DOI: http://dx.doi.org/10.7551/978-0-262-31050-5-ch039

The Role of Working Memory in an Urban Pursuit Scenario

Ransom K. Winder[1], and James A. Reggia[2]

[1]The MITRE Corporation
[2]University of Maryland, College Park
rwinder@mitre.org

Abstract

Most self-organizing models of moving agent collectives (simulated herds, bird flocks, etc.) employ reflexive agents that lack significant memory of past movements and previously encountered environmental features. Further, these agent collectives often act in fairly open environments where obstacles to movement are relatively sparse. In this work, we explore the hypothesis that a limited working memory of recently encountered environmental features, distributed throughout the collective, can improve task performance for a team of interacting agents that are operating in a highly occluded environment. Investigating a team of agents pursuing a mobile target in an "urban environment", we found that the team benefited from 1) communication that coordinated team movements, and 2) from a working memory of the environment that was distributed among the agents, despite individual agents knowing only a small part of the relevant information. These results further our understanding of methodologies that can be applied to control robotic teams and swarm optimization, and may also provide insight into herd behavior of biological populations in some densely occluded environments.

Introduction

Teams of collectively moving agents have been widely studied in artificial life for many years. Reynolds' early work established that basic agent interactions, such as avoidance, alignment and cohesion forces, could produce surprisingly realistic flock-like behaviors (Reynolds, 1987). Subsequent studies have extended these results in areas such as robotics (Atherton, et al. 2006; Bayazit, et al. 2002; Couzin, et al. 2005; McCook and Esposito, 2007) and optimization (Kennedy et al, 2001; Lapizco-Encinas, Kingsford, and Reggia, 2009), and have shown that they can be integrated effectively with goal-driven control mechanisms to support problem solving (Rodriguez and Reggia, 2005; Rodriguez and Reggia, 2009). Other formulations have been explored, such as the use of potential fields that guide agent movements (Vail and Veloso, 2003; Kurihara, et al. 2005).

While a great deal has been learned about collective movements from these past studies, most past multi-agent systems have involved movements that occur in largely open spaces in which objects that act as obstacles are relatively sparse. Much less is known about collective movements in densely occluded spaces, such as urban environments where an agent would be limited to moving between buildings and to having only very local visual information. It is not even clear a priori whether or not collective movements in the latter situation are advantageous relative to independent agent movements. Further, while maintaining a partial memory of past obstacles can be useful to agent teams in relatively open settings (Winder and Reggia, 2004), it is not clear whether or not this remains the case when movements are highly constrained by ubiquitous environmental barriers.

To help clarify these issues, here we consider a multi-agent pursuit task in a highly constrained "urban environment", or simulated city. This pursuit scenario requires multiple agents to work together to capture a moving target with capabilities on par with those of the pursuing agent team. At issue is whether or not capturing the target can be facilitated through communication between agents, coordination of their movement behaviors, and experiential knowledge (working memory) of the environment. For single agents, an episodic memory can provide a benefit (Nuxoll, 2011), but our goal here is to assess whether there is a benefit from a more volatile working memory. The urban pursuit scenario we use can be related in spirit to other past predator-prey systems (Benda, et al. 1986; Alcazar, 2004; Lenzitti, et al. 2005; Zhao and Jin, 2005; Hladek, et al. 2009; Huang, et al. 2009), but is specifically oriented towards pursuit in a setting that involves highly constraining roads and buildings. Thus, most of the environment and other agents are not visible to pursuing agents, making the utility of collective movements unclear. However, it still seems reasonable to expect that some amount of inter-agent communication and coordination will provide a benefit and raises the issue of whether recall of local road/building locations can facilitate team efforts.

In this context, we examine two hypotheses: first, that coordinated collective movements can still contribute to improving team performance as would be expected, and second, that giving individual agents even a very limited working memory provides benefits in the context of ubiquitous obstructions to movements and very limited agent visibility. For the first hypothesis, we compare the performance of coordinated versus independent movements by team agents. Since agents cannot directly see one another, coordination of movements is brought about by local broadcasting of information. For the second hypothesis, we compare situations where agents have limited local memory (working memory) versus global cumulative memory (episodic memory). Assessing these two hypotheses is important not only for theoretical reasons and intellectual curiosity, but also because of its practical importance in contemporary work with semi-autonomous robotic teams. For this reason, our agents simulate physical robots in obtaining information about the local environment from a sequence of eye-level first-person images that must be interpreted by the agents as they move through the city.

Methods

Scenario, Environment, and Agent Specifications

The specific pursuit scenario considered here uses five agents and a single moving target that the agents are trying to capture. The agents move in real-valued space, but there is a discrete 32 by 32 grid of environmental features overlaid on this space (Figure 1 shows a sample city grid populated with the agents). Because agents exist in both real-valued space and grid space, they can potentially occupy the same cell as one another, but are restricted from passing through one another since each agent occupies space in the environment. The scenario ends when one of two criteria is met. If the time counter has reached 5000 time steps, the scenario ends in failure for the agents and success for the target. At any point before that, if the agents manage to surround the target such that there is either a building or agent within one cell north, south, east, and west of it, then the target is considered "captured" and the scenario ends with success for the agents and failure for the target.

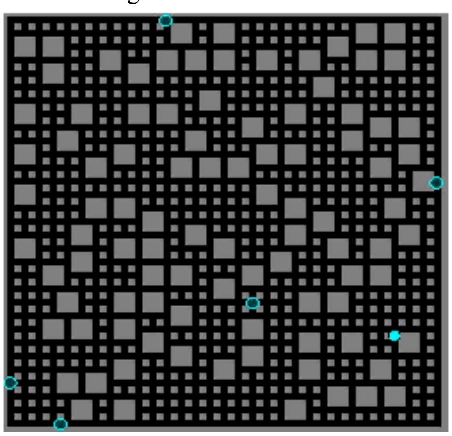

Figure 1: Sample urban environment map. The light-filled circle represents the target, the dark-filled circles represent the five agents seeking to capture it, gray regions represent small and large buildings (environmental obstacles that obstruct agent vision), and black regions represent streets.

As illustrated in Figure 1, the city consists of a grid of roads and buildings, along with a perimeter that is an unbroken wall of buildings leading to a contained setting. Given that each agent and the target use a first person perspective, they cannot see through buildings but can potentially see any distance along a road. Figure 2 displays a snapshot that is a typical example of one agent's view. As illustrated here, an agent has a relatively narrow view angle (30 degrees). The agent receives the raw image shown on the left and processes the image using self-organizing maps, trained a priori, to segment the scene as shown on the right in Figure 2 into buildings (marked "B"), roads (marked "R"), and the target object (marked "O"). Agents and target are different colors to allow for simple recognition of the target. Contiguous blocks of cells of the same type have been outlined in white in the right image. The image resolution produced by the "camera" is 128 by 128 pixels, and the scene-segmenting grid on the right has a resolution of 8 by 8 pixels per cell. When the agents move through the environment, they advance at a rate of 0.25 steps per unit time, while the target moves slightly faster with a step rate of 0.26.

Each set of trials for an agent team consists of 100 separate runs of the scenario (each with an allowed maximum of 5000 time steps). While the setup for individual runs differs from one another, the same set of trials is used for each of the different agent teams tested. Additionally, the agents and the target are not placed completely randomly in the environment. The target always begins in the center of the map, and one agent is always placed so that it can see the target from the beginning. This eliminates the need for the agents to find the target from the beginning and allows the scenarios to run more quickly. The other four agents are each placed randomly within the four different quadrants of the city, forcing them to be spread out at the beginning. None of these other agents begin the scenario seeing the target.

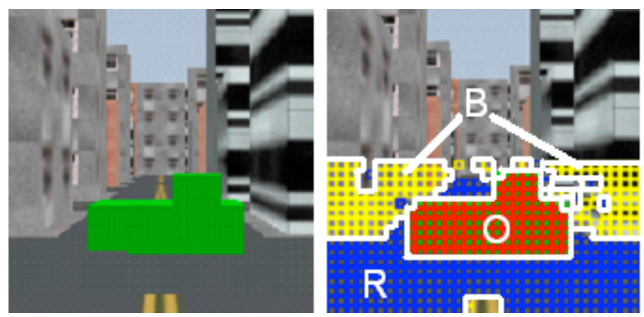

Figure 2: Sample snapshots of the pursuit environment from the perspective of an agent.

As the agents on a team receive a sequence of images of the environment as input, they follow a pursuit strategy that determines how well the team performs. Different agent teams use different strategies. These different strategies naturally show a difference in performance as certain features are included or excluded. The features that were varied between teams were communication between the agents, type of memory strategy used, and the use of movement coordination. While the strategies used by different teams varied, the target's behavior remained the same in all cases.

General Movement Strategies

Movement in the environment occurs for every agent at every time step, with a new position chosen given the agent's rate of movement and current velocity vector. Agents can move reflexively to stimuli in the environment or can place waypoints to direct their future movement towards a specific observed location, usually a point where the agent will want to make a turn. The agent is frequently required to estimate the locations of objects in the environment from its first-person view image. This is done by examining pixels along the base of a seen object. Using its height and angle of view in the vertical dimension, an agent can estimate the distance of any row of pixels in the snapshot. Using this information and the distance from the central vertical line of the snapshot, the agent is able to estimate the location in the environment.

In the simulations studied here, we are interested in assessing the relative value of agents having only a local memory (recall only information that is local in space and time) relative to agents having a cumulative memory (recall information seen since the start of a simulation. Very roughly speaking, local memory corresponds to human working memory with its very limited capacity (Cowan et al, 2005), while cumulative memory corresponds to human episodic memory (Tulving, 2002).

Accordingly, agents in a team have one of two types of basic memory available to them: local memory only (LM), or cumulative memory (CM). LM agents have a memory of what is seen in the surrounding environment, and convert some of this information into retained knowledge of the surrounding space for navigation. Thus, in addition to using the knowledge of the environment from the view window at every time step, an LM agent also keeps a local memory of its immediate surroundings. This is an n x n two-dimensional map designed as a cellular space, built by and continually updated by the agent as it moves, where $n = 3$ in the current pursuit scenario; see Figure 3. This small map is centered on the agent, but aligned with the global grid. Each cell is given a different value depending on what the agent has estimated exists in that cell from its recently observed visual information.

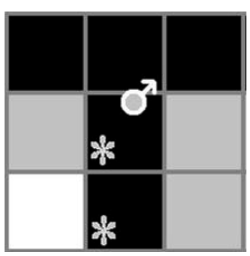

Figure 3: An example local memory map produced by a LM agent in the simulated environment, forming a three by three grid. The agent is represented by the circle with a directional arrow in the central cell. Black cells are cells estimated by the agent to be streets. Gray cells are places estimated by the agent to be buildings. White cells are unknown, meaning the agent has not acquired information about them. Black cells marked with an asterisk are areas the agent remembers passing through. As the agent moves, the center of its local memory map moves with it.

As in creating waypoints when an agent has no memory, some conversion of first-person perspective is required by LM agents to create a 2D local area map. Pixels and their locations in the environment are computed in the same way as with agents having no memory, but what is done with the information is different. For each pixel along the terrain level (approximately the lower half of a snapshot view), the location of its cell is determined relative to the agent. If this estimated location falls in a cell of the agent's local memory, then it contributes to generating a memory of the environment surrounding the agent.

The contents of LM cells (Figure 3) are determined by the agent to be in one of four possible categories: unknown (if the agent has not seen the cell's contents), visited (if the agent has been in the cell before it leaves its local memory), passable (if the cell is perceived to contain predominantly streets), and impassable (if the cell is perceived to contain predominantly buildings). These determinations shift as the agent moves through the environment. In this way, when an agent moves into a cell it has previously recognized as "passable" in its LM (i.e., when this passable region becomes a "visited" region), then all of the information currently in memory shifts to the appropriate cells, while some old information is lost for those cells that are no longer close enough to the agent, while new "unknown" areas may appear in those cells that have now become close enough.

The second CM (cumulative memory) strategy makes use of the same techniques as those of the LM agents, but it adds the ability for the agent to have a global map that it constructs from its cumulative LM memories over the course of a simulation. This global map is used to generate a sequence of planned movements from its present location all the way to an intended destination. This method makes use of much more information than the previous LM method. An example global map built by an agent is depicted in Figure 4.

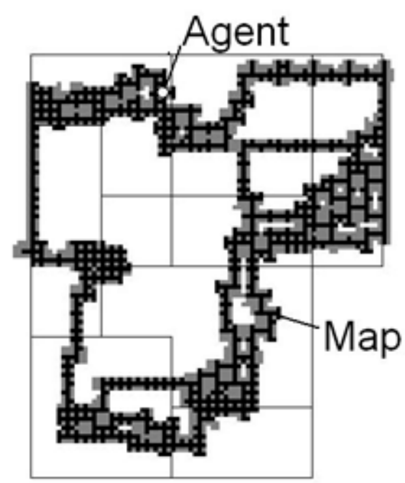

Figure 4: Representation of an example cumulative memory map produced by a single agent after touring parts of a simulated environment. Black cells are places estimated by the agent to be streets. Gray cells are places estimated by the agent to be buildings. White cells are unknown, meaning the agent has not acquired information about them. The map also includes lines denoting district boundaries of the city, which are known to the agent and can influence agent behaviors.

Like with the local memory used by LM agents, the global map constructed by a CM agent is a cellular space aligned with the world's grid. It is calibrated to be the same size as the world. It is not centered on the agent, but the agent moves through it and knows its own location at any point in time. This means the agent, aware of its own global position, is able to compute the appropriate global map changes in the cells near it, often changing nearby cell states from "unknown" to other states, as it moves through the environment. A CM agent that selects a target destination is able to plot out a sequence of waypoints across the map that will allow it to reach the target location. The planned movement path may include cells with both known and unknown content. An A* search algorithm is used to generate planned movements, where the search space is the agent's grid map, the start state is the agent's current cell, and the goal state is either a specific cell or any cell in a specified district. In generating a planned route, areas known to be impassable are excluded from the route, but known passable areas do not get priority over unknown areas, because there may be more direct routes that pass through unknown territory. This planned movement route composed of waypoints is repeatedly updated.

Agent and Target Behaviors

The mobile target and the pursuing agents have general behaviors that are programmed into each of them at the beginning of a scenario.

The target has two behaviors: patrol the city (i.e., move arbitrarily throughout the city) and evade pursuing agents. Patrolling behavior occurs when the target cannot see any pursuing agents, while evasion occurs whenever a pursuing agent is observed. Unlike the pursuing agents, the target behavior rules are the same across all test runs: use a local memory strategy while patrolling, and a cumulative memory strategy while evading. The target's memory of the environment, when using a cumulative strategy, is unlimited, while its memory of obstacles is temporary since agents rarely stay in the place the target saw them for long. The target is always moving and attempting to get out of sight of its pursuers. The target is oblivious to any communication between agents and cannot communicate with them.

The agents have two alternative behaviors: patrol the city or pursue the target. Patrolling the city occurs when an agent is unaware of the target's location. It is a high-level behavior where the agent moves about the city to cover as much unseen territory as possible. A CM agent will therefore try to move into areas it has never explored previously, while LM agents move randomly because they forget where they have previously visited. Finding the target causes an agent to announce it has seen the target when the target is first visible, and to begin pursuit.

In simulations where agents are given the ability to communicate, when an agent sees the target it will broadcast the target's location to all other agents within a broadcast range (15 grid cells). The agent will also adjust its normal patrol behavior, following the target in a manner depending on which memory strategy is in use (described below). The agent also remembers seeing the target, so if the target should turn onto a side street, the agent will continue to the last location where the target was seen and then turn to match the last remembered angle of the target. If this succeeds in returning the target to view, the agent will continue this process. If it fails, then the agent returns to a regular patrol of the environment, looking for the target it lost, unless it receives a broadcast from another nearby agent about the target's current location.

When an agent receives a broadcast of the target's location, if the agent receiving the broadcast does not also see the target, then it no longer behaves as described above. Instead, the agent treats this as a command to go to the broadcast location to assist in trapping the target. As the broadcast updates, so too does the goal location of the agent hearing the broadcast. It will then move depending on the agent memory strategy in use for this scenario. The planned movement of other agents and remembered portions of the environment's layout may also affect how the agent moves, as follows.

When an agent uses a local memory strategy, the method of pursuit is simple. If the LM agent can see the target, the agent computes the direction of the target and places a waypoint in a position in the environment that will guide it toward the target. If the agent cannot see the target, but saw it recently and remembers the target's last known location and direction, the agent will continue in the process of generating and approaching waypoints toward that location until it is less than half a cell away. When the agent reaches this threshold, it matches the target's last known direction angle in order to achieve the best chance to see if the target is still visible and to continue on the same heading that the target last had.

If the LM agent does not see or remember the target, but receives a broadcast of its current location, the agent will head to that location. The agent places a waypoint in the closest adjacent cell that takes it in the direction of the broadcast target location. As the target's position is updated in the received broadcast—assuming another agent sees it—then the agent receiving the broadcast will continue to place and approach waypoints that move it closer to the changing target location until it either sees the target or moves out of range of the broadcast. If the broadcasts cease, but the agent remembers a broadcast target location, it will continue to approach that location until a new broadcast is issued or the agent sees the target.

When the agent is using a CM strategy, the method of pursuit differs. If the agent has the target in view, it computes a path of waypoints to the target's position and follows them. If not, but the agent recently had the target in view and remembers its last known location and direction, the agent generates a path of waypoints to that location. When the agent reaches the last seen position of the target, it matches its last known direction angle. If the CM agent does not see or remember the target, but receives a broadcast of the target's current location, the agent generates a path of waypoints to that location that it will follow. Because the broadcast target location the agent receives will potentially update as an announcing agent tracks the target through the environment, the waypoint paths generated by the CM agents relying on the broadcast information are also updated. Again, if the broadcasts cease, but the agent remembers a broadcast target location, it will continue on the path to that location until a new broadcast is issued or the target enters its view.

Inter-Agent Communication and Coordination

Basic agent communication consists of broadcasting the estimated location of a target, when the target is visible, to all other agents within its broadcast radius. While not a completely accurate position of the target, especially if the agent is far away from the target, this broadcast position can still point other agents within the broadcast range toward the correct general area, giving them a greater chance of finding the target and making their own broadcast.

In more open environments, coordination between agents is usually achieved through accelerations based on direct observation of movements of other nearby agents, leading to collective movements (i.e., to agents moving in a "flock"). In contrast, due to the cramped nature of this environment, where agents are less apt to be able to see other agents, such coordination is less likely to prove beneficial. This is particularly the case with the very limited view angle of each agent. This view angle, when coupled with the many building obstacles, means that the agent can effectively see only straight ahead along the road they are using.

Nevertheless, avoidance acceleration remains useful in this scenario in situations where the agents might collide and have difficulty navigating around one another. It also helps to separate them, allowing them to spread out more and cover more of the environment. The radius of this avoidance influence is typically kept quite low (one cell length). Because this could interfere with the pursuit of a target, the avoidance influence is only factored in when the agent does not see the target and is not approaching the last place it remembered seeing the target. If the agent is close enough to another agent to experience this influence, then it alters its course to a waypoint that is in a valid adjacent cell that is closest to its avoidance vector. The avoidance vector is computed such that the agent only avoids the closest neighboring agent in its avoidance radius.

In addition to this influence, there is another method of agent coordination that can be implemented in the case of a cumulative memory strategy. The chances of a successful outcome are increased if the agents are approaching the target from different directions; this increases opportunities to surround and thus capture the target. If the paths can be coordinated so they cross as little as possible, then the agents' performance should improve. This can be achieved by having an agent broadcast its planned path when the agent has planned a path to the target's location, whether it was seen, remembered, or received through a broadcast. When planning a path, the cost of waypoints increases when a location is on another agent's path as well, making it more likely the agent will attempt to find another direction from which to approach the target. Additionally, the cost of any cell at the end of another agent's path that is adjacent to the broadcast target's current location has an even higher cost, increasing the chances that agents will look for valid paths that approach the target from another direction.

Results

In the baseline scenario, there was no communication between agents and therefore no coordination of agent positions or movements. Pursuing agents may visually perceive one another in the environment, but this does not influence their behavior. The agents use a local memory strategy for both their patrol and their pursuit behaviors. Given the limit of 5000 time steps to complete the task of capturing the target, the agents were successful in 70% of the trials (see Figure 5). The mean completion time for this set of simulations was 2904 time steps. This indicates that while agent teams can often solve the problem while the agents move independently, there is still room for improvement in that close to a third of the time they are unsuccessful.

We varied the agent behaviors away from baseline in a variety of ways to observe the effects. In a second scenario, agents were given the ability to communicate (Figure 5). Once communication is enabled, agents that see the target broadcast its location, and other agents that hear it have some knowledge of where to intercept the target. This is the only difference between this scenario and the baseline scenario, with the communicating agents experiencing a 97% success rate and a mean completion time of 804 time steps, which is a significant improvement (a paired t-test gives $p < 0.05$) over the original mean of 2904. A slight further improvement in the mean completion time is also gained when a small agent avoidance influence of radius 1 is also included in this scenario (see Figure 5); accuracy rises to 98%, and the mean completion time drops to 744. These three results support our first hypothesis that agent cooperation/communication continues to be effective even in the context of densely occluded movement spaces like those used here.

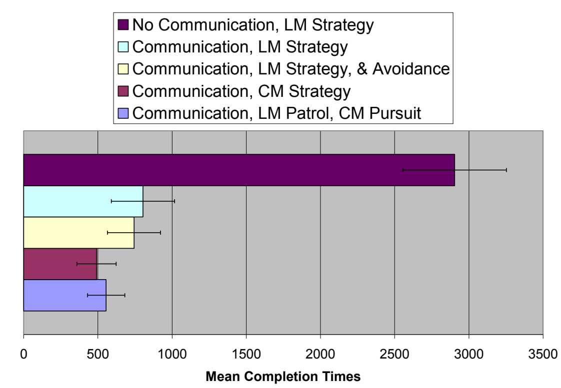

Figure 5: Mean completion times for agent teams using several different strategies as described in the text. The worst scenario was the baseline LM strategy where no communication occurs between agents (70% success rate). There is significant improvement whenever agents are given the ability to communicate. There is further improvement when the agents adopt a cumulative memory strategy for either all of their behaviors or just the pursuit behavior. The error bars here represent 95% confidence intervals.

The three scenarios described so far have used solely local memory strategies for both patrol and pursuit behavior. Either or both of these strategies can be changed to using a cumulative memory strategy. While a cumulative memory strategy would not be expected to improve patrol behavior significantly, it should improve the performance of pursuit, because agents in this latter mode are actively trying to get somewhere as quickly as possible, as opposed to simply exploring. When the cumulative memory strategy is applied to both patrol and pursuit behaviors in the context of communicating agents, the success rate increases to 99% and its mean completion time decreases to 491 time steps (Figure 5). When the cumulative memory strategy is applied to only the pursuit behavior, all trials successfully completed before the time limit, and the mean completion time was 556 time steps. Both of these mean completion times are a significant improvement over the local memory strategy, but do not have a significant difference from one another. As expected, a cumulative memory strategy was able to improve upon a strategy where agents were able to acquire less information. However, it is apparent that this is also useful in a scenario where multiple agents are performing a task together. Figure 5 also shows these additional results. The results are consistent with our second hypothesis that even a use of local memory can substantially improve agent performance. As seen in Figure 5, agents with local memory improve almost as much as those with cumulative memory relative to baseline.

To further examine whether coordination is a useful feature for pursuing agents in densely occluded environments, the scenarios that used cumulative memory were also tested to see what improvement could be gained from introducing an avoidance influence or from allowing agents to broadcast their planned movement paths so that they could coordinate in an attempt to surround the target. Figure 6 shows the results for these scenarios.

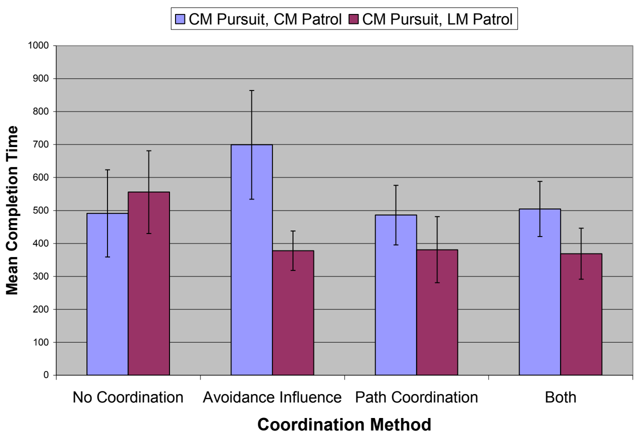

Figure 6: Mean completion times for the different coordination methods in scenarios with different memory strategies involving cumulative memory. In 100% of these simulations the pursuing agents completed the task within the time limit except for the cumulative memory pursuit and patrol strategy with only the avoidance influence, where 99% of the tasks were completed. The error bars are 95% confidence intervals.

The results here differ substantially depending on the memory strategy used. In the scenarios where cumulative memory is used for both of the two major agent behaviors (pursuit and patrol), adding these coordination methods, either had almost no effect or made the performance significantly worse (as in the case with an avoidance influence). However when a local memory strategy is used for the patrolling behavior, both coordination methods and their combined use yield a significant improvement. Thus the benefit of these coordination methods appears to be mitigated when using a cumulative memory patrol. For path coordination, this is likely due to interference between patrol paths and pursuit paths, which unexpectedly caused agents to influence each other even in situations where there is no obvious benefit from this interaction. The avoidance influence also evidently interferes with the patrol behavior, probably because the agents create distant goals for themselves when patrolling using cumulative memory. When paths cross, it is more difficult for two agents with different goals to reconcile them with just this influence. This situation is less likely to occur when agents are in pursuit of the target, because if they are in close proximity to one another, they are likely heading in the same direction and do not have conflicting paths.

The above results demonstrate that there is usually a large benefit from allowing communication between pursuing agents, a substantial benefit from allowing coordinated agent behaviors, and a substantial benefit from using a memory strategy with more information. However, path coordination relies on the use of cumulative memories, which substantially determine the details of the planned paths. Is the benefit seen with path coordination due mainly to the coordination, or does the inherent presence of memories, influencing the paths, also significantly contribute to the improvement? This issue was addressed by comparing a scenario that features the minimum amount of memory necessary for the agents (i.e., knowing the contents of four cells, the cell the agent currently occupies and the three additional cells in front of them), against scenarios with increasing memory capacities, all of which use path coordination.

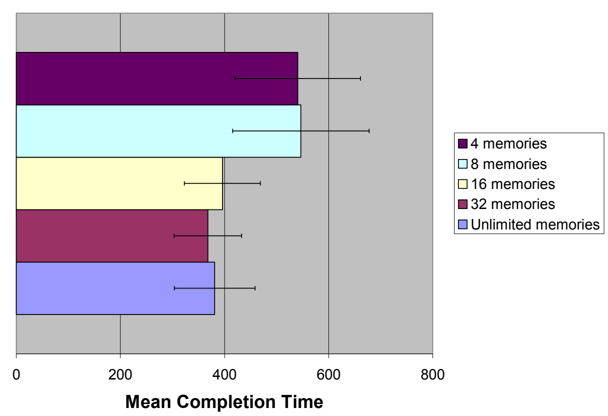

Figure 7: Mean completion times for the scenarios featuring limited memory (4 memories = contents of 4 cells recalled, etc.). Not included in this chart are the results where the agent has a limit of no memories (i.e. remembered cells). These latter agents perform much worse, with below 100% accuracy and a mean completion time in the thousands. The error bars are 95% confidence intervals.

A scenario was tested in which agents used local memory strategies for patrolling and cumulative memory strategies with path coordination for pursuit, but with a requirement that an agent remove a random old memory cell when adding a new memory based on visual data when the memory capacity is exceeded. This effectively kept a limit on the size of an agent’s memory, although memory contents could change and be updated as the agent moved through the environment. During this test, as memory capacity was increased by a factor of two, the mean time to success dropped. By the time the threshold for removing old memories is 32, there was a significant improvement (a paired t-test gives $p < 0.05$) over the scenario with minimum memory capacity, and no significant difference between its performance and that of the scenario with unlimited cumulative memory. These results, depicted in Figure 7, again demonstrate that a limited, incomplete memory of the environment can be very effective.

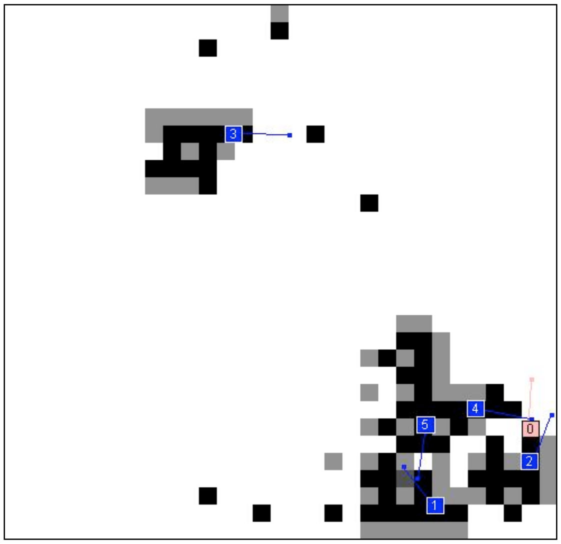

Figure 8: Composite depiction of the local maps constructed by each of the agents in the environment. An individual agent's local memory capacity limit is 24 in this case; what is shown here is the memory of the entire agent collective, which is effectively the "sum" of the individual agent memories. The target's memory, which is always unlimited, is not displayed. Black areas are streets remembered by at least one agent, gray areas are buildings remembered by at least one agent, and white areas are unknown to anyone. Some areas that are in between the normal shades of black and gray are places where the agents have differing memories of what is there. Agents are indicated by numbered boxes outlined in white. The box numbered 0 is the target. The agent numbered 2 is pursuing the target, and agents 1, 4, and 5 are in agent 2's broadcast range.

Finally, we consider the question of why agents with just a local memory perform almost as well as those with cumulative memory. Figure 8 displays a representative composite map at a single time step of the remembered cells of agents with limited memories. Even though agents do not tend to remember distant locations given the frequent updates of memory, the coordination of paths in the collective memory strategy has an impact because the agents are often trying to find ways to effectively spread out and surround an area they have recently explored and where they are told the target is estimated to be located. Knowing more than a few features is very useful for this, and shows that a memory strategy somewhere between highly local and completely cumulative is almost as effective as the fully cumulative CM strategy.

Discussion

In terms of coordination, we found that agents without any type of communication or coordination had far worse performance than agents where communication about the target location and various types of coordination (collision avoidance, planned movement path overlap avoidance) were introduced. While not consistent in improving performance, both types of coordination did make significant improvements in situations where two different memory strategies were used for pursuit and the patrol. While this is not surprising with the path coordination, which was designed so the agents would tend to surround the target, it is surprising that the avoidance influence had such a significant impact. It should be noted that when the radius of the avoidance influence is increased, the effect is detrimental, so avoidance proved effective mostly as a means of ensuring agents did not get stuck when they collided with one another.

With respect to agent memory, there are three main conclusions to emerge from the computational experiments done with this work. First, the results support the hypothesis that adding an individual working memory of the obstacles encountered by an agent team can significantly improve both its success and efficiency in accomplishing the pursuit scenario. This occurred even in this simulated environment where agent input was limited to a sequence of images from a first-person perspective, and even when the agent team was already benefiting from communication and coordination. Second, the results also support the hypothesis that even when the size of an individual's working memory is severely limited, the agent team as a whole still experiences a significant improvement in its efficacy and efficiency. Third, the results indicate that a different memory strategy for different tasks works best for the agent team. Local memory is found to be better for the patrol behavior, while a cumulative memory strategy is found to be better for the pursuit behavior. As expected, granting the agents a local communication, so that they could broadcast the estimated location of the target, improved performance greatly, allowing agents to converge on the target more easily.

More interesting than the general advantages of the cumulative over the local memory strategy is the fact that a combination of strategies tended to work quite well. In cases with coordination, it worked much better for the pursuing agents, where they would patrol using a local memory strategy, but pursue the target with a cumulative memory strategy. This gave them flexibility in searching, while allowing them to make decisions, potentially informed by memories, when trying to quickly reach the target's location upon hearing a broadcast.

Of all the observations in this study, perhaps the most important result is the finding that giving individual agents even a limited memory of the environment could give the agent team as a whole a significant improvement in performance, even when improvements were already present due to communication, path generation, and coordination. What was surprising about this is that the individual agent's memory capacity can be so small, because in viewing the agent team as a system, the resulting collective memory consists of the memories of all its agents combined with one another, albeit distributed between individuals (as illustrated in Figure 8). This is why agent teams with local memories seemed to do so well, and it suggests that much of the information stored by cumulative memory agent teams was of very limited usefulness. These results are consistent with those obtained in an earlier study with a much simpler environment and much simpler agents (Winder and Reggia, 2004).

Acknowledgement: Supported in part by funding from the United States Government.

References

Alcazar, J. (2004). A simple approach to the multi-predator multi-prey domain. In *International Conference on Complex Systems*, Boston.

Atherton, J., Hardin, B., and Goodrich, M. (2006). Coordinating a multi-agent team using a multiple perspective interface paradigm. In *Proceedings of the AAAI*, pages 47-51.

Bayazit, O., Lien, J-M., and Amato, N. (2002). Roadmap-based flocking for complex environments. In *10th Pacific Conference on Computer Graphics and Applications*, pages 104-113.

Benda, M., Jagannathan, V. and Dodhiawalla, R. (1986). On optimal cooperation of knowledge sources, Technical Report, Boeing Advanced Technology Center, Boeing Computer Services, Seattle.

Couzin, I., Krause, J., Franks, N., and Levin, S. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513-516.

Cowan, N., Elliot, E., Saults, J., et al. (2005). On the capacity of attention. *Cognitive Psychology*, 51:42-100.

Hladek, D., Vascak, J., and Sincak, P. (2009). Multi-robot control system for pursuit-evasion problem. *Journal of Electrical Engineering*. 60(3):143-148.

Huang, F., Wang, L., Wang, Q., Wu, M., and Jia, Y. (2009). Coordinated control of multiple mobile robots in pursuit-evasion games. *American Control Conference.* pp. 2861-2866.

Kennedy, J., Eberhart, R., and Shi, Y. (2001). *Swarm Intelligence*. Academic, San Diego, CA.

Kurihara, K., Nishiuchi, N., Hasegawa, J., and Masuda, K. (2005). Mobile robots path planning method with the existence of moving obstacles. In *10th IEEE Conference on Emerging Technologies and Factory Automaton, 2005*, pages 195-202.

Lapizco-Encinas, G., Kingsford, C., and Reggia, J. (2009). A cooperative Particle Swarm Optimization Method for Side-Chain Packing, *Proc. IEEE Swarm Intelligence Symposium*.

Lenzitti, B., Tegolo, D., Valenti, C. (2005). Prey-predator strategies in a multiagent system. *Proceedings of Seventh International Workshop on Computer Architecture for Machine Perception*, pages 184-189.

McCook, C., and Esposito, J. (2007). Flocking for heterogeneous robot swarms: a military convoy scenario. In *39th Southeastern Symposium on System Theory*, Macon, GA.

Nuxoll, A. and Laird, J. (2011). Enhancing intelligent agents with episodic memory. *Cognitive Systems Research*, In Press. doi:10.1016/j.cogsys.2011.10.002

Reynolds, C. (1987). Flocks, herds and schools. *Computer Graphics*, 21:25-34.

Rodriguez, A. and Reggia, J. (2005). Collective-movement teams for cooperative problem solving. *Integrated Computer Aided Engineering*, 12:217-235.

Rodriguez, A. and Reggia, J. (2009). A distributed learning algorithm for particle systems. *Integrated Computer-Aided Engineering*, 16:1-20.

Tulving, E. (2002). Episodic memory: from mind to brain. *Annual Review of Psychology*, 53:1-25.

Vail, D. and Veloso, M. (2003). Dynamic multi-robot coordination. *Multi-Robot Systems*, pages 87-100.

Winder, R. and Reggia, J. (2004). Using distributed partial memories to improve self-organizing collective movements. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(4):1697-1707.

Zhao, D., Jin, W. (2005). The study of cooperative behavior in predator-prey problem of multi-agent systems. In *Autonomous Decentralized Systems. ISADS 2005. Proceedings*, pages 90-96.