Bio-Inspired Swarm Robotics Simulation Using Python

Final Report

AINT 44052

Intelligent Autonomous Robotics

H.M. Sudith Amarasinghe

CS/2018/003

Contents

1. Abstract

This project presents a bio-inspired simulation of flocking behavior in a swarm of robots, implemented in Python using Pygame. The simulation mimics the natural flocking behavior observed in birds, where each robot (boid) follows three simple rules: alignment, cohesion, and separation. The goal is to demonstrate how simple local interactions can lead to complex global behavior in a swarm of autonomous agents.

2. Introduction

Swarm robotics is an area of multi-robot systems that focuses on the collective behavior of simple robots interacting with each other and their environment. Inspired by biological systems, such as bird flocks and fish schools, swarm robotics aims to design decentralized control systems where each robot follows simple rules, leading to emergent collective behavior.

This project simulates the flocking behavior of boids (biologically inspired robots) using Python and Pygame. The simulation demonstrates how alignment, cohesion, and separation rules enable a swarm of robots to move cohesively as a group while avoiding collisions.

3. Objectives

- To simulate the flocking behavior of a swarm of robots using bio-inspired algorithms.
- To implement the simulation in Python using Pygame for visualization.
- To demonstrate the emergent behavior resulting from simple local interactions among robots.

4. Methodology

The project is divided into several stages, including the setup of initial conditions, implementation of boid behaviors, and visualization of the simulation. The following sections describe the details of each stage.

4.1 Initial Setup

- Configuration File (config.json): Defines the simulation parameters such as the number of boids, screen dimensions, maximum speed, maximum force, and perception radius.
- Environment File (environment.json): Specifies the obstacles present in the environment.
- Initial Conditions File (initial_conditions.csv): Contains the initial positions and velocities of the boids.

4.2 Boid Class Implementation

The Boid class defines the properties and behaviors of each boid. Key methods include:

- align(): Calculates the steering force to align the boid's velocity with the average velocity of neighboring boids.
- cohesion(): Calculates the steering force to move the boid towards the average position of neighboring boids.
- separation(): Calculates the steering force to avoid collisions with neighboring boids.
- update(): Updates the boid's position and velocity based on the accumulated forces.
- edges(): Wraps the boid around the screen edges if it goes out of bounds.
- show(): Renders the boid on the screen.

4.3 Simulation Script

The simulation.py script initializes the Pygame environment, loads the configuration and initial conditions, creates the boids, and runs the main simulation loop. Within the loop, each boid's behavior is updated, and the boids are rendered on the screen. The script also handles user input for quitting the simulation.

5. Results

The simulation successfully demonstrates flocking behavior. The boids move cohesively as a group, maintaining alignment, cohesion, and separation. The presence of obstacles in the environment shows the boids' ability to navigate around them while maintaining their flocking behavior.

The simulation produces visual animations that illustrate the emergent behavior resulting from simple local interactions. The boids' movement patterns resemble natural flocking behavior observed in biological systems.

6. Discussion

The project demonstrates the effectiveness of bio-inspired algorithms in achieving complex collective behavior through simple rules. The alignment, cohesion, and separation rules enable the boids to move cohesively while avoiding collisions. The simulation highlights the potential of swarm robotics in applications where decentralized control and collective behavior are essential.

Challenges encountered during the project included ensuring numerical stability and handling edge cases where the boids' positions or velocities became invalid. These issues were addressed by adding checks and resetting invalid values.

7. Conclusion

This project successfully simulates bio-inspired flocking behavior in a swarm of robots using Python and Pygame. The emergent behavior resulting from simple local interactions demonstrates the potential of swarm robotics in achieving complex collective behavior. The project provides a foundation for further exploration of bio-inspired algorithms and their applications in robotics.

References

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. ACM SIGGRAPH Computer Graphics, 21(4), 25-34.

Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. Swarm Intelligence, 7(1), 1-41.