Flock inspired Area Coverage using Wireless Boid-like Sensor Agents

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

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

Abstract

Simulated flocking is achievable using three boid rules[13]. *We propose an area coverage model inspired by Reynolds' flocking algorithm, investigating strategies for achieving quality coverage using flocking rules. Our agents are identical and autonomous, using only local sensory information for indirect communication.*

Upon deployment, agents are in the default separation mode. The cohesion rule would then guarantee that agents remain within the swarm, covering spaces with explored neighbour spaces.

Four experiments are conducted to evaluate our model in terms of coverage quality achieved. We firstly investigate agents' separation speed before the speed with which isolated agents re-organizes is investigated. The third experiment compares coverage quality achieved using our model with coverage quality achieved using random guessing. Finally, we investigate fault tolerance in the event of agents' failures.

Our model exhibits good separation and cohesion speed, achieving high quality coverage. Additionally, the model is fault tolerant and adaptive to agents' failures.

1. Introduction

The motion of flocking birds or schooling fish is a compelling nature's delight. While the discrete members of the flock are very simply, autonomous and limited in capabilities, the aggregated motion is magnificently coherent. The flocking behaviour is emergent property of the collective effects of individual members' autonomous activities.

On the other hand, the rapid progress in wireless communication technologies have attracted a great deal of research attention due to the wide range of potential applications, most importantly, expanding people's ability to remotely interact with the physical world. In a broad sense, wireless sensor networks can transform the way we manage our homes, businesses, environment, and health.

In this work, we consider a more general area coverage model inspired by Reynolds' simulated flocks [13] in which it is indicated that, if each virtual bird followed only three rules: separation, alignment and cohesion, a realistic simulation of a flock is achievable. We investigate collections of agents modeled on flocking birds in nature, investigating a strategy for efficiently achieving quality area coverage as the resultant emergent behaviour. Our agents are all simple, small, identical, autonomous, inexpensive, and can only use local sensory information for indirect communication.

Contrary to the common distributed deployment models [2,3,4,5,6,9,12,14,16], our agents are deployed centrally mainly because it may be the most feasible option for real world application of wireless sensor networks. In addition, we do not make use of pre-specified beacons and/or landmarks nor directional information to guide agents' movements. Instead, two of Reynolds' simulated flocking rules, together with "*perching*" are sufficient. These two rules are: (a) separation; with which agents explore the region to be covered and (b) cohesion; which guarantees that agents remain within the swarm covering spaces close to explored spaces. This model will provide useful guidelines for agent based simulated design.

In the remainder of the paper, we discuss related work in section 2. The control algorithms we use are presented in section 3, emphasizing on how separation, cohesion and perching are

achieved. An in-depth experimental setup follows in section 4 and the results achieved thereafter are presented in section 5. We conclude the paper in section 6 highlighting the contributions and future directions of our work.

2. Related work

The basis for our area coverage model is two fold: (a) various wireless sensor network models suggested in literature [1,2,3,4,9,11,12,15] and (b) Reynolds' simulated flocking algorithm [13]. Commonly, agents are all simple, identical, autonomous, and can only use local sensory information for indirect communication.

Area coverage models where some agents serve as beacons/landmarks are common [3,5,7, 8,12,14,15,16]. In these, agents would use the positional properties of beacons/landmarks to establish their own coordinate system relative to the world coordinate system [3,12,16]. In other cases, beacon agents are enriched with *a prior* knowledge of the geometry of the area to be covered, influencing the movement behaviour of the rest [5, 15]. However, beacon/landmark controlled models would require agents with localization capabilities as well as memory for recording previous movement history.

Similarly, coverage models where agents are capable of performing geometric computations are also popular [6, 8, 9]. In these, an agent may establish its relative position using circle geometry [6], Voronoi diagrams, Delaunay tessellation or other triangulation techniques [9]. However, these models would require structurally complex agents with powerful geometric processing capabilities. On the other hand, agents with directional clues may also be useful, with best examples seen in the work of [2,11,12].

Our flock inspired model differs in that we neither use beacons, landmarks, leader agents, directionality nor any *a prior* knowledge of the geometry of the area to be covered. Instead, we build from Reynolds' flocking algorithm, replacing the alignment rule with perching. Existing flocking models also differ in that the simulated birds generally flock in the same direction, at varying speed, and avoiding agents' collisions. Our model works on a 2D rectangular grid similar to cellular automata. However, agents would not change their states and behavioral tendencies after every step as suggested in cellular automation models [10].

3. Agents' control routine

The "*simulator*" we implemented assumes a model that is motivated by flocking birds in nature. Agents move on a discrete rectangular grid executing the same routine in every step. Variation of agents' movement speed was not an experiment variable in this study.

The routine that controls agents' placement is as shown in Algorithm 1. The key idea is this: every agent is deployed centrally in the default separation mode. These agents are all structurally similar, with a sensing radius of two grid cells, movement and coverage radii of one grid cell. Agents explore the region avoiding spaces that are already covered. Each agent has at most eight grid cells within its movement and coverage radii. Meanwhile, agents can sense the presence of other agents as far as two grid cells away.

Firstly, an agent A_i at grid cell L determines the cardinality ($\|L\|$) of that grid cell. If $\|L\|>1$, that space is already covered and A_i has to move away. To achieve this, A_i determines all free grid cells within its movement/coverage radius. If there are such free spaces, A_i randomly relocates to one of these. However, if all spaces in the movement and coverage radii are covered, the agent determines free spaces in its sensing radius. In the event of such free spaces existing, a location within the movement/coverage radius

Algorithm 1: Control algorithm for agents placement

```
mode ← separation
while (true)
    direction ← none
    foreach agent at grid cell L (A_L)
        if ( ||L|| > 1)
            if ( ∃L*: ||L*|| == 0)
                A_L ← A_L*
            else if ( ∃L**: ||L**|| == 0)
                A_L ← A_L* : /L* - L**/ ≤ ε
            else
                A_L ← A_L* (mean free path)
        else
            mode ← cohesion
            while (true)
            foreach agent at L (A_L)
                if ( ∃L*: ||L*|| > 0)
                    A_L ← A_L (perch)
                else if ( ∃L**: ||L**|| > 0)
                    A_L ← A_L*: /L* - L**/ ≤ ε
                else
                    A_L ← A_L*-mean free path
    mode ← perching
```

that is closest to the identified free space in the sensing radius is chosen as the next destination with the hope that the agent would relocate to that free space in its next movement step. The agent's dilemma arises when all spaces both in the movement, coverage and sensing radii are covered. In this case, a "*mean free path*" is invented where agents make blind steps in any direction with the hope that the destination cell would be close to some free space.

If $\|L\|=1$, A_i is the only one covering that space and should remain doing so. The agent A_i would immediately switch to cohesion mode. In this mode, it guarantees that it remains covering some grid cells that are close to covered grid cell. This property is similarly achieved by determining the cardinalities of spaces within the agent's movement, coverage and sensing radii. If there exist L^* in the neighbourhood of A_i such that $\|L^*\|>1$, A_i should switch into perched mode. However, if for all L^*, $\|L^*\|=0$, A_i should relocate towards covered grid cells. All covered spaces within the agent's sensing radius are identified and some space within the movement radius that is closest to these covered spaces is randomly selected. In circumstances where agents are completely isolated, a mean free path is invented again.

In this study, agents that fall off the region we are covering are re-deployed by allowing them to re-surface from the opposite end of the terrain.

4. Experimental setup

We conduct four simulation experiments for comparing the efficiency of our flock inspired model with random guessing (see Algorithm 2). We evaluate our model in terms of the quality of coverage achieved as well as the time it takes to achieve complete coverage. We define the quality of coverage as the percentage of area covered in each time slot. On the other hand, complete coverage would refer to a point when the fraction of area covered in progressive time intervals is no longer changing significantly, implying no further explorations. Time in simulation is measured using the number of iterations our algorithms are executed.

4.1. Experiment I: Separation speed

This metric investigates agents' separation speed. We measure the speed using the number of agents that are successfully perched within specific time intervals. We administer each simulation until 150 iterations are executed. The cumulative quantities of perched agents in successive iterations are recorded, giving an indication of how fast agents achieve successful placement. From observations, 150 iterations are sufficiently many to influence convergence. High values indicate many agents that perch within few iterations, corresponding to better model performance.

For each test, 10 simulations are conducted. The quantities we get are averaged over results of all 10 simulations, giving a centrally placed quantity upon which separation speed is evaluated. The central measure is compared with results achieved using random guessing.

Algorithm 2: Agents' controls using random guess

---

mode ← *separation*
while (*true*)
    direction ← *none*
    foreach agent at grid cell L (A_L)
        if ($\|L\| > 1$)
            $A_L \leftarrow A_{L^*}$ (*any neighbour cells*)
        else
            $A_L \leftarrow A_L$ (*remain perched*)

4.2. Experiment II: Cohesion speed

In this experiment, we investigate agents' relocation speed after isolation. We determine the number of iterations executed from the time an agent is isolated until it perches on a cohesive space. We firstly deploy an isolated agent. The time it takes that agent to perch on a cohesive cell would indicate the model's cohesion speed. The simulations are repeated for 1000 times. The measured times are then averaged over the 1000 simulations, giving a centrally placed time value upon which cohesion speed is evaluated. Smaller measures are desirable, indicating better performance.

4.3. Experiment III: Coverage quality

Two metrics are of particular interest in this experiment: (a) the fraction of area covered at any time slot and (b) the time it takes to achieve complete coverage. A high quality model should achieve a well distributed agents' placement in a minimal number of iterations possible. However, the determination of that fraction of area covered at any time slot would require us to perform very complex geometric computations. We require a

mechanism to avoid redundant consideration of overlapping coverage. Commonly, complexity of the computations involved compromise the quality of coverage achieved.

We propose a much simpler method for determining the fraction of area covered in each time slot. In each step, we determine the number of uncovered grid cells and subtract this quantity from all grid cells in the region, giving us the number of covered grid cells. The equation below expresses the fraction as a percentage of cells in the region. An evaluation of the quality

$$Q_t \leftarrow \left(1 - \frac{\sum uncovered\ cells}{\sum all\ cells\ in\ the\ region}\right) \times 100$$

of coverage at specific time intervals would consequently provide an indication of how well and fast coverage is achieved. The overall time it takes until complete coverage would eventually be determined. Similarly, 10 simulations are conducted and the centrally placed time measures are determined. Shorter time measures with high Q_t values would indicate a high quality model and are desirable.

4.4. Experiment IV: Fault tolerance

We also assess the performance and quality of our model in a scenario where any agent may fail. Such failure may be due to environmental factors such as lava coming from a volcano or bombing in a military base [17]. Our goal is to investigate how well the model adapts to such failures. A fault tolerant model would quickly reorganize and achieve complete coverage again.

We firstly allow complete coverage to occur before we "*kill*" 40 agents covering some continuous space. The speed with which the created coverage hole is explored and covered again is recorded. We also measure the subsequent coverage quality improvements achieved thereafter. The centrally placed measures are determined and recorded. Smaller values indicate quick recovery and high fault tolerance, corresponding to better performance.

5. Results and discussions

Our simulator was implemented using a C++/OpenGL interface. All experiments were conducted using a swarm population of 1000 agents covering a 90x90 discrete grid. Geometrically, 900 agents are sufficiently many to cover the region. However, deployment of excess agents would improve the detection of any coverage holes that may arise in the event of some agents' failures late in simulation time. A few agents would remain in the separation mode, using the mean free path, in case a fault occurs. Our assumption that could agents perch after locating uncovered spaces does not therefore imply that once perched, agents would not move again. In fact, perched agents may move due to separation forces exerted by agents still in separation mode and using the mean free path.

5.1. Result I: Separation speed

Our model achieved a 50.57% successful agents' placement in 31 iterations. Contrary, only 19.14% of the agents that used random guessing were perched in the same period of time. Our model achieved complete coverage in 135 iterations. On the other hand, only 51% of agents that use random guessing could perch in 135 simulation steps. The performance gap between our model and random guessing is illustrated in Figure 1.

From observations, random guess placement suggests that agents coincidentally discover spaces to cover. Algorithm 2 does not provide any control mechanism that would prevent agents' movement towards covered spaces. This would disturb and degrade the quality of coverage already achieved. For that reason, the performance curve that corresponds to separation speed using random guessing fluctuates more often.

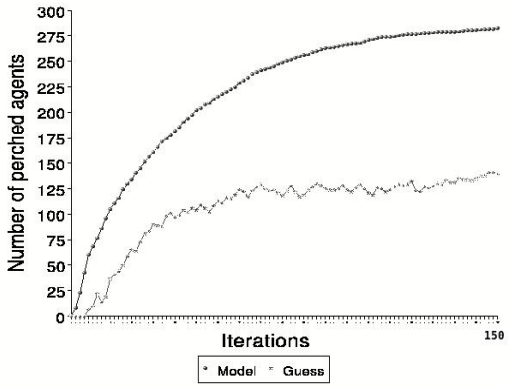

Figure 1: Comparison of separation speed

Contrary, our model avoids agent movements towards covered spaces. An agent may only move to a covered space when it is using the

mean free path. This mechanism speeds up separation with minimal need for cohesion thereafter.

5.2. Result: Cohesion speed

An agent that is completely isolated from the rest should move and cover some space close to covered spaces. Firstly, we isolated an agent to a location that is 5 grid cells away from the nearest covered space. The numbers of iterations executed until the agent perched next to some covered space are recorded in Table 1. On average, agents that used our model achieved cohesive placement in 16 iterations with a standard deviation of 3 iterations. Contrary, agents that used random guessing took an average of 39 iterations with a standard deviation of 14 iterations. From observations, the chances that agents using our model fail to locate some uncovered space are slim, 0.5%. However, 41.2% of agents that use random guessing may completely fail to perch and rather disturb the already achieved coverage.

	Our model	Random guess
Mean steps	16	39
Standard deviation	3	14
Entropy levels	0.5%	41.2%

Table 1: Steps taken before cohesive perching

5.3. Result III: Coverage quality

Figure 1 also reports that exploration of uncovered spaces may continue even after complete coverage. This is because of excess agents that would still be in separation mode. The mechanism therefore suggests that a 100% coverage quality is only achievable when we deploy the exact number of agents that are required to cover the region. However, that threshold density would depend on many unpredictable parameters such as sensing radius, size of the region, movement steps, etc. We know that excess agent deployment would compromise the coverage quality we achieve. However, those excess agents would be very useful in the event of failures as mentioned before. The coverage holes that arise as a result of these failures would easily be detected and covered again.

Figure 2 illustrates the levels of coverage quality that we achieve using our model as well as coverage using random guessing. Our model reached its maximum coverage quality of 96.23% in 135 iterations, after which we barely observed any significant coverage quality changes. Contrary, agents that used random guessing only achieved a maximum of 65.66% coverage quality.

5.4. Result IV: Fault tolerance

The performance of our model in environments where agents may fail depends mainly on two issues: (a) the density of agents deployed and (b) reliability of agents in locating

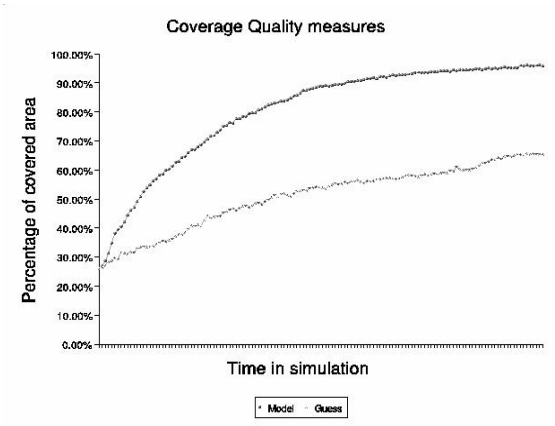

Figure 2: Coverage quality

coverage holes. Ideally, deployment of agents in excess would enhance fault tolerance.

We firstly allowed complete coverage to occur. A continuous coverage hole was formed after we killed 40 perched agents, giving a chance to those agents using the mean free path to perch in the created spaces. Our model self repaired and re-established complete coverage in an average of 34 iterations. However, the random guessing model only achieved a 46.27% coverage quality after 120 iterations.

6. Conclusion

We proposed an area coverage model inspired by Reynolds' flocking algorithm. Experiments that evaluated and compared the performances of this model with the random guessing model yield the following conclusions:

- Our model exhibits good separation and cohesion properties of simulated flocks.
- 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.

The contributions of this work are:

- We presented a novel agents control model using simulated flocking rules.
- We devised and evaluated a plausible strategy for determining coverage quality.
- We also devised new ways for measuring the performance of agent based coverage models.

The work is at the initial stages of a descriptive phenomenon aimed at providing generalized causes of emergent behaviour. Further research in formalizing and quantifying emergence is underway.

6.1. Acknowledgements

We would like to thank the entire VRSIG team at Rhodes University, without whom we would never have gotten such valuable support, suggestions and moral.

7. References

[1] S.B. Badia, U.Bernardet, A.Guanella, P. Pyk, and P. F. M.J. Verschure. "A Biologically Based Chemo-Sensing UAV for Humanitarian Demining". *In the International Journal of Advanced Robotic Systems, Volume 4, Number 2, ISSN:1729-8806, pp. 187-198, Zurich, Switzerland*, 2007.

[2] J. Ai, A. Alhussein, and Abouzeid. "Coverage by directional sensors in randomly deployed wireless sensor networks.", *In Springer Science and Business Media, Inc,* 2006.

[3] C. Savarese, J.M. Rabaey., and J. Beutel. "Locationing in distributed ad-hoc wireless sensor networks.", *In proceedings of the 2001 International Conference on Acoustics, Speech and Signal Processing*, 2002.

[4] A. Dhariwal, S.S. Gaurav, and A.G.R. Aristides. "Bacterium-inspired robots for environmental monitoring.", *In proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA,* 2004.

[5] P. Gober, A. Ziviani, P. Todorova, M. Dias de Amorim, P. Hunerberg, and S. Fdida. "Topology control and localization in wireless ad hoc sensor networks.", *In Ad hoc and Sensor Wireless Networks, OCP Science, ISSN:1551-9899,* 2005.

[6] C. Huang and Y. Tseng. "The coverage problem in a wireless sensor network.", *In the International Workshop on Wireless Sensor Networks and Applications. Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, San Diego, California, USA*. 2003.

[7] S. Hussain and L.M. Gaudette. "Quality of coverage of wireless sensor networks using energy efficient routing with genetic algorithms.", *In Jodrey School of Computer Science, Acadia University,* 2006.

[8] F. Koushanfar, S. Slijepcevic, M. Potkonjak, and A. Sangiovanni-Vincentelli. "Location discovery in ad hoc wireless sensor networks: Location discovery and sensor exp*osure", Air Force Research Laboratory, Chapter 3, pages 137-174. Kruwer Academic Publishers,* 2004.

[9] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava. "Coverage problems in wireless ad hoc sensor networks.", *In proceedings: IEEE INFOCOM'01*, 2001.

[10] D. Merkle, M. Middendorf, and A. Scheidler. "Modeling ant brood tending patterns with cellular automata.", *In Proceedings of the Workshop on modeling of complex systems by cellular automata, volume 2.2, pages 412-419*, 2005.

[11] W. Naeem, R. Sutton, and J. Chudley. "Chemical plume tracing and odour source localization by autonomous vehicles.", *In Journal of Navigation, volume 60*, *issue 02, pages 173-190*, 2007.

[12] A. Nasipuri and K.Li. "A directionality based location discovery scheme for wireless sensor networks.", *In Proceedings: ACM International Workshop on Wireless Sensor Networks and Applications, pages 105-111*, 2002.

[13] C.W. Reynolds. "Flocks, herds, and schools: A distributed behavioral model.", *In proceedings of SIGGRAPH'87 (Computer Graphics 21(4)), pages 25-34,* July 1987.

[14] K. Romer and F. Mattern. "Towards a unified view on space and time in sensor networks.", *In Wireless Sensor Networks and Applications-proceedings of the Dagstuhl Seminar 04122, volume 28 of 13, pages 1484-1497*, 2005.

[15] W. M. Spears, R. Heil, D.F. Spears, and D. Zarzhitsky. "Physicomimetics for mobile robot formations.", *In proceedings of the third International Joint conference on Autonomous Agents and Multi Agent Systems, volume 3, pages 1528 – 1529*, 2004.

[16] H. Wu, C. Wang, and N. Tzeng. "Novel self-configurable positioning technique for multi-hop wireless networks.", *In IEEE/ACM Transactions on Networking, volume 13, number 3*, 2005.

[17] O. Younis, M. Krunz, and S. Ramasubramanian. "A framework for resilient online coverage in sensor networks.", *In proceedings of IEEE conference on sensor, Mesh, and Ah hoc communications and Networks, San Diego, California,* 2007.