Self-Assembling Swarm Microrobots

Saatvik Sumanta* Bhavya Mishra†

1 Introduction

The simulation mimics the behavior of "boids," a model originally developed to simulate flocking behavior of birds. In this project, we extend the concept of flocking to simulate "self-assembling cubes" where small individual units (boids) behave based on three core principles: alignment, cohesion, and separation. The simulation visualizes how these boids can form primitive clusters or groups, which may resemble "cubes" or other shapes when they coalesce. Additionally, user-drawn boundary shapes can be followed by the boids, enhancing the simulation's interactivity and visualization.

2 Simulation Overview

- **Boids**: The individual units (or particles) that move based on velocity and acceleration.
- **Groups**: Clusters of boids that interact with each other to form structures.
- **Rules**: The movement of each boid is governed by three fundamental behaviors:
  - **Alignment**: Boids align their velocity with nearby neighbors.
  - **Cohesion**: Boids move toward the average position of nearby boids (center of mass).
  - **Separation**: Boids avoid crowding each other by maintaining a minimum distance.
- **Boundary Following**: Boids can follow boundary shapes drawn by the user. The simulation has three states: IDLE, DRAWING, and FOLLOWING, to manage boid interactions with user-defined shapes.

3 Mathematical Model

3.1 Boid Movement and Kinematics

Each boid is represented as a point with a position vector $\mathbf{p}$, a velocity vector $\mathbf{v}$, and an acceleration vector $\mathbf{a}$. These vectors are updated over time according to simple Newtonian physics:

$$\mathbf{p}_{\text{new}} = \mathbf{p}_{\text{current}} + \mathbf{v}_{\text{current}}$$
$$\mathbf{v}_{\text{new}} = \mathbf{v}_{\text{current}} + \mathbf{a}_{\text{current}}$$

The acceleration is computed based on the forces from flocking behaviors (alignment, cohesion, separation) as well as any boundary-following force if assigned.

3.2 Forces Governing Behavior

1. **Alignment**: The alignment force encourages each boid to match its velocity with the average velocity of nearby boids.

$$\text{Alignment Force} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{v}_i - \mathbf{v}_{\text{current}})$$

*Vellore Institute of Technology, Chennai. B.Tech. Computer Science and Engineering with Specialisation in Artificial Intelligence and Robotics, Class of 2025. Email: saatvik.sumanta02@gmail.com

†Vellore Institute of Technology, Chennai. B.Tech. Computer Science and Engineering with Specialisation in Artificial Intelligence and Robotics, Class of 2025. Email: bhavyamishra223@gmail.com

2. **Cohesion**: The cohesion force pulls each boid towards the center of mass of nearby boids.

$$\text{Center of Mass} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i$$

$$\text{Cohesion Force} = \text{limit}(\text{COM} - \mathbf{p}_{\text{current}})$$

3. **Separation**: The separation force ensures that boids maintain a minimum separation from their neighbors.

$$\text{Separation Force} = \sum_{i=1}^{N}\frac{\mathbf{p}_{\text{current}} - \mathbf{p}_i}{|\mathbf{p}_{\text{current}} - \mathbf{p}_i|}$$

3.3 Limiting Forces and Velocities

To keep the simulation realistic, a limit() function ensures that velocity and force vectors do not exceed a maximum value, preventing excessive speeds or forces.

$$\text{limit}(\mathbf{v}, v_{\max}) = \begin{cases} \frac{\mathbf{v}}{|\mathbf{v}|} \times v_{\max} & \text{if } |\mathbf{v}| > v_{\max} \\ \mathbf{v} & \text{otherwise} \end{cases}$$

4 Group Formation and Self-Assembly

Boids exhibit self-assembly by forming groups based on proximity. Each boid is assigned a group ID based on its distance from other boids, allowing dynamic cluster formation. In addition, boids can follow a user-defined boundary shape, adding another layer of self-organization as they move towards boundary points.

- **Group Tracking**: Each boid is assigned a group ID based on its proximity to other boids.
- **Boundary Following**: The `BoundaryController` class generates smooth boundary points from user-drawn shapes, and boids move toward their assigned boundary points.
- **Center of Mass Calculation**: For each group, the center of mass is calculated, pulling boids in smaller groups toward this center.
- **Visualization**: Groups with five or more boids display a highlighted center of mass, representing a "self-assembled" group.

4.1 Locating Empty Positions within User-Defined Boundaries

The boids are designed to move toward boundary points defined by the user, filling in a specified shape. To identify an appropriate empty position along the boundary, each boid is assigned to a boundary point based on proximity. This assignment ensures that each boid aligns with a nearby boundary point, creating a seamless filling of the user-defined boundary.

4.1.1 Mathematical Approach

Let $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M\}$ represent the set of all boundary points generated from the user's drawing. For each boid i, located at position $\mathbf{b}_i$, we calculate the nearest boundary point by solving:

$$\mathbf{p}_i = \arg\min_{\mathbf{p}\in\mathbf{P}} \|\mathbf{b}_i - \mathbf{p}\|$$

where $\|\cdot\|$ denotes the Euclidean distance. This selection minimizes the distance between the boid and boundary points, aligning each boid with the nearest unoccupied point.

Once a boid is assigned to a point $\mathbf{p}_i$, it calculates a desired vector $\mathbf{d}_i$ pointing toward this boundary point:

$$\mathbf{d}_i = \mathbf{p}_i - \mathbf{b}_i$$

The boid then applies a force in the direction of $\mathbf{d}_i$, constrained by the maximum speed and force limits:

$$\mathbf{f}_i = \text{limit}\left(\frac{\mathbf{d}_i}{\|\mathbf{d}_i\|} \cdot v_{\max}, f_{\max}\right)$$

where $v_{\max}$ is the maximum velocity and $f_{\max}$ is the maximum allowable force. The application of this force enables the boid to reach and settle at its assigned boundary position, thereby filling the shape with minimal overlap among boids.

This approach is repeated for each boid, and boundary points that are already assigned are marked to avoid reassignment. This iterative allocation ensures that the boids uniformly fill the user-defined boundary shape.

5 Zoom Functionality

The simulation includes a zoom function controlled by mouse scroll events. This allows the size of boids and their positions to scale, maintaining relative positioning during zoom actions.

$$\text{Scaled Position} = (\text{Position} - \text{Screen Center}) \times \text{Zoom Factor} + \text{Screen Center}$$

6 Simulation Observations

The boids display flocking behavior based on alignment, cohesion, and separation, forming natural groupings over time. Boids can dynamically adapt to follow user-drawn shapes, enhancing the emergent self-assembly behavior and simulating more complex, cube-like structures.

7 Conclusion

This simulation demonstrates that simple rules based on alignment, cohesion, and separation can produce complex, emergent behavior. The ability of boids to adapt to user-defined boundaries showcases their potential in applications requiring distributed organization in modular robots.