Flock Inspired Area Coverage Using Wireless Boid-like Sensor Agents

Colin Chibaya  
c.chibaya@ru.ac.za  
Rhodes University  
Computer Science Department  
Grahamstown, South Africa

Shaun Bangay  
s.bangay@ru.ac.za  
Rhodes University  
Computer Science Department  
Grahamstown, South Africa
Background

Wireless Sensor Networks

- Beacon/landmark based
  - Require memory
    - Movement history
    - Positional info (beacon)

- Geometry
  - Require complex agents with powerful computational capabilities
Background

Reynolds’ simulated flocks

Separation
- Move to avoid overcrowding

Cohesion
- Attempt to stay close to neighbours

Alignment
- Move towards the average direction of flock mates
The goal

Flock inspired area coverage model

- Wireless sensor networks
- Reynolds’ flocking rules
- Perching
- Beacons, geometry, landmarks
- Alignment

• Background
• The goal
• Movie
• Control routine
• Experiment
• Results
• Conclusion
• Contributions
Exemplary scenario

- Free neighbouring locations
- Free locations in sensing range
- The mean free path
Control routine-1

Mode separation

\textbf{FOR\_EVERY} agent $X$ at location $L_i$

\textbf{IF} (cardinality of $L_i > 1$) \textbf{THEN}

\textbf{IF} (exist free neighbouring spaces ($L^*$)) \textbf{THEN}

$X$ move to $L^*$

\textbf{ELSE IF} (free spaces, $L^{**}$ are in sensing range) \textbf{THEN}

$X$ move to $L^*$ that is closest to $L^{**}$

\textbf{ELSE}

$X$ uses the mean free path

\textbf{ELSE}

Mode cohesion
Control routine-2

FOR_EVERY agent X at location $L_i$

IF (exists a neighbouring location $L^*$ whose cardinality $> 0$) THEN

mode $\leftarrow$ perch

ELSE IF (exist some covered locations in sensing range) THEN

X moves to $L^*$ closest to the covered locations

ELSE

X uses the mean free path
Experiment 1: Separation speed

- How we measure separation speed
  - number of agents that are successfully perched within a specific time in simulation

- Centrally placed values
  - 10 simulations are averaged

- Results
  - Compared against a random guess.
Result 1: Separation speed

Fig 1: comparison of separation speed

Findings

- 50.57% of agents were perched in 31 iterations, compared to only 19.14% using a random guess.

- Our model achieved complete coverage in 135 iterations when a random guessing model was at 65.66%.
Experiment 2: Cohesion speed

- **How we determine cohesion speed**
  - Count iterations from isolation until cohesion

- **Procedure**
  - Allow coverage to occur in some continuous space
  - Deploy an isolated agent
  - Record the iterations
  - Repeat for 1000 times

- **Results**
  - Compare with random guessing
Result 2: Cohesion speed

Table 1

<table>
<thead>
<tr>
<th></th>
<th>Model</th>
<th>Guess</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean steps</td>
<td>16</td>
<td>39</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>Entropy levels</td>
<td>0.5%</td>
<td>41.2%</td>
</tr>
</tbody>
</table>

Findings

- Agents achieved cohesion in 16 ±3 steps, compared to 39 ±14 steps using random guessing.
- Chances that agents fail to perch are 0.5% in our model and 42.2% in random guess model.
Experiment 3: Coverage quality

• Metrics of importance
  – Fraction of area covered at any time slot
  – Time it takes to achieve complete coverage
    • Measured in iterations

• Results
  – Compared with results achieved
Result 3: Coverage quality

Findings

- Our model achieved complete coverage in 135 iterations
- Random guess achieved a maximum of 65.66%
Experiment 4: Fault tolerance

• Purpose of experiment
  – Model performance where agents may fail

• How we conducted the experiment
  – Allow coverage to occur
  – Kill 40 agents

• Results
  – Compare with a random guess
Result 4: Fault tolerance

- Our model self-repaired to 94.35% coverage quality in 34 iterations.
- Agents that used the random guessing model could not re-organize.
Conclusions

• We proposed an area coverage model inspired by Reynolds’ flocking algorithms.

• The model exhibits good separation speed and cohesion properties of the flocking algorithm

• The model is fault tolerant and adaptive to agents’ failures

• The model is fast, achieving high quality coverage in a relatively short period of time
Contributions

• We presented a novel sensor agents control model using simulated flocking rules

• We devised and evaluated a plausible strategy for determining coverage quality as well as fault tolerance

• This work provides a new way of measuring the performance of agent based coverage models