highlights and shadows remain visible when the rendering is scaled down or seen from a distance. It is as if these tiles borrow black color from one side of a region in the rendered map, and restore it back on the opposite side. This is certainly related to the asymmetric nature of shadow tiles, which suggested the next step in our inspection: to design a symmetric version of the shadow tile. Figure 1(g, h) show two options for symmetrizing the shadow tile, which we call "weave tile" and "negative weave tile", respectively. Figure 2(k, l), respectively, show example renderings with these tiles. Line density is 1, similar to shadow rendering, but highlights and shadows are completely eliminated. On boundaries between regions of the underlying image, lines from one region stop short of crossing lines in the other direction; specifically, the distance is half of distance between parallel lines. Unlike all other styles we have seen earlier in this paper, the edge message here is consistent and fixed at all angles: there are no secondary edge messages, which is a great advantage. The overall style looks like lines at region edges go above or beneath lines from the other direction; which is why we call it "Weave Rendering". This is more pronounced in case of negative tiles, possibly because each region is surrounded by a slim ribbon of black shadow which reinforces the feeling that crossing lines go beneath.

7.1 3-Way Weave Rendering

Analogous to what we did for Op Art rendering, it is easy to extend weave rendering to work for 3-color input maps. The tile to be used is hexagonal, with a line connecting mid-points of facing sides. See Figure 12(a). As it contains a single line, this tile yields a line density of $(4/3)^{1/4}$ (≈ 1.07), half that of its Op Art counterpart. Although lines never intersect, it is still desirable to draft CA tiles so as to add the appropriate chamfers on line ends. The rule is taken from the 2-color case: each line end should stop half-line-width short of a crossing line. See Figure 12(b). Like their 2-way counterpart, these tiles convey a singular consistent message on edges. Figure 5(c) shows an example rendering with these CA tiles. Since all lines meet at an angle, the feeling of weaving is more apparent here than in the 2-way case.

(a) (b)

Figure 12: *(a) Three orientations of a hexagonal tile for weave rendering of 3-color maps. (b) The 7 distinct layouts of the corresponding CA hexagonal tiles. Note that the triangle in the middle of tile 102 is manually inserted to avoid the large white area in this tile.*

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

Figure 13: *Developing 4-way weave tiles: (a) Two tiles which weave well along vertical edges, but there is no reasonable way to resolve priority along horizontal edges. (b) Which line goes above/beneath is clear here, but weaving is not symmetric on the sides of the horizontal edge between tiles, which would later make aligned edges appear offset. (c) Two tiles which weave perfectly along both horizontal and vertical edges, and (d) a working set of 4 tiles comprising the two in (c) and their 90° rotations.*

7.2 4-Way Weave Rendering

To extend weave rendering to 4-color input maps we need a seamlessly tile-able square tile decorated with slanted line(s), so that it can be oriented 4-ways using symmetry operations of the square. Since we are relieved from the line-end matching constraint of Op Art, we can use a 2-in-1 slope instead of 3-in-1; a change which reduces line density and also produces more even distribution of angles between orientations. The constraint we have, however, is that a tile should not reflect across an edge to produce another tile, because, then, there would be no reasonable way to resolve which line goes above and which goes beneath to make the weave. To solve this problem we combine reflection with inversion of colors to make a secondary tile. This makes lines from a primary tile match gaps of a secondary tile (and vise-versa) on the reflected edge, which makes a perfect weave along that edge. We had to inspect a few alternatives before arriving at the working set of tiles shown in Figure 13(d). Although these tiles, by design, do not have reflection symmetries, we are lucky that they still bear rotational symmetries, as this reduces the set of distinct CA tiles layouts from 256 to 70 (not 64, since some tiles map to themselves). See Figure 14.

Figure 10(b) shows an example rendering with 4-way weave tiles. It shares all the properties of 3- and 2-way renderings: scintillating effect with positive tiles, weaving effect with negative tiles, and consistent edge message at all angles. The main drawback is the excessive line density of $\sqrt{5}$ (≈ 2.24). It should be possible, however, to halve this by using position-dependent tiles, as we did earlier with inversion rendering. The idea is to use quarters of the tiles of Figure 13(d), taking care of which quarter fits in which position. Although it is possible to draft CA tiles from these quarters, it might not be a good idea, given the relatively large possibilities which arise as position is added as a new dimension. A recommended alternative is to algorithmically track individual lines, tile by tile, until a line stops short of a crossing line. It should not be difficult to do so for two reasons: First, there is no much interaction between lines in weave rendering as in, say, Op Art; all lines are single segments, just that they need to have their ends properly chamfered. Second, all quarters of tiles of Figure 13(d) can be obtained from only one prototype by applying symmetries of the square; so much of the tracking logic made for a single orientation can easily be reused for all other orientations.

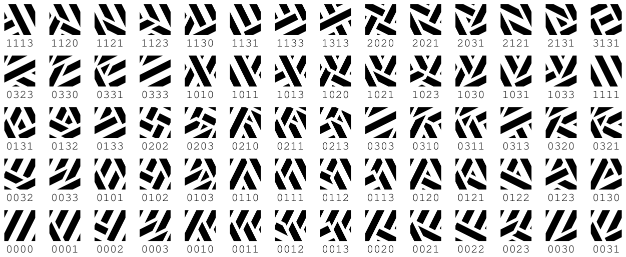


Figure 14: *The set of distinct layouts of CA tiles for 4-way weave rendering.*

(a) (b)

Figure 16: *(a) Choosing a slanted line direction for decorating 6-way weave tiles. The line shown starts at a mid-point of one edge, and crosses 2 edges before reentering back in its starting position. (b) A working set of raw tiles developed observing similar considerations to the 4-way case (Figure 13).*

7.3 6-Way Weave Rendering

Although 4-way lines should be able to render any planar map (by virtue of the 4-colors theorem), it might be desirable or helpful to use more line directions. For example, it might be useful to reserve a separate line direction to represent a background color, or for seas in a geographical map. Fortunately, the edge message of weave rendering is flexible enough to support even more line directions. We have consumed all symmetries of the square, but a hexagonal tile can give six distinct orientations if decorated with slanted lines (that is, not along a symmetry axis). Figure 16 illustrates a selection for a line orientation which should give a line density of $(196/3)^{1/4}$ (≈ 2.84). Figure 15 shows an example 6-way weave rendering. As suggested in the 4-color case, line density can be reduced further by using position-dependent tiles.

8 Relationships between Styles

We are now going to highlight inter-relationships between some of the styles we have discussed so far. These relationships will help us gain better understanding of these styles, and we will be able to mix and match between them.

8.1 Weave Rendering and Op Art

Weave rendering, specially with positive tiles, exhibit a scintillating effect similar to (but certainly less than) Op Art, especially on electronic displays, specially in the 2-way case which can be rendered perfectly (no aliasing). Let us try to explain this using the information theoretic concepts we used in Section 3. In 2-way weave rendering we have seen that lines in one direction stop half-line-width short of a line in the crossing direction, then we developed 3- and 4-way tiles to observe this. What happens if we stop them earlier, full-line-width short of a crossing line? Figure 2(m) demon-

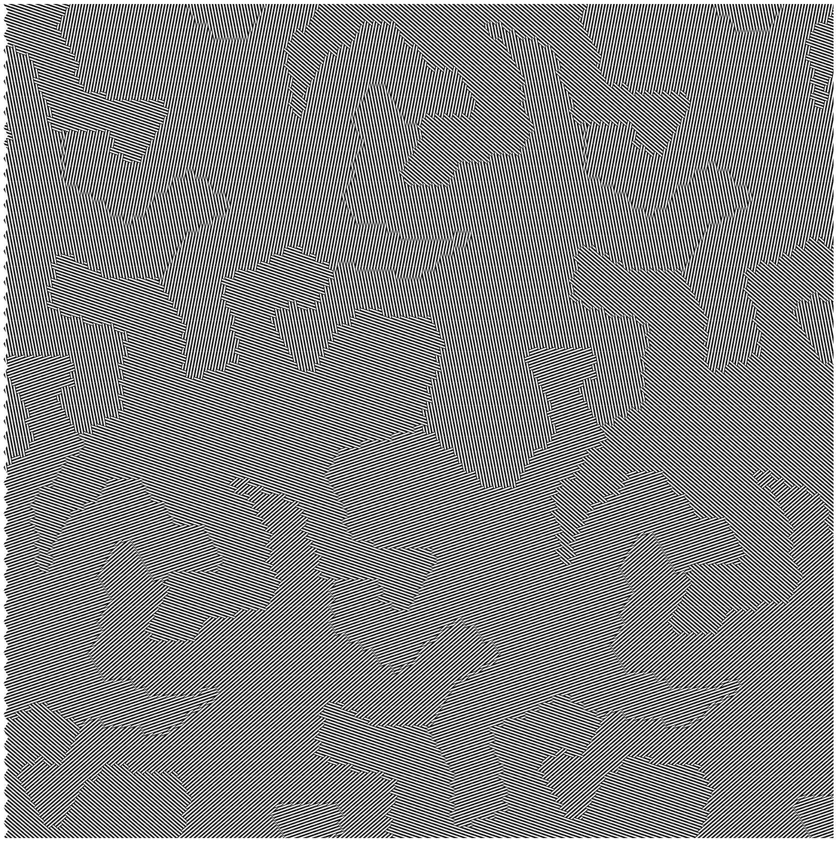


Figure 15: *Escher lizards illustration of 6-way weave rendering, scaled to give the same line width as in Figure 5. Note that although the edge message is uniform, edges do not all look the same, because of aliasing differences at different angles between lines.*

strates the effect of this: Although vertical white lines on the sides of the cottage, for example, are the same width as all other white lines, they appear "whiter". This is related to human perception and illusory contours, but we can roughly explain it using the same information theoretic approach of "surprise": parallel lines in regions prepare the eye to expect parallel black and white lines, and the fact that the highlighted white lines are open on one side to other white lines makes the eye see more white than it was expecting to see; that is, too white! Because this message of whiter lines is consistent across all edges of the underlying drawing, we can count this as a distinct rendering style; let us call it "white ink rendering". A single tile like that in Figure 1(g) can not implement it, but a modular approach to implementation is easy with the concept of context-aware tiles we used in previous sections. It is also easy to extend the rendering concept to 3-, 4-, and 6-color input maps. If instead of retracting lines of a weave rendering, we extend them to merge with the crossing lines, we obtain a too black illusion on the edges. See Figure 2(n). We call this "black ink rendering", and it can be explained the same way. It is worth noting that white and black ink renderings do not conform with the 50% shading of other Truchet-like tiles, and would therefore show edges when scaled down or looked at from a distance.

Back to weave rendering, we find that making lines stop halfway makes a perfect balance between emphasizing the white and the black color. In other words, it does meet the viewer's eye expectation built up from looking inside regions, and transition between regions now becomes smooth; just that line orientation changes. Thus, instead of emphasizing edges, weave rendering tends to hide them altogether, and rely more on change of line orientation to signal a different regions. Even though this contradicts with the fact that edges are more important, being able to hide edges this way seems to be the surprise carried by weave rendering.

After this analysis we can now compare weave rendering to Op Art. When edges are neutralized the way we explained, more emphasis goes to parallelism, which makes the characteristic scintillating effect of Op Art. We can therefore see weave rendering as Op Art less the corner-based edges. Earlier in this paper we chose stacks of corners as the preferred edge message in Op Art, and justified it by the choice of a recognized artist. We can now go further and give an explanation, that stacks of corners preserves parallelism all through the image, and therefore maximize the scintillating effect; in contrast to weave rendering, which breaks parallelism on edges. Further, rows of U-shaped line bends destroy parallelism, because they induce many highlights between the U-bends; similar to white ink mentioned above. This furnishes a justification for our earlier claim (Section 3) that U-shaped edges are more visually emphasized than stacks of corners.

8.2 Shadow and Weave Renderings

Shadow rendering relies on highlights and shadows on opposite sides to give the illusion of a 3D depth. Technically speaking, it turns out that highlights and shadows are, respectively, nothing other than the white and black ink described in Section 8.1. This suggests a method to transform weave renderings into shadow rendering by allowing lines to retreat/extend to add highlights/shadows. It should be possible to implement this functionality interactively to let the end user choose where to add highlights

Figure 17: *Rotating gates to animate route change in labyrinths based on shadow rendering.*

or shadows. Alternatively, CA tiles can be designed for automatic shadow rendering. Please note that symmetries will no longer be preserved between CA tiles, so it will be required to draw CA tiles for all combinations; but that has to be done only once. Finally, we are able to have shadow rendering guided by 4- and 6-color input maps.

There is more about black and white inks than shadows and highlights: they also mean, respectively, that lines are attached or detached, and having these shadows and highlights on opposite sides has an interesting consequence. For example, in the shadow rendering of Figure 2(h), having shadows on the West sides and highlights on the East sides of originally black regions is equivalent to having all horizontal lines with left ends attached to vertical lines, and right ends detached. Similarly, all vertical lines are free on their top ends, and their bottom ends are either attached to horizontal lines or reach the bottom edge of the drawing. If, however, we manually edit the input bitmap so that the leftmost column is white and the bottom row is black, then we will be sure that all horizontal lines are attached on the left and free on the right, and all vertical lines are attached on the bottom and free on the top. All lines would then be connected in one block. Thus, under the condition that edges of the input bitmap are edited as described, a shadow rendering represents a spanning tree which grows upwards and to the right, and reaches all pixels of the input bitmap. Via careful design of CA tiles, combined with appropriate editing of edges of input maps, it should also be possible to generate 3-, 4-, and 6-way shadow renderings which represent spanning trees. We will use this property in the following subsection to produce single loop renderings.

8.3 Shadow Rendering and Op Art

An interesting property of shadow rendering is that distance between lines is equal to line widths, and this holds as well for the free ends of lines (the highlight sides). If the lines themselves are outlined the result would look similar to Op Art, or more generally to a labyrinth rendering guided by the underlying map. Combined with the spanning tree property of shadow rendering, this suggests a modular approach for single loop rendering, as demonstrated in Figure 2(o). Note that the primary Op Art message (stack of corners) is now confined to only edges at 45° angles: edges which already bear that edge message in the underlying shadow rendering. Thus, the drawing is dominated by U-shaped edges, and it is more appropriate to call it labyrinth rendering. Outlining normally doubles line density; but this can be reduced by employing position dependent tiles for the 4- and 6-way cases. Since shadow tiles are decorated with a single line in the 2- and 3-way cases, position dependent tiles are not possible. Instead, line density can be reduced by sampling to equilateral triangular pixels for the 3-colors case, and right angled isosceles triangular pixels for the 2-color case. It is worth noting that the Hamiltonian tour length here is independent of the underlying image details, and that switching between an attached (shadow/black ink) and detached (highlight/white ink) line end in the intermediate shadow rendering is equivalent to toggling doors in the labyrinth. See Figure 17. Opening and closing such doors does not significantly change the shape of the labyrinth, but it attaches and detaches parts of it; which can prove valuable in tour puzzles and interactive maze design.

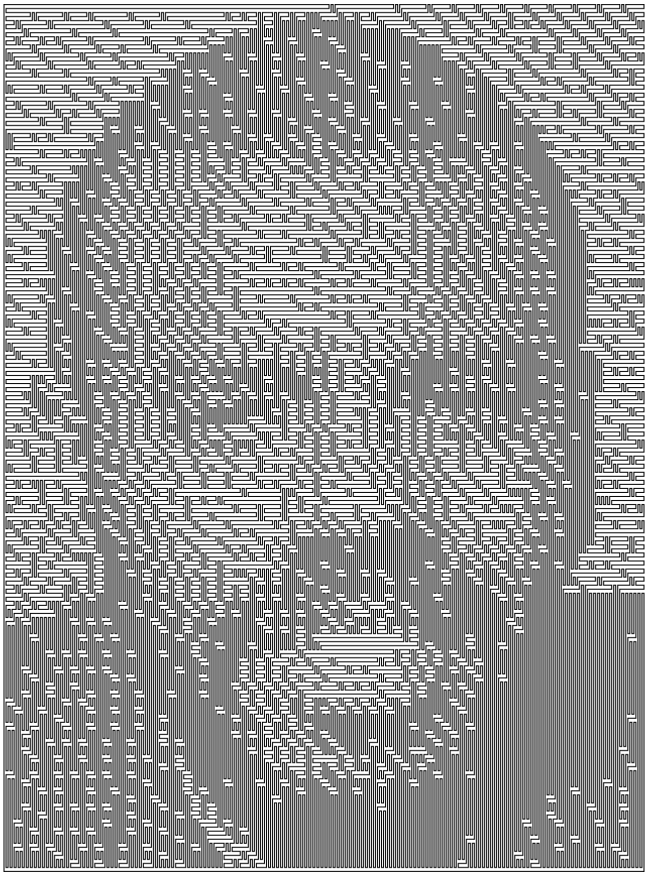


Figure 18: *A continuous line halftoning obtained by outlining rectangular shadow tiles used to pave a dithered image.*

9 From Shape to Tone

Before concluding this paper we would like to highlight how the tile-based approaches we discussed here can be adapted to convey different tones, and hence used for line-based halftoning. We have been using regular polygons for tiles throughout this paper; but what if we use, for example, rhombuses instead of squares for the Inglis-Kaplan tile (Figure 1(d)), or rectangles instead of squares for our shadow tile (Figure 1(f))? Squeezing tile shapes while preserving line widths makes decorating lines longer and closer apart in one direction than another; hence it makes different tiles convey different tones even though they use the same line width. Another approach which works with outlined shadow tiles (Section 8.3) is to use more decorating lines in one direction than another. As illustrated in Figure 18, this approach can be used for modular line-based halftoning, and by combining deformation of tiles with variation of number of decorating lines, it is possible to generate tiles with up to 6 different tones. Even though such rendering might be less beautiful than, say, TSP art [Kaplan and Bosch 2005], it has its merits, such as speed. Another important feature of this halftoning approach is that it is built from short connected line segments of fixed lengths, which is perfectly compatible with the stitching structure of a sewing machine. This suggests an application in halftoned embroidery, where line continuity is highly appreciated.

10 Conclusion

In this paper we demonstrated how 50% shaded Truchet-like tiles can be used to produce aesthetic rendering styles. We discussed cases of using known tiles for a new rendering style, designing new tiles for a known rendering style, and introducing new rendering styles along with their associated tile sets. Our research was originally motivated by the challenge of finding a general solution for producing artifact-free 3-way Op Art, but we ended up being able to produce 6-way artifact-free single-loop renderings; which is a substantial contribution of this paper; even though we found that being free of artifacts does not guarantee a nice looking Op Art rendering. Other contributions of this paper include defining some qualitative and quantitative measures for assessing tile-based renderings, and introducing the concepts of position-dependent and context-aware tiles. CA tiles can be used directly, possibly pre-rasterized or converted to a font for improved performance, and they can also prove useful in developing other implementation approaches. For example, we used them to identify the states of an automaton which generates Postscript output.

For future research we suggest exploring the application of tile-based rendering to texture synthesis, considering some recent researches which employed Truchet tiles in texture synthesis [Garigipati and Akleman 2012; Akleman et al. 2013]. After proving the existence of artifact-free Op Art solution of a general 3- and 4-color map, we plan to search for solutions with lower line densities. Indeed, all artifact-free solutions presented in this paper came at a cost in line density. We also plan to research further in tile-based maze design, as well as tile-based single-line halftoning and its application to embroidery.

References

AKLEMAN, E., XING, Q., GARIGIPATI, P., TAUBIN, G., CHEN, J., AND HU, S. 2013. Hamiltonian cycle art: Surface covering wire sculptures and duotone surfaces. *Computers & Graphics 37*, 5, 316 – 332.

BOSCH, R., AND COLLEY, U. 2013. Figurative mosaics from flexible truchet tiles. *Journal of Mathematics and the Arts 7*, 3-4, 122–135.

BOSCH, R. 2011. Opt Art: Special Cases. In *Proceedings of Bridges 2011: Mathematics, Music, Art, Architecture, Culture*, Tessellations Publishing, Phoenix, Arizona, R. Sarhangi and C. H. Séquin, Eds., 249–256. Available online at `http://archive.bridgesmathart.org/2011/bridges2011-249.pdf`.

BROWNE, C. 2008. Duotone Truchet-Like Tilings. *Journal of Mathematics and the Arts 2*, 4, 189–196.

GALE, D., PROPP, J., SUTHERLAND, S., AND TROUBETZKOY, S. 1995. Further Travels with My Ant. *Math. Intelligencer 17*, 48–56.

GARIGIPATI, P., AND AKLEMAN, E. 2012. Duotone surfaces. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, CAe '12, 99–106.

GLASSNER, A. 1998. Aperiodic tiling. *IEEE Comput. Graph. Appl. 18*, 3 (May), 83–90.

INGLIS, T. C., AND KAPLAN, C. S. 2011. Generating Op Art Lines. In *Workshop on Computational Aesthetics*, Eurographics Association, Vancouver, Canada, D. Cunningham and T. Isenberg, Eds., 25–32.

INGLIS, T. C., AND KAPLAN, C. S. 2013. Animating Line-based Op Art. In *Proceedings of Bridges 2013: Mathematics, Music, Art, Architecture, Culture*, Tessellations Publishing, Phoenix, Arizona, G. W. Hart and R. Sarhangi, Eds., 159–166. Available online at `http://archive.bridgesmathart.org/2013/bridges2013-159.pdf`.

INGLIS, T. C., INGLIS, S., AND KAPLAN, C. S. 2012. Op Art Rendering with Lines and Curves. *Computers & Graphics 36*, 6, 607 – 621.

KAPLAN, C. S., AND BOSCH, R. 2005. Tsp art. In *Renaissance Banff: Mathematics, Music, Art, Culture*, Canadian Mathematical Society, Banff, Alberta, R. Sarhangi and R. V. Moody, Eds., 301–308. Available online at `http://archive.bridgesmathart.org/2005/bridges2005-301.html`.

KRAWCZYK, R. J. 2011. Truchet Tilings Revisited. In *Hyperseeing*, 69–77.

MULLER, D. 1998. Putting the Shadow in Shadow Weave. *Handwoven XIX*, 4 (Sep/Oct), 34–40.

PICKOVER, C. A. 1989. Picturing Randomness with Truchet Tiles. *J. Recr. Math. 21(4)*, 256–259.

REIMANN, D. 2009. Text from Truchet Tiles. In *Proceedings of Bridges 2009: Mathematics, Music, Art, Architecture, Culture*, Tarquin Books, C. S. Kaplan and R. Sarhangi, Eds., 325–326. Available online at `http://archive.bridgesmathart.org/2009/bridges2009-325.pdf`.

REIMANN, D. A. 2012. Modular knots from simply decorated uniform tessellations. In *Proceedings of ISAMA 2012: Eleventh Interdisciplinary Conference of the International Society of the Arts, Mathematics, and Architecture*, 63–69.

SALOMON, D. 2004. *Data Compression. The complete reference*, 3rd edition ed. Springer, New York.

SMITH, C., AND BOUCHER, P. 1987. The Tiling Patterns of Sebastian Truchet and the Topology of Structural Hierarchy. *Leonardo 20(4)*, 373–385.