COP290

Starling Simulation(Design Document)

Yash Raj Gupta & Parth Shah

1. Abstract

Aerial displays of starlings (Sturnus vulgaris) at their communal roosts are complex: thousands of individuals form multiple flocks which are continually changing shape and density, while splitting and merging. To understand these complex displays both empirical data and models are needed. Whereas detailed empirical data were recently collected through video recordings and position measurements by stereo photography of flocks of thousands of starlings, there are as yet no models that generate these complex patterns. Numerous computer models in biology, however, suggest that patterns of single groups of moving animals may emerge by self-organisation from movement and local coordination (through attraction, alignment and avoidance of collision). In this paper, we investigated whether this approach can be extended to generate patterns resembling these aerial displays of starlings. We show in a model that to generate many of the patterns measured empirically in real starlings we have to extend the usual rules of local coordination with specifics of starling behaviour, mainly 1) their aerial locomotion, 2) a low and constant number of interaction-partners and 3) preferential movement above a 'roosting area'. Our model can be used as

a tool for the study of these displays, because it provides new integrative hypotheses about the mechanisms underlying these displays and of swarming patterns in biological systems in general.

2. Introduction

Moving animal groups (including those of humans) often exhibit complex patterns of coordination. One of the most complex patterns is shown by flocks of tens of thousands of European Starlings (Sturnus vulgaris) over the roost before nesting at dusk. Because the coordination during their wheeling and turning, splitting and merging and the changes of flock shape and density are truly amazing, this performance has been attributed to 'thought transference' (Selous, 1931). To understand how these patterns are generated is not only important in itself, but also for obtaining insight how starlings –despite their large numbers- are able to prevent collisions even when escaping a predator (Carere et al., 2009; Feare, 1984). This may bear interest in relation to traffic control.

In short we developed a model of the self-organisation of starling-like displays at the roost. We will show that patterns of our model resemble starling-like displays qualitatively in their shape, splitting and merging, movement trajectories, positioning of individuals during turning, dynamics over time.

3. Methodology

Our model will follow the methodology laid by Craig Reynolds who was the first person to simulate this behavior on a computer, which has earned him the unofficial title as the father of flocking. His first simulation took place back in 1986 and the ingenious basic structure he used still holds strong to this day. Craig proposed that flocking was an emergent behavior, that is each member or agent of a flock followed simple rules in order to create this group behavior. Craig also gave an interesting name for each member of the flock, he called them boids.

3.1 3D Rendering

To draw a boid, use a series of triangles. Use one to draw a body and another two to create wings to flap. We use camera and lighting to help in the 3D rendering.

3.2 The Rules

3.2.1 Separation, Cohesion and Alignment

Description

Separation:- This rule helps avoid collision of neighbouring boids by keeping a check on the nearest distance between them.

Cohesion:- It helps keep the neighbouring boids together and helps integrate stary boids into a flock.

Alignment:- It helps keep the boids in a flock aligned in a direction.

Implementation

Birds collective flocking behaviour is taking into account by separation, cohesion and alignment. These three act as forces to accomplish it. Firstly we create a function that takes an agent and return force on it. This function internally calls the individual functions and saves its values in variables

```
ali = alignment ( bl ) ;
coh = cohesion ( bl ) ;
sep = separation ( bl ) ;
```

Finally after each frame, the variables scaled by corresponding factor is added to acceleration vector of a boid.

```
acc += a l i* f 1
acc += coh*f 2
acc += sep*f 3
```

Cohesion works on following guidelines:-

```
Function {
  Vector AvgPos
  Vector f o r c e
  for (Boid b : boids )
    {
      d = pos  b . pos ) ;
      i f (0<d<=neighborhoodRadius )
            {
                  posSum. add (b . pos ) ;
                  Count++;
            }
   }
i f ( count>0)
{
      posSum. div ( ( float) count) ;
}
steer = PVector.sub (posSum , pos ) ;
steer.limit(maxSteerForce ) ;
return posSum;
}
```

Alignment and Separation work similarly.

3.3 Functional Specifications

3.3.1 Real time control on the degree of cohesion, separation and alignment:-

The Videopane has sliding tools on the bottom left portion of the window where you could change the degrees of cohesion, separation and alignment and observe their effects.

3.3.2 Addition and Deletion of Boids:-

The number of boids present can also be changed in real time. Pressing the '=' button adds 50 boids while pressing the '-' button deletes random boids from the screen.

3.3.3 Reversal of Motion:-

The direction of motion of the boids can also be reversed. This has been implemented by multiplying the velocity and the acceleration of the boid by -1.

3.3.4 Zoom in/out:-

We can zoom in/out of the screen using the down/up arrow keys respectively.

We can also monitor the effects of changing the various parameters on the kinetic energy of the flock the value of which will be printed on the screen.