Roadmap-Based Pursuit-Evasion in 3D Structures *

Samuel Rodriguez, Jory Denny, Aditya Mahadevan, Jeremy Vu, Juan Burgos, Takis Zourntos, and Nancy M. Amato

Parasol Lab, Dept. Computer Science and Engineering, Texas A&M University
{sor8786,jorydenny,maditya,cvu,jburgos,takis,amato}@tamu.edu

Abstract. We present an approach to the pursuit-evasion problem which is applicable to complex, multi-level environments. Studying each aspect of this problem in 3D structured environments is a distinct extension over many previous approaches. We also utilize our roadmap-based approach to multi-agent behavior when tracking agents of interest. Results are presented in three multi-level environments to highlight the search, pursuit and evasion components of the problem.

Keywords: pursuit-evasion, 3D structured environments, roadmap-based navigation

1 Introduction

The pursuit-evasion game involves one set of agents, the pursuers, locating and chasing another group of agents, the evaders, which attempt to hide and flee from the pursuers. When the agents are equipped with the ability to cooperate and a reasonable ability to track other agents, this competitive scenario becomes one often seen in the real world. Studying these aspects have a wide range of applications, and are one reason for the prevalence of the pursuit/evasion problem in a wide range of disciplines including computer graphics, games and robotics.

While a great deal of work has been done in pursuit-evasion, many solutions to the problem have a set of restrictions and assumptions about the environment or the agents involved to make it more manageable. For example, in the case of graph-based approaches, the agents are limited to moving on the nodes and the edges [1]. In geometry based approaches, the environment is often limited to being polygonal [2]. In fact, the types of visibility restrictions that agents can handle is often one of the main limiting factors [3, 4]. While providing a means for developing a more complete solution, these assumptions limit the scope of the problem and consequently the applicability of the techniques.

The main contribution of this work is our:

– Versatile approach for agent-based pursuit-evasion games

* *This research supported in part by NSF Grants CRI-0551685, CCF-0833199, CCF-0830753, IIS-096053, IIS-0917266 by THECB NHARP award 000512-0097-2009, by Chevron, IBM, Intel, Oracle/Sun and by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

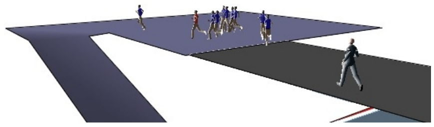


Fig. 1. An agent pursuing a group of agents attempting to flee.

– Agent tracking integrating roadmaps with probability model
– Analysis of 3D building scenarios

Our roadmap-based approach, integrated with agents equipped with heuristic behaviors, allows us to handle complex scenarios. To the best of our knowledge, the complexity of the environments that we are able to handle and the analysis done in these environments has not been done before.

2 Related Work

Here we describe a selection of work that is relevant to the pursuit-evasion problem. One major distinction between each variant of pursuit-evasion are how the problem is encoded with respect to environment representation, visibility restrictions and capture conditions. Early graph-based approaches[1, 5] limit the movement of the pursuer and evader to the graph, and often restrict visibility to the pursuer's current node or adjacent nodes and edges. Instantaneous movement between nodes is assumed. Unrealistic constraints and movement models limit the applicability of these approaches. In other work, such as in [2, 6], the environment is restricted to be composed of polygonal regions. Visibility is considered to be infinite along a straight line, limited only by obstacles and the environment boundary. These assumptions impose limitations on these approaches although it does allow for bounds to be found for the number of pursuers. However, environments cannot always be encoded as simple polygons, and it is not always desirable to do so. Evaders in these approaches are usually assumed to have an infinite velocity, thus ignoring the behavior of the evader.

Roadmap-based pursuit-evasion, where the pursuer and evader share a roadmap and play different versions of the pursuit-evasion game is described in [7]. These games include determining: 1) if the pursuer will eventually collide with the evader, 2) if the pursuer can collide with the evader before the evader reaches a goal location, and 3) if the pursuer or evader can collide with the other in a dog fight scenario. In our work, we consider capture conditions based on a pre-defined distance value.

Our roadmap-based approach to pursuit-evasion was first proposed in [8] which is applicable in 3D environments. The work focused on different types of 3D problems that could be studied using the roadmap-based framework which included terrains, multi-level environments, in crowds of agents and for actual robots. The work did not include in depth analysis and did not include a means of tracking agents when they are no longer visible.

The benefits of integrating roadmap-based path planning techniques with flocking techniques were explored in [9, 10]. A variety of group behaviors, including exploring and covering, were simulated utilizing an underlying roadmap.

The idea of using probabilistic shadow information spaces for targets which move out of a pursuing agent's field-of-view has also been considered [11]. This is done in simply connected, polygonal environments. Other probabilistic approaches to pursuit evasion attempt to reason about and find pursuers using agents with limited sensing capabilities. In [12], a probabilistic framework is described in which a swarm of autonomous agents pursues one or more evaders using a greedy policy. The environment is represented as a set of grid cells, and all events (motion, sensing) occur at equally spaced intervals. Pursuers can detect evaders occupying the same cell as themselves with perfect accuracy, and evaders at adjacent cells with some inaccuracy. A similar environment is utilized by both [13] and [14], using both aerial and ground vehicles.

Different forms of collaboration between searching agents has been considered where agents utilize frontier information [15], and with communication to switch between roles needed by a team exploring the frontier [16]. The level of visibility between agents plays a role in what the agents detect in the environment and what kinds of environments can be handled. The problem of restricting or limiting the amount of visibility is considered in [3] with the field of view being variable in [4] which allows for bounds to be found on the number of searchers needed.

This pursuit-evasion problem is similar to the camera tracking problem where an external camera has the goal of tracking target agents through virtual environments. In [17], an approach is described which extracts paths to goal locations through a roadmap with the paths weighted on a visibility criteria and distance. While tracking approaches are similar, they assume the target's location is always known, that the target is not taking evasive actions, and visibility and navigation limitations of the camera are not considered.

Our representation of the environment and the probability model we present does not suffer from most of the restrictions described. For instance, complex environments are discretized and represented in our roadmap, which also guides transitions and movements, however we compute visibility throughout the 3D environment considering agent capabilities and limitations whereas most approaches are limited to 2D.

3 Framework Overview

Our roadmap-based approach allows us to handle problems that have not been considered before. This includes agents operating in structured 3D environments such as a multi-level building. Here we describe our overall system and approach to this problem. Our framework also allows us to study each behavior in the system, including search, pursuit, and evasion.

3.1 Problem definition

We consider a general form of the pursuit-evasion problem in which one set of agents, the pursuers H, attempt to capture another set of agents, the evaders

E, within a bounded environment. The environment consists of surfaces the agents can travel on and obstacles, both of which restrict visibility. An evader $e \in E$ is considered captured if at least one pursuer $h \in H$ fulfills a capture distance requirement $d \geq 0$. The pursuers first search for a target agent and once detected, attempt to create a pursuit plan in order to make a capture. The evading agents flee from the pursuing agents they detect and find suitable hiding locations when no pursuer is detected. One of our goals of this paper is to improve agents' decisions soon after their adversaries go out of view.

3.2 Approach

We have developed a real-time simulation system in which agents are guided by roadmap paths obtained from complex pursuit/evasion strategies, with agent interactions occurring. We have designed and implemented this infrastructure for handling the following information: agents, behaviors, the environment, groupings, and relationships between groups.

At each step of the simulation, each agent updates it's sensory information. The agent's ability to detect other agents in the environment can be affected by a number of factors. In simple 2D problems, the agent's visibility to other agents may only be restricted by the obstacles in the environment. Agent capabilities can also determine the amount of the environment that can be sensed by setting a maximum view radius and angle. In 3D environments, the surfaces and obstacles composing the environments, may also block visibility from one agent to another. Visibility may also be restricted by other agents in the environment.

The behavior module that the agent executes will determine how the agent reacts. These behaviors determine the actions that the agent or groups of agents take. At each time step, all agents update their state (position, velocity) in the environment based on the last plan that was either generated or verified by the behavior rule according to their capabilities (maximum velocity and acceleration). The state is then resolved of collisions in the environment.

3.3 Roadmaps

We use the roadmap as an approximation of the environment. The roadmap allows an agent to find routes through the environment as it represents the connectivity of the free space in the environment. When selecting routes through the environments, paths can be selected that avoid certain areas or are biased towards other areas. We also use the roadmap in our model which tracks agents of interest that have been encountered.

Roadmaps are graph representations of an environment that encode feasible, collision-free paths through the environment. Agents navigating in environments consisting of multi-level surfaces (buildings) need the ability to map these spaces. Agents encode valid movements on the surfaces by first sampling collision-free nodes on each surface. Connections are allowed between nodes on the same surface in which the straight line along that surface, projected to a 2-dimensional plane, remains completely within the projected polygon and is not in collision with objects on the surface. Connections are then made between surfaces that are connected based on an input configuration.

4 Agent Tracking Probability Model

This section describes the probability model used in order to track an agent of interest. The model exploits the roadmap representation of the environment to model the potential movements of the agent of interest.

This differs from prior approaches that use a probability model. First, we use an arbitrary roadmap that is not dependent on a regular discretization of the environment. Instead, since nodes are randomly sampled, the discretization is dictated by the roadmap. Secondly, the roadmap approach allows more realistic assumptions about agents' capabilities and limitations, considering where the agent believes the agent being tracked may move.

A Voronoi region is a partitioning of a space into a discrete set of objects within it. Each region consists of points that are closer to the object representing that partition than to any of the other objects. We can consider each node of our roadmap to be the object representing a Voronoi region. Also, each node has a probability associated with it representing the probability the agent being tracked is residing in that region.

We can consider each node of the roadmap to represent a disjoint subset of points: every point in the free space of the environment maps to the node closest to it. This mapping partitions the environment into regions represented by the roadmap nodes. The edges of the roadmap represent the available transitions an agent can make between regions. There is also an implicit self-edge, meaning an agent can remain in the region during a transition. Due to agent capabilities, agents in a common region represented by a node need not be visible to one another and agents visible to one another are often not in the same cell.

An agent in the environment exists in one implicit Voronoi region at any time. Since the agent has bounded speed, it moves a finite distance in a single time step. After its movement, it is either in the same region it was previously in, or it has moved to an adjacent region. Each region can be thought of as a state. After every time step, an agent continues to be in its previous state or has transitioned to a neighboring state. Since each region is represented by a roadmap node, we can consider an edge between any pair of nodes to represent a transition between two states.

We can therefore model the movements of an unseen agent by assigning transition probabilities to each node n and its successors S. Probability is distributed in a simple manner. Some amount of the probability of a node, $P_{n,before}$ remains on a node, $P_{n,new} = P^{'}_{n,new} + 1/(|S|+1) * P_{n,before}$. The rest will be distributed equally among its successors, $P_s = P^{'}_s + 1/(|S|+1) * P_{n,before}$, where P_s is the new probability associated with the node for the updated model. This probability model is updated at each time step in our simulation to keep accuracy of the model. We define the temperature of a node n to be $T(n) = P_n + \sum_{s \in succ(n)} P_s$, where P is the probability at a node. This node is called a hotspot if $T(n) > \epsilon$, ϵ being a threshold value.

5 Search Strategies

The searching behaviors that the pursuit agents use to locate a target can greatly influence the pursuing agent's effectiveness. We have implemented a number of heuristic, roadmap-based searching behaviors which allow groups of agents to utilize the roadmap to effectively cover an environment. The goal of the searching behaviors is for the agents in the group to visit as much of the environment as possible. The effectiveness of the search behavior has an underlying dependence on the quality of the roadmap used by the agents. Knowing and analyzing the effectiveness of each type of searching behavior would allow a game designer to get a handle on the expected coverage of the environment. It adds a degree of realism if a game player were the target against adversarial searching agents.

Using *basic* roadmap exploration, agents select the next area to follow by evaluating edge weights of the node nearest to their current location. Another form of *basic* exploration allows an agent to select a node in the roadmap at random and find a path to this node using the roadmap. Areas that have been visited by agents can be communicated to influence edge weights in the roadmap and paths that are selected.

Exploring the *frontier* between what has been observed and what is still unobserved is a traditional approach to searching [18]. We extend this approach by marking nodes in the shared roadmap with labels of clear, frontier and unclear nodes. In this way, the agents benefit from work done by other agents in clearing the environment by utilizing the roadmap. This also represents a form of communication since the roadmap is shared and marked when cleared.

Probabilistic exploration of the environment is when an agent uses its tracking probability model in order to explore the environment to search for the tracking target. In this searching behavior the agent will plan its motion towards the node of the roadmap with the highest temperature as described in Section 4.

In multi-level environments, the frontier behavior may be paired with a *level-by-level* search so the agents can search each level in a coordinated way. We allow the scenario designer to specify the order which agents should search and attempt to clear the environment in order to see the effect of different strategies. This also allows agents to be assigned with searching certain portions of the environment.

6 Pursuit Behavior

We consider a general pursuing strategy for an agent. The algorithm has four stages: locating the target, creating a pursuit plan, enacting the plan, and evaluating pursuit status. In our specific pursuit behavior, when a target is visible the pursuer will create a pursuit plan in order to move toward the target. If a target is not visible then the pursuer will search for a target. Searching is defined by one of the behaviors in Sec. 5.

When using the probability model, the pursuit agent will be able to continue following the target even if the target goes out of view. If this happens, the probability model is queried to find the node where the target is most likely to be. If this node is a hotspot, it is considered a target; if not (i.e. temperature is below a threshold ϵ), it is simply a suitable location to guide searching for the target. By considering a hotspot to be a target, a more direct route will be taken to attempt a more timely capture.

7 Evasion Behavior

The general evasion algorithm attempts to reduce the likelihood of being spotted by opposing agents by utilizing an evasion strategy. If it is spotted, the behavior attempts to evade opposing agents by moving out of their view. In this behavior, the agent samples potential hiding spots from the environment. These positions are then evaluated using a scoring function, depending on whether or not pursuers are present. The new goal and path are updated if a location is found that sufficiently improves the score. The score of a hiding spot is generated considering: distance and direction to pursuing agents, distance to boundaries, and the amount the location is obscured by objects in the environment to other hiding spots. Each component can be given a weight and the score composed to create an evasion strategy.

When the probability model is integrated into the evasion behavior, we augment the evaluation process using information from the probability model. First, the agent queries the probability model to find all of the hotspots; that is, the agent determines the areas where it believes pursuers are most likely to be found. It then scores each hiding spot using a utility function that takes into account the distance between the potential hiding spot and every hotspot in the environment, the temperature of each of the hotspots, and the visibility of the potential hiding spot with respect to the hotspots. By actively taking into account the hotspots in the environment, the agent attempts to preemptively evade and hide from pursuers based on where it believes the pursuers most likely are at the time.

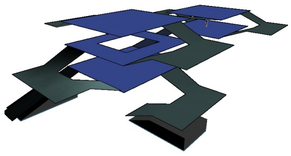
8 Experimental Results and Discussion

In this paper we present three basic comparative studies: a) a comparison of search strategies, b) a basic comparison when using probability model, and c) a more complex scenario in a two level office building. All scenarios were allowed to run a finite number of time steps in our framework. There was no other stopping criteria for the simulation.

8.1 Scenario: Immobile Targets

This scenario takes place in a two column building, shown in Table 1. This scenario involves five pursuers searching for 100 immobile targets. The immobile targets are randomly scattered throughout the environment. The pursuers' roadmap is sufficiently sampled and connected to represent the free space of the environment. The pursuers use one of the searching behaviors as described in Sec. 5. For this scenario, since the evaders are immobile we recorded the number of captures, and the percent of time these Evaders stay hidden (Averaged over all evaders).

In this scenario, we see how the various behaviors perform. The frontier behavior performed the best for this scenario. It captured on average 99.6 out of 100



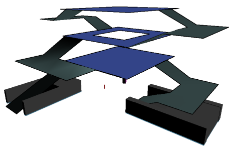
Search Behavior	Num. Captures	Per. Time Hidden
Basic	93.0	0.25
Frontier	99.6	0.21
Level	94.6	0.26
Probabilistic	83.0	0.31

Table 1. We show how each searching behavior performed by comparing the number of captures and the percent of time the evaders stay hidden in a multi-level building example with two columns of levels.

targets. The frontier behavior performed better than the other various behaviors because it uses implicit communication to better explore the environment. Both the Level and Basic search strategies perform about the same. These do not use any communication, so some of the agents end up re-exploring areas already explored by their fellow pursuers. This explains the increase in percent of time hidden as compared to the frontier behavior, 0.26 and 0.25 as compared to 0.21 for frontier. The probabilistic search performs poorly in this scenario because it is modeling moving agents, and has no communication to share models between agents. Since the behavior models moving agents, it re-explores areas which it has visited more often than the covering or the level search might do.

8.2 Scenario: Tracking

In this scenario, we compare a basic search with and without the probability model to show the base effectiveness on one pursuer. One pursuer tries to capture 10 evaders. The pursuer is given a slight advantage of speed, while the evaders have an improved view radius. This experiment was done in a simple three level building environment, in Table 2. Here we compare not only the number of captures but also the total time spent chasing agents.



Search Strategy	Prob. Model	Num Captures	Total Chase
Basic	N	6.8	2264
Basic	Y	8.6	3630
Probabilistic	Y	5.9	3011

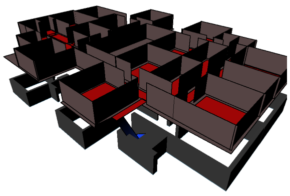
Table 2. A comparison of search with and without the probability model in a three-level building example.

From this experiment we can see the base effectiveness of the probability model for tracking. The evaders advantage in view radius allows them to move more effectively to get out of view of the pursuer, allowing the model to show its effectiveness. We see an average improvement in the number of captures of 1.8.

Also seen, is an improvement in the total chase time. Agents with the probability model can chase agents better, and we see that Basic and Probabilistic searches using the model have better chases than without the model, 3630, and 3011 as compared to 2264.

8.3 Scenario: Office Building

In the office building scenario, five pursuers attempt to capture twenty evaders. Evaders have an advantage of speed, but the view distance of the agents are equal. Evaders can effectively move around corners and out of view readily as to make capturing highly difficult. This scenario takes place in a two level office building with many rooms, in Table 3. We compare the number of captures and total chasing time of three search strategies with and without the probability model.



Search Strategy	Prob. Model	Num Captures	Time Chasing
Basic	N	14.7	16,594
Basic	Y	16.2	21,680
Frontier	N	15.6	15,816
Frontier	Y	16.8	20,426
Level	N	14.4	14,527
Level	Y	15.0	18,729

Table 3. Comparison of the probability model for searching behaviors in a complex office building.

A continued effectiveness of the tracking probability model can be seen in Table 3. In all searching behaviors, using probabilistic tracking improves the number of captures and chase time. The level search did not see as much improvement in the number of captures as the others did, but it still increased the total chase time, which shows an improvement with the model.

From each of the experiments shown, we see that while using the probability model alone may not result in a good searching strategy, when combined with traditional searching strategies, it becomes an effective method of tracking and capturing, even in highly complex multi-level environments.

9 Conclusion

By modeling realistic pursuit and evasion games in 3D, we can improve the scenarios that can be explored and extend the potential applications. In this paper we compared four search strategies. From our experiments we can see some benefits of not only implicit communication with a search strategy, such as the frontier behavior, but also the benefits of integrating a tracking model even in an approximated representation of the environment. These two could be effectively combined to make a hybrid search strategy for more interesting behaviors.

References

1. T. D. Parsons. Pursuit-evasion in a graph. In *Theory and Applications of Graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 426–441. Springer, 1978.
2. Leonidas J. Guibas, Jean claude Latombe, Steven M. Lavalle, David Lin, and Rajeev Motwani. Visibility-based pursuit-evasion in a polygonal environment. In *International Journal of Computational Geometry and Applications*, pages 17–30. Springer-Verlag, 1997.
3. V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion with limited visibility. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 1060–1069, 2004.
4. B. P. Gerkey, S. Thrun, and G. Gordon. Visibility-based pursuit-evasion with limited field of view. *Int. J. Robot. Res.*, 25(4):299–315, 2006.
5. Andreas Kolling and Stefano Carpin. Pursuit-evasion on trees by robot teams. *Trans. Rob.*, 26(1):32–47, 2010.
6. Steven M. Lavalle, David Lin, Leonidas J. Guibas, Jean claude Latombe, and Rajeev Motwani. Finding an unpredictable target in a workspace with obstacles. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 737–742, 1997.
7. V. Isler, D. Sun, and S. Sastry. Roadmap based pursuit-evasion and collision avoidance. In *Proc. Robotics: Sci. Sys. (RSS)*, 2005.
8. Samuel Rodriguez, Jory Denny, Takis Zourntos, and N. M. Amato. Toward simulating realistic pursuit-evasion using a roadmap-based approach. *Lecture Notes in Computer Science*, 6459:82–93, 2010.
9. OB Bayazit, J M Lien, and N. M. Amato. Better group behaviors in complex environments using global roadmaps. In *Artif. Life*, pages 362–370, Dec 2002.
10. OB Bayazit, JM Lien, and N. M. Amato. Roadmap-based flocking for complex environments. In *Proc. Pacific Graphics*, pages 104–113, Oct 2002.
11. J. Yu and S. M. LaValle. Probabilistic shadow information spaces. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3543–3549, 2010.
12. J. P. Hespanha, H. J. Kim, and S. Sastry. Multiple-agent probabilistic pursuit-evasion games. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2432–2437, 1999.
13. H.J. Kim, R. Vidal, D.H. Shim, O. Shakernia, and S. Sastry. A hierarchical approach to probabilistic pursuit-evasion games with unmanned ground and aerial vehicles. In *Proc. IEEE Conf. on Decision and Control*, pages 634–639, 2001.
14. R. Vidal, O. Shakernia, H.J. Kim, D. H. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation, and experimentlal evaluation. In *IEEE Transaction son Robotics and Automation*, pages 662–669, 2002.
15. W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 476–481, 2000.
16. J. W. Durham, A. Franchi, and F. Bullo. Distributed pursuit-evasion with limited-visibility sensors via frontier-based exploration. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3562–3568, 2010.
17. T. Oskam, R. Sumner, N. Thuerey, and M. Gross. Visibility transition planning for dynamic camera control. In *Proc. of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 55–65. ACM, 2009.
18. Brian Yamauchi. Frontier-based exploration using multiple robots. In *International Conference on Autonomous Agents (Agents '98)*, pages 47–53, 1998.