Pursuit/Evasion Behaviors for Multi-Agent Systems

Juan L. Burgos

August 5, 2010

Abstract

Multi-Agent Systems involve the cooperation and coordination of groups of agents in an environment while they perform tasks. Previous work has provided simulated heterogeneous agents that engage in specific behavior driven actions with a priori assumptions with respect to agent knowledge of the environment. These experiments however did not have a distributed set up. Our approach involves distributed agents that follow their own behaviors while receiving information via a virtual sensor-suite that coordinates information flow through sensors each agent possesses. In this paper, we describe the sensor-suite approach to information distribution and our attempts to apply this approach to real world robots in physical environments.

1 Introduction

The pursuit/evasion problem is the classic cops and robbers problem where pursuers chase evaders through an environment. This problem involves fast-paced dynamic decision making on the part of both parties of agents where they work toward success in their respective goals. Both parties must rely on both individual and group effort to coordinate actions while appropriately handling encounters with both environmental obstacle and agent encounters. In order to play the game successfully, agent architecture must be designed with specific mechanisms for simulating behavioral processes and information gathering/analysis/communication abilities.

Much research has been conducted in this area with approaches ranging from strict graph-based roadmap pursuit/evasion games such as to real world heuristic games. The former would implement games with specific node and edge based hunt/capture strategies where the goal of the pursuer is to be present in the same node as the evader. The later would involve more in-depth trial-and-error traversals of the environment which are more costly but would, if cost weren't the issue, be suitable for real world experimentation.

Many existing approaches to the problem tend to follow either the strict graph-based game or the heuristic game for solving more scenario specific versions of the pursuit/evasion problem. These approaches, though relevant to the overall body of research in this area are far too specific and limiting to be broadly applicable. Therefore, in order to gain insight from both approaches while managing their limitations a combination of both techniques would be the appropriate strategy. We expand upon previous work in roadmap-based flocking [FKW03, SOSB05, VSK*02] by providing distributed agents with the ability to communicate among group-mates information that has been gathered through the exploitation of a sensor-suite that has been provided to each agent. The sensor-suite is set up such that each sensor provides the agent with specific information regarding the environment and other agents that can provide input necessary for group behaviors to determine the agent's next course of action.

2 Related Work

2.1 Early Pursuit/Evasion Games

Many pursuit/evasion game experiments have been conducted over the past decades with much of the early theoretical work performed with one-on-one games where a single pursuer chased a single evader. Methods for these simplified versions of the game involved game theory [Isa65], genetic algorithms [Rey94], and neural networks [CM96]. One of the earliest versions of the pursuit/evasion games involved a one-on-one match restricted to a graph. Due to the limitation caused by having one-on-one games, applicability of the results is scarce in terms of real world applications. Over time, interesting multi-player experiments would become available through team based matches, examples of which would include [KAS03, KVS*01].

2.2 Important advances in Multi-Agent Systems

2.2.1 Distributed Flocking Behavior Model

An important advancement in the field is presented in [Rey87] with the "Distributed Behavioral Model". In a flock of birds, each agent is a separate entity that responds to local variables in its environment while maintaining focus on the group's goal. These agents will attempt to remain in formation while in flight, making sure not to collide with other agents or obstacles. Simulated flock behavior can be related to the behavior of aggregate vapor particles in an arbitrary particle system where each particle will poses its own individual behavior that relates to its immediate properties such as current state and location. Therefore, the flocking model can be applied to the behaviors of both animate and inanimate objects affected by external influences.

2.2.2 Pursuit/Evasion with Sensory Input

Past work in [YL10] provided agents with sensors that provided a field-of-view. They enabled probabilistic distortions through a Bayesian filter in order to mirror real world sensory data for robots since perfect vision sensors are impractical.

Expanding the idea of providing sensor data to agents was the work in [DFF10] where limited range finding capabilities were used to provide groups of pursuers with territory coverage information as they cleared the environment in search of the evader.

2.3 Robotic Pursuit/Evasion Games

Group behaviors need to be designed to dynamically interpret and react real world uncertainties in order to provide for robotic based pursuit/evasion games. Due to the impracticality an agent obtaining perfect knowledge of the environment most of the work in robotic pursuit/evasion is theoretical with the few implementations using probabilistic methods for obtaining providing robotic pursuers with the most likely location of the evader. One approach in [HKS07] can handle unknown environments but limits applicability to homogeneous agents with no cooperative strategies possible between agents. Other approaches were attempted but still suffered from noncooperative play limitations.

3 Overview of Approach

3.1 Problem Definition

Most of the previous implementations of the pursuit/evasion game involved agents using sensory systems that were limited in scope of perceivable information. Many approaches have provided agents with perfect knowledge of the environment. Previous work from our lab used graph-based roadmaps to provide an environmental guide for the agents' path planning techniques and group behaviors from [BLA02,LRMA05] such as flocking, herding, and homing behaviors. Our work this summer involved the development of a suitable sensor-suite for a distributed robot based implementation of the pursuit/evasion game. We define what knowledge an agent would require to make decisions regarding its next action with respect to controlled information flow through specific sensor. Our current implementation uses a global camera system that acts as a server where the robots (agents) can gain access to appropriate information about their environment through simulated sensors. This is our "centralized" implementation of the approach.

The following sections provide a basic description of our sensor-suite's architecture and the functions of each individual sensor provided.

3.2 Sensor Architecture

3.2.1 Basic Architecture

Before designing each individual sensor, we constructed a base sensor class that contains all the functions that will be common amongst the derived sensors.

The idea is to make all information that is not based on inter-agent communication to strictly come from the sensors.

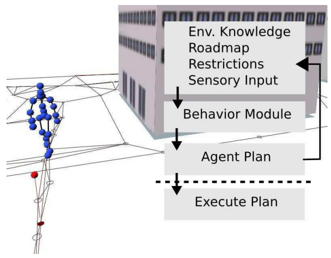

Figure 1: Though the flow of information to agents follows a specialized breakdown, all data is eventually combined to produce a picture which the group behaviors must interpret to provide instructions for the next action.

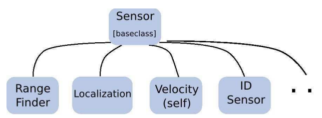

Figure 2: Every sensor to be implemented has been constructed based off of the sensor baseclass.

3.3 Derived Sensors

The following list is of the currently simulated sensors. The subsequent subsections will explain the purpose of each sensor; note that some sensors are built off of the abilities of others.

- Vision Sensor
- Compass Sensor
- Localization Sensor
- Odometry
- Self-Velocity
- External-Velocity Sensor
- Range-Finding Sensor

3.3.1 Vision Sensor

The Vision sensor provides the user agent with information regarding other visible agents, including their ID's and whether they are friend or foe.

3.3.2 Compass Sensor

The Compass provides the agent with its current orientation in the environment.

3.3.3 Localization Sensor

The Localization sensor keeps track of the agent's current position with respect to the environment's global coordinate system.

3.3.4 Odometry Sensor

The Odometer is similar to the localization sensor in that it provides location information to the agent. However, the key difference is that this sensor records the agent's current location relative to another point in the environment. Information from the localization sensor is crucial for this.

3.3.5 Self-Velocity Sensor

The Self-Velocity sensor provides the agent with its current speed and direction. This makes use of the Compass's information.

3.3.6 External-Velocity Sensor

The External-Velocity sensor provides information regarding the velocities of objects in the environment, be them agents and obstacles alike. This sensor uses localization, vision, and rangefinder results to calculate its data.

3.3.7 Range-Finding Sensor

The rangefinder is the more complex of the sensors due to its importance in future which will be discussed in the next section. The purpose of this sensor was to provide the agent with a detailed view of its local environemnt, this includes both 2-dimensional cartesian coordinates of detected objects within some maximum range of the agent along with those object's polar-coordinates relative to the robot's location in the environment and its orientation with respect to the global coordinate axis. In its current form, as shown below, the algorithm simulates a laser-rangefinder type device in digital space which relies on simulated environments in order to function properly. Only coordinate based information of collision detections or collision-free vectors is provided to the agent a given maximum range of detection.

4 Current Results

At the moment, we are in the process revisiting much of the group behavior code to bring it up to professional programming standards along with the robot-simulation integration process. This integration process is meant to take the simulation based pursuit/evasion games that we have been testing our sensor suite and group behaviors on and integrating them into robots so that robot based pursuit/evasion game experiments can be run.

Algorithm 1 RangeFinding Sensor Algorithm

Input: Agent A_0 with its state, location, and orientation data

Output: Two arrays containing 2D cartesian and polar coordinates of obstacle detections

Initialize $\Delta danger$, $\Delta\Theta$, and $\Delta step$ from simulation input file.

Create one empty array for the 2D cartesian coordinates of detected objects in the environment

Create another empty array for polar coordinates of detected objects relative to A_0

for Each Angle, from $Angle$ to $\leftarrow$ 0 to 360 degrees, incrementing by $\Delta\Theta$ **do**

for Each Step, from $Step$ to $\leftarrow$ 0 to $MaxStep$, incrementing by $\Delta step$ **do**

if $step \geq maxstep$ **then**

Record polar coordinate $[r, \Theta] \leftarrow [MaxStep, Angle]$

Record 2D cartesian coordinate $[x, y] \leftarrow getCoord()$

$breakloop()$

end if

if Obstacle Collision detected or Agent is nearby **then**

if Any obstacles or agents are within $\Delta range$ of A_0 **then**

Set $CollisionWarningFlag$

end if

Record polar coordinate $[r, \Theta] \leftarrow [Step, Angle]$

Record 2D cartesian coordinate $[x, y] \leftarrow getCoord()$

$breakloop()$

end if

end for

end for

Return2 arrays with 2D cartesian and polar coordinates of detected obstacles

5 Conclusion

In this paper, we propose a new approach to the pursuit/evasion problem involving an array of sensors that disseminate information with respect to type and purpose. The purpose of this work is to provide a proficient sensor suite for robots playing the pursuit/evasion games in any unknown environment under the directive of any multitude of group behaviors. Currently, the work is still in its early stages with the robot-simulation integration process still progressing. At the moment, our system uses a centralized server-agent model where a global camera provides data through simulated sensors. This scheme has worked to help us with the integration process. However, we are currently working on decentralizing the information dissemination process so that each individual agent will not require the overhead of server-agent communications.

References

[BLA02] Bayazit O. B., Lien J.-M., Amato N. M.: Better group behaviors in complex environments using global roadmaps. In *Artif. Life* (Dec 2002), pp. 362–370.

[CM96] Cliff D., Miller G. F.: Co-evolution of pursuit and evasion II: Simulation methods and results. In *From animals to animats 4* (Cambridge, MA, 1996), Maes P., Mataric M. J., Meyer J.-A., Pollack J. B., Wilson S. W., (Eds.), MIT Press, pp. 506–515.

[DFF10] Durham J. W., Franchi A., F.Bullo: Distributed pursuit-evasion with limited-visibility sensors via frontier-based exploration. Proceedings IEEE Internation Conference Robot Automation.

[FKW03] Fregene K., Kennedy D., Wang D.: Multi-vehicle pursuit-evasion: an agent-based framework. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)* (September 2003), pp. 2707– 2713.

[HKS07] Hollinger G., Kehagias A., Singh S.: Probabilistic strategies for pursuit in cluttered enviornments with multiple robots. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)* (2007).

[Isa65] Isaacs R.: *Differential games: A mathematical theory with applications to warfare and pursuit and optimization.* John Wiley, 1965.

[KAS03] Kim H., Antoniades A., Sastry S.: Pursuit-evasion strategies for teams of multiple agents with incomplete information. In *Proc. IEEE Conf. on Decision and Control* (2003), pp. 756–761.

[KVS*01] Kim H., Vidal R., Shim D., Shakernia O., Sastry S.: A hierarchical approach to probabilistic pursuit-evasion games with unmanned ground and aerial vehicles. In *Proc. IEEE Conf. on Decision and Control* (2001), pp. 634–639.

[LRMA05] Lien J.-M., Rodriguez S., Malric J.-P., Amato N. M.: Shepherding behaviors with multiple shepherds. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)* (April 2005), pp. 3413–3418.

[Rey87] Reynolds C. W.: Flocks, herds, and schools: A distributed behaviroal model. In *Computer Graphics* (1987), pp. 25–34.

[Rey94] Reynolds C. W.: Competition, coevolution and the game of tag. In *Artificial Life IV* (1994).

[SOSB05] Schenato L., Oh S., Sastry S., Bose P.: Swarm coordination for pursuit evasion games using sensor networks. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)* (2005), pp. 2493–2498.

[VSK*02] Vidal R., Shakernia O., Kim H., Shim D. H., Sastry S.: Probabilistic pursuit-evasion games: Theory, implementation, and experimentlal evaluation. In *IEEE Transaction son Robotics and Automation* (2002), pp. 662–669.

[YL10] Yu J., Lavalle S. M.: Probabilistic shadow information spaces. In *ICRA* (2010). Proceedings IEEE Internation Conference Robot Automation.