CS 423 Final Project

Brandon Wade*
Computer Science
University Of New Mexico

John Krukar †
Computer Science
University Of New Mexico

ABSTRACT

In this paper we seek to replicate the methods and results in the paper ”Exploring the co-evolution of predator and prey morphology and behavior”[1]. This consists of providing observational analysis of co-evolutionary cycle's between prey swarming behavior and a predators field of view. We will provide evidence in regards to the predator visual system being the driving force in this co-evolutionary cycle. We will be conducting our methodology in the Unity Programming Environment. Unity provides a useful machine learning package that gives a wealthy alternative of using genetic algorithms to evolve our prey and predators.

1 INTRODUCTION

In this paper we seek to replicate the methods and results in the paper ”Exploring the coevolution of predator and prey morphology and behavior”[1]. This paper provides observational analysis of coevolutionary cycles between prey swarming behavior and a predators field of view. We will provide evidence that cites the predators visual system as the driving force in this coevolutionary cycle. We will be conducting our methodology in the Unity programming environment. Unity provides a useful machine learning package that gives an alternative to using genetic algorithms to evolve our prey and predators.

In this paper we aren't necessarily trying to understand HOW the visual systems between predators and prey evolve to be so complex, but WHY does it evolve, and what results from this evolution. The authors of Olson et. al 2016[1], underline information that suggests certain prey evolve to form swarms, and defend themselves as a group for a variety of hypothesized reasons. In this paper we are trying to validate the hypothesis that as a predators visual system becomes broader, the prey start to swarm together as a defense mechanism. Inversely, as the predators visual system becomes more narrow, the prey become more dispersed.

In Olson et. al [1], Markov Networks (MN) are optimized via a genetic algorithm to evolve the behaviors of the predator and prey agents. We chose not to evolve the predator and prey in a coevolutionary cycle where the predator's visual system is gradually and randomly altered over time as the prey continuously adapt in order to keep up. Instead, we evolve our predator and prey using reinforcement learning. We believe using reinforcement learning will be the least biased optimization method because we do not directly define the relationships between environmental observations and agent actions.

An open source Unity plugin called Unity Machine Learning Agents enables us to use reinforcement learning to develop the complex behaviors of our predator and prey agents. In reinforcement learning, an agent perceives its environment as a series of inputs and builds a model to map those inputs to actions which will maximize a reward function. The model is optimized as the agent collects data in the form of experiences and reinforces actions that lead to more rewards. Our models of our agent's behaviors are trained with proximal policy optimization algorithms via TensorFlow. TensorFlow is tool used to execute machine learning models. Proximal policy optimization is a reinforcement algorithm which tries to maximize the reward function of a mathematical model while also ensuring that updates to that model are frequent and incremental to ensure a more stable learning process.

*e-mail: bwade@unm.edu
†e-mail:jkrukar@unm.edu

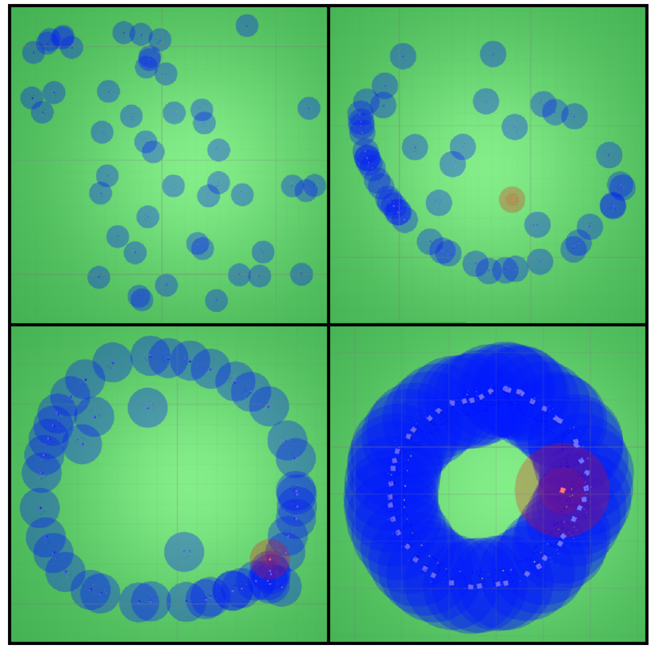

Figure 1: An example of prey swarming behavior. The order of the images goes from the top left to the bottom right. The top left image is the initial random distribution of the prey agents. Over time the prey form a loose ring which converges inward until a the prey have formed a very dense swarm. Each prey has a light blue circle around it to make it easier to distinguish them since at certain image scales they are almost too small to discern.

2 METHODS

2.1 Simulation Environment

Our simulation environment is a 3D flat arena. The scale of the arena is 512 meters by 512 meters. Each simulation is run for 2,000 timesteps. At the start of each simulation, one predator and 50 prey are placed randomly throughout the arena. At each timestep, every agent in the arena will make environmental observations and choose an action to perform.

2.2 The Predator and Prey Agents

As in Olson et. al. [1], our predator and prey agents can perform one of the following actions at each timestep of the simulation: do nothing, move forward one unit, turn right by a fixed amount while moving forward one unit, or turn left by a fixed amount while

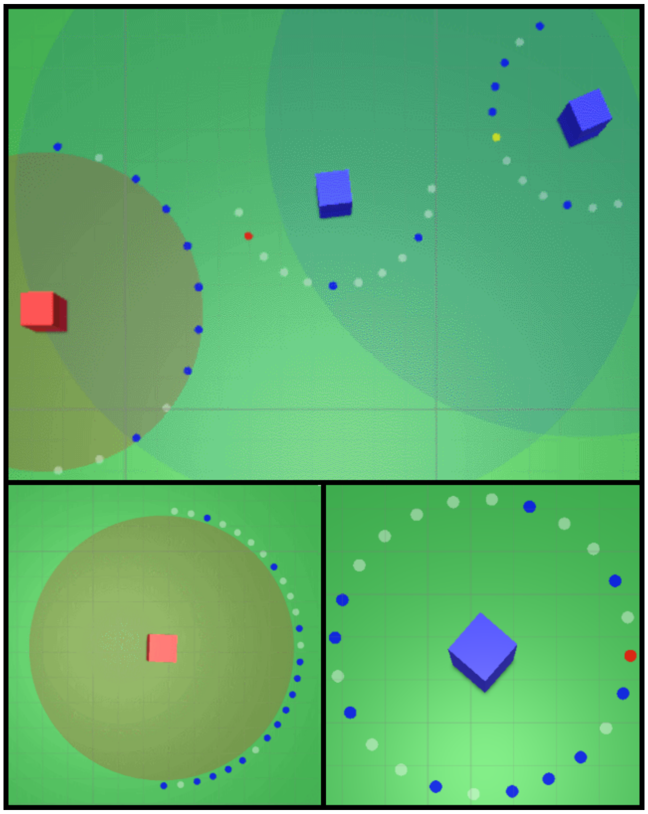

Figure 2: Examples of the predator and prey visual systems in action. In the top half of the figure, blue prey agents can be seen to the right of the red predator agent. Both the predator and prey have a 180° field of view and 12 circles indicating the status of each retinal slice. A blue indicator shows that at least one or more prey are visible to that retinal slice. A red indicator shows that the predator is visible to that retinal slice. A yellow indicator shows that the predator and at least one prey are visible to that retinal slice. The faint red circle around the predator indicates the predator's attack range. The faint blue circles surrounding the prey indicate how close another prey agent must be to be part of that agent's swarm. In this case, the two prey are close enough to be considered part of a swarm. The bottom left section of the figure shows the predator after we doubled its visual acuity. Although the predator's field of view has remained the same, there are now twice as many retinal slices. This results in a more accurate visual system without sacrificing field of view. The bottom right section of the figure shows a prey agent with a 360° field of view.

moving forward one unit. The predator can move at a rate of 1 unit/timestep and is able to turn 6°. The prey can turn 8° and is three times slower than the predator. While the predator is faster, the prey is more maneuverable. Each agent is given its distance from the edges of the arena in an (x,y) coordinate with the values normalized between [1,-1]. The goal of the predator agent is to maximize the number of prey it can capture in a 2,000 timestep simulation. If the predator is within 5 meters of a prey agent, it will attempt to capture that prey. After an attack, the predator must wait 10 timesteps before attempting another attack. As noted in Olson et. al. [CITATION], this cooldown simulates the predator consuming a prey agent or taking time to recover from an unsuccessful attack. The predator is given a reward of 1.0 point in its reward function each time it captures a prey. The predator is also penalized 0.0005 points each timestep to encourage a sense of urgency. We chose this value because it generates a cumulative penalty of 1.0 point over the course of the entire simulation. The prey agent is penalized 1.0 point if it is captured by the predator. However, the prey is awarded 0.0005 points for every timestep that it has not been captured. We used reinforcement learning to train predator agent models with fields of view ranging from 60°-360° by incrementing the field of view by 30° for each model. The predator agents were trained for 200,000 rounds with 50 static prey. Then we trained a prey agent model for 200,000 rounds against each predator agent model. Predator fitness was evaluated by average the number of prey captured over the course of one hundred simulations.

2.3 Predator and Prey Visual Systems

We implemented the visual system described in Olson et. al. [CITATION] in which an agent's field of view is divided into 15° retinal slices. Each slice outputs a value indicating if a predator and/or prey agent is visible to that slice. In the case of the predator, the slices only indicate if a prey agent is visible (since there is only one predator in the simulation). The slices do not indicate the quantity of visible agents or their distance away from the perceiving agent. The prey has a 180° field of view composed of 12 retinal slices. In our experiments, we altered the predator's field of view from 60° to 360° in 30° increments.

We performed two additional experiments. In one, we altered the prey's field of view to 360°. In the other experiment, we set the predator's field of view to 180° and doubled the number of retinal slices to simulate a more acute sense of vision while retaining a broad field of view. The predator can detect agents up to 200 meters away and the prey can detect agents up to 100 meters away. For observation purposes, we implemented visible markers to represent the perception of each retinal slice in the visual system. The markers change from translucent white to blue if a prey agent is visible in that slice. The markers turn red if the predator is visible or yellow if both the predator and a prey agent are visible in the retinal slice.

2.4 The Predator Confusion Effect

The predator confusion effect is a constraint put onto the predator agent whereby the number of prey near the predator's target during an attack has a negative effect on the predator's chance of success. We implemented this constraint using the same equation as Olson et. al. [1]:

$$Pcapture = \frac{1}{Anv} \tag{1}$$

Pcapture is the probability that a predator will successfully capture a prey, where Anv is the number of prey agents that within 30 meters of the predator's target and visible to the predator.

2.5 Prey Average Swarm Density

We measured the prey average swarm density by calculating a running average of the number of other prey within 30 meters of each prey agent at each timestep. At the end of each round, we averaged the swarm density values of each prey agent. We took the average prey swarm density from one hundred simulations and averaged those swarm density samples to get the average swarm density for each of predators models.

2.6 Predator Visual Acuity vs. Prey Swarm Density

In order to extend the research of Olson et. al. [CITATION] we doubled the predator's visual acuity by increasing the maximum number potential of retinal slices in its visual system from 24 to 48 and reducing the angle of each slice from 15° to 7.5°. We performed this experiment using a predator agent with a 180° field of view composed of 24 retinal slices. We trained a new predator agent model and tested it against the prey model previously trained against a predator with a 180° field of view. We also tested the more visually acute predator against a prey model trained against it to see how the prey would adapt.

2.7 Prey Field of View vs. Prey Swarm Density

We further extended the research of Olson et. al. [CITATION] by increasing the prey's field of view from 180° to 360° to see how average swarm density and predator fitness would be affected. First, we trained a prey agent model against a predator with a 180° field of view. Then we compared the average predator fitness and average prey swarm density derived from 100 simulations to our previously collected data.

3 Results

3.1 Predator Field of View Affect on Fitness and Prey Average Swarm Density

As can be seen in Figure 3, we found that as the predators field of view increases so does the prey average swarm density. There were a few outliers in our data where the average swarm density decreased, but we believe that the overall trend remains strong. We found it interesting that despite our outliers, the predators fitness seemed to decrease at the same slow rate as its field of view increased. In the case of a predator with a 360° field of view, the average swarm density experiences a sharp decrease. Upon further investigation, we found that the prey had learned to form a ring (such as in figure 1) with the predator inside it. The ring had a radius the size of the predators maximum sight distance. This allowed the prey to maximize the predator confusion effect while also retaining as much distance away from the predator as possible.

From our observations, we noticed that the swarm size toward the end of the simulation seemed much denser than the average swarm size that we calculated. We believe this is a result of the initial random placement of the prey in the arena. Since the prey start so dispersed, the average swarm density is weighed down throughout the rest of the simulation. If we were to repeat this experiment, we would sample the average swarm density for the first and second half of each simulation.

3.2 Prey Field of View Affect on Swarm Size

Doubling the prey's field of view resulted in a higher average swarm density. We believe that the increase in the prey's field of view allows the prey to find the relative direction containing the most prey and coalesce into a large swarm faster and maintain a more condensed swarm.In the future, we would like to spend more time testing prey agents with a 360° field of view against a variety of predator agents with various fields of view.

3.3 Predator Visual Acuity Affect on Predator Fitness and Prey Average Swarm Size

Increasing the predators visual acuity by doubling the number of retinal slices resulted in an increase in predator fitness and a decrease in prey average swarm density. Evolving increasingly complex eyes

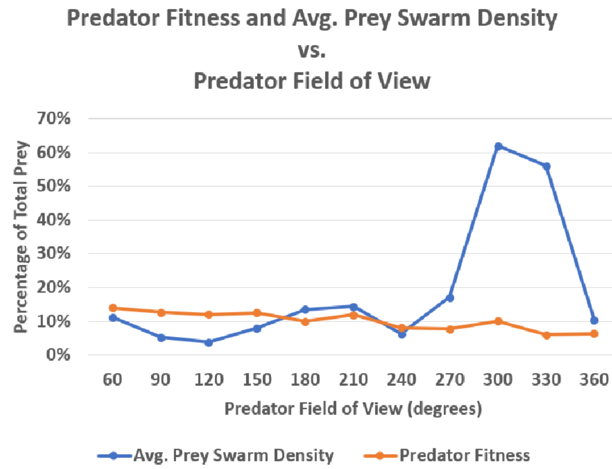

Figure 3: This chart shows the relationship between the predators field of view and both the predators fitness as well as the average prey swarm density. For the predators fitness, the values indicate what percentage of the total prey population the predator was able to capture. For the average prey swarm density, the values indicates the percentage of all prey that were in an individual prey agent's swarm. Although the average prey swarm density generally increased as the predators field of view increased, there were several outliers where the swarm density actually decreased. However, the predators fitness had a generally weak, negative slope despite the fluctuations in prey swam density.

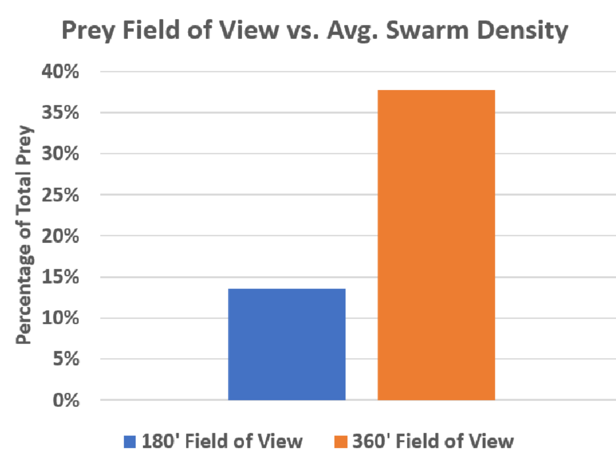

Figure 4: When we doubled the prey's field of view from 180° to 360°, the average swarm density increased dramatically. We believe the average swarm density increased so drastically because the prey were able to detect the relative direction of the majority of the other prey and form a large swarm faster and maintain the swarm density better.

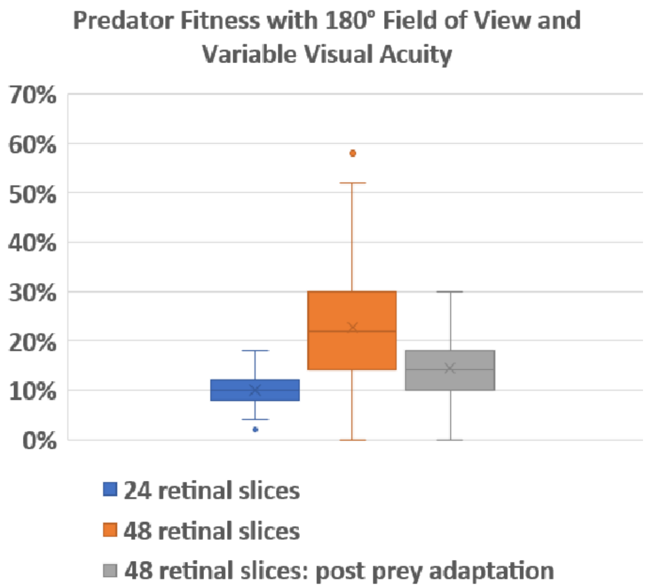

Figure 5: This figure shows the affect that increasing the predators visual acuity has on its fitness. We see that the increasing the predators visual acuity more than doubles its fitness initially. However, once the prey are given a chance to adapt, the resulting predator fitness is approximately a 50% increase from the original fitness value.

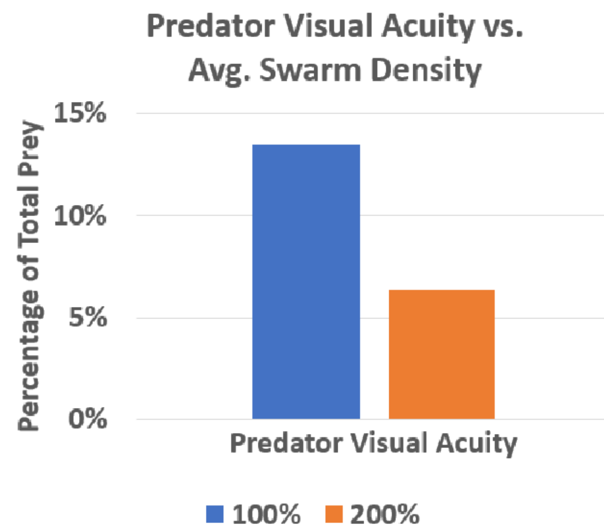

Figure 6: This figure shows the affect that increasing the predators visual acuity has on prey average swarm density. 100% visual acuity represents the normal acuity of the predators visual system. The 200% visual acuity represents the visual system once we doubled the number of retinal slices. Once the predator developed a more acute visual system, the average swarm density was reduced by about half.

could certainly be a valid strategy to maximize predator fitness. In the future, we would like to test this hypothesis by allowing the number and size of the retinal slices to be varied to see the what kinds of visual systems are developed.

4 Conclusion

We conclude that our findings were consistent with the results presented in Olson et. al. [1]. Predator field of view is positively correlated with prey average swarm density. We believe this is a robust validation of their findings given the differences between our methods: Markov Networks vs. Reinforcement Learning. We also found that increased visual acuity results in an increase in predator fitness and a decrease in prey average swarm size. Furthermore, we found that increasing the prey's field of view resulted in a higher average swarm density.

5 Author Contributions

Brandon Wade wrote the abstract, and helped write the introduction and method section. He also helped write code for the machine learning agents.

John Krukar helped write the introduction and methods section. He wrote the results and conclusion. He helped write the code for the machine learning agents in Unity. He performed the analysis of the data, generated the figures, and wrote the captions.

Acknowledgments

We wish to thank Melanie Moses and Wayne Just for being such great and patient instructors.

References

[1] S. Olson ,A. Hintze, F. Dyer, J. Moore, C. Adami *Exploring the coevolution of predator and prey morphology and behavior* , 2016

[2] Unity *Unity Machine Learning Agents*,GitHub repository https://github.com/Unity

[3] R.S. Olson, A. Hintze, F.C. Dyer, J.H. Moore, and C. Adami, (2016). ”Exploring the coevolution of predator and prey morphology and behavior,” Proceedings Artificial Life, vol. 15,

[4] V. Pierre, ’Unity ML-Agents (Beta)’, 2018. [Online]. Available: https://github.com/Unity-Technologies/ml-agents. [Accessed 16-Apr-2018].

[5] J. Schulman, O. Klimov, F. Wolski, P. Dhariwal, A. Radford. ”Proximal Policy Optimization”, 20-July-2017. [Online]. Available: https://blog.openai.com/openai-baselines-ppo/ppo. [Accessed: 18-Apr-2018].

[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al. ”Large-scale machine learning on heterogeneous distributed systems,” 20-Feb-2018. [Online]. Available: https://www.tensorflow.org/about/bib. [Accessed 17-Apr-2018].

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al. ”About TensorFlow”, 9-March-2018. [Online]. Available: https://www.tensorflow.org/about/. [Accessed 17-Apr-2018].

[8] Wikipedia, ”Reinforcement learning”, Wikipedia: The Free Encyclopedia, 2018.

[9] Wikipedia, ”Artificial neural network”, Wikipedia The Free Encyclopedia, 2018.