Target Curves for Pick-up, Carry, and Drop Mobile Automata

Gary R. Greenfield
Department of Mathematics & Computer Science
University of Richmond
Richmond, VA 23173, USA
ggreenfi@richmond.edu

Abstract

Mobile automata repeatedly sense a grid cell (or neighborhood of cells), perform an action, and then move to a new cell. We explain how to modify mobile automata that pick-up items they encounter while performing random walks and carry them to distant cells to drop in such a way that they are arranged along the circumferences of circles, so that other target curves besides circles can be specified. We apply this technique to the composition of algorithmic designs.

Introduction

Martin Gardner popularized cellular automata — automata that sense and change state — by devoting his October 1970 and February 1971 Scientific American columns to Conway's Game of Life. A.K. Dewdney popularized mobile automata — automata that sense, change state, and move — by devoting his September 1989 Scientific American column to Turk's tur-mites. Paulo Urbano's Sand Painting Artists, which model the circular nest building behavior of *T. albipennis* ants [2], can be viewed as mobile automata that pick-up, carry, and drop virtual sand grains using as targets cells that lie on circumferences of circles. By carefully specifying radii, centers, and sand grain color preferences, I have used this technique to generate compositions consisting of symmetric patterns of circles [1]. Here I consider using target curves other than circles. The stochastic algorithm for deciding how Urbano's automata navigate, pick-up, and drop is given in [1]. It suffices to recall that the drop criterion depends on the mobile automaton's radial distance r_m from its local origin. Unless otherwise specified grids are 500×500 cells, the virtual color grain density is 0.2, 2,000 automata are used, 60,000 time steps are simulated, the local origin is the center of the grid, and when given the opportunity to do so, automata will *fail* to pick-up or drop virtual sand grains only 1% of the time.

From Circles to Ellipses

Let (r_m, θ_m) where $r_m > 0$ be the position of a mobile automaton in polar coordinates when it is headed either directly towards, or directly away from, its local origin. To arrange it so that automata deposit carried items along the circumference of a circle of radius k, it suffices to have them drop their items when $r_m = k$. Suppose, instead, we want them to drop their items on the boundary of the ellipse $x^2/a^2 + y^2/b^2 = 1$. Knowing that the parametric equations of such an ellipse are $(a\cos\psi, b\sin\psi)$, if we naively direct the automata to drop their items when $r_m = \sqrt{a^2\cos^2\theta_m + b^2\sin^2\theta_m}$, then we obtain the curve shown on the left in Figure 1. Something is amiss. The explanation is that ψ is a time variable, and the angle subtended when moving counterclockwise on the ellipse from point P_0 to P_1 determined using the parametric equations with values ψ_0 and ψ_1 does not necessarily have radian measure $\psi_1 - \psi_0$. To remedy this, we need to determine the value ψ_m such that the cartesian coordinates of the intersection point of the ellipse $x^2/a^2 + y^2/b^2 = 1$ and the ray $r = \theta_m$ are $(a\cos\psi_m, b\sin\psi_m)$. To accomplish this we convert the automaton's position (r_m, θ_m) to cartesian coordinates by letting $x_m = r_m\cos\theta_m$ and $y_m = r_m\sin\theta_m$ and set $\psi_m = \tan^{-1}((y_m/b)/(x_m/a))$.

Now we direct the automaton to drop its item when $r_m = \sqrt{a^2\cos^2\psi_m + b^2\sin^2\psi_m}$. The image on the right in Figure 1 confirms this gives the desired result. Further investigation of the "mistake" caused by erroneously using $r_m = \sqrt{a^2\cos^2\theta_m + b^2\sin^2\theta_m}$ as the target curve for an ellipse reveals something unusual, namely that there are values of a and b such that this curve appears almost indistinguishable from a track and field oval whose straightaways are straight lines and whose turns are semicircles (see middle image of Figure 1).

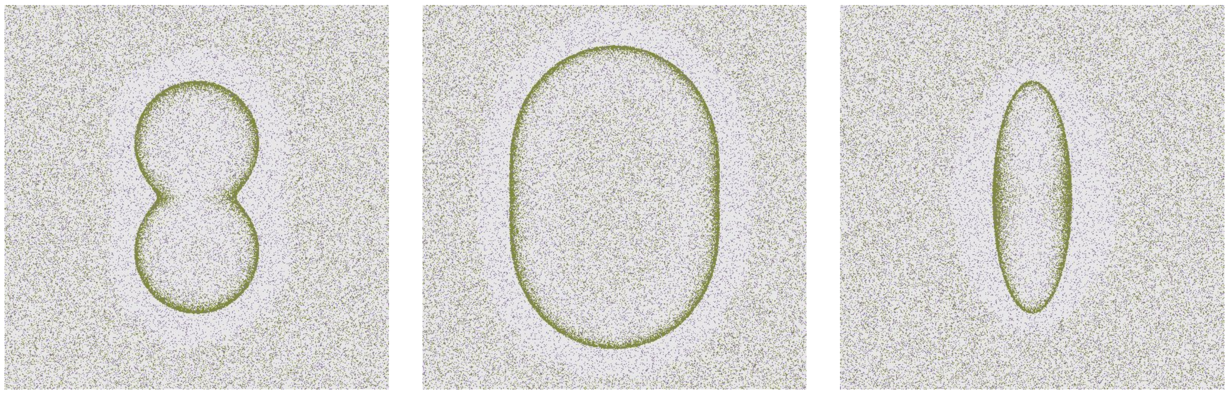

Figure 1 : *Left and Middle: The polar equation* $r = \sqrt{a^2\cos^2\theta + b^2\sin^2\theta}$ *with* $a = 50$, $b = 150$ *and* $a = 135$, $b = 195$ *respectively as the target curve. Right: The ellipse* $x^2/a^2 + y^2/b^2 = 1$ *with* $a = 50$, $b = 150$ *as the target curve.*

Polar Curves

Our ellipse mistake leads to the observation that whenever we have a polar curve given by $r = f(\theta)$ where f is periodic on the interval $[0, 2\pi]$, it is easy to direct automata to use it as a target curve. We simply make the drop criterion $r_m = f(\theta_m)$. Unfortunately, the catalog of polar curves is not as rich as one would like. Common choices are cardioids, n-leaf roses, limaçons, and lemniscates (see Figure 2 for examples).

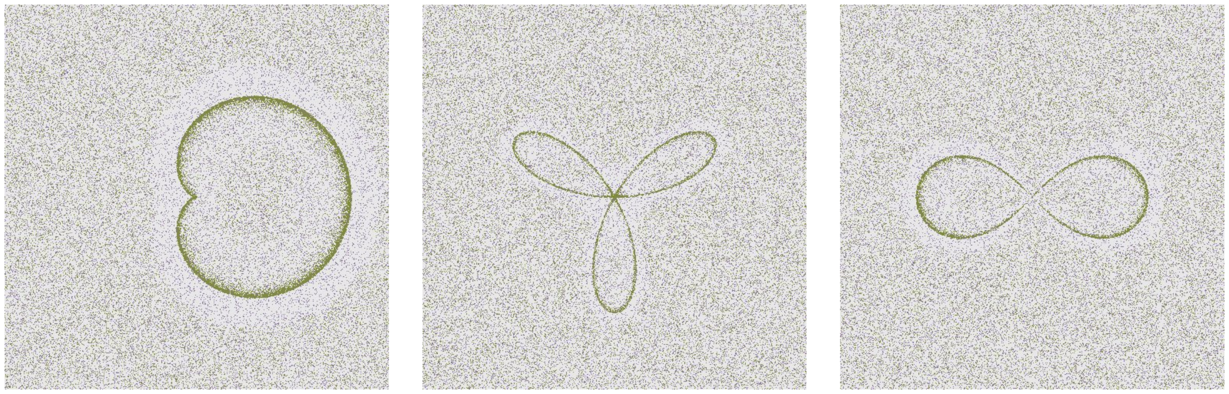

Figure 2 : *Left: The cardioid* $r = a(1 + \cos\theta)$ *with* $a = 100$. *Middle: The 3-leaf rose* $r = a\sin(3\theta)$ *with* $a = 150$. *Right: The lemniscate* $r = \sqrt{a^2\cos(2\theta)}$, *for* $\cos(2\theta) > 0$, *with* $a = 150$.

As it turns out, our implementation protects us when $f(\theta_m) < 0$ by never letting automata drop their items. This is wasteful of automata but gives the desired results (see left and middle images of Figure 3 for visual evidence that this is what is happening). Also wasteful of automata, but a nice feature we can add when using $r_m = f(\theta_m)$ as the drop criterion, is to condition the drop so that θ_m lies either inside the interval $[\theta_f, \theta_l]$ or lies outside it i.e., $\theta_m \in [0, 2\pi] \setminus [\theta_f, \theta_l]$. This allows us to cut the target curve into two pieces and isolate one of them. One piece is realized by tracing counterclockwise from $\theta_m = \theta_f$ to $\theta_m = \theta_l$ and the other is its complement obtained by tracing clockwise from $\theta_m = \theta_f$ to $\theta_m = \theta_l$. The right image of Figure 3 shows that this works even for an ellipse.

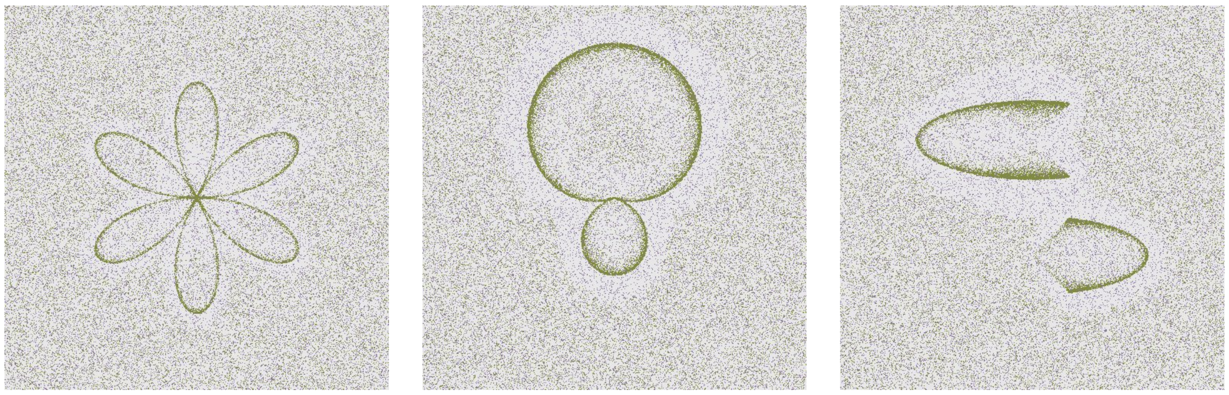

Figure 3: *Left: The 3-leaf rose* $r = a\sin(3\theta)$ *becomes a 6-leaf rose by letting* $r = |a\sin(3\theta)|$ *in order to expose negative radii. Here* $a = 150$. *Middle: The normally suppressed negative radius "inner" loop of the limaçon* $r = a + b\sin\theta$ *when* $|a/b| < 1$ *is seen reflected across the origin by letting* $r = |a + b\sin\theta|$. *Here* $a = 50$, $b = 150$. *Right: For the interval* $[\pi/4, 7\pi/4]$, *translated versions of the two segments of the ellipse* $x^2/a^2 + y^2/b^2 = 1$ *with* $a = 150$, $b = 50$ *are shown.*

Cartesisan Curves

To use curves of the form $y = f(x)$ as target curves it is necessary to modify the behavior of the automata so that when they pick up an item in a cell with cartesian coordinates (x_m, y_m) rather than trying to intersect with the target curve by returning to the local origin along a radial line, they intersect with it by following (in the proper direction) the vertical line $x = x_m$. That is, the drop criterion becomes $y_m = f(x_m)$. Note that automata are unable to drop when $f(x_m)$ is out of range. Using the standard trick from computer graphics of first rotating the automaton about the local origin by an angle λ, determining the direction vector and distance to the target curve, and then rotating the direction vector back using the angle $-\lambda$, we can make rotated cartesian curves target curves also. Figure 4 shows cartesian curve examples.

Algorithmic Designs

As an "application" of these ideas Figure 5 shows two algorithmic designs, each generated by using six scaled copies of two of the curves introduced above as target curves. The twelve different color grains were chosen using six shades of two colors in such a way that interesting interference patterns result.

References

[1] Greenfield, G., Stigmmetry prints from patterns of circles, *Bridges 2012 Conference Proceedings*, (eds. R. Bosch, D. McKenna and R. Sarhangi), 2012, 291–298.

[2] Urbano, P., The *T. albipennis* sand painting artists, Applications of Evolutionary Computation, Springer-Verlag Lecture Notes in Computer Science, LNCS 6625, 2011, 414–423.

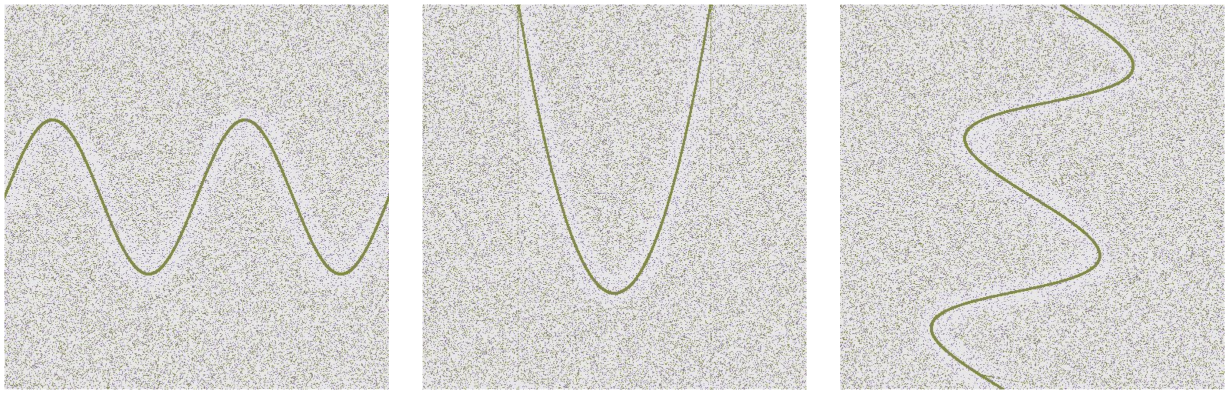

Figure 4: *Cartesian curves used as target curves. Left: The sine curve* $y = a\sin(bx)$ *with* $a = 100$*,* $b = \pi/125$*. Middle: The parabola* $y = ax^2 - b$ *with* $a = 3/125$*,* $b = 125$ *showing an out of range example. Right: The sine curve* $y = a\sin(bx)$ *with* $a = 100$*,* $b = \pi/125$ *rotated about the local origin by* $4\pi/9$ *radians.*

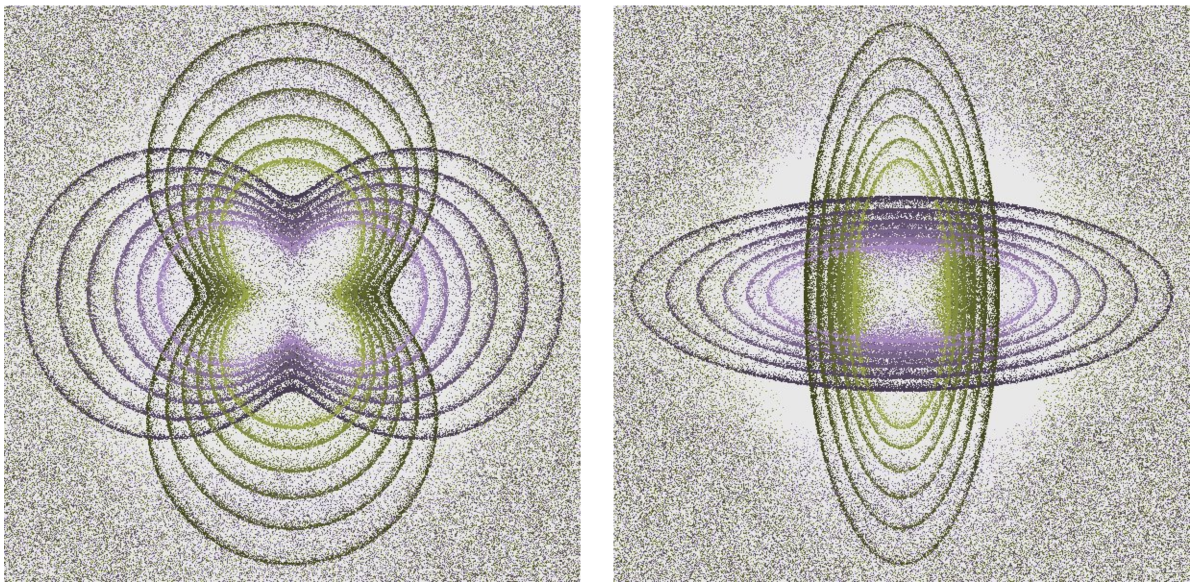

Figure 5 : *Two algorithmic designs. Six shades of two different colors are used as the virtual grain colors. Each of the twelve target curves has its own distinct grain color. The grid is* 600×600*, the grain density is* 0.4*, there are 6,000 automata, and the number of time steps is 300,000.*