Modeling Flocking Behaviour with the Boids Algorithm

Ben Wilfong

1 Introduction

Boids were first introduced in 1987 by Craig Reynolds in "Flocks, Herds, and Schools: A Distributed Behavioural Model" [2]. In his approach, the complex motion of flocking animals is modeled as an elaboration of a particle system [2]. A distributive behavioural model is implemented in which each boid chooses its own path according to its local perception of the entire flock [2]. In its simplest form, this local perception is based on only three simple rules: (1) separation, (2) alignment, and (3) Cohesion. Like Conway's Game of Life, fascinating and complex behaviours and patterns emerge from these three simple rules. In this report I will begin by taking the liberty of adding simulation boundaries and velocity limits to my baseline model. I will then add a limitation on the distance each boid can see and compare the cohesiveness of the resulting flocks.

2 Modeling

The simplest boids model initializes a number of boids at random positions with random velocities and applies the following three rules:

- *Separation:* Boids attempt to maintain a minimum distance from each other
- *Alignment:* Boids attempt to fly in the same direction
- *Cohesion:* Boids attempt to minimize the distance between themselves and other boids

subject to the settings

- *Separation Distance* (D_{sep})*:* Minimum desired distance between boids
- *Separation Strength* (S_{sep})*:* Scaling factor for the velocity corrections resulting from the separation rule
- *Alignment Strength* (S_{align})*:* Scaling factor for the velocity corrections resulting from the alignment rule
- *Cohesion Strength* (S_{coh})*:* Scaling factor for the velocity corrections resulting from the cohesion rule

- V_{max}: Maximum boid velocity

along with the obvious N, the number of boids to simulate, W, the width of the simulation box, and H, the height of the simulation box to update their positions. Algorithm 2.1 details how each boids position is updated based on the addition of velocities which are detailed in Algorithms 2.5 through 2.5 [1].

This rudimentary treatment of interactions yields surprisingly realistic flocking behaviour. The behaviour can be made even more realistic by adding a maximum vision distance (D_{vis}), that is, for each boid to only include boids within a certain radius when updating velocities. As one might expect, this results in flocks that break apart and join together at different times in the simulation.

Algorithm 2.1 Update Boid Positions

Initialize: Vectors $\vec{v}_{\text{sep}}, \vec{v}_{\text{align}}, \vec{v}_{\text{coh}}$
Given: $S_{\text{sep}},\ S_{\text{align}},\ S_{\text{coh}}$

for all Boids B_i **do**
  $\vec{v}_{\text{sep}}$ = calculateVseparation(B_i)
  $\vec{v}_{\text{align}}$ = calculateValignment(B_i)
  $\vec{v}_{\text{coh}}$ = calculateVcohesion(B_i)

  B_i.velocity = B_i.velocity + $S_{\text{sep}}\vec{v}_{\text{sep}} + S_{\text{align}}\vec{v}_{\text{align}} + S_{\text{coh}}\vec{v}_{\text{coh}}$
  B_i = boundPosition(B_i)
  B_i.velocity = limitVelocity(B_i.velocity)

  B_i.position = B_i.position + $\Delta t \text{B}_i$.velocity
end for

Algorithm 2.2 calculateVseparation(B_i)

Initialize: Scalars $d,\ D_{\text{sep}}$
Initialize: Vector v_{sep}

for all Boids B_j **do**
  $d = ||\text{B}_i$.position - B_j.position$||$
  if $d < D_{\text{sep}}$ **then**
    $v_{\text{sep}} = v_{\text{sep}} + d$
  end if
end for

return v_{sep}

In order to quantify the difference in flocking dynamics for different separation, alignment, and cohesion strengths and maximum vision distances, I define the

Algorithm 2.3 calculateValignment(B_i)

Initialize: Vectors $v_{\text{align}}, v_{\text{flock}}$
Given: N

```
for all Boids B_j do
    if i ≠ j then
        v_flock = v_flock + B_j.velocity
    end if
end for
v_flock = v_flock/(N − 1)
v_align = v_flock − B_i.velocity

return v_align/||v_align||
```

Algorithm 2.4 calculateVcohesion(B_i)

Initialize: Vectors $p_{\text{flock}},\ v_{\text{coh}}$
Given: N

```
for all Boids B_j do
    if i ≠ j then
        p_flock = p_flock + B_j.position
    end if
end for
p_flock = p_flock/(N − 1)
v_coh = p_flock − B_i.position

return: v_coh/||v_coh||
```

Algorithm 2.5 limitVelocity(B_i.velocity)

Given: $V_{\max}$

```
if ||B_i.velocity|| > V_max then
    B_i.velocity = V_max(B_i.velocity/||B_i.velocity||)
end if
```

Algorithm 2.6 boundPosition(B_i)

Initialize: Scalars $d_{\text{threshold}},\ v_{\text{correction}}$
Given: $W,\ H$

$p_x = \mathrm{B}_i.\text{position}_x$
$p_y = \mathrm{B}_i.\text{position}_y$

if $p_x < d_{\text{threshold}}$ **then**
  $\mathrm{B}_i.\text{velocity}_x = \mathrm{B}_i.\text{velocity}_x + v_{\text{correction}}$
else if $p_x > W - d_{\text{threshold}}$ **then**
  $\mathrm{B}_i.\text{velocity}_x = \mathrm{B}_i.\text{velocity}_x - v_{\text{correction}}$
end if
if $p_y < d_{\text{threshold}}$ **then**
  $\mathrm{B}_i.\text{velocity}_y = \mathrm{B}_i.\text{velocity}_y + v_{\text{correction}}$
else if $p_y > H - d_{\text{threshold}}$ **then**
  $\mathrm{B}_i.\text{velocity}_y = \mathrm{B}_i.\text{velocity}_y - v_{\text{correction}}$
end if

Table 1: Baseline Parameters

N	W	H	D_{sep}	S_{sep}	S_{align}	S_{coh}	$V_{\max}$	D_{vis}
100	200	200	1	4	0.04	0.075	10	∞

mean x-position of the flock, $\bar{P}_x$, and the mean y-position of the flock, $\bar{P}_y$, as

$$\bar{P}_x = \frac{1}{N}\sum_{\text{Boids}} P_x \quad \text{and } \bar{P}_y = \frac{1}{N}\sum_{\text{Boids}} P_y$$

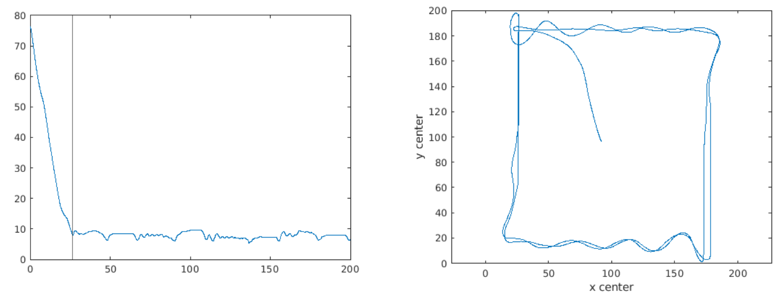
respectively. A flock diameter $\bar{D}$ is then defined as

$$\bar{D} = \frac{1}{N}\sum_{\text{Boids}} ||\langle \bar{P}_x, \bar{P}_y\rangle - \langle P_x, P_y\rangle||$$

where $\langle\ \rangle$ denotes a two component vector. When a flock of boids separates due to a maximum view distance, the quantity $\bar{D}$ will increase, and then decrease when these sub-flocks join back together.

3 Results

Finding a baseline set of parameters that yielded natural flock like behavior was nontrivial. Reasonable parameters were determined by adding rules to the model one at a time and tweaking the parameters slightly until a baseline could be achieved. This process yielded the set of baseline parameters in Table 1. An animation of this flock is available on YouTube[1]. Once the initial equili-



(a) Average Flock Diameter

(b) Average Flock Path

Figure 1: Average Flock Parameters for the Baseline Model

bration from the random initialization of each boid, the emergent behaviour is surprisingly natural. The average path of the flock defined by $\langle \bar{P}_x, \bar{P}_y \rangle$ is shown in Figure 1b and the flock diameter as a function of time is shown in Figure 1a. Because the boids are able to see each other from infinitely far away, once they equilibrate into a flock at $t = 26.55$, they stay together as a flock for the remainder of the simulation.

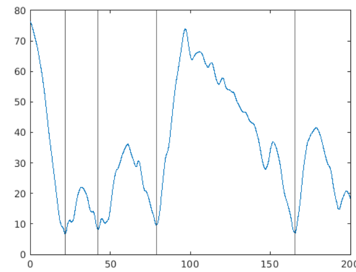
Limiting D_{vis} to 10 yields a different trend in average flock diameters versus time shown in Figures 2. Without visualizing the data, it is reasonable to guess that the up and down trends in average flock diameter suggest the breaking apart of and joining together of a main flock. The local minimums at $T = 21.8$, 42.4, 78.75, and 165.55 are good estimates of when these events occur. Some of this behaviour is seen in the animation available on YouTube, but the boids appear to get stuck on the left boundary in an unnatural way [2]. It is also interesting to note that the average flock center remains relatively close to the bottom of the simulation box once equilibration occurs.

If we instead increase the vision distance to 25, and the alignment strength to 0.06, we get the average path and diameter as shown in Figure 3. In these case we doo get the coming together at $T = 33.3$ and breaking apart at $T = 81.8$ as suggested by Figure 3a and shown in the animation on YouTube [3]. Unfortunately, some of the boids do get stuck along the bottom and left walls once they separate from the flock, but some breaking apart and joining together behaviour is seen.

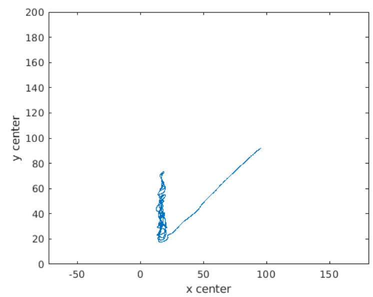
4 Conclusion

Despite its simple rule-set, Craig Reynolds' Boids Algorithm generates complex, natural looking emergent behaviour. The addition of a maximum sight

[1] `https://youtu.be/79HfNRK3_TU`
[2] https://youtu.be/SDfxQrzPY5A
[3] https://youtu.be/qkSqA5eVwZA

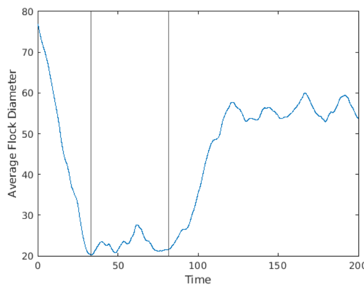


(a) Average Flock Diameter

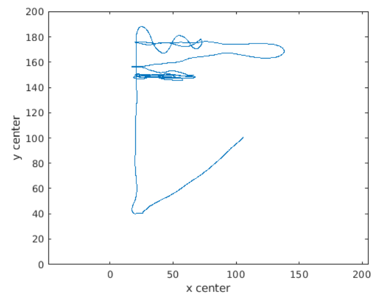


(b) Average Flock Path

Figure 2: Average Flock Parameters for Model with $D_{\text{vis}} = 10$



(a) Average Flock Diameter



(b) Average Flock Path

Figure 3: Average Flock Parameters for Model with $D_{\text{vis}} = 25$ and $S_{\text{align}} = 0.06$

distance further increases the realistic nature of the simulation, however more careful treatment of the simulation boundary is required to prevent boids from getting stuck at the simulation boundary when they separate from the main flock. Regardless, the results strongly suggest that a similar rules set might be subconsciously used by living organisms that form flocks, herds, and schools.

References

[1] Conrad Parker. *Boids Pseudocode*. en. URL: https://vergenet.net/~conrad/boids/pseudocode.html.

[2] Craig W. Reynolds. “Flocks, herds and schools: A distributed behavioral model”. In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987).