Two-Frame Animations in Conway's Game of Life

Robert Bosch
Dept. of Mathematics
Oberlin College
Oberlin, OH 44074
(`rbosch@oberlin.edu`)

Abstract

Using integer programming, we designed a set of period-2 oscillators in Conway's Game of Life. Our goal was to produce oscillators that can be joined together to form large period-2 oscillators that resemble one grayscale image in one phase and a second grayscale image in their other phase.

Introduction

In the late 1960s, John Horton Conway invented Life, a one-player game played on a grid of squares [5,6]. At each time $t = 0, 1, 2, \ldots$ each square cell is either alive or dead. Starting from an initial pattern, the "time 0" pattern, the player repeatedly follows Conway's rules to produce the "time 1" pattern, the "time 2" pattern, and so on. On each update, the fate of each cell depends on its current state and on the states of its eight neighbors. Conway's rules can be denoted B3/S23, as they mandate the following: If at time t cell (i, j) is dead and exactly three of its eight neighbors are alive, then at time $t+1$ cell (i, j) will become alive (a birth). If at time t cell (i, j) is alive and has either two or three living neighbors, then at time $t+1$ cell (i, j) will remain alive (survive). Otherwise at time $t+1$ cell (i, j) will be dead.

Recently Bosch and Olivieri introduced two Game of Life mosaicking systems [2,3]. Their *still life* tiles are stable under Conway's rules, and their *phoenix* tiles are period-2 oscillators that have no cells that are alive in both phases (implying that each living cell dies and then is immediately reborn). In both cases, the tiles are modular, which means that they can be joined together to form larger patterns that retain their defining characteristic (being stable or being a phoenix). And in both cases, the tiles have various brightness levels (due to having various numbers of living cells), which means that they can be used to construct larger patterns that resemble user-supplied grayscale images.

Here we extend the work of Bosch and Olivieri by presenting a collection of period-2 oscillators that can be joined together to form two-frame animations.

HL, LH, LL, and HH Tiles

Using an integer programming formulation similar to those described in [1,3], we searched for period-2 oscillators with maximum *constrast*, the largest possible difference between the number of living cells in phase one and the number of living cells in phase two. Our formulation has two indicator variables per cell: $x_{i,j}$ is either 0 or 1 depending on whether cell (i, j) is dead or alive during the first phase, and $y_{i,j}$ is either 0 or 1 depending on whether cell (i, j) is dead or alive during the second phase. In terms of these variables, the contrast is $\sum_i \sum_j (x_{i,j} - y_{i,j})$. We maximize this linear function subject to linear constraints that enforce Conway's rules. After considerable experimentation with the formulation, we discovered a pattern that we were then able to extend into the pattern shown in Figure 1. This pattern fits in a 28×28 square, has $D4$ symmetry, and has 352 living cells in one phase and 288 living cells in the other. Due to its having $22.\overline{2}\%$ more living cells in one phase that the other, this oscillator appears noticeably brighter in one phase than the other. We call it the HL tile. "HL" is short for "high-low." By swapping the phases, we obtain the LH

tile. The border pattern (which is shaded light gray) and the "moat" of half cells that surrounds it allow us to place two copies of these tiles side by side and be sure that the resulting pattern is also a period-2 oscillator.
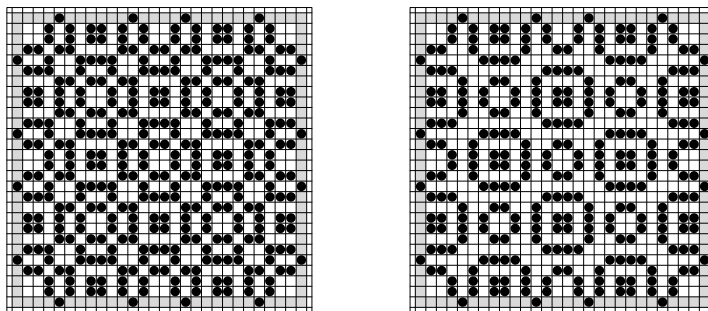
Figure 1: *The high and low phases of the HL tile, a* 28×28 *period-2 oscillator. The high phase has* 352 *living cells, and the low phase has* 288*.*

We then used integer programming to search for LL (low-low) tiles: 28×28 period-2 oscillators that have precisely 288 living cells in each phase, have $D4$ symmetry (to reduce the size of the search space), and have the same border pattern as the HL tile. One is shown in Figure 2. Here, our goal was to maximize the *heat* of the pattern, the number of cells that change state. Here, we used a formulation that has many more indicator variables per cell. For example, in addition to the $x_{i,j}$ and $y_{i,j}$ variables, there are $AL_{i,j}$, $AO_{i,j}$, $LA_{i,j}$, $OA_{i,j}$ variables (and others as well). The $AL_{i,j}$ variable is 1 if cell (i,j) is alive in the first phase but dies from loneliness (from having too few living neighbors). The $OA_{i,j}$ variable is 1 if cell (i,j) is alive in the second phase but dies from overcrowding (from having too many living neighbors). Here, we maximize the linear function $\sum_i \sum_j (AL_{i,j} + AO_{i,j} + LA_{i,j} + OA_{i,j})$ subject to linear constraints that enforce Conway's rules.
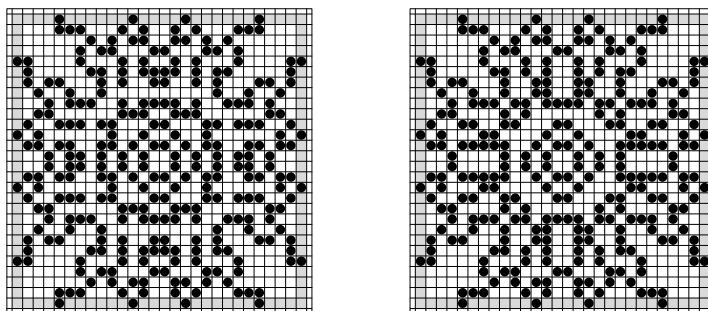

Figure 2: *The high and low phases of one of the LL tiles. Both phases of this* 28×28 *period-2 oscillator have* 288 *living cells.*

By including constraints that prohibit previously discovered solutions, we were able to find additional LL tiles, and by repeating this process over and over again, we ended up with a large collection of them. The search took a considerable amount of computer time (many days worth), and we aborted it when we had decided that we had accumulated enough tiles. The integer programs are not easy to solve, as their linear programming relaxations do not provide good upper bounds on the maximum heat.

Finally, we used integer programming to search for HH (high-high) tiles: 28×28 period-2 oscillators that have precisely 352 living cells in each phase, have $D4$ symmetry (again to reduce the size of the search

space), and have the same border pattern as the HL tile. Again, when solving our integer programs, we maximized heat. Here, we were disappointed to learn that for HH tiles, the maximum heat is zero. This means that all of our HH tiles are actually stable. They are not true period-2 oscillators. One is shown in Figure 3.
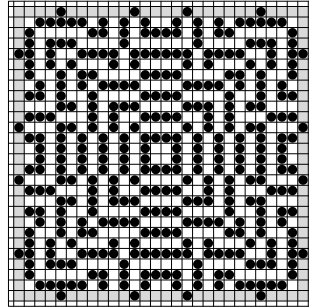
Figure 3: *One of the HH tiles. This* 28×28 *still life (stable pattern) has* 352 *living cells.*

Creating Two-Frame Animations

We start with two grayscale images I_1 and I_2 of identical size and partition each one into m rows and n columns of square ($k \times k$) blocks of pixels. If necessary, we crop the images. For each block of pixels in image I_1, we compute the average brightness value $\beta_{i,j}^1$ of the pixels in the block, using a 0-to-1 black-to-white scale. We do the same for image I_2 to obtain its brightness values (denoted $\beta_{i,j}^2$).

To build a period-2 oscillator that resembles I_1 in phase one and I_2 in phase two, we construct a two-dimensional array of HL, LH, LL, and HH tiles. If we draw living cells in black ink on a white background, then we use the following table to select the tile for block (i, j).

Image I_1	Image I_2	Tile
$\beta_{i,j}^1 < 0.5$	$\beta_{i,j}^2 < 0.5$	HH
$\beta_{i,j}^1 < 0.5$	$\beta_{i,j}^2 \geq 0.5$	HL
$\beta_{i,j}^1 \geq 0.5$	$\beta_{i,j}^2 < 0.5$	LH
$\beta_{i,j}^1 \geq 0.5$	$\beta_{i,j}^2 \geq 0.5$	LL

Figure 3 displays the frames of a period-2 oscillator/animation based on photographs taken by Eadward Muybridge. Links to videos and RLE patterns (for running the patterns on the Golly Life simulator [7]) can be found at the DominoArtwork.com website [4].

References

[1]   R.A. Bosch, *Integer programming and Conway's game of Life*, SIAM Review 41(3) (1999), pp. 594-604.

[2]   R. Bosch and J. Olivieri, *Game of Life mosaics*, Proceedings of Bridges Seoul: Mathematics, Music, Art, Architecture, Culture, (2014), G. Greenfield, G. Hart, and R. Sarhangi, eds., Tessellations Publishing, Phoenix, AZ, pp. 325-328.

[3]   R. Bosch and J. Olivieri, *Designing Game of Life mosaics with integer programming*, Journal of Mathematics and the Arts 8 (3-4) (2014), pp. 120-132.
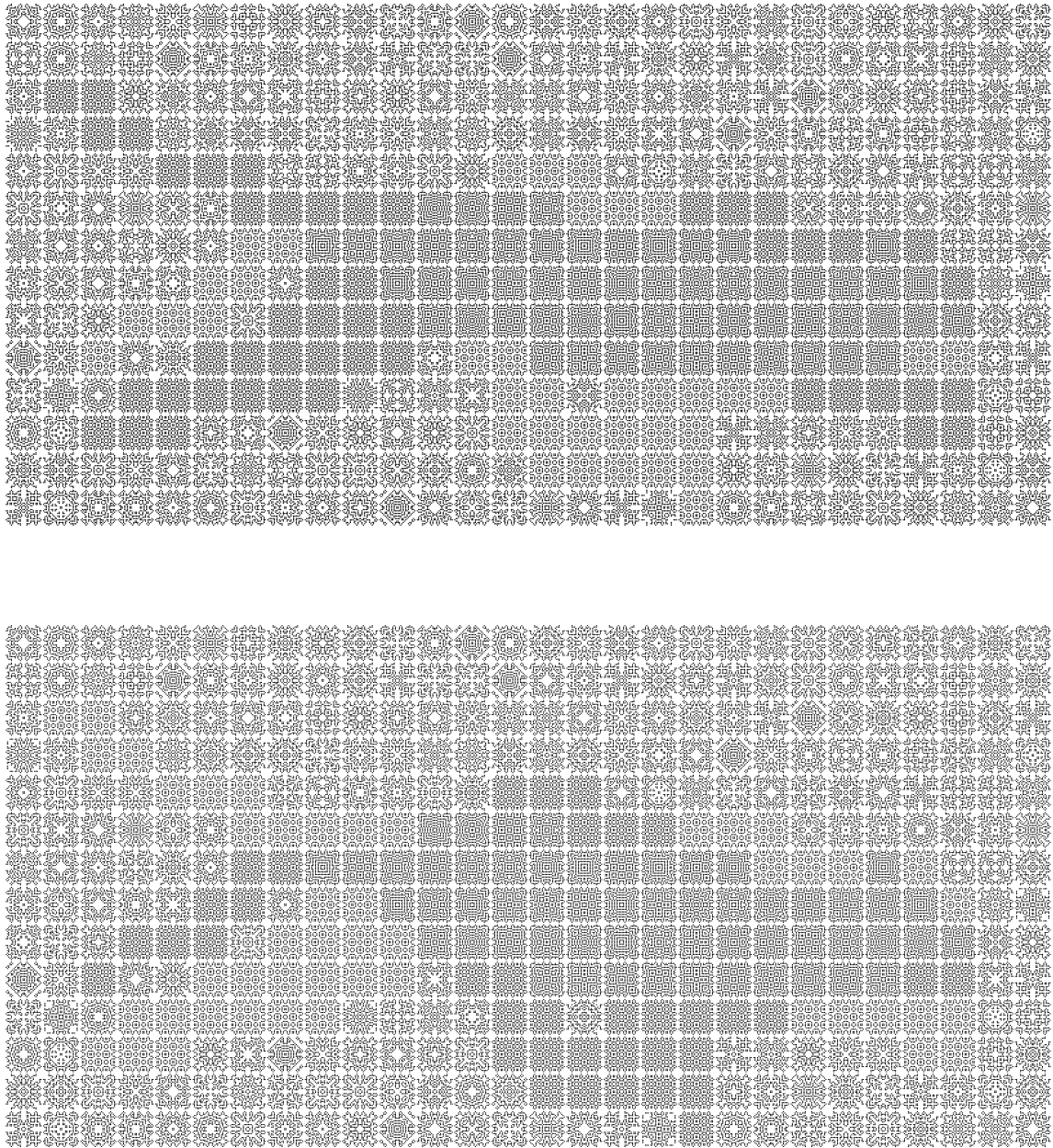
Figure 4: *The two phases/frames of the period-2 oscillator/animation "Running Cat (after Muybridge)".*

[4]   Domino Artwork. `www.dominoartwork.com`. Accessed 18 March 2015.

[5]   M. Gardner, *The fantastic combinations of John Conway's new solitaire game "life"*, Sci. Am. 223 (1970), pp. 120-123.

[6]   M. Gardner, *On cellular automata, self-reproduction, the Garden of Eden and the game "life"*, Sci. Am., 224 (1971), pp. 112–117.

[7]   Golly. Golly Game of Life Home Page. `golly.sourceforge.net`. Accessed 18 March 2015.