Flowing, Organic Forms Using Adaptive Line-Drawing Agents

David Chappell
Department of Math, Physics and Computer Science
University of La Verne, La Verne, CA, USA
dchappell@laverne.edu

Abstract

We present a method of generating organic, 2D forms using a collection of adaptive line-drawing agents. The agents follow random-curvature walks when roaming in isolation. Each walk branches with a probabilistic rule. When an agent "sees" a neighboring path or an old segment of its own path, it attempts to follow it. The "following algorithm" generates rhythmic undulations as the agent pursues the path.

Introduction

Organic forms constructed from ensembles of line-drawing agents have been explored by Annunziato [1], McCormack [4], Greenfield [3], and many others. This form of mathematical art is often referred to as generative art because the resulting patterns and designs are generated from the interaction of many nonlinearly interacting agents. This approach mimics natural growth processes, often resulting in self-similar textures that fill the image space. The current study uses agent-based drawing methods to create self-bounded forms with organic, flowing curves. We explore ways in which the line-drawing agents can interact with each other to form rhythmic repeating designs.

The method presented in this paper is an extension of the line-drawing agents described in [2]. We use the same random-curvature walks and following rules, but omit the self-avoiding reversing turns presented in that work. We also allow the paths to branch at random intervals. When a line-drawing agent is unable to avoid a path and thus collides with it, it comes to a stop. When the last agent stops, the design is finished. The paths are rendered by laying down translucent feathered marks to one side of the path. Figures 1, 3, 4, 5 show examples of designs created using this method.

Random-Curvature Walks

In [2] we defined a random-curvature walk (RCW) as a path whose curvature undergoes a sequence of random fluctuations. Such curves smoothly transition between bends and have an natural, organic quality. Let s be the arc length along the path and θ be the tangential angle of the path relative to the x axis. The curvature κ experiences a sequence of random jumps according to $\kappa(s + \Delta s) = \kappa(s) + \kappa_0 X_i$, where X_i is a stochastic variable that returns $+1$ and -1 with equal probability and κ_0 is a free parameter that controls how tightly wound the curve is. Throughout this paper, the unit length is defined as the length of one step, i.e. $\Delta s = 1$. In order to generate statistically similar curves with different step sizes, κ_0 must scale with the step size Δs according to $\kappa_0 \propto \sqrt{\Delta s}$. Once the curvature $\kappa(s)$ is specified, the Cartesian coordinates of the curve may be found through integration:

$$\theta(s) = \int_0^s \kappa(t)dt \qquad x(s) = \int_0^s \cos\theta(t)dt \qquad y(s) = \int_0^s \sin\theta(t)dt.$$

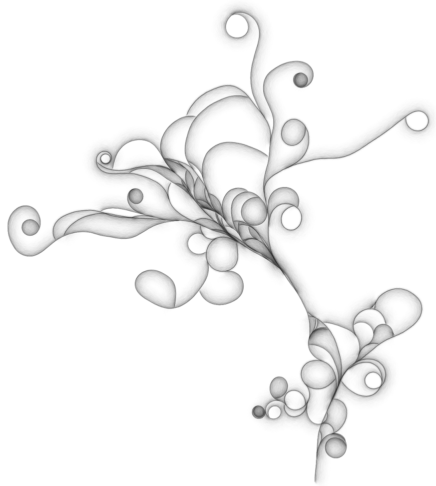

Figure 1: $P_B = 0.01$, $\kappa_0 = 0.003$, $\eta = 0$, $N_0 = 1$, $N_{max} = 150$.

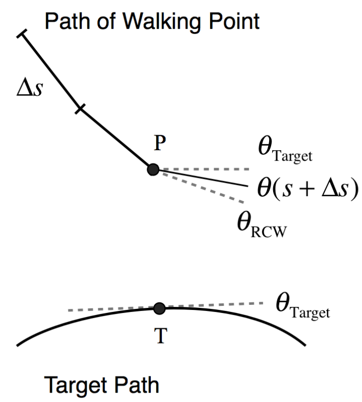

Figure 2: Illustration of the following rule (see text for details). An agent P encounters a path, the nearest point of which is marked T.

Following Rule

As a means of generating rhythmic variations between neighboring curves, we adopt the "following" method presented in [2]. When a walking agent wanders within a distance D of a path (called the "target path" in the following discussion), it will attempt to align its motion with the target path (see Figure 2). On each time step, the tangential angle of the agent's path will be a linear combination of the angle produced by a random curvature walk θ_{RCW} and the tangential angle of the target path θ_{Target} (see the third line of the update equations below). The parameter η controls how strongly the agent "snaps to" the target path: when $\eta = 0$ the walking agent executes a random curvature walk unaffected by the presence of target path and when $\eta = 1$ the agent's motion is determined solely by the target path.

$$\begin{aligned}
\kappa_{RCW} &= \kappa(s) + \kappa_0 X_i \\
\theta_{RCW} &= \theta(s) + \kappa_{RCW}\Delta s \\
\theta(s+\Delta s) &= (1-\eta)\theta_{RCW} + \eta\theta_{Target} \\
\kappa(s+\Delta s) &= \frac{\theta(s+\Delta s) - \theta(s)}{\Delta s}
\end{aligned}$$

The first two equations define the random curvature walk and the last equation is a simple first-order approximation to the curvature derivative.

When the target path is curved, the point along the curve that is closest to the walking agent is used in calculating θ_{target}. If multiple paths fall within the search distance D, the closest path is chosen. The walking agent is blind to the direction that the target path was originally traversed by its agent. In other words the agent attempts to follow the target path either "upstream" or "downstream", whichever direction is closest to the direction of motion of the walking agent. The search for neighboring paths is optimized using an indexed, linked-list as described in [2].

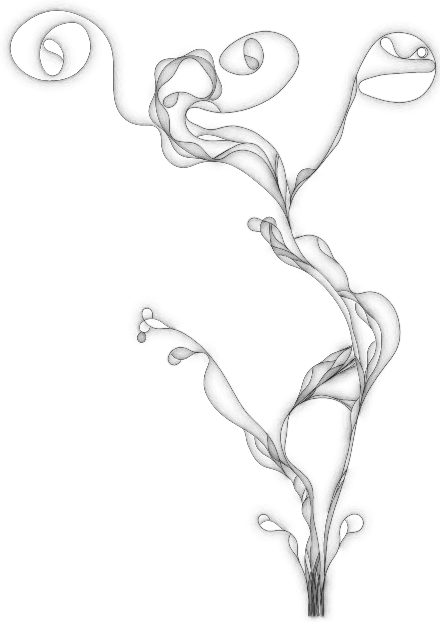

Figure 3: $P_B = 0.004$, $\kappa_0 = 0.0003$, $\eta = 0.001$, $N_0 = 13$, $N_{max} = 100$.

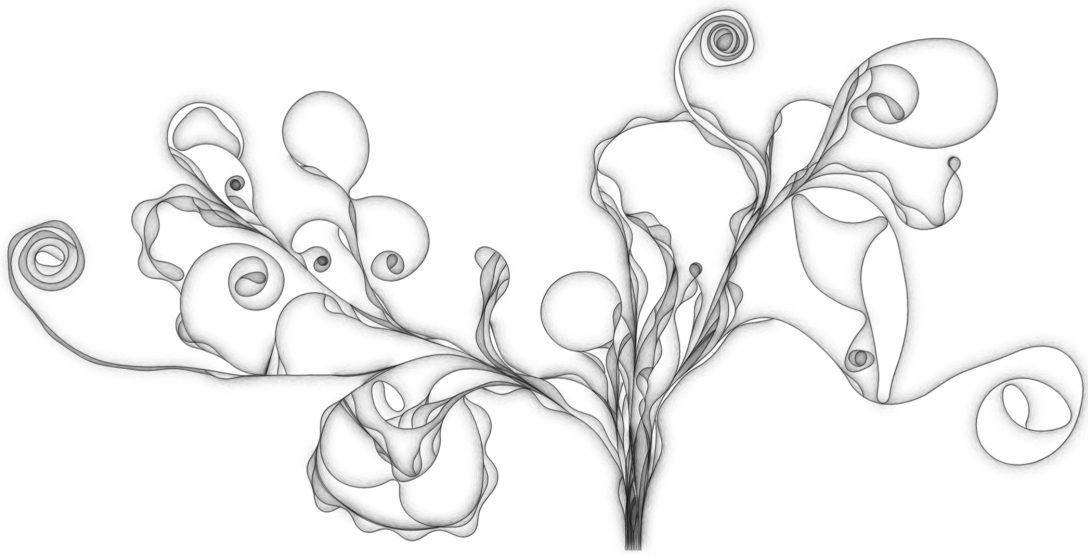

Figure 4: $P_B = 0.003$, $\kappa_0 = 0.002$, $\eta = 0.001$, $N_0 = 13$, $N_{max} = 200$.

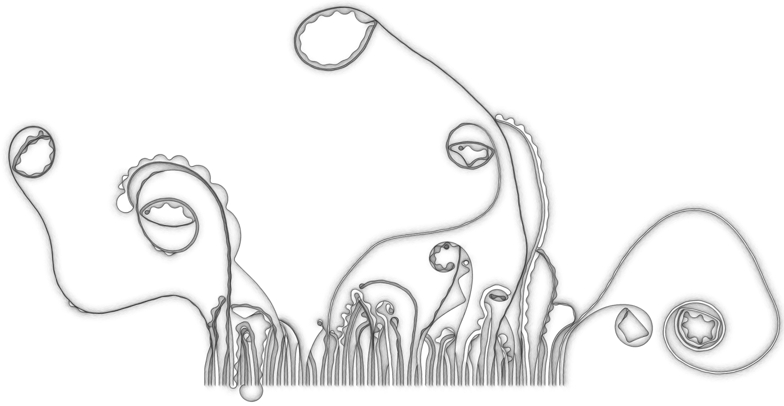

Figure 5: $P_B = 0$, $\kappa_0 = 0.00015$, $\eta = 0.002$, $N_0 = 100$, $N_{max} = 100$.

Branching

Each path branches with probability P_B per step Δs. Each branch is created with the opposite curvature of the parent path, i.e. $\kappa_{branch} = -\kappa(s)$. The tangential angle of the branch is equal to the tangential angle of the "parent" mark, i.e. $\theta_{branch} = \theta(s)$. When the branching probability is small, the design will run to completion after a finite number of branches. We find that when the branching probability is above a critical threshold, which depends on the curvature and coupling parameters, the number of branches will grow without limit. We choose to limit the total number of branches to a fixed value, N_{max}, as a way of creating bounded forms. Once the number of branches reaches this value, no additional branches are created. The design is finished once the existing paths are followed to completion.

In summary, the patterns presented in this paper are controlled by the following parameters:

Curvature parameter (κ_0) sets the length scale of the random curvature walk
Branching probability (P_B) is the probability that a given path will branch on a given step
Initial number of branches (N_0) is the initial number of line-drawing agents
Maximum number of branches (N_{max}) is the maximum number of branches in the design
Coupling strength (η) controls how strongly neighboring paths affect the motion of the line-drawing agent
Coupling distance (D) is the maximum distance that an agent "sees" when looking for neighboring paths.

References

[1] M. Annunziato, "The Nagual experiment", (1998) *Proceedings 1998 International Conference on Generative Art* (ed. C. Soddu), pp. 241-251.

[2] D. Chappell, "Taking a point for a walk: pattern formation with self-interacting curves", in Bridges 2014 Conference Proceedings. G. Greenfield, et al., eds., Tessellations Publishing, Phoenix, AZ, 2014, pp. 337-340.

[3] G. Greenfield, "On Evolving Multi-Pheromone Ant Paintings", in 2006 IEEE Congress on Evolutionary Computation, 2006, pp. 2072-2078.

[4] J. McCormack, "Creativity with Niche Construction," in *Proc. of the Alife XII Conference, Odense, Denmark, 2010*, (eds. J. McCormack and M. d'Inverno), Springer, Heidelberg, pp. 39-60.