Staying Hidden: An Analysis of Hiding Strategies in a 2D Level with Occlusions

Navjot Singh and **Clark Verbrugge**
School of Computer Science
McGill University
Montréal, Québec, Canada
navjot.singh2@mail.mcgill.ca, clump@cs.mcgill.ca

Abstract

The need to stay hidden from opponents is a common feature of many games. Defining algorithmic strategies for hiding, however, is difficult, and thus not usually a non-player character activity outside of very simple or scripted behaviours. In this work we explore several algorithmic approaches for ensuring a character can remain hidden with respect to another, moving agent. We compare these strategies with an upper-bound solution based on a known opponent path, giving us a mechanism for evaluating both relative efficacy and for understanding the different factors that affect success. Experimental evaluation considers multiple levels, including ones adapted from commercial games, and also examines the impact of relative movement speed and different observer movements. Our analysis shows that simple cost-effective approaches to hiding are feasible, but success strongly depends on level geometry, with a large gap remaining between heuristic and optimal performance.

Introduction

The ability to hide from opponents is central to stealth-oriented games, but also found in most modern combat and role-playing games. In most games, however, hiding is primarily a player activity, and not usually a function of non-player characters (NPCs). Enemies and allies may seek cover in combat, temporarily avoiding direct line-of-sight, but more sophisticated uses of hiding that attempt to avoid observation in the first place, or remain hidden for long times are computationally expensive, and thus rare. Even when co-operating with a sneaking player, NPC companions rely on scripted contexts for hiding, or are included in the player's stealth mode, irrespective of their actual visibility.

In this work we examine feasible algorithmic strategies for hiding as a means of expanding NPC behaviours. We consider a basic greedy approach, as well as extensions that attempt to take better advantage of occlusion while still being computationally cheap. To evaluate relative success we develop an upper-bound model, describing an optimal approach for hiding from a known observer path. Using multiple game levels, as well as different observer paths and speeds, we show our heuristics can be effective, although with a large variance in success depending on the properties of the level structure and NPC position. Comparison with the optimal also shows that there remain significant opportunities for further improvement to approach the best performance. Contributions of this work include,

- the design of two novel greedy heuristics; these strategies extend a basic greedy approach, while still preserving computational efficiency;
- a formal approach to estimating an optimal hiding strategy for a given observer path; and
- a detailed evaluation of heuristic hiding in several game levels.

Hiding Strategies

Our game context is intended to model generic situations in which hiding may be necessary; this includes active hide-and-seek, as well as more incidental uses in combat games for avoiding over-powered enemies, or even in game design, as a means of determining appropriate ambush points for expected player movements.

We thus abstract a 2D game level containing obstacles as a polygon with holes. We assume the level is populated with two types of characters: a (single) observer, with motion forming some path through the game level, and one or more agents, whose sole purpose is to hide from the observer. In a hide-and-seek or exploratory context the observer path will be complex and exhaustive, while in other contexts the observer will be unaware of the agents, and so we do not make strong assumptions about the path taken, and simply treat it as a route from a start position to a goal position in the game level, through which the observer moves at a constant speed.

Computing visibility from a path is a form of *weak visibility*, for which only high complexity solutions are known (Ghosh 2007; Ghosh and Goswami 2013). Instead, we discretize the observer's path, and compute point visibility from each path point using a simple angular sweep algorithm to construct a star visibility, as shown in the figure 1. For this we consider the observer to have a full 360° Field of View (FoV) and infinite range, although these factors can be easily adjusted.

The agent characters aim to hide from the observer by remaining in the non-visible, or shadow part of the level. For this we explore 4 different hiding strategies, which we will describe below. These strategies all make use of a dis-

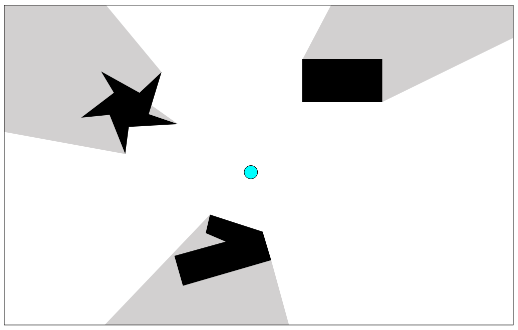

Figure 1: Visibility and shadow regions in a level. The blue dot represents an observation point, grey areas represent non-visible regions, and black areas represent obstacles.

cretized environment (time and space), and depend on basic parametrizations that model the speed of an agent relative to the observer in order to determine where an agent can move in order to stay hidden. We assume agent speed is the same for all agents, and does not change. More formally, we assume we are given a series of regular observer path points $p_1 \ldots, p_{max}, \ldots$ for an observer moving at unit speed, and allow a range of possible agent movements as a radius d_{agent}, where $d_{agent} > 1$ means the agent moves faster than the observer, $d_{agent} < 1$ slower, and $d_{agent} = 1$ gives both agents and observer equal speeds.

Simple Greedy

Our first strategy is a straightforward greedy approach. For this, an agent remains at its current position as long as it continues to be in shadow. As soon as the shadow adjusts such that it is no longer hidden, it locates the position closest to it which would be in shadow in the next observer step. Note that this assumes there is sufficient window of opportunity for the agent to move upon shadow changes without being detected, although this could also be done based on predicted next positions.

This strategy works well when the change in the visibility polygon between each observer step is slow and gradual, giving an agent enough time to move until it is out of immediate danger of being found. This approach also requires relatively little calculation, and it is our fastest strategy. However, it easily fails when the visibility polygon changes dramatically, such as when an observer comes around a corner.

Max Shadow Greedy

The choice of the closest hidden spot in the simple greedy strategy is clearly naive in being minimally responsive to impending detection. Our *max shadow* greedy approach is slightly more sophisticated, in aiming to select a new position that is not only in shadow, but as "deep" in shadow as possible within the agent's movement radius. This does not improve the situation for gradual changes in shadow, but can better accommodate sudden larger changes.

Figure 2 shows the design. The agent is currently positioned at the orange circle at the center of the light green d_{agent} radius, but will be exposed given the new shadow for

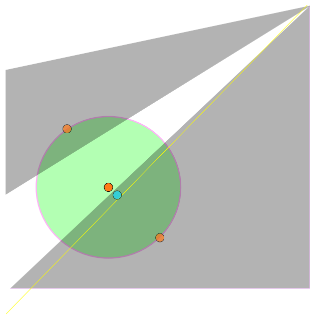

Figure 2: Max shadow greedy.

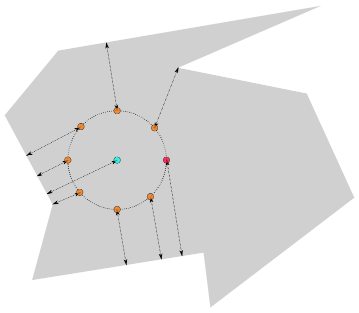

Figure 3: The shadow assisted computation.

p_{n+1} indicated by the grey area. A simple greedy approach would move the agent to the blue circle, minimally within the shadowed region. This point can tolerate further incremental changes in its distance to a shadow edge of $\leq d_{agent}$, but a larger subsequent change will expose this point. The two light orange positions on the perimeter of the d_{agent} radius, on the other hand, are potentially safer, in that a sudden change in the shadow line of $d_{agent} + k$ may be needed to expose them, where k is the current distance to the shadow. This of course does depend on the shape and evolution of the shadow region, and even our simplified example shows multiple reasonable candidates that are deeper in shadow than the blue dot. To avoid a more complicated determination of which point is "deepest" in shadow (which is the basis of our next heuristic), here we compute the set of movement-perimeter arcs contained within shadow, and move to the center of the largest arc (i.e., the bottom right orange dot).

Shadow Centering

Both the greedy and max shadow greedy strategies are "last moment" hiding strategies, only ever reacting to imminent discovery. We thus also introduce a strategy that aims to constantly improve an agent's position by continually looking for a safer position inside the current shadow.

Heuristically, an agent that stays deep within a shadowed region has improved opportunity for staying in shadow over someone at the periphery, as a larger portion of their reachable future positions are likely to be in shadow in the future. Our approach is thus to have the agent try to remain "centered" inside the shadow.

The center of a non-convex polygon is not well-defined,

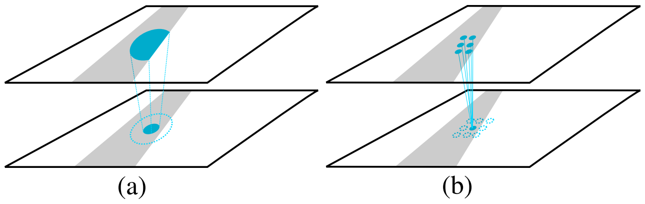

Figure 4: Optimal computation.

and so we use a heuristic calculation. In this we consider moving to future positions within the shadow polygon, prioritizing choices by a weighted combination of their distance to (a) the closest edge of the shadow polygon, (b) the furthest visible edge of the shadow polygon, and (c) the nearest vertex of the shadow polygon. The motivation behind this approach is that the agent has the best chance of remaining hidden if it moves away from the closest shadow edge, heads towards the furthest shadow edge, and moves away from the nearest vertex. Figure 3 shows the (a) part of this calculation. The small light-blue circle marks the original position of the agent within a newly adjusted shadow polygon, the orange dots future positions on the perimeter of d_{agent}, and the red dot as the one that would maximize (a). In practice we choose the best point based on a sum of all three distances for each point, scaling them by factors $w_a > w_b > w_c$ to reflect their relative importance.

Optimal

In order to better understand our hiding strategies we compare them to an optimal strategy. This approach assumes a known deterministic movement for the observer, and is computationally expensive, but can determine the full set of ideal motions and positions that can stay hidden.

The basic idea using a discretized time model is illustrated in Figure 4 (a). The bottom layer shows a point in the level domain, currently within shadow. The dotted radius around it indicates the set of possible positions this point can reach within a given time unit; the faster the agent can move the larger this area, but in general some of this area will be in shadow, and some not. The layer above it shows the next time unit, where the resulting set of possible positions has been intersected with the shadow polygon in that time frame to give us a new set of points, each reachable from the initial point, and each still in shadow. This process is then repeated to compute a set of points in each subsequent time unit, each time expanding the set of hidden points according to reachability, and intersecting it with the shadows available in that time, as computed given the observer's next position. Note that this model assumes time is discretized with sufficient granularity to ensure the shadow area changes in a simple, linear fashion, and so allows us to guarantee a motion from a hidden point in one time frame can be connected to any reachable point still in shadow in the next frame.

The results of this process gives us a complex, 3D manifold that encodes all successful motions and positions from a given starting region. Our actual process further discretizes this, however, both in order to avoid the need to explicitly

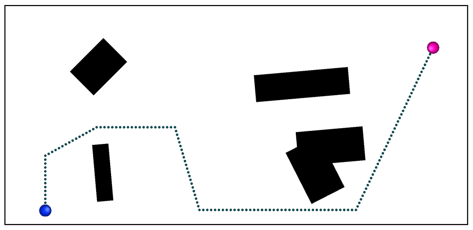

Figure 5: Level I setup

model arbitrary polyhedra, and to also give us a simple graph form we can use for easy pathfinding choices. Figure 4 (b) shows our discrete analog. Here we assume the level space is organized as a 2D grid, and instead of a polygonal region we compute a reachable set of points, pruning them in the next layer of any points now outside the new shadow region. We use this to construct a 3D graph, connecting each point in the lower layer to each point in the next layer up that is both reachable and within shadow. The result is a graph where the number of vertices depends on the time and space granularity, and the number of edges depends on agent speed as a branching factor, but restricted to shadow locations in each layer.

If a path exists in this graph going from a starting point at t_0 to a point in the layer at t_{max}, then an agent at that point has a set of movements that allows them to be safely hidden for the entirety of player movement. The number of points which have such a successful strategy can also be computed, giving us a measure of the maximum amount of level space that can be filled with hiding agents (ignoring agent movement collisions), and the maximum length of a path through time from each initial point can also be computed to give a relative scale of the longevity of each hiding spot.

Experimental Results

Evaluation of our strategies is performed on several game levels, including ones first-person shooter (FPS) and role-playing games (RPG). For each experiment an observer will move along a deterministic path and agents try to hide behind obstacles. The parameters we vary in our tests are the strategy engaged, agent speed relative to the observer's speed, and observer paths. We do not measure execution time, as it is dominated mainly by the number of agents, which varies throughout our experiments.

Level I

Our first test level is shown in the figure 5, and consists of multiple obstacles including a non-convex one. This game level acts as a proof of concept for our strategies and the experimental setup. The starting position of the observer is the blue circle while it ends its path at the pink circle.

Figure 6 (left) shows the number of still unobserved agents remaining as a function of observer path length, with agents restricted to 70% of the speed of the observer. Here we can see a quick culling as the observer begins moving and discovers agents near the periphery of the initial shadows, followed by a plateau as the observer moves through

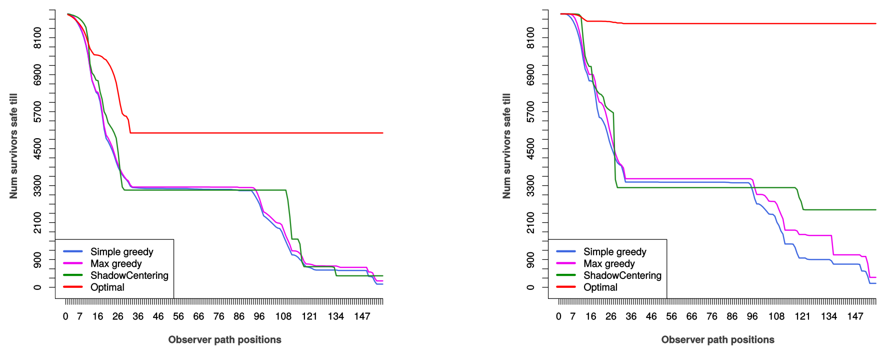

Figure 6: Number of agents remaining in *Level I* versus observer path length with agents at 70% (left) and 150% (right) of observer speed.

the middle of the level, only gradually changing shadow shapes. The latter shows strategies can be effective when movement speeds provide enough reaction time to follow shadow changes. A uniformly larger drop for all heuristics occurs when the observer rounds the non-convex obstacle. Under optimal conditions agents can transition from hiding behind the non-convex obstacle to the other side, or the one above it, but heuristics are limited by both speed and lack of foresight.

Figure 6 (right) shows results when agents are faster, 1.5 times the speed of the observer. In this case we see some separation between heuristics, and agents applying the *Shadow centering* strategy are more capable of remaining hidden till the end of the observer path than the two greedy strategies. A faster speed compensates for the speed of change in the shadow region, at least for agents that try to remain in the central regions of shadows. In particular, agents initially in the alcove region in the bottom right obstacle are able to get out and follow the shadow around the obstacle with *shadow centering*, while ones using greedy strategies end up herded deeper into the alcove, where they are eventually seen. We can see here that speed is an important success factor, but not in all cases, and there remains great room for improvement.

Level II

Now we will examine a game level from *Wasteland 2*, a turn-based role-playing game. We built a level based on the *Temple of Titan-the underground*, a typical RPG level consisting of corridors and rooms, with relatively little open space. Here we consider only short, partial observer paths, mainly due to the cost of computing optimal results on this large level, but also since the level structure means results are dominated by corridor/room enters and exits, and not overall path structure.

The path we consider, shown in figure 7, explores a portion of the right side of the level. The observer starts from the bottom right at the blue circle and follows the dotted line to the pink circle. The level is discretized and color-coded to show the result of the optimal strategy: green regions indicate starting positions that would be safe for the entire observer path, while red and yellow regions are discovered by the observer very early, or later in the path. These results are based on a 36% agent speed—higher speeds show generally similar results in that the amount of occluded space in this level allows the optimal strategy to preserve close to 100% of initially unobserved agents, but interesting behaviours are best revealed at lower speeds.

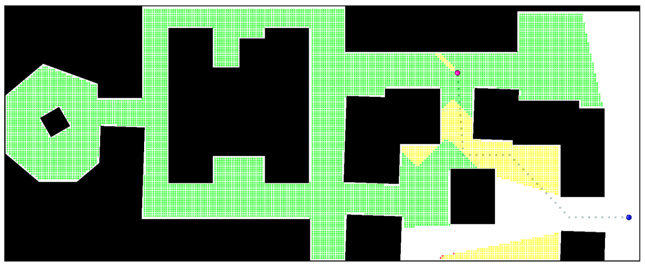

Figure 7: *Level II* setup; color-coding shows relative success for initial agent positions at 36% of the observer speed with an *optimal* approach.

In figure 7 we see a complex structure to the yellow regions, and in particular an odd, narrow strip of sub-100% success near the observer's terminus. Closer analysis revealed that agents surrounding this region are able to move to the left or right to maintain a safe position as the observer approaches, but points in the region are not quite able to do so. The structure of this space depends on the evolving shadows, and the nature of these regions is not necessarily evident without performing the experiments.

Figure 8 compares results for our different strategies at a speed of 36%, although results are nearly identical at 100%. All heuristics are elevated due to the large, unobserved space of the level, but similar to the low-speed results of our previous level the 3 heuristics do not show much difference. Again we observe large, stepped drops in agents hidden, especially towards the end, as the observer enters a fairly large corridor. Here *shadow centering* is even less effective than greedy strategies: our strategy for moving toward a "cen-

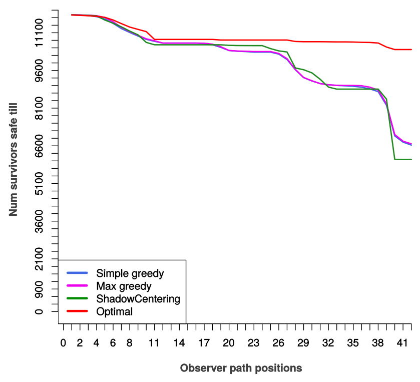

Figure 8: Number of agents remaining versus observer path length at 36% of observer speed for *Level II.*

tral" area of a shadow polygon in such a complex, maze-like space tends to cause agents to go to move away from walls and cluster in room or corridor centers, leaving no options when the observer comes out of the narrow pathway.

Further experiments were conducted on this level with other observer paths and varying speeds. We do not present these for space reasons, but they give similar results, and it is clear that by far the most important factor is the size and shape of obstacles and the structure of game level through which the observer moves. In this case long, narrow pathways with sharp corners mean changes in shadows are infrequent but sudden, and for heuristics to approach optimal results would require some element of path prediction.

Level III

This experiment is based on the *Crash* game level from *Call of Duty 4: Modern Warfare,* a 2007 first-person shooter video game. Different from our underground, RPG level, this level is relatively open, but also well filled with obstacles and alcoves for cover and ambush purposes. Again, we consider only short observer paths. Figure 9 shows the level and one of the observer paths we analyzed, starting from the top at the blue circle and following the dotted line to the pink circle. Note that we exclude the right third of the original level from our image, experiments and analysis as, like the *Wasteland* experiments, the abundance of relatively static area raises all results uniformly.

Figure 10 show results at 60% and 120% speeds. Surprisingly we find at either speed that the greedy strategies perform significantly better than *shadow centering.* This is counter-intuitive given results from *Level I,* which suggested that *shadow centering* performs better when the change in shadow regions is gradual, shifting incrementally around independent obstacles. In this case, however, the obstacles are mostly small, making their shadows relatively long and thin. Agents attempting to stay in the shadow center end up far away from the obstacle itself, but as the rate at which an obstacle's shadow can shift increases as one moves further away from the obstacle, these agents are prone to being

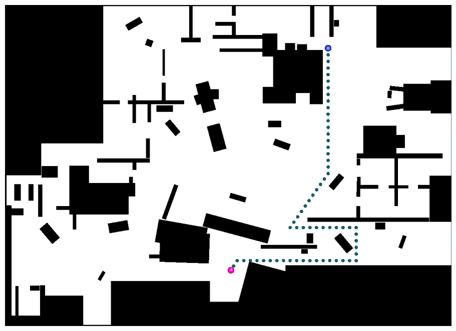

Figure 9: Level III setup

quickly revealed by small changes in observer position. Increasing agent speed is thus effective at helping the greedy models stay on the opposite side of obstacles, but has almost no impact on *shadow centering* agents, which are still not able to follow the rapid shadow changes.

Related Work

Prior work in analyzing and optimizing stealthy behaviours has focused on pathing problems, attempting to compute a route that either guarantees unobservability or at least heuristically minimizes the probability of detection. The problem space is also framed by a variety of theoretical results in the general area of *pursuit-evasion*, wherein one or more pursuers tries to detect or capture one or more evaders.

Stealth and pursuit-evasion problems tend to be computationally hard. Even a basic problem with a single pursuer and single evader in a confined but otherwise empty space (Rado's famous "Lion and Man" scenario) has significant complexity, with ideal strategies for both parties depending on relative speed, time bounds, and arena shape assumptions (Noori and Isler 2014). The addition of simple obstacles adds further complication (Karnad and Isler 2009) to movement choices. Suzuki & Yamashita show that the ability of a single searcher with k "flashlights" (rays) to illuminate an evader depends on the polygonal shape, giving conditions for $k = 1$ searchable polygons, and defining a class of polygons for which $k = 2$ is sufficient (Suzuki and Yamashita 1992). Gerkey *et al.* extended the rays to ϕ-radian cones and show that computing the minimum number of ϕ-searchers is NP-hard (Gerkey, Thrun, and Gordon 2006).

The more directly related problem of staying hidden has mainly been considered from the perspective of the searcher. Even in a simple polygon a single searcher cannot always locate an arbitrarily fast evader (LaValle et al. 1997), and $O(\log(n) + \sqrt{h})$ pursuers may be needed for an n-vertex, h-hole polygon (Guibas et al. 1997). Bounded-speed assumptions can help (McGee and Hedrick 2006), but in general the problem is still quite difficult for single pursuers (Tovar and LaValle 2008). A combination of equal movement speeds and multiple pursuers can be efficient though, with 3 pursuers sufficient to capture an evader in a polygon with holes if equal movement speed is assumed (Bhadauria and Isler 2011). Chung *et al.*'s survey gives more information on the

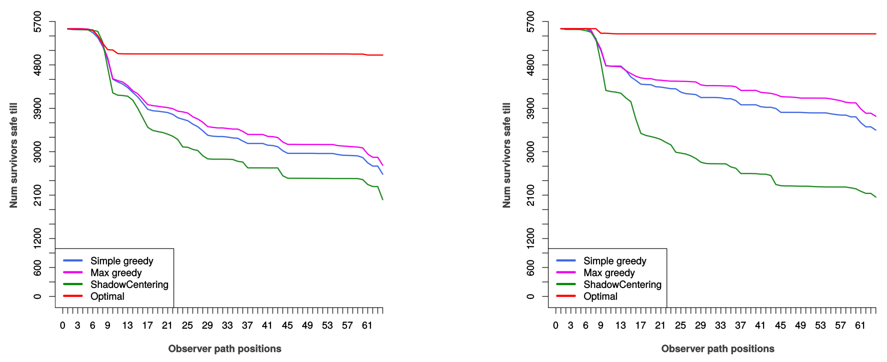

Figure 10: Number of agents remaining in *Level III* at 60% (left) and 120% (right) relative speeds.

many different formulations and results for pursuit-evasion problems (Chung, Hollinger, and Isler 2011).

Work exists with some similarity to ours, in aiming to find reachable, unobserved positions given a fixed observer path. Teng *et al.* follow the same basic construction we use in our optimal solution approach, but going backward in time: starting at the end-goal position, a single point is dilated by the subject's movement speed and the resulting area intersected with the unobserved region of the prior time slice (Teng, DeMenthon, and Davis 1993). The subject's movement is then constructed by selecting the point in this area closest in a straight line sense to the starting position, repeating this process to assemble a full path from start to goal if possible. Park *et al.* use a continuous state space and forward search, computing a discrete time-ordered set of undetected segments in the configuration space and selecting the next path state as the midpoint of each unobserved segment (Park et al. 2009). Both these heuristics aim at path generation, finding a stealthy path between two given points rather than optimal hiding, and so are not directly usable in our context.

Other work on stealthy paths has been at least partly probabilistic, defining variations on potential fields in order to compute a covert path from one location to another. Tews *et al.*, for instance, assign a risk factor to each point, combining the resulting global minima with a strategy for remaining within shadow-frontiers to make pathing decisions (Tews, Mataric, and Sukhatme 2004). Marzouqi & Jarvis describe covert navigation, finding a path of minimal risk given some number of either known or unknown static observers (Marzouqi and Jarvis 2006).

These works have mainly aimed at robotics. In a games context, both deterministic and probabilistic approaches have been considered. Covert paths can be heuristically computing on an aggregate probability map, statically summarizing the potential for being seen at different locations (Johansson and Dell'Acqua 2010). A similar technique is found in *Third Eye Crime*, where gameplay is created by showing the resulting probability field to the player (Isla 2013). Work by Tremblay *et al.* uses a randomized pathfinding search to compute stealthy path solutions, treating it as a motion problem in a 3D state space (Tremblay et al. 2013). Their design has the advantage of extending to non-stealthy activities, such as combat (Tremblay, Torres, and Verbrugge 2014), but requires deterministic observer behaviours.

Conclusions & Future Work

The potential for characters to stay hidden in a level offers interesting, additional opportunities for game design. A player may be required to find an actively hiding character, extending more common uses of simple pre-defined hiding positions in hide-and-seek scenarios (Gamezinvaders 2015), or hiding can be used to enable wounded characters to better flee from combat. Our work evaluates some basic strategies for hiding, and gives a technique for measuring their quality relative to a theoretical best solution. The latter is further helpful in showing the relatively quality of hiding spots within a level, and could be extended to other uses, such as defining to what degree a level is hiding-friendly.

Our comparison with the optimal approach suggests that while greedy approaches can be effective, they are also limited by their inherent lack of foreknowledge—alcoves and shadow "centers" act as local minima in hiding solutions, but are clearly not globally good. Improved strategies are likely possible with even small amounts of lookahead, and perhaps also by considering level structure in more detail, as sharp obstacle corners imply fast shadow movements. The heuristic–optimal gap may also be reduced in the presence of collisions between hiding agents, although that also implies a non-trivial, multi-agent path scheduling problem. Finally, success of hiding heuristics under probabilistic knowledge of observer position would be interesting to explore.

Acknowledgements

Research supported by the Natural Sciences and Engineering Research Council of Canada, Application ID #249902.

References

Bhadauria, D., and Isler, V. 2011. Capturing an evader in a polygonal environment with obstacles. In *IJCAI'11: Proceedings of the Twenty-Second International Joint Confer-*

ence on Artificial Intelligence - Volume Volume Three, 2054–2059. AAAI Press.

Chung, T. H.; Hollinger, G. A.; and Isler, V. 2011. Search and pursuit-evasion in mobile robotics. *Autonomous Robots* 31(4):299–316.

Gamezinvaders. 2015. Find the hidden children using your Witcher senses WITCHER 3 walk through 25. https://www.youtube.com/watch?v=OVuJci8-FRw. Accessed June 6, 2016.

Gerkey, B. P.; Thrun, S.; and Gordon, G. 2006. Visibility-based pursuit-evasion with limited field of view. *International Journal of Robotics Research* 25(4):299–315.

Ghosh, S. K., and Goswami, P. P. 2013. Unsolved problems in visibility graphs of points, segments, and polygons. *ACM Computing Surveys* 46(2):22:1–22:29.

Ghosh, S. K. 2007. *Visibility Algorithms in the Plane*. Oxford University Press.

Guibas, L. J.; Latombe, J.-C.; Lavalle, S. M.; Lin, D.; and Motwani, R. 1997. Visibility-based pursuit-evasion in a polygonal environment. In Dehne, F.; Rau-Chaplin, A.; Sack, J.-R.; and Tamassia, R., eds., *Algorithms and Data Structures*, volume 1272 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 17–30.

Isla, D. 2013. Third Eye Crime: Building a stealth game around occupancy maps. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Johansson, A., and Dell'Acqua, P. 2010. Knowledge-based probability maps for covert pathfinding. In Boulic, R.; Chrysanthou, Y.; and Komura, T., eds., *Motion in Games*, volume 6459 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 339–350.

Karnad, N., and Isler, V. 2009. Lion and man game in the presence of a circular obstacle. In *IROS'09: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5045–5050. Piscataway, NJ, USA: IEEE Press.

LaValle, S. M.; Lin, D.; Guibas, L. J.; Latombe, J.-C.; and Motwani, R. 1997. Finding an unpredictable target in a workspace with obstacles. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, volume 1, 737–742 vol.1.

Marzouqi, M. S., and Jarvis, R. A. 2006. New visibility-based path-planning approach for covert robotic navigation. *Robotica* 24(6):759–773.

McGee, T., and Hedrick, J. 2006. Guaranteed strategies to search for mobile evaders in the plane. In *Proceedings of the 2006 American Control Conference*, 2819–2824.

Noori, N., and Isler, V. 2014. Lion and man with visibility in monotone polygons. *International Journal of Robotics Research* 33(1):155–181.

Park, J.-H.; Choi, J.-S.; Kim, J.; and Lee, B.-H. 2009. Roadmap-based stealth navigation for intercepting an invader. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, 2402–2407. Piscataway, NJ, USA: IEEE Press.

Suzuki, I., and Yamashita, M. 1992. Searching for a mobile intruder in a polygonal region. *SIAM Journal of Computing* 21(5):863–888.

Teng, A. Y.; DeMenthon, D.; and Davis, L. S. 1993. Stealth terrain navigation. *IEEE Transactions on Systems, Man and Cybernetics* 23(1):96–110.

Tews, A.; Mataric, M.; and Sukhatme, G. 2004. Avoiding detection in a dynamic environment. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, 3773–3778 vol.4.

Tovar, B., and LaValle, S. M. 2008. Visibility-based pursuit-evasion with bounded speed. In Akella, S.; Amato, N.; Huang, W.; and Mishra, B., eds., *Algorithmic Foundation of Robotics VII*, volume 47 of *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg. 475–489.

Tremblay, J.; Torres, P. A.; Rikovitch, N.; and Verbrugge, C. 2013. An exploration tool for predicting stealthy behaviour. In *The Second Workshop on Artificial Intelligence in the Game Design Process*.

Tremblay, J.; Torres, P. A.; and Verbrugge, C. 2014. An algorithmic approach to analyzing combat and stealth games. In *IEEE Conference on Computational Intelligence and Games*, 1–8.